MATHEMATICAL TECHNIQUES AND PROBLEMS IN TELECOMMUNICATIONS

Tomar, September 8-12, 2003

Carlos Fernandes, Joaquim Júdice, Carlos Salema

24

CENTRO INTERNACIONAL de MATEMÁTICA

Introduction

Bridging the communication gap between engineers and mathematicians is a well-known and old problem. Suffice to recall Oliver Heaviside the engineer who, in spite of being the creator of operational calculus, was unable to demonstrate it in a manner that was acceptable to the mathematicians of his time. The result was that operational calculus was regarded with suspicion and was only fully accepted many years later.

Another example, now in the opposite direction, involves Claude Shannon, the mathematician known as the father of information theory. Given the considerable difficulties many telecommunication engineers found in understanding his papers, particularly the later ones, only recently and 40 years after being published are Shannon results being fully understood and applied.

It is widely recognized that most problems telecommunication engineers face, present a wide range of mathematical difficulties spanning from calculus to number theory, from statistics to topology, from transforms to heuristics. In order to create and diffuse new knowledge in the field of telecommunications and to foster a much required and fruitful cooperation between mathematicians and telecommunications engineers, "Instituto de Telecomunicações" organized in Tomar, from 8 to 12 of September 2003, the Summer School "MTPT – Mathematical Techniques and Problems in Telecommunications".

MTPT model was drawn from an earlier event that took place in 1997 and consisted of a competition of mathematical problems in telecommunications, which, after refereeing and selection, were made available to the community of mathematicians with a request for solutions. The presentation and discussion of the selected problems and solutions was complemented by a set of lectures, which introduced some of the newer mathematical techniques with potential applications in telecommunications.

This time, with the generous support of the "Centro Internacional de Matemática" (CIM) and "Instituto Politécnico de Tomar", it was possible to extend the duration of MTPT to a full working week and to format it as a Summer School. Each day of this school was concerned with an area of mathematics that finds important applications in telecommunications: Stochastic Processes, Transforms, Partial Differential Equations, Optimization and Evolutionary Computing. The morning session of 3 hours was devoted to an extended lecture by an expert (usually a mathematician), aiming to introduce the subject to an audience of engineers and mathematicians. The afternoon session started with the presentation and discussion of two problems and their solutions (when available) and ended with a one-hour conference on applications in telecommunications of the techniques discussed in the morning session.

This volume contains a number of articles concerning some of the morning and afternoon lectures and the solutions of a few problems that have been presented at the MTPT Summer School. We sincerely hope that it can help to bridge the gap between mathematicians and telecommunication engineers. Special thanks should be given to the invited speakers, to the authors of the problems and to all participants that worked on the problems and presented and discussed possible solutions to them. We are also quite grateful to "Instituto Politécnico de Tomar" and to "Centro Internacional de Matemática" for providing the opportunity to organize such an interesting event. Finally, our warmest thanks to Luís Merca Fernandes and João Patrício for their excellent work in the organization of the School and of this volume.

The Editors

Programme

Monday, 8

Topic: Stochastic Processes
Chairman: Carlos Belo

09:00 - 09:30	Welcome Session (CIM Director, Luís Trabucho and School Director, Carlos Salema).
09:30 - 11:00	Ivette Gomes, Course on Stochastic Processes, Part 1.
11:00 - 11:30	Coffee Break.
11:30 - 13:00	Ivette Gomes, Course on Stochastic Processes, Part 2.
13:00 - 15:00	Lunch Break.
15:00 - 16:30	Discussion of Problems 1 and 2.
16:30 - 17:00	Coffee Break.
17:00 - 18:00	Conference: Rui Valadas, *Statistical Characterization and Modelling of IP Traffic.*
18:45	Reception by the Mayor of Tomar.

Tuesday, 9

Topic: Transforms
Chairman: Mário Figueiredo

09:30 - 11:00	Joana Soares, Course on Transforms, Part 1.
11:00 - 11:30	Coffee Break.
11:30 - 13:00	Joana Soares, Course on Transforms, Part 2.
13:00 - 15:00	Lunch Break.
15:00 - 16:30	Discussion of Problems 3 and 4.
16:30 - 17:00	Coffee Break.
17:00 - 18:00	Conference: Aníbal Ferreira, *Dynamic Window Switching in Audio Coding Using the MDCT Transform.*

Wednesday, 10

Topic: Partial Differential Equations
Chairman: Carlos Fernandes

09:30 - 11:00	Enrique Zuazua, Course on Partial Differential Equations, Part 1.
11:00 - 11:30	Coffee Break.
11:30 - 13:00	Enrique Zuazua, Course on Partial Differential Equations, Part 2.
13:00 - 15:00	Lunch Break.
15:00 - 16:30	Discussion of Problems 5 and 6.
16:30 - 17:00	Coffee Break.
17:00 - 18:00	Conference: Carlos Alves, *Mathematical and Numerical Problems on Wave Scattering.*
20.00	School Dinner

Thursday, 11

Topic: Optimization
Chairman: Joaquim Júdice

09:30 - 11:00	Maurício Resende, Course on Optimization, Part 1.
11:00 - 11:30	Coffee Break.
11:30 - 13:00	Maurício Resende, Course on Optimization, Part 2.
13:00 - 15:00	Lunch Break.
15:00 - 16:30	Discussion of Problems 7 and 8.
16:30 - 17:00	Coffee Break.
17:00 - 18:00	Conference: José Craveirinha, *Application of Multicriteria Analysis to Planning and Design Problems – Issues and Trends.*

Friday, 12

Topic: Evolutionary Computing
Chairman: Pedro Oliveira

09:30 - 11:00	Eckart Zitzler, Course on Evolutionary Computing, Part 1.
11:00 - 11:30	Coffee Break.
11:30 - 13:00	Eckart Zitzler, Course on Evolutionary Computing, Part 2.
13:00 - 15:00	Lunch Break.
15:00 - 16:30	Discussion of Problems 9 and 10.
16:00 - 17:00	Conference: Agostinho Rosa, *Application of Evolutionary Computing to Games and Scheduling.*

Courses

- Ivette Gomes (DEIO Univ. Lisboa), *Stochastic Processes in Telecommunication Traffic.*
- Joana Soares (DM Univ. Minho), *Transforms, Algorithms and Applications.*
- Enrique Zuazua (Universidad Autonoma, Madrid), *Propagación numérica de ondas: Una introducción.*
- Maurício Resende (ATT, USA), *Some Applications of Combinatorial Optimization in Telecommunications.*
- Eckart Zitzler (Comp. Eng. Comm. Network Lab, Zurich), *Evolutionary Algorithms, Multiobjective Optimization, and Applications.*

Problems

1. Adolfo Cartaxo, *Efficient and Accurate Numerical Solution of Stochastic Partial Differential Equations.*
2. Bárbara Coelho, António Navarro, *Optimal M-QAM/DAPSK Allocation in Narrowband OFDM Radio Channels.*
3. Vitor Silva, Fernando Perdigão, *Computational Complexity of Discrete Fourier Transform.*
4. Henrique Silva, *Optimization of the Dispersion Profile in Solution Links With Dispersion-Varying Compensating Fiber.*
5. António Almeida, *Finding a Stability Region for a Congestion Control Algorithm.*
6. Mário Silveirinha, Carlos Fernandes, *Band Structure of Media With Highly Localized Permittivity Distributions.*
7. Paulo Monteiro, Luis Sá, *Optimal Globality in Time-Domain Digital Arma Filter Design.*
8. Victor Anunciada, *Cost Minimization of a Multiple Section energy Cable Supplying Remote Telecom Equipments.*
9. Carlos Salema, *Numeric Integration of Rapidly Oscillating Functions.*
10. Antoni Zabludowski, *Analysis of Chordal Rings.*

Stochastic Processes in Telecommunication Traffic

M. Ivette Gomes*

Abstract

The main goal of this paper is to review various recent models, within the queueing framework, which have been suggested for teletraffic data. Those models intend to capture the specific features of the data, such as variability of arrival rates, heavy-taildeness of on-periods and off-periods, as well as long-range dependence in teletraffic transmission.

1 Specific features of the data: an introduction

The statistical properties of computer network traffic seem to differ significantly from the voice traffic in the telephone system (see, e.g., Fowler and Leland, 1991, or Willinger and Paxson, 1998), and have revealed to be a great challenge to engineers and statisticians. Telephone calls can be modeled by a Poisson process, i.e., their inter-arrival times are roughly exponentially distributed. The lengths of telephone calls have an exponentially bounded right tail. This implies that the autocorrelations of the network workload decrease exponentially in the time between observations. Moreover, on a sufficiently large time scale the workload smooths out, i.e., the number of call arrivals is approximately equal to the long-term arrival rate of the Poisson process.

But these properties are not usually observed in computer network traffic. On the contrary, file lengths, transmission durations and connection lengths are heavy-tailed, workload processes exhibit long-range dependence (*LRD*) and show "burstiness" across an extremely wide range of time scales (i.e., traffic does not smooth out). Workload measurements in computer networks (i.e., packet or byte counts) show a high level of variability on every time scale that is considered, from milliseconds to minutes. For instance, for the Bellcore measurements, such a conclusion has been drawn by several authors. In Leland et al. (1994) and Willinger et al. (1995) the variability of the workload on the Bellcore *Local Area Networks* (*LAN*s) is shown to be roughly the same on five different time scales. And this invariance under scaling in time and space is taken to be evidence of self-similarity in the workload measurements. That's why computer networks are a far greater challenge to an

[*] DEIO Universidade de Lisboa. E-mail:ivette.gomes@fc.ul.pt

engineer than the telephon system.

In these notes we shall briefly review, in section 2, some concepts in the general field of *stochastic processes*, together with some classical *short-range dependent* models for *inter-arrival times.* Since empirical studies of computer network traffic suggest that there are three properties always present in the data, *heavy tails*, *long-range dependence* and *self-similarity*, we shall discuss these three concepts in Section 3. Finally, in section 4 we review the *ON-OFF* and the *infinite source Poisson* processes, which exhibit *LRD*.

2 A brief introduction to stochastic processes

A stochastic process is a collection of random variables (r.v.'s) $\{X_t\}_{t\geq 0}$, usually indexed in a time t. More specifically:

Definition 1. *A stochastic process with parameter space T is a collection of r.v.'s $\{X_t\}_{t\in T}$ defined on the same sample space Ω. If T is an interval of real numbers, the process is said to have a continuous time space; if T is a sequence of integers, the process is said to have a discrete time space.*
The possible values of X_t are the states of the process. The set $\mathcal{S}$ of all possible states is the state space. The state space may also be either discrete or continuous.

Possible *realizations* or *sample paths* of the same stochastic process are different, but it is possible to detect a clear pattern in their behaviour, i.e., their behaviour is governed by a predictable random mechanism.

We next introduce some standard definitions:

Definition 2. *A stochastic process $\{X_t\}_{t\geq 0}$ has independent increments if and only if for all $0 \leq t_1 \leq t_2 \leq t_3$, $X_{t_3} - X_{t_2}$ and $X_{t_2} - X_{t_1}$ are independent r.v.'s (occurrence in disjoint intervals are independent of each other).*

Definition 3. *A stochastic process $\{X_t\}_{t\geq 0}$ has stationary increments if and only if for all $t \geq 0$ and $h \geq 0$, $X_{t+h} - X_t$ and X_h are equally distributed r.v.'s (the distribution of the increments depends only on the difference in time).*

2.1 Markov processes

Definition 4. *Let $\{X_t\}_{t\geq 0}$ be a time continuous stochastic process which assumes non-negative integer values. The process is called a discrete Markov process if for every $n \geq 0$,* *time points* $0 \leq t_0 \leq t_1 \cdots \leq t_n \leq t_{n+1}$ *and states* $i_0, i_1, \cdots, i_n, i_{n+1}$,

$$P\left(X_{t_{n+1}} = i_{n+1} | X_{t_n} = i_n, \cdots, X_{t_0} = i_0\right) = P\left(X_{t_{n+1}} = i_{n+1} | X_{t_n} = i_n\right),$$

holds, i.e., the future (t_{n+1})*, given the present* (t_n) *and the past* $(t_0, t_1, \cdots, t_{n-1})$*, depends only on the present.*

We shall here be essentially interested in time-homogeneous stochastic processes:

Definition 5. *Let* $\{X_t\}_{t\geq 0}$ *be a discrete Markov process. If the conditional probabilities* $P(X_{t+s} = j|X_s = i)$*,* $s, t \geq 0$ *do not depend on* s*, the process is said to be time homogeneous. Let us then define the transition probabilities*

$$p_{ij}(t) = P(X_t = j|X_0 = i).$$

The $n \times n$ *matrix* $\mathbf{P}(t) = [p_{ij}]$ *is the so-called transition matrix.*

Note that $p_{ii}(0) = 1$ and $p_{ij}(0) = 0$ for $i \neq j$. Consequently $\mathbf{P}(0) = I$, the identity matrix. Notice also that the rows of the transition matrix $\mathbf{P}$ sum up to one — this is what we call a *stochastic matrix.*

Definition 6. *Occurrence times of our Markov stochastic process* $\{X_t\}_{t\geq 0}$ *are the random times* $0 \leq T_1 < T_2 < T_3 \cdots$ *where the process makes a transition from one state to another. The duration between occurrences are the r.v.'s* $Y_n = T_n - T_{n-1},\ n \geq 1$ $(T_0 = 0)$*.*

Another characteristic of a Markov process is its *intensity* or *generator matrix.*

Definition 7. *Let* $\{X_t\}_{t\geq 0}$ *be a discrete time Markov process. Assume there exists* $q_{ij} = p'_{ij}(0) \geq 0$ *for* $i \neq j$*, and* $q_{ii} \leq 0$ *such that, as* $h \to 0$*,*

$$p_{ij}(h) = q_{ij}h + o(h) \quad \textit{and} \quad 1 - p_{ii}(h) = -q_{ii}h + o(h) =: q_i h + o(h), j \neq i,$$

where $q_i = -q_{ii} = \sum\limits_{j\neq i} q_{ij}$*. The probability* q_{ij}*,* $i \neq j$ *is called the transition intensity from state* i *to state* j*. The intensity matrix (or generator matrix) is the matrix* $\mathbf{Q}(t) = [q_{ij}]$*.*

The rows of $\mathbf{Q}$ sum up to zero.

Remark 1. *Notice that the concept of intensity matrix is related to the concept of instantaneous failure rate in Reliability Theory. Such a concept comes to give an answer to the question: "If a system is still working at time* t *what is the probability that it fails immediately?" In the context of streams of events, we may place the question: "If we know that an event has not yet occurred at time t, what is the probability* $\lambda(t)$ *that it occurs immediately?". Mathematically, denoting* T *the time of occurrence of the event:*

$$\lambda(t) = \lim_{dt\to 0} \frac{1}{dt}\ P(T \leq t + dt|T > t).$$

The notion of conditional probability enables us to write,

$$P(T \leq t + dt | T > t) \quad = \quad \frac{P(t < T \leq t + dt)}{P(T > t)} = \frac{F(t + dt) - F(t)}{1 - F(t)}.$$

If F is differentiable with probability density function $f = F'$, then

$$\lambda(t) = \frac{f(t)}{1 - F(t)}.$$

The instantaneous failure rate in Reliability Theory is here called the intensity function for T. Intuitively, if the event does not occur up to time t, the probability that such an event is going to occur in the interval $(t, t + dt]$ for small dt is approximately proportional to dt, with a proportionality constant given by $\lambda(t)$.

2.2 Birth-and-death processes

These processes have revealed to be a useful class of Markov processes whenever we need to analyze queueing systems. In this kind of processes the only possible state transitions are from i to $i - 1$ or from i to $i + 1$. The transition intensity from state i to $i + 1$ (a *birth*) is usually denoted $\lambda_i \geq 0$ for $i \geq 0$, and the transition intensity from i to $i - 1$ (a *death*) is denoted $\mu_i \geq 0$ for $i \geq 1$.

The state space of a birth process is $\Omega = \{0, 1, 2, \cdots\}$. The intensity matrix of such a process is thus given by:

$$\mathbf{Q} = \begin{pmatrix} -\lambda_0 & \lambda_0 & 0 & 0 & 0 & \cdots \\ \mu_1 & -(\lambda_1 + \mu_1) & \lambda_1 & 0 & 0 & \cdots \\ 0 & \mu_2 & -(\lambda_2 + \mu_2) & \lambda_2 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

A great diversity of queueing systems are suitably modelled by birth-and-death processes. The numbers $\{\lambda_i\}$ and $\{\mu_i\}$ are interpreted as the arrival rate of the queue and the server rate, respectively.

2.3 Poisson arrivals

For a complete description of Poisson processes see for instance Gut (1995). A Poisson process is a counting process:

$$X_t = \text{number of occurrences in the interval } (0, t],$$

which may be formally defined in the following way:

Definition 8. *A stochastic process $\{X_t\}_{t\geq 0}$ is a Poisson process with intensity $\lambda > 0$, and we use the notation $X_t \frown Poisson(\lambda)$ if and only if*

(a) $\{X_t\}_{t\geq 0}$ *is integer-valued, increasing and* $X_0 = 0$*;*

(b) $\{X_t\}_{t\geq 0}$ *has independent and stationary increments;*

(c) $P(X_t = x) = e^{-\lambda t}\ (\lambda t)^x/x!$ *for* $x = 0, 1, 2, \cdots$*, and for every* $t \geq 0$*.*

We shall briefly refer some of the properties of a Poisson process:

1. $E[X_t] = Var[X_t] = \lambda\ t$;

2. Let $s \geq 0$ be a fixed point of time. Then $Z_t = X_{t+s} - X_s \overset{d}{=} X_t$, a Poisson$(\lambda t)$ r.v., for all $s,\ t \geq 0$.

Remark 2. *Notice that the Poisson process, with intensity λ is a Markovian process, with state space $\mathcal{S} = \{0, 1, \cdots\}$, parameter space $\mathcal{T} = \mathbb{R}^+$, and such that*

(a') $p_{k,k+1}(h) = \lambda\ h + o(h)$*.*

(b') $p_{k,k}(h) = 1 - \lambda\ h + o(h)$*.*

(c') $p_k(0) = \begin{cases} 1 & if \quad k = 0 \\ 0 & if \quad k \neq 0 \end{cases}$.

Consequently, the Poisson process is a pure birth-process, with constant birth-rate equal to λ, the intensity of the process.

An important property of the Poisson process, that we shall present without proof, is the following:

Theorem 1. *$\{X_t\}_{t\geq 0}$ is a Poisson process with intensity λ if and only if the inter arrival times are independent exponential r.v.'s with mean value equal to $1/\lambda$.*
In other words, if we denote $T_1, T_2, \cdots$ the times of occurence of the Poisson events, the r.v.'s $Y_j = T_j - T_{j-1},\ j \geq 1$ ($T_0 = 0$) are independent, identically distributed (i.i.d.), with distribution function (d.f.) $F_Y(t) = 1 - e^{-\lambda t},\ t \geq 0$. The converse also holds true.

The Poisson process exhibits thus a *lack of memory* property (directly related to the exponential inter-arrival times. Indeed:

Theorem 2. *If T is Exponential $(1/\lambda)$, then*

$$P(T > t + s | T > s) = e^{-\lambda\ t} = P(T > t).$$

2.3.1 Superposition of Poisson processes

It is a straighforward result that the superposition of Poisson processes is still a Poisson process. Indeed,

Theorem 3. *Let $\{X_i(t)\}_{t\geq 0}$, $i = 1, 2, \cdots, k$, denote k independent Poisson processes with intensities $\lambda_1, \lambda_2, \cdots, \lambda_k$. Then the superposition*
$Z(t) = \sum_{i=1}^{k} X_i(t)$ is a Poisson process with intensity $\lambda = \sum\limits_{i=1}^{k} \lambda_i$.

2.4 Renewal arrival streams

A large class of stochastic processes are *renewal processes.*

Definition 9. *Let $Y_1, Y_2, \cdots$ be i.i.d. and positive r.v.'s, and let $T_n = Y_1 + Y_2 + \cdots + Y_n$ and*

$$X_t := \max\{n : T_n \leq t\}.$$

Then, the process $\{X_t\}_{t\geq 0}$ is called a renewal process.

Remark 3. *It is obvious that such a simple definition leads to the possible description of many types of stochastic processes as renewal processes. It is often true that a complex stochastic model has one or more embedded renewal processes: this is indeed the basic idea of regeneration, which allows a process to be decomposed into i.i.d. blocks of random length.*

The name "renewal process" is motivated by the fact that every time there is an occurence, the process "starts over again", i.e., it renews itself.

It is possible to prove some important results for renewal processes, as $t \to \infty$. Among them, we state the following ones (Taylor and Karlin, 1998):

Theorem 4. *Let $\{X(t)\}_{t\geq 0}$ be a renewal process with durations Y, with finite variance $\sigma^2 = \mathbb{V}ar\ Y$. Let us denote $\mu = \mathbb{E}\ Y$. Then:*

$$\frac{X(t)}{t} \xrightarrow[t\to\infty]{a.s.} \frac{1}{\mu}, \quad \frac{\mathbb{E}\ X(t)}{t} \xrightarrow[t\to\infty]{} \frac{1}{\mu}, \quad \frac{\mathbb{V}ar\ X(t)}{t} \xrightarrow[t\to\infty]{} \frac{\mathbb{V}ar\ Y}{\mathbb{E}^3\ Y}.$$

Remark 4. *Note that the Poisson(λ) process is obviously a renewal process, where the generators Y_i are exponentially distributed with mean value equal to $1/\lambda$.*

Remark 5. *Although the superposition of Poisson processes is a Poisson process, the superposition of renewal processes is not necessarily a renewal process.*

2.5 Arrivals and the ON/OFF periods

2.5.1 Deterministic inter-arrivals

Sometimes, the inter-arrivals may be considered deterministic, being T the inter-arrival time. In the ON-periods, the busy periods in which arrivals happen, let us denote N_b the number of arrivals. It is often assumed that N_b is geometrically distributed, with probability function

$$P(N_b = k) = p^{k-1}(1-p), \quad k = 1, 2, \cdots . \tag{2.1}$$

We then have $\mathbb{E}N_b = 1/(1-p)$, and consequently the ON-periods have an expectation

$$\alpha = T/(1-p) =: nT,$$

where T is the fixed inter-arrival time, and n is the expected number of arrivals in an ON-period.

It is also often assumed that the OFF-periods (period of time without arrivals) are exponentially distributed with mean $1/\beta$, and such a source may be viewed as a two state birth-and-death process.

2.5.2 Exponentially distributed inter-arrivals

To be coherent with the deterministic source, we shall assume that in the ON-periods the times Y between consecutive events are exponentially with mean value $T = 1/\lambda$. This means that the events occur according to a Poisson Process with intensity λ. Let N_b denote the number of arrivals in an ON-period, and let us assume that N_b is geometrically distributed with mean $n = 1/(1-p)$. Let us also assume that Y and N_b are independent. It is then easy to show that the ON-periods U are also exponential distributed. Indeed, the conditional r.v.,

$$U|_{N_b=k} = \sum_{i=1}^{k} Y_i \overset{d}{=} Gama(k, T),$$

with p.d.f.

$$f_{U|_{N_b=k}}(t) = \frac{1}{\Gamma(k)} \, \frac{t^{k-1}}{T^k} \, e^{-t/T}, \ t \geq 0.$$

Consequently, the law of total probability enables us to write

$$\begin{aligned} f_U(t) &= \sum_{k\geq 1} P(N_b = k) f_{U|_{N_b=k}}(t) = (1-p) \, e^{-t/T} \sum_{k\geq 1} \frac{(pt)^{k-1}}{(k-1)! \, T^k} \\ &= \frac{(1-p) \, e^{-t/T}}{T} \, e^{pt/T} = \frac{1-p}{T} \, e^{-t(1-p)/T}, \ t > 0, \end{aligned}$$

i.e., U is an exponential r.v. with mean value $T/(1-p) = nT$, the same as in the deterministic inter-arrival source.

We have here again an underlying birth-and-death process. The deterministic source of events is now replaced by a Poisson source.

2.5.3 A more general source of events

Let us introduce the r.v. W, exponentially distributed with mean value $1/\beta$, which also describes the length of the OFF-periods. Let us also assume that the inter-arrival time in the ON-period, Z, is such that

$$Z = \begin{cases} T & \text{with probability} \quad p = \frac{n-1}{n} \\ T + W & \text{with probability} \quad p = \frac{1}{n} \end{cases},$$

whenever we have deterministic inter-arrivals, and

$$Z = \begin{cases} Y & \text{with probability} \quad p = \frac{n-1}{n} \\ Y + W & \text{with probability} \quad p = \frac{1}{n} \end{cases},$$

whenever we have exponential inter-arrivals Y.

The consideration of sequences of i.i.d. r.v.'s distributed as Z leads us to a single source of events, which can be seen as a renewal process, generalizing the cases considered before.

2.5.4 The superposition of independent sources

The superposition of arrival sources can be viewd as a birth-and-death process where the states represent the number of sources that are currently in the ON-state. Consequently, state i represents that i sources are active. Such a birth-and-death process is often referred in the literature as the *Phase Process*, $J(t)$. The birth rate is again given by the mean $1/\beta$ of exponentially distributed iddle periods. The death rate is usually denoted $1/\alpha$, and the probability that the source is ON is $p_{ON} = \alpha/(\alpha+\beta)$.

The limiting probabilities π_i, that the Phase Process is in state i is obviously Binomial,

$$\pi_i = \binom{N}{i} p_{ON}^i \left(1 - p_{ON}\right)^{N-i}, \quad i = 0, 1, \cdots, N, \tag{2.2}$$

where N is the number of superpositioned sources. The intensity matrix of the Phase process is given by

$$\mathbf{Q} = \begin{pmatrix} -N\beta & N\beta & 0 & 0 & 0 & \cdots \\ \alpha & -(\alpha + (N-1)\beta) & (N-1)\beta & 0 & 0 & \cdots \\ 0 & 2\alpha & -(2\alpha + (N-2)\beta) & (N-2)\beta & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \cdots & \cdots & \cdots & \cdots & 0 & N\alpha \end{pmatrix}$$

Markov Modulated Rate Process (*MMRP*)

The $MMRP$ process is a superposition of sources with deterministic inter-arrival times. When the phase process is at state i we have an arrival rate equal to i/T per second, let us say.

This process is difficult to describe mathematically, because the different sources may be unsynchronized, and consequently, we no longer have a deterministic equal inter-arrival time property.

Markov Modulated Poisson Process ($MMPP$)

The $MMPP$ is a widely used tool for the analysis of teletraffic data. Since the superposition of Poisson processes is also a Poisson Process, this process is much simpler to be treated mathematically than the $MMRP$.

2.5.5 Index of Dispersion

To describe the dependence between successive arrivals of an arrival process it is usual to consider the following *Index of Dispersion* (ID), a measure of burstiness of a signal. Let $Y_k,\ k \geq 1$ denote the inter-arrival times associated to our arrival stream. Let us assume that the process of inter-arrival times is stationary, and let $T_k = Y_1 + Y_2 + \cdots + Y_k$ denote the time of occurrence of the k-th event.

Definition 10. *The Index of Dispersion (ID) of the arrival stream is defined as*

$$ID_k := \frac{k \ \mathbb{V}ar \ T_k}{\mathbb{E}^2 \ T_k}$$

For a Poisson arrival stream, we may state:

Theorem 5. *The ID of a Poisson process is equal to 1, for all $k \geq 1$.*

More generally,

Theorem 6. *The ID of a renewal process is constant and equal to $ID = \mathbb{V}arY/\mathbb{E}^2Y$ for all $k \geq 1$.*

Theorem 5 leads us to the use of the indicaror ID_k as a measure of deviation from a Poisson process. Also, Theorem 6 enables the same relatively to a general renewal process — fluctuation in the ID_k sequence enable us to detect deviations from the renewal property.

3 Heavy tails, long-range dependence and self-similarity

Empirical studies of computer network traffic suggest that there are three properties always present in the data: *heavy tails*, *long-range dependence* and *self-similarity.* We shall here briefly review these three concepts, following closely Stegeman (2002).

Let F be the d.f. of a positive r.v. X. We shall denote $\overline{F} = 1 - F$ the tail function of F.

3.1 Regularly varying tails

Definition 11. *The d.f. F (or the r.v. X) has a heavy tail, as $x \to \infty$, if and only if*

$$\overline{F}(x) = P(X > x) = x^{-\alpha} L(x) \quad (\alpha > 0), \tag{3.1}$$

where $L(x)$ is a slowly varying function, i.e.

$$\frac{L(tx)}{L(t)} \underset{t\to\infty}{\longrightarrow} 1, \qquad \textit{for all } x > 0.$$

Remarks:

1. The function $\overline{F}$ in (3.1) is said to be *regularly varying* with *index of regular variation* equal to $-\alpha$.

2. A slowly varying function is a regularly varying function with index of regular variation equal to 0.

3. If $\alpha < 2$ the variance of X is infinite. Indeed, some authors consider that a model is heavy-tailed only if $\alpha \in (0, 2)$.

4. If $\alpha < 1$ the mean value is infinite.

5. A particular important case of the model in (3.1) is provided by the the slowly varying functions $L(x) = C(1 + o(1))$, i.e. $L(x) \to C$, $0 < C < \infty$, as $x \to \infty$. Notice however that a slowly varying function may converge to zero, like $1/\ln x$, or diverge to infinity, like $\ln x$.

Examples of heavy-tailed models

1. The clasical heavy-tailed model is the Pareto model, with a d.f.

$$F(x) = 1 - \left(1 + \frac{x}{\delta}\right)^{-\alpha}, \ x \geq 0 \quad (\alpha,\ \delta > 0).$$

2. Another important heavy-tailed model is the Fréchet model, with a d.f.

$$F(x) = \exp\left\{-\left(\frac{x}{\delta}\right)^{-\alpha}\right\} \ x \geq 0 \quad (\alpha,\ \delta > 0).$$

3. Some other models, also common in the literature related to heavy tails, are the so called *stable models*, which appear as limits related to sum's schemes.

Definition 12. *A stable d.f., $S_\alpha(\sigma, \beta, \mu)$, is characterized by four parameters: the index of stability $\alpha \in (0, 2]$, a scale parameter $\sigma > 0$, a skewness parameter $\beta \in [-1, 1]$ and a location parameter $\mu \in \mathbb{R}$. The characteristic function $\mathbb{E}\left(e^{itx}\right)$ of a stable r.v. is given by*

$$\begin{aligned} \exp\left\{-\sigma^\alpha |t|^\alpha (1 - i\beta \operatorname{sign}(t) \tan(\pi\alpha/2)) + i\mu t\right\} \quad &\textit{if} \quad \alpha \neq 1 \\ \exp\left\{-\sigma |t| (1 + 2i\beta\pi^{-1} \operatorname{sign}(t) \ln|t|) + i\mu t\right\} \quad &\textit{if} \quad \alpha = 1 \end{aligned}$$

Although any stable distribution has a density, in general an explicit expression of the density in terms of elementary functions is unknown. Exceptions (excluding the degenerate case) are the Lévy distribution $S_{1/2}(\sigma, 1, \mu)$, the Cauchy d.f. $S_1(\sigma, 0, \mu)$ and the Gaussian d.f. $S_2(\sigma, 0, \mu) \equiv N(\mu, 2\sigma^2)$.

The parameter β is a skewness parameter, and if $\beta = 0$ the distribution is symmetric around μ. The mean of of a stable r.v. is μ if $\alpha \in (1, 2]$, and for $\alpha \leq 1$, $\mathbb{E}|X| = \infty$. For $\alpha \in (0, 2)$ the variance is infinite. Then (Samorodnitsky and Taqqu, 1994, Property 1.2.15), as $x \to \infty$, S_α has tails given by

$$P(X > x) \sim C_\alpha \sigma^\alpha \, \frac{1+\beta}{2} \, x^{-\alpha}, \quad \text{and} \quad P(X \leq x) \sim C_\alpha \sigma^\alpha \, \frac{1-\beta}{2} \, x^{-\alpha},$$

where

$$C_\alpha = \begin{cases} \frac{1-\alpha}{\Gamma(2-\alpha)\cos(\pi\alpha/2)} & \text{if} \quad \alpha \neq 1 \\ \frac{2}{\pi} & \text{if} \quad \alpha = 1 \end{cases}.$$

Consequently, if $\alpha < 2$ a stable d.f. has heavy tails.

3.2 Sub-exponential tails

Some authors consider a class of heavy-tailed distributions, much larger than the class of regularly varying functions with a negative index of regular variation, the class of *sub-exponential* distribution functions:

Definition 13. *The d.f. F (or the r.v. X) is sub-exponential if*

$$\lim_{x\to\infty} \frac{P(X_1 + \cdots + X_n > x)}{P(\max(X_1, \cdots, X_n) > x)} = 1 \textit{ for some (equivalently all) } n \geq 2. \qquad (3.2)$$

Remark 6. *This is a much larger class of models, containing the class of regularly varying tails. Further examples of sub-exponential d.f.'s are the lognormal and the Weibull d.f.'s, $F(x) = 1 - \exp(-x^\beta), x \geq 0, \ \beta \in (0, 1)$.*

Remark 7. *Equation (3.2) has a possible physical interpretation: the sum of i.i.d. subexponential r.v.'s is likely to be large if and only if their maximum is. This fact accounts for large values in subexponential samples.*

For relations among different classes of heavy-tailed distributions see Embrechts et al. (1997), section 1.4. A full treatment of regular variation may be found in Bingham et al. (1987).

3.3 How to detect heavy tails?

We may use different graphical tools, usually based on the notion of regularly varying tails. We shall refer two of those methods:

1. The *log-log cumulative distribution* plot (LLCD plot) is frequently used. The idea is to plot $1-\widehat{F}_n$ on log-log scales, where $\widehat{F}_n$ is the empirical cumulative distribution function,

$$\widehat{F}_n(x) = \frac{1}{n}\sum_{i=1}^{n} I_{\{X_i \leq x\}}.$$

If the theoretical d.f. has a heavy right tail, for large n and moderately large values of x, the LLCD plot should consist of points randomly scattered around a straight line with slope equal to $-\alpha$. An estimate of α may be obtained through least squares regression.

2. The most frequently used method for estimating the tail parameter α is the Hill estimator

$$\widehat{\alpha}_n(k) := \left(\frac{1}{k}\sum_{i=1}^{k} \ln\frac{X_{n-i+1:n}}{X_{n-k:n}}\right)^{-1},$$

where $(X_{1:n} \leq X_{2:n} \leq \cdots \leq X_{n:n})$ denotes the sample of the ascending order statistics associated to our sample $(X_1, X_2, \cdots, X_n)$. In case of a heavy tail, the plot of $\ln k$ versus $\widehat{\alpha}_n(k)$ for small k will stabilize around a value $\alpha > 0$. For other alternatives to Hill's estimator of $\gamma = 1/\alpha$, which exhibit smaller bias for large k, enabling thus their plot as functions of k, see for instance Gomes and Martins (2002).

3.4 Long-range dependence

Whenever we are working with a weakly stationary (finite variance) stochastic process $\{X_t\}$, $t = 0, 1, 2, \cdots$, dependence between observations at times t and $t+k$ is usually measured by the autocorrelation function (ACF) $\rho(k)$ at lag k, given by

$$\rho(k) = \frac{Cov(X_t, X_{t+k})}{Var(X_t)} = \frac{Cov(X_0, X_k)}{Var(X_0)}, \; k = 0, 1, 2, \cdots . \tag{3.3}$$

The plotting of $\rho(k)$ versus k gives us an idea of the second order structure of the process. We obviously expect $|\rho(k)|$ to be decreasing in k. The notion of *LRD* (*long-range dependence*) depends on the size of $\rho(k)$ for large k. A possible definition of *LRD* (Beran, 1994) is,

Definition 14. *The stationary process $\{X_t\}$ exhibits LRD if, as $k \to \infty$, the ACF in* (3.3) *is such that*

$$\rho(k) \sim C\ k^{-\beta}, \tag{3.4}$$

where C is a positive constant and $\beta \in (0,1)$. Equivalently, we may say that $\sum\limits_k \rho(k)$ is not summable.

This property contrasts to *short-range dependent* processes like the *autoregressive* (AR) or more generally the *autoregressive moving average* ($ARMA$) processes, where $\rho(k)$ decays at an exponential rate. Popular processes, like the fractional ARIMA (Brockwell and Davis, 1991, section 13.2) and the *fractional Gaussian noise*, introduced later on, satisfy (3.4) with positive values $\rho(k)$.

3.4.1 Dectection of long-range dependence

Several heuristic graphical tools may be used to detect *LRD* in a time series. A statistical evaluation of these exploratory methods may be found in Taqqu and Teverovsky (1995, 1996). *LRD* seems to be present in most of the teletraffic data.

We shall here describe essentially two methods:

1. Suppose we have observed a stationary time series ($X_t,\ t = 1, 2, \cdots, n$). One way to detect *LRD* makes use of the sample autocorrelations

   $$\widehat{\rho}(k) = \frac{\widehat{\gamma}(k)}{\widehat{\gamma}(0)}, \quad \widehat{\gamma}(k) = \frac{1}{n}\sum_{t=1}^{n-k}\left(X_t - \overline{X}_n\right)\left(X_{t+k} - \overline{X}_n\right),\ 0 \le k \le n-1.$$

   If *LRD* is present, a plot $\ln\widehat{\rho}(k)$ agains $\ln k$ (*the log-log correlogram*) will provide points randomly scattered around a straight line with slope $-\beta$ for appropriate values of k and large n. A disadvantage of this method is that the estimate $\widehat{\rho}(k)$ is unreliable for large k with respect to n (see Brockwell and Davis, Section 7.2)

2. The oldest and perhaps better known *LRD* technique detection is the R/S method. This method has its origins in the field of hydrology and was first used by Hurst when he discovered *LRD*-like features in the yearly minimal water levels of the Nile river (Hurst, 1951). For a stationary process ($X_t,\ t = 1, 2, \cdots, n$), let us consider partial sums $Y_k = \sum_{t=1}^{k} X_t$ and the sample variance $S_k^2 = \sum_{t=1}^{k} X_t^2/k - (Y_k/k)^2$. The R/S statistic is

   $$(R/S)(k) := \frac{1}{S_k}\left\{\max_{0\le t\le k}\left(Y_t - \frac{tY_k}{k}\right) - \min_{0\le t\le k}\left(Y_t - \frac{tY_k}{k}\right)\right\}.$$

   It can be shown that if X_t is Gaussian, stationary, ergodic and (3.4) holds, then, with $H = 1 - \beta/2$,

   $$k^{-H}\ \left(R/S\right)(k) \xrightarrow{d} Z,\ \text{as } k \to \infty, \tag{3.5}$$

where Z is a non-degenerate r.v. (see Mandelbrot, 1975, Theorems 5 and 11, as well as Taqqu, 1975). The parameter H is the so-called *Hurst parameter* and is frequently used as a measure of strength of the LRD present in the data: H close to 1 corresponds to a strong presence of LRD. For various short memory processes, (3.5) holds with $H = 1/2$. For fractional Gaussian noise and fractional ARIMA (with gaussian innovations),

$$\mathbb{E}(R/S)(k) \sim c\ k^H, \text{ as } k \to \infty \quad (c > 0). \tag{3.6}$$

For an observed time series $(X_t,\ t = 1, 2, \cdots, n)$ one can try to verify (3.6) as follows:

- partition the series in $[n/m]$ blocks of size m;
- then, for each k, compute $(R/S)_{m_i}(k)$, starting at points $m_i = im+1$, $i = 0, 1, \cdots$, such that $m_i + k \leq n$;
- for values of $k \leq m$ we get $[n/m]$ different estimates of $(R/S)(k)$. For values of k approaching n, we get fewer values, as few as 1 as $k \geq n - m$;
- next plot $\ln(R/S)_{m_i}(k)$ against $\ln k$ and get, for each k several values on the plot;
- for large k, the points shoud lie around a straight line with slope H.

Another popular method to detect LRD (see Brockwell and Davis, 1975, section 10.3) is based on the *periodogram*

$$I(\lambda) = \frac{1}{2\pi\ n} \left| \sum_{t=1}^{n} X_t\ e^{-it\lambda} \right|, \quad \lambda \in [-\pi, \pi]. \tag{3.7}$$

It is common use to evaluate the periodogram at the *Fourier frequencies* $\lambda_j = 2\pi j/n,\ j = -[(n-1)/2], \cdots, [n/2]$. If LRD is present, a plot of $\ln I(\lambda_j)$ against $\ln(\lambda_j)$ would result in an approximately straight line with slope $\beta - 1$ for small frequencies λ_j.

3.4.2 Long-range dependence or non-stationarity?

It is clear that the concept of LRD, as given in (3.4), applies only to stationary processes, and we may say that no general test for the stationarity of an observed time series is indeed available in the literature. More than that: the graphical methods used to detect LRD in a time series are not fully reliable, and it has been observed that non-stationarities, like shifts in the mean or a slowly decaying trend can also be the cause of slowly decaying autocorrelations. It seems sensible to assume that "traffic is stationary only over short periods": at different time scales different factors may induce non-stationarities in the measurement series. Usually a portion of measurements containing not more than one hour of data traffic is considered and the workload is defined per second. We shall come to this feature later on.

3.4.3 *ARIMA* models and long-range dependence

The class of auto-regressive moving averages (*ARMA*) processes is the class of models most frequently applied to time series that exhibit no apparent deviations from stationarity and have rapidly decreasing autocorrelations. *ARMA* processes are defined as follows: for non-negative integers p and q, let the polynomials ϕ and θ be given by

$$\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p \quad \text{and} \quad \theta(z) = 1 + \theta_1(z) + \cdots + \theta_q z^q.$$

Let us define the *backward shift operator* B, by

$$B^j X_t = X_{t-j}, \quad j = 0, 1, 2, \cdots .$$

Definition 15. *The process $\{X_t, t = \pm 1, \pm 2, \cdots\}$ is an ARMA(p, q) process if $\{X_t\}$ is stationary and if for every t*

$$X_t - \phi_1 X_{t-1} - \cdots - \phi_p X_{t-p} = Z_t + \theta_1 Z_{t-1} + \cdots + \theta_q Z_{t-q}, \tag{3.8}$$

where $\{Z_t, t = \pm 1, \pm 2, \cdots\}$ is a sequence of white noise, i.e., uncorrelated and identically distributed r.v.'s with mean 0 and variance σ_Z^2. Compactly we may write (3.8) *as*

$$\phi(B)X_t = \theta(B)X_t,$$

where ϕ is the autoregressive part and θ the moving average part of the process.

Notice that the *ARMA* process has mean equal to 0. We say that $\{X_t\}$ is an $ARMA(p, q)$ with mean μ if $\{X_t - \mu\}$ satisfies (3.8).

Definition 16. *An ARMA(p, q) process is called casual if the polynomials ϕ and θ have no common zeros and ϕ has no zeros inside or on the unit circle in the complex plane. A causal ARMA process can be written as an infinite moving average*

$$X_t = \left(\frac{\theta}{\phi}\right)(B) Z_t,$$

which is the unique stationary solution of (3.8)*.*

An *ARMA* process has *short memory* in the sense that the autocorrelation function $\rho(k)$ satisfies

$$|\rho(k)| \leq b a^k, \text{ for some constants } b \in (0, \infty) \text{ and } a \in (0, 1). \tag{3.9}$$

The *ARMA* class is very convenient for modelling stationary short memory time series. However, in practice, stationarity of a time series is not always observed. Often, the series is transformed to "make it look more stationary". A frequently applied method is differencing, i.e., instead of the original series one looks at the series

$$(1-B)X_t = X_t - X_{t-1},$$

which will usually remove a global linear trend. Differencing twice will remove second degree polynomials, and usually we stop here, because, on a finite interval, many functions can be well approximated by polynomials of a reasonably low degree.

Definition 17. *A process that, after differencing finitely many times, reduces to an ARMA process, is called an autoregressive integrated moving average (ARIMA) process. An ARIMA process that is ARMA(p, q) after d times differencing is denoted ARIMA(p,d,q), i.e., if $\{X_t\}$ is ARIMA(p,d,q) then $(1-B)^d X_t$ is ARMA(p,q).*

Note that if $d \geq 1$ the *ARIMA* process is not stationary, and may appear as a model for series exhibiting a "*LRD*" type property.

3.4.4 Fractional Brownian motion

The fractional Brownian motion is a stochastic process exhibiting *LRD* in the sense of (3.4):

Definition 18. *A process $\{\sigma_0 B_H(t),\ t \geq 0\}$ with $\sigma_0 > 0$ is called a fractional Brownian motion if*

1. $B_H(t) \sim N\left(\sigma^2 t^{2H}\right)$.
2. $\mathbb{C}ov\left(B_H(s), B_H(t)\right) = \frac{1}{2}\left(s^{2H} + t^{2H} - |t-s|^{2H}\right)$ *for some* $H \in (0,1)$.
3. B_H *has continuous sample paths a.s.*

Since fractional Brownian motion is a Gaussian process, its finite-dimensional distributions are completely determined by *1.* and *2.* Using *2.* it can be shown that for $s < t$,

$$\mathbb{V}ar(B_H(t) - B_H(s)) = \mathbb{V}ar(B_H(t-s)).$$

Hence, fractional Brownian motion has stationary increments. Notice that for $H = 1/2$, B_H is an ordinary Brownian motion. Fractional Brownian motions have continuous sample paths which become smoother as H increases.

The increment process $Y_t = B_H(t) - B_H(t-1),\ t = 1, 2, \cdots$, of a fractional Brownian motion is called a *fractional Gaussian noise.* It is a mean-zero stationary Gaussian process with ACF

$$\rho_Y(k) = \frac{1}{2}\left(|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}\right), \quad k = 0, 1, 2, \cdots.$$

For $H = 1/2$, Y is a sequence of i.i.d. $N(0, \sigma^2)$ variables. If $H \neq 1/2$, Y is a dependent sequence. It has been shown (Samorodnitsky and Taqqu, 1994, Proposition 7.2.10) that for $H \neq 1/2$, as $k \to \infty$,

$$\rho_Y(k) \sim H(2H-1)k^{2H-2}.$$

Hence, for $H \in (1/2,\ 1)$ the fractional Gaussian noise exhibits *LRD*. If $H \in (0, 1/2)$ the autocorrelations $\rho_Y(k)$ are absolutely summable, i.e., the process has short memory.

3.5 Self-similarity

3.5.1 Distributional self-similarity

Definition 19. *A stochastic process $\{X_t\}_{t \geq 0}$ is said to be self-similar if the finite-dimensional distributions of $\{X_{at}\}$ and $\left\{a^H X_t\right\}$ are identical for any $a > 0$ and some $H \in (0,1)$, i.e., if*

$$X_{at} \overset{d}{=} a^H X_t \quad \forall t \geq 0. \tag{3.10}$$

The parameter H is called the index of self-similarity.

The concept of self-similarity became popular due to the work of Mandelbrot and van Ness (1968). A thorough mathematical description of self-similarity is given by Samorodnitsky and Taqqu (1994).

An important stochastic processes in this area is the *stable Lévy motion*, directly related to a *stable d.f.*:

Definition 20. *A process $\{\Lambda_{\alpha,\sigma,\beta}(t),\ t \geq 0\}$ is a α-stable Lévy motion if*

1. *$\Lambda_{\alpha,\sigma,\beta}$ has independent increments.*

2. *$\Lambda_{\alpha,\sigma,\beta}$ has stationary increments.*

3. *$\Lambda_{\alpha,\sigma,\beta} \overset{d}{\frown} S_\alpha(\sigma t^{1/\alpha}, \beta, 0)$, with S_α a stable d.f. for sums for some $\alpha \in (0, 2]$, $\beta \in [-1, 1]$ and $\sigma > 0$.*

4. *$\Lambda_{\alpha,\sigma,\beta}$ has right-continuous sample paths a.s.*

Examples.

(a) The α-*stable Lévy motion* is self-similar with $H = 1/\alpha$.

(b) The fractional Brownian motion is self-similar with index H and, obviously, the Brownian motion is self-similar with $H = 1/2$.

3.5.2 Second-order self-similarity

A different notion of self-silimarity is given by Cox (1984): given a stochastic process $\{X_t,\ t \geq 0\}$, consider its averaging through adjacent blocks of size m, i.e., consider

$$X_t^{(m)} := \frac{1}{m}\left(X_{tm-m+1} + \cdots + X_{tm}\right), \quad m \geq 1.$$

Definition 21. *The weakly stationary stochastic process $\{X_t,\ t \geq 0\}$ is second-order self-similar if X and $X^{(m)}$ have the same ACF for every $m \geq 1$. If the autocorrelation structures are equal only when $m \to \infty$, the process is called asymptotically second-order self-similar.*

The fractional Gaussian noise is second-order self-similar. If the ACF shows a slow decay, in the sense of (3.4), the associated stochastic process is asymptotically second-order self-similar; this is the case of the fractional $ARIMA$.

3.5.3 Self-similarity "by picture"

It is sometimes tempting to argue that a sample path of a self-similar process, with index H, on [0,1] will look qualitatively as a sample path on [0,100] where the realizations are divided by 100^H. Notice however that self-similarity means that the distribution of the process is invariant under the transformation, and not necessarily the sample paths. For instance in Leland et al. (1994) and Willinger et al. (1995) a time series of measured packet arrivals per time unit in the Ethernet LAN at Bellcore is plotted on five different time scales, the time units ranging from 0.01 till 100 seconds. From the plots it can be seen that the relative variability of the arrival process remains roughly the same in four of the five plots. The authors conclude then that evidence has been found of "self-similarity" of the measured Ethernet traffic. However, the same conclusion can be drawn for an appropriate time series, which is not self-similar, in the way it is done by Stegeman (2001). This author starts with a realization from an $ARIMA$(1,1,1) model with $\phi_1 = 0.4$, $\theta_1 = -0.95$ and $\sigma_Z^2 = 800$, with a length $n = 10,000.000$. From this sequence he constructs four new sequences by taking sums over consecutive blocks of sizes $n = 10, 100, 1000$ and 10000, respectively. He then notices that the relative variability of the four plots remains roughly the same, but the realization is from an $ARIMA$ model, which is not self-similar. There may however exhist an explanation for this fact: the $ARIMA$ model is asymptotically second-order self-similar in the sense of Cox.

4 Modeling the workload in computer networks

At the application level, file sizes, connection lengths and transmission durations are found to be heavy-tailed. Also, the heavy-tailed file sizes are related to the *LRD* observed in packet inter-arrival times, silent times and packet sizes. For example, suppose a source is transmitting an extremely large size file to a destination host. Due to the observed heavy-tails the probability of extremely large files is non-negligible. Before transmission, the file is decomposed into small packets, on which the bandwidth of physical medium imposes a certain maximum packet size. Since the file is extremely long it is more efficient if it is decomposed into packets of this maximum size. Hence, a long stream of packets of the same size occurs. Moreover, if there is no interference from other transmissions, the inter-arrival times and silence times between the packets will also be the same. This explains how the transmission of extremely large files causes dependence over a long range of observations (i.e. *LRD*) in the sequences of packet inter-arrival times, silent times and packet sizes.

The idea of heavy tails as the cause of *LRD* in workload measurements has been captured in two popular models:

1. The *ON/OFF model* proposed by Willinger et al. (1995). Here, traffic is generated by M i.i.d. ON/OFF sources. If a source is ON it transmits data at unit rate (e.g. 1 byte per time unit). If it is OFF it remains silent.

   In this way, an individual ON/OFF source generates a binary ON/OFF process W_t, where
   $$W_t = \begin{cases} 1 & \text{if the source is } ON \text{ at time } t \\ 0 & \text{if the source is } OFF \text{ at time } t \end{cases}.$$
   The lengths of periods in which the source is ON, the ON-periods X_i, are independently drawn from a heavy-tailed distribution. Analogously, the OFF-periods Y_i are also heavy-tailed. The X and Y sequences are assumed to be independent.

   It has been shown by Heath et al. (1998) that the stationary version of the ON/OFF process W_t exhibits *LRD*. Moreover, since the M sources are independent, the sum of their ON/OFF processes, i.e., the total workload generated by the M sources, also exhibits *LRD*.

2. The *infinite source Poisson model*, sometimes called the *$M/G/\infty$ input model*. Here, the number of sources in the network is taken infinite. Traffic is generated by independent connections arriving according to a Poisson process, i.e., with exponential inter-arrival times. During a connection, traffic is generated at a unit rate. The lengths of the connections are independent and taken from a heavy-tailed distribution. Also, the

connection lengths are independent from the connection inter-arrival times. Cox (1984) shows that the workload process generated by this model exhibits LRD.

4.1 The ON/OFF model

Consider first a single ON/OFF source such as a workstation. During an ON-period, the source generates traffic at a constant rate 1, e.g., 1 byte per unit time. During an OFF-period, the source remains silent; we assign the value 0 to it. Let $X_0, X_1, X_2, \cdots$ be i.i.d. non negative rvs representing the lengths of the ON-periods, X_{on}, and $Y_0, Y_1, Y_2, \cdots$ be i.i.d. non-negative rvs representing the lengths of the OFF-periods, X_{off}. We also write

$$Z_i = X_i + Y_i, \ i \geq 0.$$

The X and Y sequences are supposed to be independent. For any df F we write $\overline{F} = 1 - F$. We denote F_{on}/F_{off} the distribution of ON/OFF periods, and we shall assume that

$$1 - F_{on} \in RV_{-\alpha_{on}} \quad \text{and} \quad 1 - F_{off} \in RV_{-\alpha_{off}}, \ \alpha_{on}, \alpha_{off} \in (1, 2).$$

Consequently, both distributions F_{on} and F_{off} have finite means μ_{on} and μ_{off}, respectively, but infinite variances.

It is often assumed that $\alpha := \alpha_{on} < \alpha_{off}$. And this assumption makes the results for Models *1.* and *2.* almost identical.

Consider the renewal sequence generated by the alternating ON- and OFF-periods. Renewals happen at the beginnings of the ON-periods, the interarrival distribution is the convolution $F_{on} * F_{off}$ and the mean inter-arrival time

$$\mu = \mathbb{E} Z_1 = \mu_{on} + \mu_{off}.$$

To make the renewal sequence stationary (see Resnick, 1992, page 224, for a definition), we need to introduce a delay rv T_0, which is independent of the X_i's and the Y_i's. A stationary version of the renewal sequence $\{T_n\}$ is then given by

$$T_0, \quad T_n = T_0 + \sum_{i=1}^{n} Z_i, \ n \geq 1. \tag{4.1}$$

One way to construct the delay variable T_0 is as follows. Let B, $X_{on}^{(0)}$ and $Y_{off}^{(0)}$ be independent r.v.s, independent of $\{X_n\}, \{Y_n\}$, such that B is Bernoulli with

$$P(B = 1) = \frac{\mu_{on}}{\mu} = 1 - P(B = 0),$$

and

$$P\left(X_{on}^{(0)} \leq x\right) = \frac{1}{\mu_{on}} \int_0^x \overline{F}_{on}(s)ds =: F_{on}^{(0)}(x),$$
$$P\left(Y_{off}^{(0)} \leq x\right) = \frac{1}{\mu_{off}} \int_0^x \overline{F}_{off}(s)ds =: F_{off}^{(0)}(x).$$

Define

$$T_0 = B\left(X_{on}^{(0)} + Y_{off}^{(0)}\right) + (1 - B)\, Y_{off}^{(0)}.$$

The renewal sequence (4.1) is then stationary.

The ON/OFF process W is a binary process with $W_t = 1$ if t is in an ON-period and $W_t = 0$ if t is in an OFF-period. The stationarity of the renewal sequence (4.1) implies strict stationarity of the process W with mean

$$\mathbb{E}W_t = P(W_t = 1) = \mu_{on}/\mu.$$

The precise rate of decay for $\gamma_W(k)$, the covariance function of the stationary process W, under the regular variation assumptions and $\alpha_{on} < \alpha_{off}$, is, as $k \to \infty$,

$$\gamma_W(k) := \mathbb{C}ov(W_t, W_{t+k}) \sim \frac{\mu_{off}^2 k^{-(\alpha-1)} L_{on}(k)}{(\alpha-1)\mu^3} = C\ k\ \overline{F}_{on}(k). \tag{4.2}$$

The process W exhibits LRD in the sense that

$$\sum_k |\gamma_W(k)| = \infty,$$

which is equivalent to (3.4). Intuitively, this can be explained in the following way: since the lengths of the ON and OFF periods follow a heavy tailed distribution, they can assume extremely large values with non-negligible probability. Such an extremely large ON or OFF periods may contain both $W(t)$ and $W(t+k)$, even if k is extremely large, yielding the non-negligible covariance in (4.2).

Now consider a superposition of M i.i.d. ON/OFF sources feeding a server, $\left(W_t^{(m)},\ m = 1, \cdots, M;\ t \geq 0\right)$. The number of active sources at time t, or equivalently, the total traffic at the network at time t, is

$$N(t) = M_M(t) = \sum_{m=1}^{M} W_t^{(m)}, \quad t \geq 0.$$

Note that $N(t)$ is the input rate to the server at time t and is usually referred to as the *workload processs.* Since the sources are i.i.d., (4.2) implies that N exhibits LRD in the spirit of (3.4), since the stationary version of N satisfies

$$\gamma_N(k) = \sum_{i=1}^{M} \gamma_{W^{(i)}}(k) = CMk\overline{F}_{on}(k).$$

The *cumulative input* of work to the server or *total accumulated work* by time t is

$$A(t) = A_{_M}(t) = \int_0^t N(s)ds, \quad t \geq 0.$$

The behaviour of the cumulative input process $A(t)$ for the superposition of a large number of i.i.d. ON/OFF sources has already been studied by Willinger at al. (1995) and Taqqu et al. (1997), and it has been found that the cumulative input process (properly normalized) of an increasing number of i.i.d. ON/OFF sources converges to a *fractional Brownian motion* in the sense of convergence of the finite dimensional distributions. Their result is formulated as a double limit: first, the number M of sources goes to infinity and then the time-scaling parameter T converges to infinity. This order of taking limits is crucial for obtaining fractional Brownian motion as limit. When limits are taken in the reversed order, the limits of the finite-dimensional distributions are those of an infinite variance stable Lévy motion. Indeed, Mikosch et al. (2002?) consider the ON/OFF model when M and T go simultaneously to infinity and they get, after adequate normalization, *α-stable Lévy motion.*

4.1.1 Local Area Networks (*LAN*s) traffic: the *ON/OFF* model and reality

The introduction of the ON/OFF model into the networking community has been accompanied by a detailed statistical analysis at the source level of traffic generated in the Ethernet LAN at Bellcore. ON and OFF periods are defined for the traffic between individual source-destination pairs. It is found, using the LLCD plot and the Hill plot that the distributions of the lengths of ON and OFF periods are heavy tailed, with parameter between 1 and 2. Also it is mentioned that no evidence is found for dependence in (or between) the sequences of ON and OFF periods. The other independence assumption, of the M sources in the model, is less likely to hold in a real-life network. Also, the ON/OFF model does not involve queuing or congestion control, which make it seem simplistic. However, this model seems to have been successful in capturing some of the characteristics of real-life LAN traffic.

4.2 The infinite source Poisson model

This is a common model of incoming traffic to a communication network. We have a homogeneous Poisson process on $[0, \infty)$ with rate λ, and let $T_i, \ i \geq 1$ denote the times of occurrence of such a process, so the $T_i - T_{i-1}, \ i \geq 1$, $(T_0 \equiv 0)$ are i.i.d. exponential r.v.'s with mean value $1/\lambda$. We imagine that a communication system has an infinite number of nodes, sometimes called *sources*. Suppose that at time T_i, some node turns on and begins a transmission, or a source starts a transmission to the server, and continues to transmit for a period of length L_i, at a possibly time varying rate. It is often assumed a constant rate, taken equal to unity, and we shall do it here. As said before, heavy tails are common for this type of data, i.e., it is sensible to assume that the tail function of L_i is regularly varying with index of regular variation equal to $-\alpha$, i.e.,

$$P(L_i > x) = 1 - F_{on}(x) = x^{-\alpha}\ L(x), \quad 1 < \alpha < 2,\ x > 0, \tag{4.3}$$

for some slowly varying function $L(x)$.

The first quantity of interest is $N(t)$, the number of active sources at time t. For each t, $N(t)$ is a Poisson r.v. with parameter $\lambda\mu_{on}$, where $\mu_{on} = \mathbb{E}L$. Due to the memoryless property of the exponential distribution and the independence between the Poisson process and the connection lengths, the process $N(t)$ is stationary. The process is considered on large time scales, i.e., for large T, we consider $N(t) = N_T(t)$, a family of Poisson processes with intensity $\lambda = \lambda T \to \infty$ as $T \to \infty$. The intensity $\lambda = \lambda T$ is often referred as the *connection rate*.

During a transmission, the transmitting node is sending data to the server at unit rate. The *total accumulated input* in $[0,t]$ for the T-th model is

$$A(t) = A_T(t) = \int_0^t N(s)ds.$$

Again, heavy tailed transmission time L_k induce *LRD* in N. It can be shown (Cox, 1984) that

$$\mathbb{C}ov(N(t), N(t+k)) = C\ k^{-(\alpha-1)}L(k),\ \text{as } k \to \infty.$$

Mathematically speaking: If $\overline{F}_{on}$ is regularly varying with index $-\alpha$, $1 < \alpha < 2$, a slow growth condition may be written as

$$\lim_{T\to\infty} \lambda T\overline{F}_{on}(T) = 0.$$

The fast growth condition may be written as

$$\lim_{T\to\infty} \lambda T\overline{F}_{on}(T) = \infty.$$

The cumulative input is well approximated by a *stable Lévy motion*, a process with independent increments, when the connection rate is *”slow”*, or equivalently when the dependence in the T-th model disappears as $T \to \infty$.

If the connection rate is *”fast”*, or equivalently the dependence in the T-th model remains strong as $T \to \infty$, the *fractional Brownian motion* is the adequate approximation.

4.2.1 Wide Area Networks (WANs) traffic: the infinite source Poisson model and reality

The construction of superimposed connections resembles traffic generation in a WAN: first a connection is set up, and afterwards data is transmitted. The assumptions of Poisson connection arrivals and heavy-tailed connection lengths are consistent with the findings of Paxson and Floyd (1995), in their analysis of WAN traffic. More recently, Guerin et al. (2000) consider HTTP sessions at different universities and find evidence for heavy-tailed connection lengths with tail parameter between 1 and 2. But the data are inconclusive regarding the assumption of exponential inter-arrival times. In general, they state that the infinite source Poisson model does not adequately describe some of the datasets considered. They refer to the assumption of a constant transfer rate as the center of the problem. Indeed, their measurements show widely varying transfer rates. The assumption of a constant transfer rate has been relaxed in Kurtz (1996) and in several recent papers, among which we refer Resnick and van den Berg (2000). Resnick and Rootzén (2000) consider the infinite source Poisson model for very heavy-tailed connection lengths, with tail parameter between 0 and 1.

References

[1] Beran, J. (1994). *Statistics for Long-Memory Processes.* Chapman and Hall.

[2] Binghman, N. H. Goldie, C. M. and J. Teugels 81987). *Regular Variation.* Cambridge Univ. Press.

[3] Brockwell, P. J. and R. A. Davis (1991). *Time Series: Theory and Methods*, 2nd edition. Springer-Verlag.

[4] Cox, D. R. (1984). Long-range dependence: a review. In David; H.A. and H.T. David (eds.). *Statistics: An Appraisal.* Iowa State University Press, 55-74.

[5] Embrechts, P. Kluppelberg, C. and T. Mikosch (1997). *Modelling Extremal Events for Insurance and Finance.* Springer-Verlag.

[6] Fowler, and Leland (1991). Local area network traffic characteristics, with implications for broadband network congestion management. *IEEE J. on Selected Areas in Communications* **9**, 1139-1149.

[7] Gomes, M. I. and M. J. Martins (2002). "Asymptoticallt unbiased" estimators of the tail index based on the second order parameter. *Extremes* **5**:1, 5-31.

[8] Gut, A. (1995). *An Intermediate Course in probability.* Springer-Verlag.

[9] Hurst, H. E. (1951). Long-term storage capacity of reservoirs. *Trans. American Soc. Civil Engineers* **116**, 770-799.

[10] Leland, W., Taqqu, M. S., Willinger, W. and D. Wilson (1994). On the self-similar nature of Ethernet traffic. *IEEE/ACM Transactions on Networking* **2**, 1-15.

[11] Mandelbrot, B. B. (1975). Limit theorems on the self-normalized range for weakly and strongly dependent processes. *Zeitschrift für Wahrscheinlichkeitstheorie and verwandte Gebiete* **31**, 271-285.

[12] Mandelbrot, B. B. and J. W. van Ness (1968). Fractional Brownian motions, fractional noises and applications. *SIAM Review* **10**, 422-437.

[13] Mikosch, T., Resnick, S., Rootzén, H. and A. W. Stegeman (2002?) Is network traffic approximated by stable Lévy motion or fractional Brownian motion? *Ann. Appl. Probab.*

[14] Paxson, V. and S. Floyd (1995). Wide area traffic: the failure of Poisson modeling. *IEEE and ACM Transactions on Networking* **3**, 226-244.

[15] Resnick, S. (1992). *Adventures in Stochastioc Processes.* Birkhauser.

[16] Resnick, S. I. (1997). Heavy tail modelling and teletraffic data. *Annals of Statistics* **25**, 1805-1869.

[17] Resnick, S. and E. van den Berg (2000). Weak convergence of high-speed netwoek traffic models. *J. Applied Probab.* **37**, 575-597.

[18] Resnick, S. and H. Rootzén (2000). Self-similar communication models and very heavy tails. *Ann. Applied Probab.* **10**, 753-778.

[19] Samorodnitsky, G. and M. S. Taqqu (1994). *Stable Non-Gaussian Random Processes. Stochastic Models with Infinite Variance.* Wiley, New York.

[20] Stegeman, A. (2002). *The Nature of the Beast — Analyzing and Modeling Computer Network Traffic.* Rijksuniversiteit Groningen.

[21] Taqqu, M. S. (1975). Weak convergence of fractional Brownian motion to the Rosenblatt process. *Zeitschrift für Wahrscheinlichkeitstheorie and verwandte Gebiete* **31**, 287-302.

[22] Taqqu, M. S. and V. Teverovsky (1995). Estimators for long-range dependence: an empirical study. *Fractals* **3**, 785-798.

[23] Taqqu, M. S. and V. Teverovsky (1996). On estimating the intensity of long-range dependence in finite and infinite variance time-series. In Adler et al. 8eds.). *A Practical Guide to heavy tails: Statistical Techniques and Applications.* Birkhauser, 177-217.

[24] Taqqu, M., Willinger, W. and R. Sherman (1997). Proof of a fundamental result in self-similar traffic modelling. *Computer Comm. Review* **27**, 5-23.

[25] Taylor, H. M. and S. Karlin (1998). *An Introduction to Stochastic Modeling*, 3rd ed.. Academic Press.

[26] Willinger, W., Taqqu, M. S., Sherman, R. and D. V. Wilson (1995). self-similarity through high-variability: statistical analysis of Ethernet *LAN* traffic at the source level. *Computer Commun. Review* **25**, 100-113.

[27] Willinger, W. and V. Paxson (1998). Where mathematics meet the internet. *Notices of the Americam Mathematical Soc.* **45**, 961-970.

Transforms, Algorithms and Applications

M. J. Soares*

Abstract

Fourier transforms and other related transforms are an essential tool in applications of science, engineering and technology. In fact, much of the work currently being done in mathematics, physics and engineering has its roots in Fourier's pioneering idea of representing an arbitrary function as the sum of a trigonometric series. The main purpose of these notes is to give a brief overview of some Fourier-related transforms, namely: continuous Fourier transform, Fourier series, discrete Fourier transform, fast Fourier transform (FFT), sine and cosine transforms, Z-transform, Laplace transform, windowed Fourier transform, continuous and discrete wavelet transforms. Our aim is simply to present a summary of these transforms and to describe their main properties and possible applications, and so most of the results are presented with no proof. References containing the proofs and other details about the transforms are always indicated.

Keywords : Fourier transforms, Fourier series, FFT, wavelet transforms.

1 Notations

We start by introducing the main notations that will be used throughout these notes.

• If X is a measurable subset of the real line $\mathbb{R}$, in particular the whole of $\mathbb{R}$, we denote by $L^p(X)$ $(0 < p < \infty)$, the Banach space of the (equivalence classes of) measurable functions f defined in X such that

$$\|f\|_p := \left(\int_X |f(t)|^p dt \right)^{1/p} < \infty. \tag{1}$$

When $p = 2$, this is a Hilbert space with respect to the inner product

$$\langle f, g \rangle := \int_X f(t)\overline{g(t)}dt. \tag{2}$$

(Here and throughout, $\overline{u}$ denotes the complex conjugate of u.)

• When X is a finite interval $X = [a, a + \Omega]$ of length Ω, $\Omega > 0$, we can identify the above space with the space of functions which are periodic of period Ω, i.e. satisfy $f(t + k\Omega) = f(t)$,

*DM Universidade do Minho. E-mail:jsoares@math.uminho.pt

for all $k \in \mathbb{Z}$ and for almost all t, and are such that $\int_a^{a+\Omega} |f(t)|^p dt < \infty$. In fact, any Ω-periodic function is totally determined by its behaviour on any interval of length Ω and, reciprocally, any function which is only defined on an interval of length Ω can always be periodically extended (with period Ω) to the whole line. We can also think of this space as a space of functions defined on the Ω-torus $\mathbb{T}_\Omega = \mathbb{R}/\Omega\mathbb{Z}$; see Section IV if you are unfamiliar with this type of notation. In this case, it is more convenient to normalize the inner product (2) as

$$\langle f, g\rangle = \frac{1}{\Omega}\int_a^{a+\Omega} f(t)\overline{g(t)}dt. \tag{3}$$

The norm $\|.\|_p$ will also be redefined as

$$\|f\|_p := \left(\frac{1}{\Omega}\int_a^{a+\Omega} |f(t)|^p dt\right)^{1/p}. \tag{4}$$

In order to simplify the notation, we will always write $\int_{\mathbb{T}_\Omega}$ to designate $\frac{1}{\Omega}\int_a^{a+\Omega}$. This means, for example, that the inner product (3) will be written simply as

$$\langle f, g\rangle = \int_{\mathbb{T}_\Omega} f(t)\overline{g(t)}dt. \tag{5}$$

• When X is the discrete set $\mathbb{Z}$, the functions defined on X will simply be two-sided sequences, and we use for them a notation of the type $f = (f[k])_{k\in\mathbb{Z}}$, following the tradition of signal processing literature of using square brackets around a discrete variable. In this case, the integrals in (1) and (2) should be understood with respect to the *discrete measure*, i.e. the norm and inner product are defined, respectively, by

$$\|f\|_p := \left(\sum_{k\in\mathbb{Z}} |f[k]|^p\right)^{1/p} \tag{6}$$

and

$$\langle f, g\rangle := \sum_{k\in\mathbb{Z}} f[k]\overline{g[k]}. \tag{7}$$

These spaces are referred to as the spaces of *p-summable sequences* and denoted by $\ell^p(\mathbb{Z})$.

• Finally, when the set X is discrete and finite, e.g. $X = \{0, 1, \ldots, N-1\}$, the functions on X, which are simply vectors $f = f([k])_{k=0}^{N-1}$, can also be “viewed” as N-periodic sequences on $\ell^p(\mathbb{Z})$ (any p) if we define, for $k \in \mathbb{Z}$, $f[k] = f[k \bmod N]$, where k mod N denotes the remainder of the division of k by the integer N. This space can be identified with the space $\ell(\mathbb{Z}_N)$ with $\mathbb{Z}_N = \mathbb{Z}/N\mathbb{Z}$; more details, again, in Section IV. Here, naturally, the inner product and norm are the usual Euclidean inner product and norm of vectors in $\mathbb{C}^N$, i.e. they are, respectively

$$\langle f, g\rangle := \sum_{k=0}^{N-1} f[k]\overline{g[k]} \tag{8}$$

and

$$\|f\| := \{\sum_{k=0}^{N-1} |f[k]|^2\}^{1/2}. \tag{9}$$

• A family $\{e_k : k \in \mathbb{Z}\}$ of elements in a Hilbert space H (with inner product $\langle \cdot, \cdot \rangle$ and corresponding norm $\| \cdot \|$) is said to be an *orthogonal basis* of H if it satisfies:

1. $\langle e_i, e_j \rangle = 0, \quad i \neq j$;

2. for any $x \in H$, there is a unique sequence of scalars $\hat{x}[k]$ such that

$$\lim_{N\to\infty} \|x - \sum_{k=-N}^{N} \widehat{x}[k] e_k\| = 0. \tag{10}$$

The orthogonality condition implies that the coefficients $\hat{x}[k]$ are necessarily given by

$$\hat{x}[k] = \frac{\langle x, e_k \rangle}{\|e_k\|^2},$$

and we will write (10) simply as

$$x = \sum_{k\in\mathbb{Z}} \frac{\langle x, e_k \rangle}{\|e_k\|^2} e_k. \tag{11}$$

When each vector e_k has unit norm, the basis is said to be *orthonormal* (o.n.). In this case, *Plancherel formula*, which sates an energy conservation, holds:

$$\|x\|^2 = \sum_{k\in\mathbb{Z}} | \langle x, e_k \rangle |^2. \tag{12}$$

We will frequently refer to *Fourier transform* to designate several different mathematical transformations, depending on the nature of the spaces on which they are defined (in other words, depending on the type of *signals* on which they are acting). When necessary, we will be more specific and use terms like *continuous time Fourier transform*, *continuous time Fourier series*, etc. A small table summarizing the Fourier transforms for various settings is given below.

Table of Fourier Transforms

Name	**Domain** of f	**Transform** $F = \hat{f}$ (and Inverse)	**Domain** of $\hat{f}$
CTFT	$\mathbb{R}$	$\hat{f}(\omega) = \int_{\mathbb{R}} f(t)e^{-2\pi i\omega t}dt$ $f(t) = \int_{\mathbb{R}} \hat{f}(\omega)e^{2\pi it\omega}d\omega$	$\mathbb{R}$
CTFS	$\mathbb{T}_\Omega$	$\hat{f}[k] = \int_{\mathbb{T}_\Omega} f(t)e^{-2\pi ikt/\Omega}dt$ $f(t) = \sum_{k\in\mathbb{Z}} \hat{f}[k]e^{2\pi ikt/\Omega}$	$\mathbb{Z}$
DTFT	$\mathbb{Z}$	$\hat{f}(\omega) = \sum_{k\in\mathbb{Z}} f[k]e^{-2\pi ik\omega/\Omega}$ $f[k] = \int_{\mathbb{T}_\Omega} \hat{f}(\omega)e^{2\pi ikt\omega/\Omega}d\omega$	$\mathbb{T}_\Omega$
DTFS	$\mathbb{Z}_N$	$\hat{f}[n] = \sum_{k=0}^{N-1} f[k]e^{-2\pi ikn/N}$ $f[k] = \frac{1}{N}\sum_{n=0}^{N-1} \hat{f}[n]e^{2\pi ikn/N}$	$\mathbb{Z}_N$

CT-continuous time; DT-discrete time; FT-Fourier transform; FS-Fourier series

In Section IV, we wil give a more unified view of these different transforms, briefly describing how they all fit in the more general framework of *Fourier transforms on groups*. For the moment, we will study in more detail each of the above transforms, discussing, in particular the conditions (and the different interpretations) for the *inverse formulas* to hold.

2 Continuous Time Fourier Transform (CTFT)

2.1 Fourier transform in $L^1(\mathbb{R})$

We start by defining the Fourier transform of functions in the space $L^1(\mathbb{R})$.

The Fourier transform (also called *continuous-time Fourier transform* or *integral Fourier transform*) of a function $f \in L^1(\mathbb{R})$ is the function $\hat{f}$ defined by

$$\hat{f}(\omega) := \int_{\mathbb{R}} f(t)e^{-2\pi i\omega t}dt, \quad \omega \in \mathbb{R}. \tag{13}$$

For simplicity, to indicate the correspondence between a function f and its Fourier transform, we use the notation $f \longrightarrow F$.

We consider the following three operators, defined for $a \in \mathbb{R}$:

Translation: $T_a f(t) = f(t-a)$

Modulation: $E_a f(t) = e^{2\pi iat} f(t)$

Dilation: $D_a f(t) = |a|^{-1/2} f(t/a), \ (a \neq 0)$.

The main algebraic and analytic properties of the Fourier transform are summarized in the following two theorems; the proofs can be seen, e.g. in [1].

Theorem 1

1. Linearity $c_1 f_1 + c_2 f_2 \longrightarrow c_1 F_1 + c_2 F_2$.
2. Conjugation $\overline{f}(t) \longrightarrow \overline{F}(-\omega)$.
3. Time shifting $T_a f \longrightarrow E_{-a} F$.
4. Modulation $E_a f \longrightarrow T_a F$.
5. Time dilation $D_a f \longrightarrow D_{1/a} F$.

Theorem 2

Let $f \in L^1(\mathbb{R})$ and let F be its Fourier transform. Then, we have

1. Boundedness *For each $\omega \in \mathbb{R}$, $|F(\omega)| \leq \|f\|_1$.*
2. Continuity *F is (uniformly) continuous on $\mathbb{R}$.*
3. Riemann-Lebesgue Lemma $\lim\limits_{|\omega| \to \infty} F(\omega) = 0$.
4. Time differentiation *Let $f \in C^m(\mathbb{R}) \cap L^1(\mathbb{R})$ be such that $f^{(k)}; k = 1\ldots, m$, are in $L^1(\mathbb{R})$. Then*
$$f^{(k)}(t) \longrightarrow (2\pi i\omega)^k F(\omega).$$
5. Frequency differentiation *Suppose that $t^m f(t) \in L^1(\mathbb{R})$. Then, $F^{(k)}; k = 1, \ldots, m$, exist and*
$$(-2\pi it)^k f(t) \longrightarrow F^{(k)}(\omega).$$

Another important property of Fourier transform is its behaviour with respect to *convolution*. Recall that the *convolution* $f * g$ of two functions f and g is the function defined by

$$f * g(t) = \int_{\mathbb{R}} f(u) g(t-u) du. \tag{14}$$

We then have the following result:

Theorem 3 (Convolution) *If $f, g \in L^1(\mathbb{R})$, then $f * g \in L^1(\mathbb{R})$ and*

$$f * g \longrightarrow FG.$$

2.2 Inversion

Given a function $g \in L^1(\mathbb{R})$, we define its *inverse Fourier transform* $\check{g}$ by

$$\check{g}(t) := \int_{\mathbb{R}} g(\omega) e^{2\pi i \omega t} d\omega, \quad t \in \mathbb{R},$$

i.e. $\check{g}(t)$ is simply $\hat{g}(-t)$. The name *inverse* Fourier transform is justified by the following theorem, which shows that the function f can be recovered from its Fourier transform, by applying to it the inverse Fourier transform.

Theorem 4 *Let $f \in L^1(\mathbb{R})$ and let $\hat{f}$ denote its Fourier transform. If $\hat{f} \in L^1(\mathbb{R})$, then f is continuous and $f = \check{\hat{f}}$, i.e.*

$$f(t) = \int_{\mathbb{R}} \hat{f}(\omega) e^{2\pi i \omega t} d\omega. \tag{15}$$

Note: This theorem establishes a pointwise inversion formula for the Fourier transform under the assumption that $\hat{f} \in L^1(\mathbb{R})$. It should be interpreted in the following sense: the integral on the r.h.s. is defined for every $t \in \mathbb{R}$ and defines a continuous function which coincides with f almost everywhere (a.e.); the pointwise equality is valid for the continuous representative of f.

2.3 Fourier transform in $L^2(\mathbb{R})$

The formula (13) as it stands can not be applied directly to functions in the space $L^2(\mathbb{R})$ (if they are not in $L^1(\mathbb{R})$), so the definition of the Fourier transform for functions in this space (the important space of signals of finite energy) has to be suitably adapted.

The following result is essential for establishing a natural definition for the Fourier transform in $L^2(\mathbb{R})$.

Theorem 5 (Plancherel-Parseval) *If $f, g \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$, then*

$$\langle f, g \rangle = \langle \hat{f}, \hat{g} \rangle \quad (\textit{Parseval identity}). \tag{16}$$

In particular, we have

$$\|f\| = \|\hat{f}\| \quad (\textit{Plancherel formula}). \tag{17}$$

The extension of the Fourier transform to $L^2(\mathbb{R})$ is based on the use of the above formulae and the fact that $L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ is *dense* in $L^2(\mathbb{R})$. This means that, given a function $f \in L^2(\mathbb{R})$, there is a sequence of functions $(f_n)_{n \in \mathbb{N}}$ in $L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ converging to f (with convergence taken with respect to the norm in $L^2(\mathbb{R})$). This implies that $\|f_m - f_n\|_2 \to 0$ when $m, n \to \infty$. By the linearity of the Fourier transform and the Plancherel formula, we immediately conclude that the sequence $(\widehat{f_n})_{n\in\mathbb{Z}}$ converges to a certain function in $L^2(\mathbb{R})$. This limit function will be called the Fourier transform of f (sometimes called the Plancherel transform) and will also be denoted, as before, by $\hat{f}$ or F. It can be shown that the limit function F does not depend on the choice of the sequence f_n converging to f and, naturally, that it coincides with the usual Fourier transform of f when $f \in L^1(\mathbb{R})$. A standard way of selecting the sequence f_n is to take $f_n = f\,\mathbf{1}_{[-n,n]}$, where $\mathbf{1}_{[a,b]}$ denotes the characteristic function of the interval $[a, b]$, i.e.

$$\mathbf{1}_{[a,b]}(t) = \begin{cases} 1, & t \in [a, b] \\ 0, & \text{otherwise.} \end{cases}$$

If we write l.i.m.$g_n(t) = g(t)$ to indicate that $\|g_n - g\|_2 \to 0$ when $n \to \infty$, we can thus write, for $f \in L^2(\mathbb{R})$,

$$\hat{f}(\omega) := \text{l.i.m.} \int_{-n}^{n} f(t)e^{-2\pi i\omega t}dt. \tag{18}$$

Note: With a convenient abuse of notation we will still write, when $f \in L^2(\mathbb{R})$,

$$\hat{f}(\omega) = \int_{\mathbb{R}} f(t)e^{-2\pi i\omega t}dt,$$

with the understanding that this is a limiting process as defined above.

It is important to observe that the main properties stated for the Fourier transform of functions in $L^1(\mathbb{R})$ also hold for this extension to $L^2(\mathbb{R})$. The extension of the definition of the inverse Fourier transform $\check{g}$ to functions $g \in L^2(\mathbb{R})$ is, naturally, done in manner analogous to the process described for the Fourier transform, and we also have an inversion theorem for this case.

Theorem 6 (Inversion in $L^2(\mathbb{R})$) *The Fourier transform is a bijective linear operator from $L^2(\mathbb{R})$ into $L^2(\mathbb{R})$. Given $f \in L^2(\mathbb{R})$, we have*

$$f = \check{\hat{f}}.$$

The definition of the Fourier transform can also be extended to a wider class of "objects", the so-called *tempered distributions*; as an example of a tempered distribution we have the Dirac-delta δ. This is a linear functional which acts on a (sufficiently well-behaved function) f by giving its value at zero, i.e.

$$\delta(f) := f(0).$$

The Fourier transform of a tempered distribution is another tempered distribution. In the case of the Dirac-delta, the Fourier transform can be identified with the constant function 1, i.e

$$\hat{\delta} = 1.$$

For more details on Fourier transforms of tempered distributions, see, e.g. [2] or [1].

3 Continuous Time Fourier Series (CTFS)

We now consider the case where the function f to be transformed is in $L^2(\mathbb{T}_\Omega)$, where $\mathbb{T}_\Omega = \mathbb{R}/\Omega\mathbb{Z}$ is the Ω-torus ($\Omega > 0$). It can be shown that the set of functions

$$\gamma_k(t) := e^{2\pi ikt/\Omega}, \quad k \in \mathbb{Z}, \tag{19}$$

is an orthonormal basis of $L^2(\mathbb{T}_\Omega)$ (with respect to the inner product defined by (5)). This means that every function $f \in L^2(\mathbb{T}_\Omega)$ can be written as

$$f(t) = \sum_{k\in\mathbb{Z}} \hat{f}[k] e^{2\pi ikt/\Omega}, \tag{20}$$

where the coefficients $\hat{f}[k]$ are given by

$$\begin{aligned} \hat{f}[k] =& \langle f, \gamma_k \rangle \\ =& \int_{\mathbb{T}_\Omega} f(t) e^{-2\pi ikt/\Omega} dt. \end{aligned} \tag{21}$$

The coefficients $\hat{f}[k],\ k \in \mathbb{Z}$ given by (21), are called the *Fourier coefficients* of the function f and the series on the r.h.s. of (20) is the called the *Fourier series of* f.

The equality (20) is to be interpreted as (cf. 10)

$$\lim_{N\to\infty} \int_0^\Omega |f(t) - \sum_{k=-N}^{N} \hat{f}[k] e^{2\pi ikt/\Omega}|^2 dt = 0$$

and does not necessarily mean that, for every $t \in \mathbb{R}$, the series on the r.h.s. of (20) converges to the value $f(t)$. The problems associated with the pointwise (and uniform) convergence of Fourier series, namely the discussion of the minimum conditions which ensure this type of convergence, have attracted the attention of mathematicians for more than two centuries and had a profound impact on the evolution of the foundations of Analysis; an accessible reference on this subject, with an interesting historical perspective, is [3].

The equality (20) is also known to hold for almost all t; moreover, if the function f is sufficiently well-behaved (e.g. piecewise smooth) then the series converges, at every point t, to the average value

$$\frac{f(t^+) + f(t^-)}{2}.$$

The Fourier series has a typical behaviour near the points of discontinuity; its partial sums overshoot and undershoot the true values $f(t^+)$ and $f(t^-)$, respectively, by about 9% of the total jump $f(t^+) - f(t^-)$. This is the famous *Gibbs phenomenon*, and was observed by Gibbs, for a particular function, in a letter to *Nature* (vol.59, p.606), in 1899.

Note: In fact, this phenomenon had already been described by H. Wilbraham, 51 years earlier [4], although Gibbs was not aware of this. In 1906, M. Bôcher, an American mathematician, proved that this behaviour is a general property of Fourier series in the vicinity of a jump discontinuity; [5].

The computation of the sequence of the Fourier coefficients $\hat{f}[k]$ in the case where f is a periodic function can be seen as the analogue of the computation, for a function f with no periodicity, of $\hat{f}(\omega)$, for all $\omega \in \mathbb{R}$. This corresponds, in both cases, to the *analysis* phase of the given signal f; the inversion formula (15) and the series expansion (20) then correspond to the *synthesis* or *reconstruction* phase of the signal.

Since $\gamma_k(t) = e^{2\pi ikt/\Omega}$ form an orthonormal basis of $L^2(\mathbb{T}_\Omega)$, Pareseval's identity gives us

$$\sum_{k\in\mathbb{Z}} |\langle f, \gamma_k\rangle|^2 = \sum_{k\in\mathbb{Z}} |\hat{f}[k]|^2 = \|f\|_2^2. \tag{22}$$

It is also important to state the following result (which should be compared with the result 3. in Theorem 2).

Lemma 1 (Riemann-Lebesgue) *If $f \in L^2(\mathbb{T}_\Omega)$, then its Fourier coefficients $\hat{f}[k]$ satisfy*

$$\lim_{|k|\to\infty} \hat{f}[k] = 0. \tag{23}$$

Note: The above result is also valid for functions in the wider class $L^1(\mathbb{T}_\Omega.)$

The analogies between the Fourier transforms and series can also be extended to results on convolutions, provided an appropriate definition for convolution is given. Given two functions $f, g \in L^1(\mathbb{T}_\Omega)$ we define its convolution as

$$f * g(t) = \int_{\mathbb{T}_\Omega} f(u)g(t-u)du.$$

We then have the following result (cf. Theorem 3).

Theorem 7 *Let $f, g \in L^1(\mathbb{T}_\Omega)$, with corresponding sequences $(\hat{f}[k])_{k\in\mathbb{Z}}$ and $(\hat{g}[k])_{k\in\mathbb{Z}}$ of Fourier coefficients. Then, $f * g \in L^1(\mathbb{T}_\Omega)$ and the sequence of its Fourier coefficients is the product of the two sequences $(\hat{f}[k])_{k\in\mathbb{Z}}$ and $(\hat{g}[k])_{k\in\mathbb{Z}}$, i.e.*

$$\widehat{f * g}\,[k] = \hat{f}[k]\,\hat{g}[k], \quad k \in \mathbb{Z}.$$

4 Discrete Time Fourier Transform (DTFT)

The equality (22) shows that, given a function in $L^2(\mathbb{T}_\Omega)$, the sequence of its Fourier coefficients is in the space $\ell^2(\mathbb{Z})$. One can also "move" the other way around. Let $f = (f[k])_{k\in\mathbb{Z}}$ be a given sequence in $\ell^2(\mathbb{Z})$. Then, for any chosen $\Omega > 0$, the trigonometric series

$$\sum_{k\in\mathbb{Z}} f[k]e^{-2\pi ik\omega/\Omega} \tag{24}$$

converges (with respect to the $\|\cdot\|_2$ norm defined by (4)), to a certain function in the space $L^2(\mathbb{T}_\Omega)$. We call this function the *discrete time Fourier transform* (corresponding to Ω) of the sequence $f = (f[k])$ and denote it by $\hat{f}(\omega)$. That is, we have

$$\hat{f}(\omega) = \sum_{k\in\mathbb{Z}} f[k]e^{-2\pi ik\omega/\Omega}. \tag{25}$$

One can show that the Fourier coefficients of this function $\hat{f}$ are precisely the given numbers $f[k]$, that is, we have

$$\int_{\mathbb{T}_\Omega} \hat{f}(\omega)e^{2\pi ik\omega/\Omega}d\omega = f[k], \tag{26}$$

which can be seen as an inversion result. The equality (25) is also known to hold for almost all ω. Moreover, if the given sequence is known to decrease "faster" than just being in $\ell^2(\mathbb{Z})$, namely if $f = (f[k])_{k\in\mathbb{Z}} \in \ell^1(\mathbb{Z})$, then the series on the r.h.s. of (25) converges uniformly and defines a continuous function $\hat{f}(\omega)$, for all $\omega \in \mathbb{R}$.

If $f, g \in \ell^1(\mathbb{Z})$, we define the convolution $f * g$ of these two sequences by

$$(f * g)[k] := \sum_{l\in\mathbb{Z}} f[l]g[k-l]. \tag{27}$$

We again have a result concerning the behaviour of the (discrete) Fourier transform with respect to convolution.

Theorem 8 *Let $f, g \in \ell^1(\mathbb{Z})$ and let $\hat{f}, \hat{g}$ denote their discrete Fourier transforms. Then, $f * g \in \ell^1(\mathbb{Z})$ and*

$$\widehat{f * g}(\omega) = \hat{f}(\omega)\,\hat{g}(\omega) \tag{28}$$

5 Discrete Fourier Transform (DFT)

We now concentrate on the case where our signal is simultaneously discrete in time and finite, $f = (f[k])_{k=0}^{N-1}$. As already mentioned, we can also think of f as a periodic sequence $f = (f[k])_{k\in\mathbb{Z}}$ of period N (i.e. as an element in $\ell(\mathbb{Z}_N)$) by letting $f[k] = f[k \bmod N]$, for all $k \in \mathbb{Z}$.

It is easy to show that the set of N vectors $\gamma_k; k = 0, \ldots, N-1$, defined by

$$\gamma_k[n] := e^{2\pi ikn/N};\ n = 0, \ldots, N-1, \tag{29}$$

is an orthogonal basis of $\ell(\mathbb{Z}_N)$ and that $\|\gamma_k\|^2 = N$. Hence, any signal $f \in \ell(\mathbb{Z}_N)$ admits the following expansion

$$f[n] = \frac{1}{N}\sum_{k=0}^{N-1} \hat{f}[k]e^{2\pi ikn/N};\quad n = 0, 1, \ldots, N-1, \tag{30}$$

where the coefficients $\hat{f}[k]$ are given by

$$\begin{aligned}\hat{f}[k] &= \langle f, \gamma_k \rangle \\ &= \sum_{n=0}^{N-1} f[n] e^{-2\pi i k n/N};\ k = 0, 1, \ldots, N-1. \end{aligned} \tag{31}$$

Formula (31) defines the so-called *discrete time Fourier series* or *discrete Fourier transform* (DFT) of f and formula (30) the *inverse discrete transform*. Naturally, the following *Parseval's identity* holds

$$\sum_{k=0}^{N-1} |f[k]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |\hat{f}[k]|^2.$$

Because of the N-periodicity of the functions $e^{-2\pi ikn/N}$, we can also see (31) as a function defined on $\mathbb{Z}_N$. This means that the discrete Fourier transform can be seen either as a map from $\mathbb{C}^N$ into $\mathbb{C}^N$ or as a map from $\ell(\mathbb{Z}_N)$ into $\ell(\mathbb{Z}_N)$. Let's introduce the following standard notation

$$W_N := e^{-2\pi i/N}. \tag{32}$$

Then, the discrete Fourier transform of $f = (f[n])_{n=0}^{N-1}$ can be defined by

$$\hat{f}[k] = \sum_{n=0}^{N-1} f[n] W_N^{kn}. \tag{33}$$

The discrete Fourier transform (as a linear transformation from $\mathbb{C}^N$ into $\mathbb{C}^N$) can also be defined using the $N \times N$ matrix (called the $N^{\underline{\text{th}}}$ *order DFT matrix*),

$$M = (m_{kn}), \quad m_{kn} = W_N^{kn}; \quad k, n = 0, \ldots, N-1.$$

It is simply given by

$$\hat{f} = Mf.$$

Given two sequences $f, g \in \ell(\mathbb{Z}_N)$, we define its convolution by

$$(f * g)[k] = \sum_{l=0}^{N-1} f[l] g[k-l], k = 0, 1, \ldots, N-1. \tag{34}$$

(Recall that the sequences are periodic of period N, i.e. $g[k] = g[k \,\mathrm{mod} N]$.)

Once more, we have the usual property relating the Fourier transform of convolutions and the product of Fourier transforms.

Theorem 9 *Let $f, g \in \ell(\mathbb{Z}_N)$ and let $\widehat{f}$ and $\widehat{g}$ denote their DFT's. Then, we have*

$$\widehat{f * g}[k] = \hat{f}[k] \hat{g}[k].$$

- *Relation of DFT to Fourier coefficients*

Assume that we know the period Ω of a certain function f as well as N of its values $y[n] := f(t_n)$ at the equally spaced points

$$t_n := n\frac{\Omega}{N}; n = 0, 1 \ldots, N-1, \tag{35}$$

and that we want to make use this information to approximate the Fourier coefficients $\hat{f}[k]$ of f. In other words, we want to compute

$$\hat{f}[k] = \frac{1}{\Omega}\int_0^{\Omega} f(t)e^{-2\pi ikt/\Omega}dt. \tag{36}$$

If we approximate the integral in (36) by a left-endpoint, uniform Riemann sum, based on the points t_n, we obtain

$$\begin{aligned}\hat{f}[k] &\approx \frac{1}{\Omega}\sum_{n=0}^{N-1} f(t_n)e^{-2\pi ikt_n/\Omega} \times \frac{\Omega}{N} \\ &= \frac{1}{N}\sum_{n=0}^{N-1} y[n]e^{-2\pi ikn/N}. \end{aligned} \tag{37}$$

The above formula shows that the $k^{\underline{\text{th}}}$ Fourier coefficient of the function f is approximately given by $\frac{1}{N}\hat{y}[k]$, where $(\hat{y}[k])_{k=0}^{N-1}$ is the N-point discrete Fourier transform of the vector $(y[n])_{n=0}^{N-1} = (f(n\frac{\Omega}{N}))_{n=0}^{N-1}$.

Note: The approximation described by the formula (37) has to be interpreted very carefully. Note that the r.h.s of (37) has period N in the variable k and the same is not true for the sequence $(\hat{f}[k])$ (typically, $\hat{f}[k] \to 0$, as $k \to \infty$). The approximation (37) will usually be used only to calculate coefficients $\hat{f}[k]$ for $|k| << N$, e.g. for $|k| \leq N/8$; for a justification of this "rule of thumb", see e.g. [6].

6 Transforms in several dimensions

All the transforms referred so far were given for the one-dimensional case, i.e. for functions of a single variable. The extension of these transforms to higher dimensions is straightforward. For example, the Fourier transform of a function $f \in L^1(\mathbb{R}^d)$ is defined by

$$\hat{f}(\boldsymbol{\omega}) = \int_{\mathbb{R}^d} f(\boldsymbol{x})e^{-2\pi i\langle\boldsymbol{\omega},\boldsymbol{x}\rangle}d\boldsymbol{x}, \quad \boldsymbol{\omega} \in \mathbb{R}^d. \tag{38}$$

In the particular case of dimension $d = 2$ (this is of special importance due to its applications in image processing), we have

$$\hat{f}(\omega, \xi) = \int_{\mathbb{R}}\int_{\mathbb{R}} f(x, y)e^{-2\pi i(\omega x+\xi y)}dx\,dy, \quad (\omega, \xi) \in \mathbb{R}^2. \tag{39}$$

The evaluation of the Fourier transform of a 2D-function is especially simple when the function is separable, i.e. can be written as

$$f(x,y) = g(x)h(y).$$

In that case, its Fourier transform is simply given by

$$\hat{f}(\omega,\xi) = \hat{g}(\omega)\hat{h}(\xi),$$

where $\hat{g}$ and $\hat{h}$ are the one-dimensional transforms of g and h. The basic transformational properties of a d-dimensional Fourier transform are essentially the same as in one dimension, with one new feature: the Fourier transform commutes with rotations, i.e. if R denotes a rotation in $\mathbb{R}^d$, then

$$f(R\boldsymbol{x}) \longrightarrow \hat{f}(R\boldsymbol{\omega}).$$

- *Fourier transforms of radial functions*

The fact that the Fourier transform commutes with rotations has the following interesting consequence. A function f defined in $\mathbb{R}^d$ is called *radial* if $f(R\boldsymbol{x}) = f(\boldsymbol{x})$ for all rotations R, i.e. $f(\boldsymbol{x})$ depends only on $|\boldsymbol{x}|$, where we used the simplified notation $|\cdot|$ for the Euclidean norm$\|\cdot\|_2$ in $\mathbb{R}^d$. If f is radial – say $f(\boldsymbol{x}) = g(|\boldsymbol{x}|)$ – then so is its Fourier transform – $\hat{f}(\boldsymbol{\omega}) = h(|\boldsymbol{\omega}|)$, say. In this case the integral formula relating f and $\hat{f}$ can be written in polar coordinates to yield h directly in terms of g. Let us illustrate with the two-dimensional case. With $\boldsymbol{x} = r(\cos\theta, \sin\theta)$ and $\boldsymbol{\omega} = \rho(\cos\phi, \sin\phi)$, we have $\langle x, \omega\rangle = r\rho\cos(\theta - \phi)$, and hence

$$\begin{aligned}
\hat{f}(\omega) &= \int_{\mathbb{R}^2} f(\boldsymbol{x}) e^{-2\pi i\langle x,\omega\rangle} d\boldsymbol{x} \\
&= \int_0^\infty \int_0^{2\pi} g(r) e^{-2\pi i r\rho\cos(\theta-\phi)} r d\theta dr \\
&= \int_0^\infty g(r) \left[\int_0^{2\pi} e^{-2\pi i\rho r\cos\theta} d\theta\right] r dr
\end{aligned}$$

By recalling the definition of the zero-order Bessel function of the first kind

$$J_0(z) = \frac{1}{2\pi}\int_0^{2\pi} e^{-iz\cos\theta} d\theta,$$

we obtain

$$h(\rho) = 2\pi \int_0^\infty g(r) J_0(2\pi\rho r) r dr. \tag{40}$$

The integral on the r.h.s (without the factor 2π) is called the *Hankel transform of order zero* of g.

- *Projection*

Suppose that we project a two-dimensional function $f(x, y)$ onto the x-axis, i.e we form

$$p(x) = \int_{\mathbb{R}} f(x, y)dy$$

Then, the (one-dimensional) Fourier transform of p is

$$\begin{aligned}\hat{p}(\omega) &= \int_{\mathbb{R}} \left[\int_{\mathbb{R}} f(x, y)dye^{-2\pi i\omega x} \right] dx \\ &= \int_{\mathbb{R}} \int_{\mathbb{R}} f(x, y)e^{-2\pi i(\omega x+0y)}dxdy = \widehat{f}(\omega, 0)\end{aligned}$$

So, the transform of the projection of $f(x, y)$ onto the x-axis is $\widehat{f}(\omega, \xi)$ evaluated along the ω-axis. This, together with the rotation property, implies that the Fourier transform of the projection onto a a line at an angle θ with the x-axis is just the Fourier transform computed along a line at an angle θ with the ω-axis. This projection property can be used e.g. in computerized axial tomography; see, e.g. [7].

7 Fourier Transform on Groups

It is possible to give a unified view of all of the different Fourier transforms described above. This is done by considering them as particular cases of a more general theory of *Fourier transforms on groups*. To present this theory in full detail requires ideas from topology and measure theory which are beyond the scope of these notes. We will, however, try to give a very brief idea of the main points (for simplicity, we will concentrate in the 1-D case).

7.1 Groups, Subgoups, Cosets

We start by recalling the notion of a *group*. A set G forms a *group* with respect to a certain binary operation $\oplus$, if the following properties hold:

1. *Closure* $\forall f, g \in G, \quad f \oplus g \in G$

2. *Associativity* $\forall f, g, h \in G, \quad (f \oplus g) \oplus h = f \oplus (g \oplus h)$

3. *Identity* $\exists 0_G \in G : \forall g \in G \quad 0_G + g = g + 0_G = g$

4. *Inverse* $\forall g \in G \exists -g \in G : g \oplus -g = -g \oplus g = 0_G$

As examples of groups especially important for our work, we have:

1. the set of real numbers $\mathbb{R}$, under addition;

2. the set of integers $\mathbb{Z}$, under addition;

3. the set $N\mathbb{Z}$, N fixed integer, under addition;
4. the set $\Omega\mathbb{Z}$, $\Omega > 0$, under addition;
5. the unit circle S^1 in the complex plane (i.e. the set of complex numbers of modulus 1), under multiplication.

A group is called Abelian if the operation $\oplus$ is commutative, i.e. $f \oplus g = g \oplus f$, for all $f, g \in G$.

A group G is *locally compact* if it has a topological structure such that the map $(f, g) \to f \oplus -g$ is continuous and every point in G has a compact neighbourhood. The group $\mathbb{R}$ is naturally a locally compact group (with the usual topology on $\mathbb{R}$). In fact, all the groups referred to in our examples are locally compact Abelian (LCA) groups.

A *subgroup* K of G is a subset of G which is also a group with respect to the same group operation. We use the notation $K \leq G$ (respectively $K < G$) to indicate that H is a subgroup of G (not equal to G itself). For example, for any N, $N\mathbb{Z}$ is a subgroup of $\mathbb{Z}$; the integers $\mathbb{Z}$ also form a subgroup of the additive group $\mathbb{R}$.

If $K < G$ and $g \in G$, we define the *coset* $g \oplus K$ of K in G as the set

$$g \oplus K = \{g + k : k \in K\}.$$

If G is an Abelian group with subgroup $K < G$, then the set of all cosets of K in G is a group under the following operation inherited from $\oplus$ (for which we use the same symbol $\oplus$):

$$(f \oplus K) \oplus (g \oplus K) := (f \oplus g) \oplus K. \tag{41}$$

This group is denoted by G/K (the *quotient group* of G modulo K).

It is easy to see that the group $\mathbb{Z}/N\mathbb{Z}$ is finite and has exactly N distinct elements. A set of *coset representatives* of G/K is a set S of elements of G such that every coset in G/K contains exactly one element of S. For example, a set of coset representatives of $\mathbb{Z}/N\mathbb{Z}$ can be taken to be $\{0, 1, \ldots, N-1\}$. When we use coset representatives instead of writing the full coset notation itself, we must remember that the operation involved is *modular*. In this sense, we can identify the group $\mathbb{Z}/N\mathbb{Z}$ with the group formed by the set $\{0, 1, \ldots, N-1\}$ with the operation of addition *modulo* N. Similarly, the group $\mathbb{T}_\Omega := \mathbb{R}/\Omega\mathbb{Z}$ can be identified with the group whose set of elements is $[0, \Omega)$ (or any other interval of length Ω) and whose operation is addition modulo Ω.

Let G and H be two groups with operations $\oplus_G$ and $\oplus_H$, respectively. A *homomorphism* from G to H is a map $\phi : G \to H$ such that

$$\phi(f \oplus_G g) = \phi(f) \oplus_H \phi(g), \quad \forall f, g \in G.$$

Is the homomorphism is bijective, we call it an *isomorphism*. For example, the function $\phi : \mathbb{T}_\Omega \to S^1$ defined by

$$\phi(t + \Omega\mathbb{Z}) = e^{2\pi i t/\Omega}, \quad t \in \mathbb{R}$$

is an isomorphism between the additive group $\mathbb{T}_\Omega$ and the multiplicative group S^1. In some sense, we can view the two groups S^1 and T_Ω as the *same* group. Another important example of two isomorphic groups is given by the group $\mathbb{Z}/N\mathbb{Z}$ and the group S_N of all the $N^{\underline{\text{th}}}$ roots of unity,

$$S_N := \{1, W_N, W_N^2, \ldots, W_N^{N-1}\}, \quad W_N := e^{2\pi i/N},$$

under multiplication. An isomorphism between the two groups is given by

$$\phi(k + N\mathbb{Z}) = e^{2\pi ik/N} = W_N^k.$$

7.2 Characters of a group

For any Abelian group G, a *character* γ of G is a homomorphism of G into the group S^1. The set of all *continuous* characters of G is denoted by $\hat{G}$ and is itself an Abelian group under the operation of pointwise multiplication. This is called the *dual* group of G.

For example, when G is the additive group $\mathbb{R}$, one can show that the characters are the functions

$$\gamma_\omega(x) = e^{2\pi i\omega x}, \quad x \in \mathbb{R},$$

for all $\omega \in \mathbb{R}$.

Note: The choice of the exponent 2π associated with the "index" ω is just for convenience; any other real number would do, which would correspond to a simple renaming of the functions γ.

This is easily seen to be isomorphic to $\mathbb{R}$ itself (the mapping $\omega \mapsto \gamma_\omega$ defining an isomorphism). In that sense, we say tat $\mathbb{R}$ is self-dual and write $\hat{\mathbb{R}} = \mathbb{R}$. One can also identify the characters of the group $\mathbb{Z}_N = \mathbb{Z}/N\mathbb{Z}$: they are the functions $\gamma_k; k = 0, 1\ldots, N-1$ defined by

$$\gamma_k[m] = e^{2\pi ikm/N}, \quad m \in \{0, \ldots, N-1\}.$$

The function $\phi : k \mapsto \gamma_k$ is an isomorphism between $\mathbb{Z}_N$ and $\hat{\mathbb{Z}}_N$ and, in that sense, $\mathbb{Z}_N$ is also self-dual.

Finally, we describe the characters of the group $\mathbb{T}_\Omega$. They are the functions

$$\gamma_k(t) := e^{2\pi ikt/\Omega}, \quad t \in [0, \Omega),$$

for all $k \in \mathbb{Z}$. The dual group of $\mathbb{T}_\Omega$ is thus isomorphic to $\mathbb{Z}$, the mapping $\phi : k \mapsto \gamma_k$ defining an isomorphism.

Since the dual group is also an Abelian group it is possible to define its set of characters, i.e. to define its dual $\hat{\hat{G}}$. It turns out that this group is always isomorphic to G. Hence, the dual group of $\mathbb{Z}$ is (isomorphic) to the Ω-torus $\mathbb{T}_\Omega$ (the particular choice of $\Omega > 0$ is not important, since all these groups are isomorphic to each other).

7.3 Integration on groups and Fourier transform

In order to be able to give an adequate definition of the Fourier transform on a (LCA) group, one has to introduce an appropriate concept of measure and corresponding integration on the group. It can be shown that in every LCA group, there exists a non-negative and regular measure, which is not identically zero and is *translation invariant.* This measure is unique (up to the multiplication by a positive constant) and is called the *Haar measure* of G. Integration on the group G will always be understood with respect to such measure. For the construction of such a measure, see e.g. [8] or [9]. In our cases, we simply refer that this measure is:

1. the usual Lebesgue measure, for the cases $G = \mathbb{R}$ and $G = \mathbb{T}_\Omega$ (with the "normalization" constant $\frac{1}{\Omega}$ in the latter case) ;

2. the usual counting measure for the discrete cases $G = \mathbb{Z}$ and $G = \mathbb{Z}/N\mathbb{Z}$.

Having defined integration on G, we can also introduce, in a natural way, the $L^p(G)$ spaces.

We can then define the *Fourier transform* of any function $f \in L^1(G)$: it is the function $\widehat{f}$, defined on $\widehat{G}$ by

$$\widehat{f}(\gamma) = \int_G f(t)\overline{\gamma(t)}dt, \quad \gamma \in \widehat{G}. \tag{42}$$

Note that the Fourier transform of a function defined on G is actually a function defined on $\hat{G}$. This means, for example, that in the case of $G = \mathbb{R}$, we should have written the Fourier transform of a function f as $\hat{f}(\gamma_\omega)$. However, due to the identification of $\mathbb{R}$ with $\hat{\mathbb{R}}$, this is naturally shortened to $\hat{f}(\omega)$.

Having identified previously the characters $\gamma \in \widehat{G}$ for all the cases $G = \mathbb{R}$, $G = \mathbb{T}_\Omega$, $G = \mathbb{Z}$ and $G = \mathbb{Z}_N$, it is now simple to verify that the definitions (13), (21) , (25) and (31) all fit into this framework.

We also have an inversion theorem (cf. formulae (15), (20), (26) and (30)).

Theorem 10 *Let $f \in L^1(G)$ be such that $\hat{f} \in L^1(\hat{G})$. If the Haar measure of G is fixed, the Haar measure of $\hat{G}$ can be normalized so that the following inversion formula holds*

$$f(t) = \int_{\hat{G}} \hat{f}(\gamma)\gamma(t)d\gamma, \quad t \in G.$$

If we define the convolution of any two functions $f, g \in L^1(G)$ by

$$f * g(t) = \int_G f(u)g(t-u)du$$

we have the following result from which the results of theorems 3,7,8 and 9 are specific examples:

Theorem 11 *Let $f, g \in L^1(G)$ and let $\widehat{f}, \widehat{g}$ be their Fourier transforms. Then, $f * g \in L^1(G)$ and*

$$\widehat{f * g}(\gamma) = \hat{f}(\gamma)\hat{g}(\gamma).$$

Finally, we would like to remark that there is a natural way of extending the definition of the Fourier transform from $L^1(G)$ to $L^2(G)$ and that the Parseval formula holds

$$\int_G f(t)\overline{g(t)}dt = \int_{\widehat{G}} \hat{f}(\gamma)\hat{g}(\gamma)d\gamma.$$

For more details on this fascinating topic of Fourier transforms on groups see, e.g. [10]. We now turn to the problem of describing efficient algorithms for the computation of Fourier transforms.

8 Fast Fourier Transform

The Fast Fourier Transform – the most valuable algorithm of our lifetime.

Strang, 1993

A direct calculation of a N-point DFT requires $(N-1)^2$ multiplications and $N(N-1)$ additions, i.e. it involves a number of operations of order $O(N^2)$. For large N, this can be extremely time consuming. In 1965, Cooley and Tukey [11] proposed an algorithm to compute the DFT reducing the number of operations involved to $O(N \log_2 N)$, when $N = 2^r$. This algorithm, which has become known as the *Fast Fourier Transform*, had a tremendous impact and is responsible for the widespread use of DFT's in almost all branches of scientific computation, with particular emphasis on digital signal processing.

Note: In fact, as referred in [12], the basic idea of the FFT had already been discovered by Gauss, in 1805, as an efficient means of interpolating asteroid orbits. However, it was the Cooley and Tukey publication which popularized the use of the discrete Fourier transform; see [13].

Many variants of the basic FFT algorithm have also appeared subsequently. Here, we will briefly describe one of the most widely used of these algorithms, the so-called *decimation in time, radix 2 FFT*; for other variants the reader is referred to, e.g. [6], [14] or [15]. FTT programs in various computer languages can be found in [16]. The article by Burrus [17] gives an excellent summary and contains an extensive list of references on efficient algorithms to compute the DFT. A compiled bibliography on this topic (with more than 3400 entries!) is given in [18].

8.1 Decimation in Time Radix 2 FFT

We will assume that N is a power of 2, say $N = 2^r$, where r is a positive integer. Let us start by recalling the formula for the N-point DFT transform of a sequence $f = (f[k])_{k=0}^{N-1}$,

$$\hat{f}[n] = \sum_{k=0}^{N-1} f[k] W_N^{kn}, \tag{43}$$

where $W_N = e^{2\pi i/N}$. For simplicity, we will introduce the notation $F_n := \hat{f}[n]$. We can halve the N-point DFT in (43) in two sums, each of which is a $N/2$-point DFT:

$$\begin{aligned} F_n &= \sum_{k=0}^{N/2-1} f[2k] W_N^{2kn} + \sum_{k=0}^{N/2-1} f[2k+1] W_N^{2kn} W_N^n \\ &= \sum_{k=0}^{N/2-1} f[2k] W_{N/2}^{kn} + \sum_{k=0}^{N/2-1} f[2k+1] W_{N/2}^{kn} W_N^n \end{aligned}$$

We can thus write

$$\left. \begin{aligned} F_n &= F_n^0 + W_N^n F_n^1 \\ F_{n+N/2} &= F_n^0 - W_N^n F_n^1 \end{aligned} \right\} ; n = 0, \ldots, N/2 - 1, \tag{44}$$

where, for $j = 0, 1$,

$$F_n^j = \sum_{k=0}^{N/2-1} f[2k+j] (W_{N/2})^{kn}, \tag{45}$$

and where we have used the fact that $W_N^{j(N/2)+n} = \pm W_N^n$, depending on wether $j = 0, 1$. The DFT $(F_n)_{n=0}^{N-1}$ written in terms of the calculations (44)-(45) can be visualized as

$$\begin{array}{c} F_n^0 \longrightarrow F_n^0 + W_N^n F_n^1 \\ \times \\ F_n^1 \longrightarrow F_n^0 - W_N^n F_n^1 \end{array} \tag{46}$$

where $n = 0, 1, \ldots, N/2 - 1$. This diagram is called a *butterfly.* The butterfly (46) can be viewed as a construction of the DFT in $\mathbb{Z}_N$ in terms of two DFTs, F^0 and F^1, on $\mathbb{Z}_{N/2}$. In the same way, each F^0 and F^1 can be constructed in terms of a pair of two of DFTs on $\mathbb{Z}_{N/4}$. For example,

$$F_n^0 = F_n^{00} + W_{N/2}^n F_n^{01}$$

and

$$F_{n+N/4}^0 = F_n^{00} - W_{N/2}^n F_n^{01},$$

for $n = 0, 1, \ldots, N/4 - 1$, where F^{00} is the DFT of $(f[0], f[4], \ldots, f[N-4])$ and F^{01} is the DFT of $(f[2], f[6], \ldots, f[N-2])$. Since $N = 2^r$, this procedure can be repeated and after $r = \log_2 N - 1$ stages we reach a point where we are performing $N/2$ 2-point DFTs, which consist of adding and subtracting two points. Computationally, it is convenient to compute the 2-point DFTs first, then the 4-point DFts, etc.

8.2 Bit Reversal

Let $f : \mathbb{Z}_N \to \mathbb{C}$ be given and suppose we want to compute the DFT F in the natural ordering $(F_0, \ldots, F_N)$. From (45), it is clear that if we begin with the DFTs of the pairs $(f[0], f[1]), (f[2], f[3]), \ldots$ we will not obtain F in the natural ordering. For example, when $N = 8$, the input indices must be ordered as $(0, 4, 2, 6, 1, 5, 3, 7)$ so that the output sequence will appear in the natural order. This ordering is obtained by *bit reversal.* Bit reversal (at *level* r or of *order* r) is defined recursively as follows. For $r = 1$, the bit reversal ordering (of the set $\{0, 1\}$) is the ordered pair $(0, 1)$. At level $r; r = 2, 3, \ldots$, the bit reversal ordering of the set $\{0, 1, \ldots, 2^r - 1\}$ is the 2^r-tuple

$$(2b_0, \ldots, 2b_{M-1}, 2b_0 + 1, \ldots, 2b_{M-1} + 1), \tag{47}$$

where $M = 2^{r-1}$ and $(b_0, b_1, \ldots, b_{M-1})$ is the bit reversal ordering at level $r - 1$. For example, bit reversal orderings at levels 2 and 3 are $(0, 2, 1, 3)$ and $(0, 4, 2, 6, 1, 5, 3, 7)$, respectively. The term bit reversal comes from the following observation. If $k \in \{0, 1, \ldots, 2^r - 1\}$ has the binary expansion

$$k = \sum_{j=0}^{r-1} \epsilon_j 2^j$$

then the number in the position $k; k = 0, \ldots, 2^{r-1}$ of the bit reversal ordering is obtained by "reversing" the order of the coefficients ϵ_j in the above expansion. It is important to observe that there are efficient algorithms for obtaining bit-reversed indices; see e.g. [14] or [6]. This last reference also describes efficient ways of performing the butterfly calculations involved in each step of the FFT algorithm.

9 Fourier Related Transforms

9.1 Cosine and Sine Transforms

9.1.1 Fourier Sine and Cosine Transform

Using Euler's formula, we can write the Fourier transform of f as

$$\begin{aligned}\hat{f}(\omega) &= \int_{\mathbb{R}} f(t)e^{-2\pi i\omega t}dt \\ &= \int_{\mathbb{R}} f(t)\cos(2\pi\omega t)dt - i\int_{\mathbb{R}} f(t)\sin(2\pi\omega t)dt \\ &:= \mathcal{C}f(\omega) - i\mathcal{S}f(\omega), \end{aligned} \tag{48}$$

where $\mathcal{C}f(\omega)$ and $\mathcal{S}f(\omega)$ are called, the *Fourier cosine transform* and *Fourier sine transform* of f, respectively. Observe that if the function f is real-valued, then its Fourier transform can found by evaluating two real integrals. Also, if f is an even function, then the Fourier transform of f is simply its Fourier cosine transform and can be computed simply

$$\begin{aligned}\hat{f}(\omega) &= \int_{\mathbb{R}} f(t)\cos(2\pi\omega t)dt \\ &= 2\int_0^\infty f(t)\cos(2\pi\omega t)dt.\end{aligned}$$

Similarly, if f is an odd function,

$$\begin{aligned}\hat{f}(\omega) &= -i\int_{\mathbb{R}} f(t)\sin(2\pi\omega t)dt \\ &= -2i\int_0^\infty f(t)\sin(2\pi\omega t)dt.\end{aligned}$$

9.1.2 Fourier Sine and Cosine Series

It is simple to establish that the set of functions

$$\gamma_n(t) := \cos(\pi nt/\Omega), \quad n \in \mathbb{N}_0,$$

is an orthogonal basis of the space $L^2(\mathbb{T}_\Omega)$, and that $\|\gamma_0\|_2^2 = 1$ and $\|\gamma_n\|_2^2 = 1/2$, $n \in \mathbb{N}$. Hence, every function $f \in L^2(\mathbb{T}_\Omega)$ admits an expansion

$$f(t) = \frac{1}{2}A_0 + \sum_{n=1}^{\infty} A_n \cos(\pi nt/\Omega)\,, \tag{49}$$

where

$$A_n = \frac{2}{\Omega}\int_0^\Omega f(t)\cos(\pi nt/\Omega)dt. \tag{50}$$

The series (49), with the coefficients given by (50), is called the *Fourier cosine series of f*.

Note: The above series is, in fact, the Fourier series of the *even* extension of f to $L^2(\mathbb{T}_{2\Omega})$.

In a similar manner, we define the Fourier sine series of f:

$$f(t) = \sum_{n=1}^{\infty} B_n \sin(\pi nt/\Omega) dt,$$

where

$$B_n = \frac{2}{\Omega} \int_0^{\Omega} f(t) \sin(\pi nt/\Omega) dt.$$

9.1.3 Discrete Cosine Transforms

There are also discrete versions of the sine and cosine transforms. Here, we refer to four established discrete cosine transforms (DCT-I through DCT-IV). The (two-dimensional version) of DCT-II and DCT-IV are constantly applied in image processing and have a FFT implementation, which makes them especially useful. The DCTs (in fact DCT-II) was only discovered in 1974, [19]. All four types of DCT are orthogonal transforms and use bases for the space $\mathbb{C}^N$ (or $\ell(\mathbb{Z}_N)$) that involve only *cosines*. For $k, n = 0, 1, \ldots, N-1$ the nth component of the kth basis vector is

DCT-I	$\cos\left(nk\frac{\pi}{N-1}\right)$	(divide by $\sqrt{2}$ when $k, n = 0, N-1$)
DCT-II	$\cos\left((n+\frac{1}{2})k\frac{\pi}{N}\right)$	(divide by $\sqrt{2}$ when $k = 0$)
DCT-III	$\cos\left(n(k+\frac{1}{2})\frac{\pi}{N}\right)$	(divide by $\sqrt{2}$ when $n = 0$)
DCT-IV	$\cos\left((n+\frac{1}{2})(k+\frac{1}{2})\frac{\pi}{N}\right)$	

If we consider the matrices $C_{\mathrm{I}}, C_{\mathrm{II}}, C_{\mathrm{III}}$ and C_{IV} whose columns are the above vectors, then each of the DCT-T transforms $\widehat{f}_{\mathrm{T}}$ of a vector $f \in \mathbb{C}^N$ is defined by

$$\hat{f}_{\mathrm{T}} = C_{\mathrm{T}} f; \quad \mathrm{T} = \mathrm{I}, \mathrm{II}, \mathrm{III}, \mathrm{IV}. \tag{51}$$

All vectors have norm $\sqrt{N/2}$; hence, we have, for example (using the DCT-IV transform), that any vector $f \in \mathbb{C}^N$ can be written as

$$f[n] = \frac{2}{N} \sum_{k=0}^{N-1} \hat{f}_{\mathrm{IV}}[k] \cos\left((n+\frac{1}{2})(k+\frac{1}{2})\frac{\pi}{N}\right)$$

where

$$\hat{f}_{\mathrm{IV}}[k] = \sum_{n=0}^{N-1} f[n] \cos\left((n+\frac{1}{2})(k+\frac{1}{2})\frac{\pi}{N}\right).$$

Similar expressions for the transform and corresponding inverse for the DCT-I – DCT-III are easily written.

9.2 Hartley Transform

The *Hartley transform* $\mathcal{H}f$ is obtained by combining the sine and cosine transforms replacing $-i$ by 1, i.e.

$$\begin{aligned}\mathcal{H}f(\omega) &= \mathcal{C}f(\omega) + \mathcal{S}f(\omega) \\ &= \int_{\mathbb{R}} f(t)\mathrm{cas}(2\pi\omega t)dt, \qquad (52)\end{aligned}$$

where $\mathrm{cas}(t) := \cos t + \sin t$. The Hartley transform was initially proposed by Hartley in 1942 [20], but was virtually ignored until it was reintroduced by Bracewell [21] in 1983. The Hartley transform has the advantage that is real-valued for a real-valued signal, but it lacks some of the important properties of the Fourier transform; a thorough investigation of Hartley transforms can be found in [22]. There is also a discrete version of the Hartley transform and fast algorithms for its computation (Fast Hartley Transform).

9.3 Laplace Transform

Fourier transforms were defined for *real* values of the frequency variable. A more general class of transforms can be obtained if the frequency variable is allowed to be complex.

We define the (bilateral) *Laplace transform* of a function f by

$$\mathcal{L}f(s) = \int_{\mathbb{R}} f(t)e^{-st}dt \qquad (53)$$

where $s \in \mathbb{C}$. Note that, when $s = 2\pi i\omega$, $\mathcal{L}f(s) = \hat{f}(\omega)$ and so, as it might be expected, the Laplace transforms has many important properties similar to those of the Fourier transform. When $s = \sigma + 2\pi i\omega$, then $\mathcal{L}f(s)$ is the Fourier transform of $g(t) = f(t)e^{-i\sigma t}$, i.e. is the transform of an exponentially weighted signal.

Note: The more frequently used *unilateral* Laplace transform can be defined as the Laplace transform of $f(t)u(t)$, where $u(t)$ is the unit-step function defined by $u(t) = 1$, for $t \geq 0$ and $u(t) = 0$ otherwise.

The above transform does not, in general, converge for all values of s. The set of values for which (53) converges is called the region of convergence (ROC). The ROC has the following important poperties:

1. it consists of strips in the complex plane parallel to the to the $i\omega$ axis i.e. is of the form $A \leq \mathrm{Re}(s) \leq B$ where A and B may be $-\infty$ and $+\infty$, respectively; (In the extreme cases, the $\leq$ sign might have to be replaced by $<$);

2. if $f(t)$ is right-sided (left-sided), i.e. is zero for $t < T_0$ (i.e is zero for $t > T_1$), then $B = +\infty$ ($A = -\infty$).

3. if $f(t)$ is time-limited (i.e. $f(t) = 0$ for $T_0 < t < T_1$, then its ROC is the whole complex plane (provided it converges at some point);
4. if the $i\omega$ axis is contained in the ROC, then the Fourier transform of f exists.

The Laplace transform can be inverted. Its inverse is given by

$$f(t) = \frac{1}{2\pi i} \int_{\sigma - i\infty}^{\sigma + i\infty} e^{st} \mathcal{L}f(s) ds,$$

where σ is chosen inside the ROC.

The (unilateral) Laplace transform is particularly useful for solving initial value problems. For a comprehensive treatment of the Laplace transforms and its applications, we refer the reader to [23] or [24].

9.4 z-Transform

Just as the Laplace transform was a generalization of the Fourier transform, the *z-transform* can also be introduced as a generalization of the discrete time Fourier transform. For a given sequence $f = (f[k])_{k \in \mathbb{Z}}$, we define its z-transform as

$$Z(f[k]) := F(z) := \sum_{k \in \mathbb{Z}} f[k] z^{-k}, \tag{54}$$

where $z \in \mathbb{C}$. Again, the transform is only defined for the values of z for which the above series converges, these values defining its region of convergence (ROC). On the unit circle $z = e^{2\pi i\omega}$, this is the discrete-time Fourier transform ($\Omega = 1$), and for $z = \rho e^{2\pi i\omega}$, it is the discrete-time Fourier transform of the sequence $f[k]\rho^{-k}$. The ROC of the z-transform has properties "analogous" to the ROC of Laplace transforms:

1. it consists of a ring in the complex plane, i.e. is a set of the form $A \leq |z| \leq B$, where A may be zero and B may be $+\infty$. (In the extreme cases, the $\leq$ sign might have to be replaced by $<$).
2. if the sequence $f([k])$ is causal (i.e. $f[k] = 0$ for $k < 0$), then $B = +\infty$ ($\leq$ possibly replaced by $<$); if the sequence is anti-causal (i.e $f[k] = 0$ for $k > 0$), then $A = 0$ ($\leq$ possibly replaced by $<$);
3. if the sequence is of finite length and causal, the ROC is the entire plane, except possibly $z = 0$;
4. if the sequence is of finite length and anti-causal, the ROC is the entire z-plane except, possibly, the "point" $z = \infty$;
5. the discrete time Fourier transform of the sequence $f([k])$ converges absolutely if and only the ROC contains the unit circle.

The inverse z-transform involves the contour integration in the ROC and Cauchy's integral theorem. We have

$$f[n] = \frac{1}{2\pi i} \oint_C Z(f[k]) z^{n-1} dz$$

where C denotes a contour around the origin lying in the ROC. The z-transform is very useful for the study of difference equations and discrete-time filters; more details can be seen, e.g. in [25] or [26].

9.5 Mellin Transform

The *Mellin transform* $\mathcal{M}f$ of a function f is defined by

$$\mathcal{M}f(z) = \int_0^\infty f(t) t^{z-1} dt. \tag{55}$$

If we make the change of variable $x = \log t$, we find that

$$\mathcal{M}f(z) = \int_{\mathbb{R}} f(e^x) e^{xz} dx \tag{56}$$

which shows that $\mathcal{M}f(-2\pi i\omega)$ is the Fourier transform (at ω) of the composition $f \circ \exp$; a good reference to read about Mellin transforms is the book by Bracewell [27].

9.6 Hilbert Transform

The *Hilbert transform* $\mathrm{H}f$ of $f \in L^2(\mathbb{R})$ is defined by

$$\mathrm{H}f(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{f(u)}{t-u} du, \tag{57}$$

interpreting the integral as a Cauchy principal value, i.e. as

$$\lim_{\epsilon \to 0} \int_{|t-u|>\epsilon} \frac{f(u)}{t-u} du.$$

This transform is invertible, its inverse being simply $-H$, i.e.

$$f(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{\mathrm{H}f(u)}{u-t} dt. \tag{58}$$

- *Analytic signals and Hilbert transform*

A function $f \in L^2(\mathbb{R})$ is said to be a (strong) *analytic signal* if its Fourier transform is zero for negative frequencies, i.e $\hat{f}(\omega) = 0$ for $\omega < 0$. If f is real valued, one can associate with f an analytic signal f_a in the following manner: f_a is the signal whose Fourier transform is given by

$$\hat{f}_a(\omega) = \begin{cases} 2\hat{f}(\omega), & \omega \geq 0 \\ 0, & \omega < 0. \end{cases} \tag{59}$$

One can show that if f_a if is the analytic signal associated with the real signal f, then $\operatorname{Re} f_a = f$ and $\operatorname{Im} f_a = \mathrm{H}f$, i.e.

$$f_a = f + i\mathrm{H}f. \tag{60}$$

Let

$$f_a(t) = A(t)e^{i\phi(t)}$$

The *envelope* $E(t)$ of the signal $f(t)$ is defined as $|A(t)| = \sqrt{f(t)^2 + \mathrm{H}f(t)^2}$ and $E(t)^2$ is the the so-called *instantaneous power*; the *instantaneous frequency* $\omega(t)$ is defined by $\omega(t) = \phi'(t)$. Hence, Hilbert transform analysis provides a method of determining the "instantaneous" frequency and power of a signal. This technique is widely used in communications systems analysis; see, e.g. [28].

9.7 Haar and Walsh Transforms

We consider again the space $L^2(\mathbb{T}_\Omega)$ and take, for simplicity, $\Omega = 1$, i.e. consider functions in the space $L^2[0,1]$. Besides the basis functions $\gamma_k(t) := e^{2\pi ikt}, \quad k \in \mathbb{Z}$, used in the Fourier series expansion, one may consider the use of other orthonormal bases for this space. We describe here two such bases, consisting of step functions.

Let $H(t)$ be the function defined by

$$H(t) = \begin{cases} 1, & 0 < t < \frac{1}{2} \\ -1, & 1/2 \leq t < 1 \end{cases} \tag{61}$$

This is called the *Haar function*. Then, the set of functions obtained by dyadic dilation and translation of this function, i.e.

$$H_{jk}(t) := 2^{j/2}H(2^j t - k),\ j \geq 0,\ k = 0, 1, \ldots, 2^j - 1, \tag{62}$$

together with the function $H_0 := \mathbf{1}_{[0,1)}$ (the characteristic function on the interval $[0,1)$), form an orthonormal basis for $L^2[0,1]$. Thus, every function $f \in L^2[0,1]$ admits a Haar series expansion

$$f(t) = f_H[0] + \sum_{j\geq 0} \sum_{k=0}^{2^j-1} f_H[j,k]H_{jk}(t)$$

where the *Haar coefficients* $f_H[0]$ and $f_H[j,k]$ are given by

$$f_H[0] = \int_0^1 f(t)H_0(t)dt; \qquad f_H[j,k] = \int_0^1 f(t)H_{jk}(t)dt$$

To introduce the other basis consisting of step functions, we start by defining the so-called *Rademacher functions* r_n. For $n \geq 0$, consider the division of the interval $[0,1]$ into 2^n subintervals of equal length. Then, $r_n(t)$ is the function which takes the values $+1$ and -1, alternately, in each of these subintervals, starting with +1; in other words, if $d_n(t)$ is the $n^{\underline{\text{th}}}$ digit in the binary representation of t $(0 \leq t < 1)$, then

$$r_n(t) = (-1)^{d_n(t)}.$$

If $n \geq 0$ and $b_1, \ldots, b_k$ are the digits in the binary representation of n, i.e. $n = (b_k \ldots b_2 b_1)_2$, then the $n^{\underline{\text{th}}}$ *Walsh function* w_n is defined by

$$w_n(t) = r_1(t)^{b_1} r_2(t)^{b_2} \ldots r_k(t)^{b_k}.$$

Then, the set of Walsh functions $\{w_n; n \geq 0\}$ is an orthonormal basis of $L^2[0,1]$. For an account of the applications of the Haar and Walsh functions in signal and image processing and other related fields see, e.g. [29]. Naturally, there are are also discrete versions of these transforms.

The Haar function is the first example (constructed by Haar in 1910 [30]) of an *orthogonal wavelet*, i.e. of a function $\psi \in L^2(\mathbb{R})$ whose dyadic dilations and translations $\psi_{jk} = 2^{j/2}\psi(2^j t - k)$; $j, k \in \mathbb{Z}$ constitute an orthonormal basis of $L^2(\mathbb{R})$; we will come back to this topic of wavelets in a little more detail in Section XI.

10 Windowed Fourier Transform

Recalling the expression for the Fourier transform

$$\hat{f}(\omega) = \int_{\mathbb{R}} f(t) e^{-2\pi i \omega t} dt,$$

we see that $\hat{f}(\omega)$ depends on the values $f(t)$ for all time $t \in \mathbb{R}$. Hence, it is difficult to read any local behaviour of f from $\hat{f}$. In many applications, such as analysis of non-stationary signals or real time signal processing, the simple use of a Fourier transform may not be appropriate. In fact, one would like to dispose of an analytic tool that provides information both in time and frequency. One of the first ideas was simply to truncate the signal and to analyze only what happens on a finite interval $[-A, A]$. Mathematically, this corresponds to multiplying f by the characteristic function of this interval, $\mathbf{1}_{[-A,A]}$, and taking the Fourier transform of the product. We then have

$$\widehat{\mathbf{1}_{[-A,A]} f}(\omega) = (S_A * \hat{f})(\omega),$$

where $S_A(\omega) = \frac{\sin 2\pi A\omega}{\pi\omega}$. Thus, truncating the function results in convolving its spectrum with a cardinal sine. However, the cardinal sine decays slowly and has important lobes near the origin (hence there is poor localization in frequency). To avoid these problems, we can replace $\mathbf{1}_{[-A,A]}$ with more regular functions $W(t)$, called *windows*. Some typical choices include:

Bartlett or triangle window

$W(t) = (1 - \frac{|t|}{A})\mathbf{1}_{[-A,A]}$

Hamming and Hanning windows

$$W(t) = [\alpha + (1-\alpha)\cos(\pi t/A)]\mathbf{1}_{[-A,A]}$$

For $\alpha = 0.54$ we have Hamming's window and for $\alpha = 0.50$ we have Hanning's window.

Blackman window

$$W(t) = [0.42 + 0.5\cos(\pi t/A) + 0.08\cos(2\pi t/A)]\mathbf{1}_{[-A,A]}$$

Gaussian window

$$W(t) = Ce^{-\alpha t^2} \quad (C, \alpha > 0).$$

For more details and other choices of window functions, see e.g. [31] or [16]. All the windows described above are concentrated around the origin. We can then "slide" the window along the real axis and analyze the whole function. We then define the so-called *windowed Fourier transform* or *short time Fourier transform* (associated with the specific window W) as:

$$\mathcal{F}_W f(\omega,\tau) = \int_{\mathbb{R}} f(t)\overline{W}(t-\tau)e^{-2\pi i\omega t}dt. \tag{63}$$

Note: When a Gaussian window is used in the short time Fourier transform, this is usually referred as *Gabor transform*. If we define the family of functions $W_{\omega,\tau}$ by the result of two simple operations – translation by τ and modulation by ω – on the basic window W, i.e.

$$W_{\omega\tau}(t) := W(t-\tau)e^{2\pi i\omega t}, \tag{64}$$

we can view the windowed Fourier transform simply as the inner product of f with each of these functions:

$$\mathcal{F}_W f(\omega,\tau) = \langle f, W_{\omega,\tau}\rangle. \tag{65}$$

We then also have, by Plancherel formula,

$$\mathcal{F}_W f(\omega,\tau) = \langle \hat{f}, \hat{W}_{\omega,\tau}\rangle. \tag{66}$$

If W and $\hat{W}$ are localized around the origin, then $W_{\omega,\tau}$ is localized around the instant τ, while $\hat{W}_{\omega,\tau}$ is localized around the frequency ω. The value $\mathcal{F}_W f(\omega,\tau)$ thus provides an indication of how the function behaves around time τ and frequency ω.

The function f can always be recovered (in the L^2 sense), by a double integral

$$f(t) = \iint_{\mathbb{R}^2} \mathcal{F}_W f(\omega,\tau)W_{\omega\tau}(t)d\omega\, d\tau, \tag{67}$$

where we have assumed that the window W was chosen satisfying $\|W\|_2 = 1$. There is also an energy conservation property for the windowed Fourier transform:

$$\iint_{\mathbb{R}^2} |\mathcal{F}_W f(\omega,\tau)|^2 d\omega d\tau = \|f\|_2^2. \tag{68}$$

The windowed Fourier transform is even more familiar to signal analysis in its discrete version, where τ and ω are assigned regularly spaced values: $\tau = n\tau_0$ and $\omega = m\omega_0$, where $m, n \in \mathbb{Z}$ and $\tau_0, \omega_0 > 0$ are fixed. That is, we let

$$W_{m,n}(t) := e^{2\pi i m\omega_0 t} W(t - n\tau_0) \tag{69}$$

and compute the values

$$C_{m,n} = \langle f, W_{m,n} \rangle. \tag{70}$$

The question naturally arises of whether it is possible to reconstruct the given function f from its transform coefficients $C_{m,n}$ in a *numerically stable* way (i.e. in a manner not too "sensitive" to the unavoidable errors in the computed values). The answer is positive, provided the functions $W_{m,n}$ given by (69) constitute a *frame*, i.e. satisfy

$$A\|f\|_2^2 \leq \sum_{m,n\in\mathbb{Z}} |\langle f, W_{m,n}\rangle|^2 \leq B\|f\|_2^2, \tag{71}$$

for all $f \in L^2(\mathbb{R})$, with constants $0 < A \leq B < \infty$. The following theorem, whose proof can be seen, e.g. in [32], establishes necessary conditions on the parameters ω_0 and τ_0 for the set functions $\{W_{m,n} : m, n \in \mathbb{Z}\}$ to be a frame of $L^2(\mathbb{R})$.

Theorem 12 *Let $W \in L^2(\mathbb{R})$ be such that $\|W\|_2 = 1$. The windowed Fourier family $\{W_{m,n} : m, n \in \mathbb{Z}\}$ can only be a frame if*

$$\omega_0\tau_0 \leq 1. \tag{72}$$

The frame bounds A and B necessarily satisfy

$$A \leq \frac{1}{\omega_0\tau_0} \leq B. \tag{73}$$

In particular, a necessary condition for the functions (69) to be an orthonormal basis of $L^2(\mathbb{R})$ is that $\omega_0\tau_0 = 1$.

We also have the following important theorem, whose proof can again be seen in [32].

Theorem 13 (Balian-Low) *If $\|W\|_2 = 1$ and $\{W_{m,n} : m, n \in \mathbb{Z}\}$ is a windowed Fourier frame with $\omega_0\tau_0 = 1$, then*

$$\int_{\mathbb{R}} t^2 |W(t)|^2 dt = +\infty \quad \textit{or} \quad \int_{\mathbb{R}} \omega^2 |\hat{W}(\omega)|^2 d\omega = +\infty.$$

This theorem shows, in particular, that we can not construct an ortogonal windowed Fourier basis with a differentiable window of compact support.

11 Wavelet Transform

11.1 Continuous Wavelet Transform

The windowed Fourier transform computes the inner product of the function f with a family of functions $W_{\omega,\tau}$ obtained by translating and modulating the basic window W. The functions of this family are all of the "same size" (i.e. they all have the same spread in time and frequency). If the signal to be studied has components which are almost stationary associated with sudden variations, then the windowed Fourier analysis is not the appropriate tool, due the above fixed size of the windows. We now study a different transform which overcomes the above limitations, by using windows whose size naturally adjusts to frequencies. The idea of the *continuous wavelet transform* is again to compute the inner product of the function to be analyzed with a family of functions $\psi_{a,\tau}$ dependent on two parameters. In this case, however, these functions are obtained from a basic function (the *analyzing* or *mother wavelet*) by contractions or dilations (i.e. changes of scale)– controlled by the parameter a, and translations – controlled by the parameter τ. The mother wavelet ψ used for the analysis has to satisfy a certain technical condition, known as the *admissibility condition*. More precisely, we say that $\psi \in L^2(\mathbb{R})$ is a *wavelet* if it satisfies

$$C_\psi := \int_{\mathbb{R}} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < +\infty \tag{74}$$

In practice, we want to use a function ψ which behaves like a time window, i.e. we select ψ with a fast decay property in time (e.g. ψ and $t\psi(t) \in L^1(\mathbb{R})$). In this case, the admissibility condition (74) turns out to be equivalent to the condition

$$\int_{\mathbb{R}} \psi(t) dt = 0. \tag{75}$$

This indicates that ψ must "oscillate" above and below the t axis, i.e. must behave like a wave; this, together with the constraint that ψ decays fast (i.e. is "small") justifies the name *wavelet* adopted for these functions. Given a certain wavelet ψ (normalized so that $\|\psi\|_2 = 1$), we define the family of functions

$$\psi_{a,\tau}(t) := \frac{1}{\sqrt{|a|}} \psi(\frac{t-\tau}{a}); \quad a \in \mathbb{R}^* = \mathbb{R} \setminus \{0\}, \tau \in \mathbb{R}. \tag{76}$$

Then, the *continuous wavelet transform* (associated with the wavelet ψ) of f is defined by

$$\begin{aligned} \mathcal{W}_\psi f(a,\tau) &= \langle f, \psi_{a,\tau} \rangle \\ &= \frac{1}{\sqrt{|a|}} \int_{\mathbb{R}} f(t) \overline{\psi}(\frac{t-\tau}{a}) dt, \ a \in \mathbb{R}^*, \tau \in \mathbb{R}. \end{aligned} \tag{77}$$

As in the windowed Fourier case, there is an inversion formula and a conservation of energy result, which can be stated as follows:

$$f(t) = \frac{1}{C_\psi} \iint_{\mathbb{R}^2} \mathcal{W}_\psi f(a,\tau) \psi_{a,\tau}(t) \frac{da\, d\tau}{a^2} \tag{78}$$

and

$$\frac{1}{C_\psi} \iint_{\mathbb{R}^2} |\mathcal{W}_\psi f(a,\tau)|^2 \frac{da\, d\tau}{a^2} = \|f\|_2^2. \tag{79}$$

If the function ψ is localized around $t = 0$ and $\hat{\psi}$ is localized around $\omega = 1$, then $\psi_{a,\tau}$ will be localized around τ whilst $\hat{\psi}_{a,\tau}$ will be localized around $\omega = \frac{1}{a}$. When $|a| > 1$ ($|\omega| = |\frac{1}{a}| < 1$), the function $\psi_{a,\tau}$ becomes a stretched version of ψ (less localized in time, more localized in frequency); on the contrary, when $|a| < 1$ ($|\omega| = |\frac{1}{a}| > 1$), $\psi_{a,\tau}$ will be a function more localized in time (a compressed version of ψ) and less localized in frequency; this is the already mentioned flexibility of the wavelet windows: their size naturally adjusts to the frequencies.

11.2 Multiresolution Analysis (MRA)

As usual, we might like to use a discretized version of the wavelet transform, i.e. to compute $W_\psi(a,\tau)$ only for a discrete set of values of a and τ. A very common choice is to take the dyadic points in the plane

$$a = 2^{-j}, \quad \tau = 2^{-j}k; \quad j,k \in \mathbb{Z}. \tag{80}$$

For the above choice of values, we thus consider the family of functions

$$\psi_{j,k}(t) := 2^{j/2}\psi(2^j t - k); \ j,k \in \mathbb{Z} \tag{81}$$

and compute the wavelet values

$$C_{j,k} = \langle f, \psi_{j,k} \rangle. \tag{82}$$

A natural challenge for the earlier researchers was to find ψ such that the corresponding set of functions (81) was an orthonormal basis of $L^2(\mathbb{R})$, in which case every function $f \in L^2(\mathbb{R})$ could be decomposed in a double series

$$f(t) = \sum_{j,k\in\mathbb{Z}} C_{j,k}\psi_{j,k}(t), \tag{83}$$

with the coefficients $C_{j,k}$ given by (82). A function with this property is called an *orthogonal wavelet*. In section VIII, we already mentioned the existence of one such function: the *Haar wavelet* (61). This is, however, a discontinuous function, and the converge of the series (83) is extremely slow. In the 80's, other orthogonal wavelets, with better properties, were discovered by J. O. Strömberg [33], Y. Meyer [34], G. Battle [35] and P. G. Lemarié [36].

These first constructions of wavelets seem a bit "miraculous"; Y. Meyer confesses "*I found my wavelets by trial and error; there was no underlying concept.*" In the end of 1986, Stéphane Mallat, in collaboration with Yves Meyer, introduce the important concept of *multiresolution analysis* (MRA). This structure gives a complete understanding of all the wavelet constructions obtained up to then, and allows the construction of new orthogonal wavelets. It is based on this concept, that I. Daubechies introduces a new class of wavelets (the so called *Daubechies wavelets*) which became of great importance in applications; these wavelets have important properties: they have compact support, are smooth (smoothness increasing with the size of support) and have a certain number of zero moments.

Another important consequence of the introduction of the AMR paradigm was the discovery of efficient computational algorithms for the decomposition and reconstruction of a function in a wavelet basis, the *fast wavelet transforms*.

A *multiresolution analysis* (MRA)(V_j, ϕ) of $L^2(\mathbb{R})$ is a sequence of closed subspaces of $L_2(\mathbb{R})$ and an associated function ϕ, called the *generator* or *scaling function*, satisfying:

1. $V_j \subset V_{j+1}, \quad \forall j \in \mathbb{Z}$
2. $\displaystyle\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$
3. $\displaystyle\overline{\bigcup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R})$
4. $v(t) \in V_j \iff v(2t) \in V_{j+1}$
5. The integer translates of ϕ, $\phi(t-k), k \in \mathbb{Z}$, form an orthonormal basis of the space V_0.

Note: The concept here introduced is sometimes referred as *orthogonal* AMR; in fact, Condition 5. can be replaced by the less stringent assumption that the $\phi(t-k)$ are a Riesz basis of V_0; in that case, an "orthogonalized" function $\phi^{\perp}$ such that $\phi^{\perp}(t-k)$ forms an o.n. basis of V_0 can always been obtained by a well-defined procedure; see, e.g. [32, pp. 139-140].

It follows from the properties of an AMR that, for each j, the set of functions $\{\phi_{j,k} := 2^{j/2}\phi(2^j. - k) : k \in \mathbb{Z}\}$ is an o.n. basis for the space V_j (the so-called nodal basis). Wavelets are associated with *detail spaces*, i.e. with complementary spaces W_j satisfying $V_{j+1} = V_j \oplus W_j$, where $\oplus$ denotes the orthogonal complement of V_j in V_{j+1}. The properties of the multiresolution analysis imply that $\bigoplus_{j\in\mathbb{Z}} W_j = L_2(\mathbb{R})$. Hence, if we can find a function ψ whose integer translates form an o.n. basis of W_0, then the collection $\{\psi_{j,k} := 2^{j/2}\psi(2^j \cdot - k) : j,\ k \in \mathbb{Z}\}$ will be an o.n. basis for the space $L_2(\mathbb{R})$ (a so-called *wavelet basis*), i.e. ψ will be an orthogonal wavelet. The basic principle of a multiresolution analysis is that ψ always exists and can be explicitly determined (from ϕ). In fact, we have the following theorem.

Theorem 14 *Let $(V_j)_{j\in\mathbb{Z}}$ be a MRA of $L^2(\mathbb{R})$ with scaling function ϕ. Then*

1. *there exists a sequence of scalars $(h_k) \in \ell^2(\mathbb{Z})$ such that*

$$\phi(t) = \sqrt{2}\sum_{k\in\mathbb{Z}} h_k \phi(2t-k) \tag{84}$$

2. *the function ψ defined by*

$$\psi(t) = \sqrt{2}\sum_{k\in\mathbb{Z}} g_k \phi(2t-k) \tag{85}$$

where the coefficients g_k are given by

$$g_k = (-1)^k \overline{h}_{1-k} \tag{86}$$

is an orthogonal wavelet., i.e. the set of functions $\{\psi_{j,k}(t) := 2^{j/2}\psi(2^j t - k),\ j,k \in \mathbb{Z}\}$ is an o.n. basis of $L^2(\mathbb{R})$.

Notes:

1. Equation (84), which is known as the *refinement equation* or the *two-scale equation* for the scaling function ϕ follows immediately by observing that $\sqrt{2}\phi(2t-k)$ is an o.n. basis of V_1 and hence the function $\phi \in V_0 \subset V_1$ must have a representation in that basis.

2. The sequence of coefficients $(h_k)_k \in \mathbb{Z}$ in (84) is called the *filter* of ϕ. These coefficients are, naturally, given by

$$h_k = \langle f, \phi_{1,k}\rangle = \sqrt{2}\int_{\mathbb{R}} f(t)\overline{\phi(2t-k)}dt. \tag{87}$$

3. tThere are other possible ways to define the coefficients g_k (in terms of h_k) so that (85) is an orthogonal wavelet; the different wavelets are, however, all closely related to each other; further details can be seen, e.g. in [32, pp. 135-136].

11.3 Fast Wavelet Transforms

We now show how the MRA structure leads to a very efficient iterative scheme for computing the coefficients of the expansion of a function f in a wavelet basis.

Let $(V_j)_{j\in\mathbb{Z}}$ be a MRA of $L^2(\mathbb{R})$, with scaling function ϕ and corresponding wavelet ψ. Properties 1. and 2. of the MRA show that any function $f \in L^2(\mathbb{R})$ can be arbitrarily well approximated by a function v_j in a certain space V_j, provided j is taken sufficiently large, i.e.

$$\forall \epsilon > 0 \quad \exists J \in \mathbb{Z} \quad \exists v_J \in V_J : \quad \|f - v_J\|_2 < \epsilon. \tag{88}$$

Let, as before, denote by W_j the orthogonal complement of V_j into V_{j+1} and let P_j and Q_j denote the orthogonal projectors of $L^2(\mathbb{R})$ into V_j and W_j, respectively; since $V_j \subset V_{j+1}$, we have that $Q_j = P_{j+1} - P_j$. Moreover, $P_j P_{j+1} = P_j$ and $Q_j P_{j+1} = Q_j$.

For each j, let v_j and w_j be the projections of f into V_j and W_j, respectively, i.e. let v_j and w_j be given by

$$v_j = P_j f \qquad \text{and} \qquad w_j = Q_j f. \tag{89}$$

We thus have,

$$\begin{aligned} v_J = P_J f &= P_{J-1} f + (P_J - P_{J-1}) f \\ &= v_{J-1} + w_{J-1} \\ &= v_{J-2} + w_{J-2} + w_{J-1} \\ &= \cdots = v_{J-M} + w_{J-M} + \cdots + w_{J-1}, \qquad M > 0, \end{aligned} \tag{90}$$

Property 2. of AMR ensures that, provided M is sufficiently large, one has

$$\|v_{J-M}\|_2 < \epsilon. \tag{91}$$

We can therefore conclude that any function in $L^2(\mathbb{R})$ can be reasonably well represented as a finite sum of functions belonging to the subspaces W_j and a remainder v_{J-M} in a space V_{J-M} which can be interpreted as a very coarse version of f. The decomposition (90) tells us the *details* that must be added to this *blurred* version of f to obtain the *fine* approximation v_J to f.

Let us assume that we know the approximation $v_J = P_J f \in V_J$ to f and that we want to obtain the decomposition (90). Since, for every j, $\{\phi_{j,k} : k \in \mathbb{Z}\}$ and $\{\psi_{j,k} : k \in \mathbb{Z}\}$ are o.n. bases of V_j and W_j, respectively, to know the functions v_J and $v_{J-M}, w_{J-M}, \ldots, w_{J-1}$, is equivalent to know their coefficients in these bases. Let $\boldsymbol{c}^j = (c_k^j)_{k\in\mathbb{Z}}$ be the sequence of the coefficients of $v_j = P_j f$ in the basis $\{\phi_{j,k} : k \in \mathbb{Z}\}$, i.e. let

$$c_k^j = \langle f, \phi_{j,k} \rangle, \quad k \in \mathbb{Z}, \tag{92}$$

and let $\boldsymbol{d}^j = (d_k^j)_{k\in\mathbb{Z}}$ be the sequence of the coefficients of $w_j = Q_j f$ in the basis $\{\psi_{jk} : k \in \mathbb{Z}\}$, i.e. let

$$d_k^j = \langle f, \psi_{jk} \rangle, \quad k \in \mathbb{Z}. \tag{93}$$

Hence, we aim to obtain the decomposition

$$v_J = \sum_{k\in\mathbb{Z}} c_k^{J-M} \phi_{J-M,k} + \sum_{j=J-M}^{J-1} \sum_{k\in\mathbb{Z}} d_k^j \psi_{j,k}. \tag{94}$$

Recall that ϕ satisfies the dilation equation, i.e. that

$$\phi(t) = \sum_{n\in\mathbb{Z}} h_n \phi_{1,n}(t)$$

Hence, we have

$$\begin{aligned}
\phi_{j-1,k}(t) &= 2^{(j-1)/2}\phi(2^{j-1}t - k) \\
&= 2^{(j-1)/2}\sum_{n\in\mathbb{Z}} h_n \phi_{1,n}(2^{j-1}t - k) \\
&= 2^{j/2}\sum_{n\in\mathbb{Z}} h_n \phi(2^j t - (2k+n)) \\
&= \sum_{n\in\mathbb{Z}} h_n \phi_{j,2k+n}(t) \\
&= \sum_{n\in\mathbb{Z}} h_{n-2k} \phi_{j,n}(t). \qquad (95)
\end{aligned}$$

Thus, one gets

$$\begin{aligned}
c_k^{j-1} &= \langle f, \phi_{j-1,k}\rangle \\
&= \langle f, \sum_{n\in\mathbb{Z}} h_{n-2k}\phi_{j,n}\rangle \\
&= \sum_{n\in\mathbb{Z}} \overline{h_{n-2k}}\,\langle f, \phi_{j,n}\rangle \\
&= \sum_{n\in\mathbb{Z}} \overline{h_{n-2k}}\,c_n^j. \qquad (96)
\end{aligned}$$

In a totally similar manner, by making use of the equations (85) and (86), one gets

$$d_k^{j-1} = \sum_{n\in\mathbb{Z}} \overline{g_{n-2k}}\,c_n^j. \qquad (97)$$

Starting from the sequence $\boldsymbol{c}^J = (c_n^J)$, formulae (96) and (97)above can be used, recursively, to obtain the sequences $\boldsymbol{c}^{J-M}, \boldsymbol{d}^{J-1}, \ldots, \boldsymbol{d}^{J-M}$, i.e. to obtain the desired decomposition for v_j; see the scheme in Figure 1.

$$\begin{tikzcd}
\boldsymbol{c}^J \arrow[r] \arrow[dr] & \boldsymbol{c}^{J-1} \arrow[r] \arrow[dr] & \boldsymbol{c}^{J-2} \arrow[r] & \cdots \arrow[r] \arrow[dr] & \boxed{\boldsymbol{c}^{J-M}} \\
 & \boxed{\boldsymbol{d}^{J-1}} & \boxed{\boldsymbol{d}^{J-2}} & & \boxed{\boldsymbol{d}^{J-M}}
\end{tikzcd}$$

Figure 1: Decomposition Scheme

The above transform can be easily inverted; starting from the sequences $\boldsymbol{c}^{J-M}, \boldsymbol{d}^{J-1}, \ldots, \boldsymbol{d}^{J-M}$, we can obtain the initial sequence of coefficients $\boldsymbol{c}^J$. We have, for each j,

$$\begin{aligned} P_j f &= v_j = v_{j-1} + w_{j-1} \\ &= \sum_{l\in\mathbb{Z}} c_l^{j-1}\phi_{j-1,l} + \sum_{l\in\mathbb{Z}} d_l^{j-1}\psi_{j-1,l} \end{aligned}$$

Therefore,

$$\begin{aligned} c_k^j &= \langle f, \phi_{j,k}\rangle \\ &= \langle P_j f, \phi_{j,k}\rangle \\ &= \sum_{l\in\mathbb{Z}} c_l^{j-1}\langle \phi_{j-1,l}, \phi_{j,k}\rangle + \sum_{l\in\mathbb{Z}} d_l^{j-1}\langle \psi_{j-1,l}, \phi_{j,k}\rangle. \end{aligned} \tag{98}$$

But,

$$\begin{aligned} \langle \phi_{j-1,l}, \phi_{j,k}\rangle &= \langle \sum_{n\in\mathbb{Z}} h_{n-2l}\phi_{j,n}, \phi_{j,k}\rangle \\ &= \sum_{n\in\mathbb{Z}} h_{n-2l}\langle \phi_{j,n}, \phi_{j,k}\rangle \\ &= \sum_{n\in\mathbb{Z}} h_{n-2l}\delta_{n,k} = h_{k-2l}. \end{aligned} \tag{99}$$

On the other hand,

$$\langle \psi_{j-1,l}, \phi_{j,k}\rangle = \langle \sum_{n\in\mathbb{Z}} g_{n-2l}\phi_{j,n}, \phi_{j,k}\rangle = g_{k-2l}. \tag{100}$$

Hence, we get

$$\begin{aligned} c_k^j &= \sum_{l\in\mathbb{Z}} h_{k-2l}c_l^{j-1} + \sum_{l\in\mathbb{Z}} g_{k-2l}d_l^{j-1} \\ &= \sum_{l\in\mathbb{Z}} \left(h_{k-2l}c_l^{j-1} + g_{k-2l}d_l^{j-1} \right); \end{aligned} \tag{101}$$

see the scheme in Fig.2.

$$\begin{tikzcd}
\boldsymbol{d}^{J-M} \arrow[dr] & \boldsymbol{d}^{J-M+1} \arrow[dr] & & \boldsymbol{d}^{J-1} \arrow[dr] & \\
\boldsymbol{c}^{J-M} \arrow[r] & \boldsymbol{c}^{J-M+1} \arrow[r] & \cdots \arrow[r] & \boldsymbol{c}^{J-1} \arrow[r] & \boxed{\boldsymbol{c}^J}
\end{tikzcd}$$

Figure 2: Reconstruction Scheme

Notes

1. Naturally, when implementing the algorithms, all the infinite sequences have to be truncated. Hence, when we apply the decomposition scheme, the initial sequence is always a finite sequence, i.e. a vector of certain length N, $(c_0^J, c_1^J, \ldots, c_{N-1}^J)$. Also, either the filter $(h_k)_{k\in\mathbb{Z}}$ is finite or, if we are working with wavelets that do not have compact support, will have to be truncated for a vector of a certain size L: $(h_{-m}, h_{-m+1}, \ldots, h_{-m+L-1})$.

2. Since the initial sequence has finite length it is necessary to know how to deal with the boundary points. For example, the formulae for c_0^{J-1} and $c_{N/2-1}^{J-1}$ are

$$c_0^{J-1} = \sum_{n=-m}^{-m+L-1} \overline{h_n} c_n^J$$

and

$$c_{N/2-1}^{J-1} = \sum_{n=-m+N-2}^{n=N+L-m-3} \overline{h_{n-N+2}} c_n^J$$

Hence, it is necessary to add m components at the left of the vector $\boldsymbol{c}^J$ and $L-2-m$ components at the end. This can be done in several ways; see, e.g. [37, pp. 282-290] for a discussion on different choices of these boundary conditions.

3. With an appropriate choice of the boundary conditions, formulae (96) and (97) show that in the first step of the decomposition we compute approximately $N/2$ coefficients c_k^{J-1} and $N/2$ coeficientes d_k^{J-1}. The next decomposition step is only applied to the coefficients c_k^{J-1} which represnt the part in V_{J-1} and so on. Hence, as the decomposition proceeds, less operations are involved If the filter length is L, the number of operations involved is of the order of

$$L \times \left(N + \frac{N}{2} + \frac{N}{4} + \cdots\right) < 2NL.$$

Hence, the number of operations involved in the fast wavelet transform is $O(N)$; cf. with $O(N \log N)$ for the FFT.

There are many important variants of the basic wavelet theory. Since it is impossible to present here a reasonable description (even very brief) of these variants, we just refer to some of these developments and indicate some references for the interested reader:

- Biorthogonal wavelets, introduced by Cohen, Daubechies and Feaveau in [38];

- *wavelet-packets*, introduced in [39] and applied in signal compression in [40]; we also recommend the book by Wickerhauser [41] and the article [42];
- Wilson bases– [43];
- local sine and co-sine bases – [44], [45];
- multiwavelets – [46];
- interpolatory wavelets – [47];
- lifting scheme and second generation wavelets – [48, 49, 50].

12 Conclusion

The idea of transforming or decomposing an object (e.g. a function) in order to extract more "relevant" (for a specific purpose) information, and then reconstituting it, pervades all the areas of mathematics. This makes the subject of *mathematical transforms* extremely vast and impossible to cover, even in condensed form, in a set of notes. We were, therefore, forced to make a personal selection of topics. Our idea has been to focus on the most popular transforms, having also in mind their relevance in applied areas, such as signal processing.

We sincerely hope that these notes can be useful as a quick reference and a starting point for studying, more deeply, this fascinating area of mathematics.

References

[1] J. J. Benedetto, *Harmonic Analysis and Applications.* New York: CRC Press, 1997.

[2] C. Gasquet and P. Witomski, *Fourier Analysis and Applications.* New York: Springer-Verlag, 1998.

[3] R. O. Gandulfo, "Séries de Fourier e convergência," *Matemática Universitária*, no. 11, pp. 27–52, 1990.

[4] H. Wilbraham *Camb. and Dublin Math. J.*, vol. 3, p. 198, 1848.

[5] M. Bôcher *Annals of Math.*, vol. 7, 1906.

[6] J. S. Walker, *Fast Fourier Transforms.* Boca Raton: CRC Press, 1996.

[7] K. R. Castleman, *Digital Image Processing.* New Jersey: Prentice-Hall, 1996.

[8] P. Halmos, *Mesaure Theory.* Princeton, N.J.: Van Nostrand, 1950.

[9] H. Loomis, *An Introduction to Abstact Harmonic Analysis.* New York: Van Nostrand, 1953.

[10] W. Rudin, *Fourier Analysis on Groups.* New York: John Wiley & Sons, 1990.

[11] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. of. Comput.*, vol. 19, no. 90, pp. 297–301, 1965.

[12] M. T. Heideman, D. H. Johnson, and C. S. Burrus, "Gauss and the history of the Fast Fourier Transform," *IEEE ASSP Magazine*, pp. 14–21, 1984.

[13] J. W. Cooley, "How the FFT gained acceptance," *IEEE Signal Processing Magazine*, vol. 9, pp. 10–13, 1990.

[14] E. O. Brigham, *The Fast Fourier Transform and Its Applications.* New Jersey: Prentice-Hall, 1988.

[15] O. Ersoy, ed., *Fourier Realated Transfroms, Fast Algorithms and Applications.* New Jersey: Prentice Hall, 1997.

[16] W. H. Press, B. P. Flannery, S. A. Teukolski, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing.* Cambridge: Cambridge University Press, 2 ed., 1992. There are editions of this book with computer programs in Fortran, Pascal or C.

[17] C. S. Burrus, "Notes on the FFT." Department of Electrical and Computer Engineering, Rice University, September 1997.

[18] H. V. Sorensen, C. S. Burrus, and M. T. Heideman, *Fast Fourier Transforms Database.* Boston: PWS Publishing, 1995.

[19] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. on Computers*, no. C-23, pp. 90–93, 1974.

[20] R. V. Hartley, "A more simmetrical Fourier analysis applied to transmission problems," in *Proceedings of the IRE*, vol. 30, pp. 142–150, 1942.

[21] R. N. Bracewell, "Discrete Hartley Transform," *J. Opt. Soc. Amer.*, vol. 73, pp. 1832–1835, 1983.

[22] R. Bracewell, *The Hartley Transform.* New York: Oxford Press, 1987.

[23] G. Doetsch, *Introduction to the Theory and Application of the Laplace Transform.* Berlin: Springer-Verlag, 1974.

[24] R. E. Belmman and R. S. Roth, *The Laplace Transform.* Singapore: World-Scientific, 1984.

[25] E. I. Jury, *Theory and Applications of the z-Transfrm.* New York: John Wiley & Sons, 1964.

[26] A. V. Oppenheim and R. W. Shafer, *Discrete-Time Signal Processing.* Englewood Cliffs, New Jersey: Prentice Hall, 1989.

[27] R. Bracewell, *The Fourier Transform and its Applications.* New York: McGraw-Hill, 1965.

[28] L. Cohen, *Time-Frequency Analysis.* Englewood Cliffs, New Jersey: Prentice-Hall, 1995.

[29] K. G. Beauchamp, *Applications of Walsh and Related Functions.* London: Academic Press, 1984.

[30] A. Haar, "Zur theorie der orthogonalen Funktionen Systeme," *Mat. Ann.*, vol. 69, pp. 331–371, 1910.

[31] F. J. Harris, "On the use of windows for Harmonic Analysis with the Discrete Fourier Transform," *Proc. IEEE*, vol. 66, pp. 51–83, 1978.

[32] I. Daubechies, *Ten Lectures on Wavelets*, vol. 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics.* Philadelphia: SIAM, 1992.

[33] J. O. Strömberg, "A modified Franklin system and higher order spline systems on $\mathbb{R}^n$ as unconditional bases for Hardy spaces," in *Conf. in honor of Antoni Zygmund* (W. Beckner, A. P. Calderón, R. Fefferman, and P. W. Jones, eds.), vol. II, (New York), pp. 475–493, Wadsworth Math. Series, 1981.

[34] Y. Meyer, "Principe d'incertitude, bases Hilbertiennes et algèbres d'opérateurs," *Séminaire Bourbaki*, vol. 662, pp. 1–15, 1985-1986.

[35] G. Battle, "A block spin construction of ondelettes. Part I: Lemarié functions," *Comm. Math. Phys.*, vol. 110, pp. 601–615, 1987.

[36] P. G. Lemarié, "Une nouvelle base d' ondelettes de $L^2(\mathbb{R}^n)$," *J. de Mat. Pures et Appl.*, vol. 67, pp. 227–236, 1988.

[37] S. Mallat, *A Wavelet Tour of Signal Processing.* New York: Academic Press, 1998.

[38] A. Cohen, I. Daubechies, and J.-C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Comm. Pure Appl. Math.*, vol. 45, pp. 486–560, 1992.

[39] R. Coifman, Y. Meyer, S. Quake, and M. V. Wickerhauser, "Signal processing and compression with wavelt packets," in *Proceedings of the International Conference on Wavelets, Marseille* (Y. Meyer and S. Roques, eds.), (Paris), Masson, 1989.

[40] M. V. Wickerahauser, "Acoustic signal compression with wavelet-packets," in *Wavelts: A Tutorial of Theory and Applications* (C. K. Chui, ed.), vol. 2 of *Wavelet Analysis and Its Applications*, pp. 679–700, Boston: Academic Press, 1992.

[41] M. V. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software.* Massachusets: A K Peters, 1994.

[42] B. Torrésani, "Time-frequency representations: Wavelet packets and optimal decomposition," *Ann. Inst. H. Poincaré, Physique Théorique*, vol. 56, pp. 215–234, 1992.

[43] I. Daubechies, S. Jaffard, and J. Journé, "A simple Wilson orthonormal basis with exponential decay," *SIAM J. of Math. Anal.*, vol. 22, pp. 554–572, 1991.

[44] R. Coifman and Y. Meyer, "Remarques sur l'analyse de Fourier à fenêtre," *C. R. Acad. Sci.*, vol. 4312, pp. 259–261, 1991.

[45] P. Auscher, G. Weiss, and M. V. Wickerhauser, "Local sine and cosine bases of Coifman and Meyer and the construction of smooth bases," in *Wavelets: A Tutorial in Theory and Applications* (C. K. Chui, ed.), vol. 2 of *Wavelet Analysis and Its Applications*, pp. 181–216, Boston: Academic Press, 1992.

[46] J. Geronimo, D. Hardin, and P. Massopust, "Fractal functions and wavelet expansions based on several scaling functions," *J. Approx. Theory*, vol. 78, pp. 373–401, 1994.

[47] D. L. Donoho, "Interpolating wavelet transforms," *Appl. Comp. Harm. Anal.*, 1994.

[48] W. Sweldens, "The lifting scheme: a new philosophy in biorthogonal wavelet constructions," in *Wavelet Applications in Signal and Image Processing III*, vol. 2569, pp. 68–79, SPIE, 1995.

[49] W. Sweldens, "The lifting scheme: a custom-design construction of biorthogonal wavelets," *Appl. Comp. Harmon. Anal*, vol. 3, no. 2, pp. 186–200, 1996.

[50] W. Sweldens, "The lifting scheme: a construction of second generation wavelets," *SIAM J. Math. Anal*, vol. 29, no. 2, pp. 51–546, 1997.

Propagación y aproximación numérica de ondas: Una introducción

Enrique Zuazua[*]

Resumen

En estas notas introducimos la ecuación de transporte y la ecuación de ondas en una dimensión espacial. Se trata de modelos sumamente simples de propagación de ondas pero tales que muchas de sus propiedades cualitativas más importantes son ubicuas en prácticamente todos los modelos más sofisticados. Realizamos un análisis cuidadoso de sus propiedades cualitativas más importantes para después abordar el problema del análisis numérico a través de la transformada discreta de Fourier. La dificultad de los esquemas discretos para reproducir las propiedades de propagación de los modelos continuous a altas frecuencias queda claramente de manifiesto a través del estudio cuidadoso de las velocidades de fase y de grupo que ilustran la dispersión numérica que estos esquemas introducen.

1. La ecuación de ondas

En esta sección mencionamos la ecuación de ondas y sus variantes y algunos contextos de la Mecánica y de la Ingeniería en los que intervienen. Normalmente cuando nos referimos a la ecuación de ondas la incógnita es una función escalar $u = u(x, t)$ que depende tanto del espacio $x = (x_1, \cdots, x_n) \in \mathbf{R}^n$ como de la variable tiempo $t \in \mathbf{R}$. En las aplicaciones más relevantes las dimensiones que intervienen son $n = 1, 2$ y 3. *La ecuación de ondas* es

$$u_{tt} - \Delta u = 0, \tag{1.1}$$

donde $u_t = \partial u / \partial t$ denota la derivación temporal y Δ es el clásico operador de Laplace:

$$\Delta = \sum_{i=1}^{n} \frac{\partial^2}{\partial x_i^2}. \tag{1.2}$$

La ecuación de ondas en dimensiones $n = 1$ y 2 describe las vibraciones de cuerdas y membranas mientras que en $n = 3$ es válida para la propagación de ondas acústicas.

[*] DM Universidade Autonoma, Madrid. E-mail:enrique.zuazua@uam.es

Para simplificar la presentación en esta sección introduciremos las ecuaciones en su forma más sencilla. En particular, supondremos que los coeficientes son constantes (lo cual equivale a suponer que el medio considerado es homogéneo) y los normalizamos al valor unidad, lo cual en este caso no supone ninguna pérdida de generalidad como se puede comprobar mediante una simple dilatación/contracción de la variable temporal o espacial.

En el ámbito de las frecuencias, como es habitual en acústica y en el estudio de las vibraciones, la ecuación de ondas puede también reducirse a *la ecuación de Helmholtz*

$$-\Delta u = \lambda u. \tag{1.3}$$

La ecuación de transporte lineal

$$u_t + \sum_{i=1}^{n} b_i u_{x_i} = 0 \tag{1.4}$$

y *la ecuación de Liouville*

$$u_t - \sum_{i=1}^{n} (b_i u)_{x_i} = 0 \tag{1.5}$$

están también intimamente ligadas a la ecuación de ondas. En efecto, en una dimensión espacial, la ecuación de ondas

$$u_{tt} - u_{xx} = 0 \tag{1.6}$$

puede también escribirse como

$$(\partial_t + \partial_x)(\partial_t - \partial_x) u = 0, \tag{1.7}$$

o

$$(\partial_t - \partial_x)(\partial_t + \partial_x) u = 0, \tag{1.8}$$

o, lo que es lo mismo, el operador de d'Alembert

$$\partial_t^2 - \partial_x^2 \tag{1.9}$$

puede factorizarse de las dos siguientes maneras

$$\partial_t^2 - \partial_x^2 = (\partial_t + \partial_x)(\partial_t - \partial_x) = (\partial_t - \partial_x)(\partial_t + \partial_x). \tag{1.10}$$

Vemos pues que el operador de d'Alembert es la composición de dos operadores de transporte.

Conviene también señalar que, cuando los coeficientes b_i son constantes, la ecuación de transporte y de Liouville sólo difieren en un signo, diferencia que puede ser eliminada invirtiendo el sentido del tiempo.

Utilizando las notaciones habituales

$$\begin{aligned} \nabla u &= (\partial_1 u, \cdots, \partial_n u) && (1.11)\\ \operatorname{div} \vec{u} &= \sum_{i=1}^{n} \frac{\partial u_i}{\partial x_i} && (1.12) \end{aligned}$$

para los operadores gradiente y divergencia y denotando mediante $\cdot$ el producto escalar en $\mathbf{R}^n$ las ecuaciones de transporte y Liouville se pueden escribir respectivamente como

$$u_t + \vec{b} \cdot \nabla u = 0 \tag{1.13}$$

y

$$u_t - \operatorname{div}\left(\vec{b}u\right) = 0. \tag{1.14}$$

La *ecuación de Schrödinger* de la Mecánica Cuántica, que también interviene en el estudio de fibras ópticas es también una ecuación que, en muchos sentidos, se asemeja a la ecuación de ondas:

$$iu_t + \Delta u = 0. \tag{1.15}$$

En este caso, la incógnita u toma valores complejos.

La *ecuación de las placas vibrantes*

$$u_{tt} + \Delta^2 u = 0 \tag{1.16}$$

es también muy similar a la ecuación de ondas. Además puede factorizarse en dos operadores de Schrödinger conjugados

$$\partial_t^2 + \Delta^2 = -\left(i\partial_t + \Delta\right)\left(i\partial_t - \Delta\right). \tag{1.17}$$

En una dimensión espacial la ecuación

$$\partial_t^2 + \partial_x^4 u = 0 \tag{1.18}$$

describe las vibraciones de una viga.

Las siguientes son también variantes de la ecuación de ondas:

$$u_{tt} - u_{xx} + d\, u_t = 0 \quad \textit{(ecuación del telégrafo)} \tag{1.19}$$

$$u_t + u_{xxx} = 0 \quad \textit{(ecuación de Airy)}, \tag{1.20}$$

$$u_{tt} - \Delta u + u = 0 \quad \textit{(ecuación de Klein-Gordon)}, \tag{1.21}$$

El *sistema de Lamé* para las vibraciones de un cuerpo tridimensional elástico puede también entenderse como un sistema de ecuaciones de ondas acopladas:

$$u_{tt} - \lambda\Delta u - (\lambda + \mu)\nabla \operatorname{div} u = 0. \tag{1.22}$$

En este caso la incógnita u es un vector de tres componentes $u = (u_1, u_2, u_3)$ que describe las deformaciones del cuerpo elástico.

Las ecuaciones que hemos descrito son *lineales* y provienen de ecuaciones y sistemas más complejos de la Mecánica, de carácter no-lineal, a través de linealizaciones, lo cual las hace válidas sólo para pequeños valores de la incógnita u.

El *sistema de Maxwell* para las ondas electromagnéticas posee también muchas de las características de las ecuaciones de ondas:

$$\left\{ \begin{aligned} E_t &= \operatorname{rot} B \\ B_t &= -\operatorname{rot} E \\ \operatorname{div} B &= \operatorname{div} E = 0. \end{aligned} \right. \tag{1.23}$$

Aqui rot denota el rotacional de un campo de vectores.

Con el objeto de entender la semejanza de este sistema con la ecuación de ondas (1.6) conviene observar que esta última también puede escribirse en la forma de un sistema hiperbólico de ecuaciones de orden uno:

$$\left\{ \begin{aligned} u_t &= v_x \\ v_t &= u_x. \end{aligned} \right. \tag{1.24}$$

Sin embargo, muchas ecuaciones relevantes que intervienen en el estudio de las ondas tienen un carácter no-lineal. Por ejemplo, *la ecuación eikonal*

$$\mid \nabla u \mid = 1 \tag{1.25}$$

interviene en el cálculo de soluciones de ecuaciones de ondas mediante métodos de la Óptica Geométrica.

Lo mismo ocurre con *ecuación de Hamilton-Jacobi*:

$$u_t + H(\nabla u, x) = 0. \tag{1.26}$$

La ecuación de Korteweg-de Vries (KdV) es una versión no-lineal de la ecuación de Airy que permite analizar la propagación de ondas en canales:

$$u_t + uu_x + u_{xxx} = 0 \tag{1.27}$$

y da lugar a los célebres *solitones*.

En el contexto de la Mecánica de Fluidos los dos ejemplos más relevantes son sin duda *las ecuaciones de Navier-Stokes* para un fluido viscoso incompresible

$$\left\{ \begin{array}{l} u_t - \Delta u + u \cdot \nabla u = \nabla p \\ \operatorname{div} u = 0 \end{array} \right. \tag{1.28}$$

y *las ecuaciones de Euler* para fluidos perfectos

$$\left\{ \begin{array}{l} u_t + u \cdot \nabla u = \nabla p \\ \operatorname{div} u = 0. \end{array} \right. \tag{1.29}$$

En estos sistemas u denota el campo de velocidades del fluido y p es la presión.

Las siguientes ecuaciones, conocidas como *de Burgers* viscosa e inviscida son, en algún sentido, versiones unidimensionales de estas ecuaciones

$$u_t + uu_x - u_{xx} = 0, \tag{1.30}$$

$$u_t + uu_x = 0. \tag{1.31}$$

En esta última las soluciones desarrollan ondas de choque en tiempo finito.

Las ecuaciones que hemos citado, aunque numerosas, no son más que algunos de los ejemplos más relevantes de ecuaciones en las que intervienen de un modo u otro fenómenos de propagación de ondas y en las que los contenidos que desarrollaremos en este curso resultarán de utilidad.

2. La fórmula de D'Alembert

Consideramos la ecuación de ondas unidimensional $(1-d)$ en toda la recta real

$$u_{tt} - u_{xx} = 0,\ x \in \mathbf{R},\ t > 0. \tag{2.1}$$

D'Alembert observó que las soluciones de (2.1) pueden escribirse como superposición de dos ondas de transporte

$$u(x,t) = f(x+t) + g(x-t). \tag{2.2}$$

Es fácil comprobar que toda función de la forma (2.2) es solución de (2.1).

La fórmula (2.2) muestra que la velocidad de propagación en el modelo (2.1) es uno. En efecto, según (2.2), las soluciones de (2.1) son superposición de ondas de transporte que viajan en el espacio $\mathbf{R}$ a velocidad uno a izquierda y derecha.

Para comprobar que toda solución de (2.1) es de la forma (2.2) basta observar que el operador de d'Alembert $\partial_t^2 - \partial_x^2$ puede descomponerse del modo siguiente:

$$u_{tt} - u_{xx} = (\partial_t - \partial_x)(\partial_t + \partial_x) u = 0. \tag{2.3}$$

Introduciendo la variable auxiliar

$$v = (\partial_t + \partial_x) u \tag{2.4}$$

la ecuación se escribe como

$$(\partial_t - \partial_x) v = v_t - v_x = 0 \tag{2.5}$$

de modo que

$$v = h(x+t) \tag{2.6}$$

La ecuación (2.4) se reduce entonces a

$$u_t + u_x = h(x+t). \tag{2.7}$$

Para su resolución observamos que la función

$$w(t) = u(t + x_0, t)$$

verifica

$$w'(t) = h(2t + x_0)$$

cuya solución es

$$w(t) = \frac{H(2t+x_0)}{2} + w(0) = \frac{H(2t+x_0)}{2} + u(x_0, 0), \tag{2.8}$$

donde H es una primitiva de h.

Por lo tanto, como

$$u(x,t) = w(t)$$

con $x_0 = x - t$ obtenemos

$$u(x,t) = \frac{H(x+t)}{2} + u(x-t,\, 0) \tag{2.9}$$

lo cual confirma la expresión (2.2).

Esta fórmula permite calcular explícitamente la solución del problem de Cauchy:

$$\begin{cases} u_{tt} - u_{xx} = 0, & x \in \mathbf{R}, \quad t > 0 \\ u(x,0) = \varphi(x),\, u_t(x,0) = \psi(x), & x \in \mathbf{R}. \end{cases} \tag{2.10}$$

En efecto, en vista de la expresión (2.2), e identificando los perfiles f y g en función de los datos iniciales φ y ψ obtenemos que

$$u(x,t) = \frac{\varphi(x+t) + \varphi(x-t)}{2} + \frac{1}{2}\int_{x-t}^{x+t} \psi(y)dy \tag{2.11}$$

es la única solución de (2.10).

3. Resolución de la ecuación de ondas mediante series de Fourier

Consideramos la ecuación de ondas unidimensional $(1-d)$:

$$\begin{cases} u_{tt} - u_{xx} = 0, & 0 < x < \pi, \quad t > 0 \\ u(0,t) = u(\pi,t) = 0, & t > 0 \\ u(x,0) = u_0(x), u_t(x,0) = u_1(x), & 0 < x < \pi. \end{cases} \tag{3.1}$$

Se trata de un modelo sencillo para las vibraciones de una cuerda unidimensional flexible de longitud π, fijada en sus extremos $x = 0, \pi$.

Es fácil representar las soluciones de (3.1) mediante series de Fourier. Para ello escribimos el desarrollo de Fourier de los datos iniciales:

$$u_0(x) = \sum_{k=1}^{\infty} a_k \operatorname{sen}(kx),\ u_1(x) = \sum_{k=1}^{\infty} b_k \operatorname{sen}(kx) \tag{3.2}$$

donde los coeficientes de Fourier vienen dados por las clásicas fórmulas:

$$a_k = \frac{2}{\pi} \int_0^{\pi} u_0(x) \operatorname{sen}(kx) dx;\ b_k = \frac{2}{\pi} \int_0^{\pi} u_1(x) \operatorname{sen}(kx) dx, \quad k \geq 1. \tag{3.3}$$

La solución de (3.1) viene entonces dada por la fórmula

$$u(x,t) = \sum_{k=1}^{\infty} \left(a_k \cos(kt) + \frac{b_k}{k} \operatorname{sen}(kt) \right) \operatorname{sen}(kx). \tag{3.4}$$

Conviene observar que la evolución temporal de cada uno de los coeficientes de Fourier

$$u_k(t) = a_k \cos(kt) + \frac{b_k}{k} \operatorname{sen}(kt) \tag{3.5}$$

obedece la ecuación del muelle

$$u_k'' + k^2 u_k = 0. \tag{3.6}$$

Para cada una de estas ecuaciones la energía

$$e_k(t) = \frac{1}{2} \left[\mid u_k'(t) \mid^2 + k^2 \mid u_k(t) \mid^2 \right] \tag{3.7}$$

se conserva en tiempo.[1]

Superponiendo cada una de las leyes de conservación de las energías e_k, $k \geq 1$, de las diferentes componentes de Fourier de la solución obtenemos la ley de conservación de la energía de las soluciones de (3.1):

$$E(t) = \frac{1}{2} \int_0^{\pi} \left[|u_x(x,t)|^2 + |u_t(x,t)|^2 \right] dx. \tag{3.8}$$

Se trata de la energía total de la vibración, suma de la energía potencial y de la energía cinética.

Se cumple efectivamente que

$$E(t) = E(0),\ \forall t \geq 0 \tag{3.9}$$

para las soluciones de (3.1).

Esta ley de conservación de energía puede probarse de, al menos, dos modos distintos:

[1] Para comprobarlo basta multiplicar (3.6) por u_k' y observar que $u_k'' u_k' = \frac{1}{2} \left((u_k')^2 \right)'$ y $u_k u_k' = \frac{1}{2} \left(u_k^2 \right)'$.

- *Series de Fourier:*

Si utilizamos las propiedades clásicas de ortogonalidad de las funciones trigonométricas

$$\int_0^\pi \operatorname{sen}(kx)\operatorname{sen}(jx)dx = \frac{\pi}{2}\delta_{jk}, \int_0^\pi \cos(kx)\cos(jx) = \frac{\pi}{2}\delta_{jk}, \tag{3.10}$$

donde δ_{jk} denota la delta de Kronecker, la ley de conservación (3.9) se deduce efectivamente de la conservación de las energías e_k de (3.7) para cada $k \geq 1$.

- *Método de la energía:*

La ley de conservación (3.9) puede también obtenerse directamente de (3.1). Basta para ello multiplicar por u_t e integrar con respecto a $x \in (0, \pi)$. Tenemos entonces

$$\int_0^\pi (u_{tt} - u_{xx})\, u_t dx = 0.$$

Por otra parte,

$$\int_0^\pi u_{tt}u_t dx = \frac{1}{2}\frac{d}{dt}\int_0^\pi |u_t(x,t)|^2\, dx$$

y

$$-\int_0^\pi u_{xx}u_t dx = \int_0^\pi u_x u_{xt} dx = \frac{1}{2}\frac{d}{dt}\int_0^\pi |u_x(x,t)|^2\, dx.$$

En la última identidad hemos utilizado la fórmula de integración por partes y las condiciones de contorno de modo que, como $u(\cdot, t) = 0$ para $x = 0, \pi$, necesariamente también se tiene $u_t(\cdot, t) = 0$ para $x = 0, \pi$.

Los argumentos anteriores son formales pero la ley de conservación y la estructura de la energía E en (3.8) indican en realidad cual es el espacio natural para resolver la ecuación de ondas. En efecto, se trata del espacio de Hilbert, también denominado espacio de energía,

$$H = H_0^1(0, \pi) \times L^2(0, \pi). \tag{3.11}$$

La norma natural en este espacio es

$$|(f, g)|_H = \left[\| f \|^2_{H_0^1(0,\pi)} + \| g \|^2_{L^2(0,\pi)}\right]^{1/2} = \left[\int_0^\pi \left(f_x^2 + g^2\right) dx\right]^{1/2}. \tag{3.12}$$

Conviene observar que, salvo un factor multiplicativo 1/2 la energía E coindice con el cuadrado de la norma H de (u, u_t).

Deducimos que la norma H de la solución[2] (u, u_t) se conserva a lo largo del tiempo. Esto sugiere que H es el espacio natural para resolver el sistema (3.1). Esto es así y se tiene el siguiente resultado de existencia y unicidad:

[2] En este punto abusamos un tanto de la terminología. En efecto, la solución de (3.1) es la función $u = u(x, t)$. Ahora bien, como (3.1) es una ecuación de orden dos en tiempo es natural escribirla como un sistema de dos ecuaciones de orden uno en tiempo, con dos incógnitas. En este caso el par (u, u_t) puede ser considerado como la solución, lo cual es coherente con el hecho de haber introducido dos datos iniciales en el sistema (3.1).

"Para todo par de datos iniciales $(u_0, u_1) \in H$, i.e. $u_0 \in H_0^1(0, \pi)$ y $u_1 \in L^2(0, \pi)$, existe una única solución $(u, u_t) \in C([0, \infty); H)$ de (3.1). Esta solución pertenece por tanto a la clase

$$u \in C\left([0, \infty);\, H_0^1(0, \pi)\right) \cap C^1\left([0, \infty);\, L^2(0, \pi)\right) \tag{3.13}$$

y la energía correspondiente $E(t)$ de (3.8) se conserva en el tiempo".

En lo que respecta al desarrollo en serie de Fourier (3.2)-(3.3) de los datos iniciales, el hecho de que estos pertenezcan a $H_0^1(0, \pi) \times L^2(0, \pi)$ significa que

$$\sum_{k=1}^{\infty} [k^2 |a_k|^2 + |b_k|^2] < \infty. \tag{3.14}$$

De hecho

$$E(0) = \frac{1}{2} \int_0^{\pi} [|u_{0,x}|^2 + |u_1|^2] dx = \frac{\pi}{4} \sum_{k=1}^{\infty} [k^2 |a_k|^2 + |b_k|^2] < \infty. \tag{3.15}$$

Este resultado de existencia y unicidad puede probarse de al menos dos maneras adicionales, además del método de series de Fourier que acabamos de desarrollar:

- la teoría de semigrupos;
- el método de Galerkin.

El mismo tipo de análisis puede ser desarrollado con muy pocas modificaciones en el caso de varias dimensiones espaciales. Basta para ello utilizar los resultados clásicos sobre la descomposición espectral de la ecuación de Laplace.

Con el objeto de presentar los resultados fundamentales en el caso de varias dimensiones consideramos un *abierto* Ω de $\mathbf{R}^n$, $n \geq 1$. En este punto la regularidad de Ω no es relevante. Con el objeto de desarrollar las soluciones en series de Fourier es sin embargo importante suponer que Ω es *acotado.*

Consideramos entonces la ecuación de ondas

$$\begin{cases} u_{tt} - \Delta u = 0, & x \in \Omega, \quad t > 0 \\ u = 0, & x \in \partial\Omega, \quad t > 0 \\ u(x, 0) = u_0(x),\, u_t(x, 0) = u_1(x), & x \in \Omega. \end{cases} \tag{3.16}$$

Aquí y en lo sucesivo Δ denota el clásico operador de Laplace

$$\Delta u = \sum_{i=1}^{n} \frac{\partial^2 u}{\partial x_i^2}. \tag{3.17}$$

Para $n \geq 1$, (3.16) es claramente una generalización de la ecuación de la cuerda vibrante (3.1). Cuando $n = 2$, (3.16) es un modelo para las vibraciones de una membrana que, en reposo, ocupa el dominio Ω del plano. Cuando $n = 3$, (3.16) describe la propagación de la presión de las ondas acústicas. Sin embargo, desde un punto de vista matemático, la ecuación (3.16) puede tratarse de modo semejante en cualquier dimensión espacial.

Consideramos ahora el problema espectral:

$$\begin{cases} -\Delta\varphi = \lambda\varphi & \text{en} \quad \Omega \\ \quad \varphi = 0 & \text{en} \quad \partial\Omega. \end{cases} \tag{3.18}$$

Es bien sabido (véase [2] o [6], por ejemplo) que los autovalores $\{\lambda_j\}_{j\geq 1}$ de (3.18) constituyen una sucesión creciente de números positivos que tiende a infinito

$$0 < \lambda_1 < \lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_n \leq \cdots \to \infty.$$

El primero de los autovalores es simple. Es habitual repetir el resto de acuerdo a su multiplicidad. De este modo existe una sucesión de autofunciones $\{\varphi_j\}_{j\geq 1}$, donde φ_j es una autofunción asociada al autovalor λ_j, que constituye una base ortonormal de $L^2(\Omega)$. Es decir, se tiene, en particular,

$$\int_\Omega \varphi_j \varphi_k dx = \delta_{jk}. \tag{3.19}$$

De acuerdo a (3.19), multiplicando la ecuación (3.18) correspondiente a λ_k por φ_j e integrando en Ω, gracias a la fórmula de Green obtenemos que

$$\int_\Omega \nabla\varphi_j \cdot \nabla\varphi_k dx = \lambda_j \int_\Omega \varphi_j \varphi_k dx \delta_{jk} = \lambda_k \int_\Omega \varphi_j \varphi_k dx \delta_{jk}. \tag{3.20}$$

De este modo se deduce que las autofunciones son también ortogonales en $H_0^1(\Omega)$. Más concretamente, la sucesión $\left\{\varphi_j/\sqrt{\lambda_j}\right\}_{j\geq 1}$ constituye una base ortonormal de $H_0^1(\Omega)$.

Utilizando esta base de funciones propias del Laplaciano podemos resolver la ecuación de ondas (3.16) como lo hicimos en una variable espacial. Para ello desarrollamos los datos iniciales (u_0, u_1) de (3.16) del modo siguiente

$$u_0(x) = \sum_{k=1}^{\infty} a_k \varphi_k(x); \; u_1(x) = \sum_{k=1}^{\infty} b_k \varphi_k(x). \tag{3.21}$$

Buscamos entonces la solución u de (3.16) en la forma

$$u(x,t) = \sum_{k=1}^{\infty} u_k(t) \varphi_k(x). \tag{3.22}$$

Observamos entonces que los coeficientes $\{u_k\}$ han de resolver la ecuación diferencial:

$$u_k''(t) + \lambda_k u_k(t) = 0, \; t > 0, \; u_k(0) = a_k, \; u_k'(0) = b_k, \tag{3.23}$$

de modo que

$$u_k(t) = a_k \cos\left(\sqrt{\lambda_k} t\right) + \frac{b_k}{\sqrt{\lambda_k}} \operatorname{sen}\left(\sqrt{\lambda_k} t\right). \tag{3.24}$$

De este modo obtenemos que la solución u de (3.16) admite la expresión

$$u(x,t) = \sum_{k=1}^{\infty} \left(a_k \cos\left(\sqrt{\lambda_k} t\right) + \frac{b_k}{\sqrt{\lambda_k}} \operatorname{sen}\left(\sqrt{\lambda_k} t\right)\right) \varphi_k(x). \tag{3.25}$$

La similitud de la expresión (3.4) del caso de una dimensión espacial con la fórmula (3.25) del caso general es evidente. En realidad (3.4) es un caso particular de (3.25). Basta observar que cuando $\Omega = (0, \pi)$, el problema de autovalores para el Laplaciano se convierte en un problema clásico de Sturm-Liouville. El espectro es por tanto explícito:

$$\lambda_k = k^2,\ k \geq 1;\ \varphi_k(x) = \sqrt{\frac{2}{\pi}} \operatorname{sen}(kx),\ k \geq 1. \tag{3.26}$$

Con estos datos las expresiones (3.4) y (3.25) coinciden efectivamente.

La energía de las soluciones de (3.16) es en este caso

$$E(t) = \frac{1}{2} \int_{\Omega} \left[\mid \nabla u(x,t) \mid^2 + |u_t(x,t)|^2 \right] dx \tag{3.27}$$

y también se conserva en tiempo. Nuevamente la energía es proporcional al cuadrado de la norma en el espacio de la energía $H = H_0^1(\Omega) \times L^2(\Omega)$.

En este caso el resultado básico de existencia y unicidad de soluciones dice que:

"Si $(u_0, u_1) \in H_0^1(\Omega) \times L^2(\Omega)$, *existe una única solución* $(u, u_t) \in C([0,\infty); H)$, *i.e.*

$$u \in C\left([0,\infty);\ H_0^1(\Omega)\right) \cap C^1\left([0,\infty);\ L^2(\Omega)\right) \tag{3.28}$$

de (3.16). *La energía* $E(t)$ *en* (3.27) *es constante en tiempo".*

Conviene también señalar que, si bien la regularidad (3.28) de las soluciones débiles permite interpretar la ecuación de ondas en un sentido débil, el hecho que u sea solución con la regularidad (3.28), junto con la propiedad del operador de Laplace con condiciones de contorno de Dirichlet de constituir un isomorfismo de $H_0^1(\Omega)$ en $H^{-1}(\Omega)$, permite deducir que $u \in C^2\left([0,\infty);\ H^{-1}(\Omega)\right)$. De este modo se concluye que la ecuación (3.16) tiene sentido para cada $t > 0$ en el espacio $H^{-1}(\Omega)$. Acabamos de ver cómo se puede aplicar el método de Fourier para la resolución de la ecuación de ondas. Basta para ello conocer la descomposición espectral del Laplaciano con condiciones de Dirichlet (3.18).

El método de Fourier puede ser adaptado a muchas otras situaciones:

- Condiciones de contorno de Neumann, o mixtas en las que la condición de Dirichlet y Neumann se satisfacen en subconjuntos complementarios de la frontera.

- Ecuaciones más generales con coeficientes dependientes de x de la forma:

$$\rho(x)u_{tt} - \operatorname{div}(a(x)\nabla u) + q(x)u = 0$$

donde ρ, a y q son funciones medibles y acotas y ρ y a son uniformemente positivas, i.e. existen $\rho_0,\, a_0 > 0$ tales que

$$\rho(x) \geq \rho_0,\; a(x) \geq a_0,\; \text{p.c.t.}\, x \in \Omega.$$

Pero es cierto también que el método de Fourier tiene sus limitaciones. En particular no permite abordar ecuaciones no lineales, con coeficientes que dependen de x y t, etc. En estos últimos casos los métodos de Galerkin y la teoría de semi-grupos se muestran mucho más flexibles y útiles.

4. Series de Fourier como método numérico

En la sección anterior hemos visto que la ecuación de ondas puede ser resuelta mediante series de Fourier obteniéndose la expresión

$$u(x,t) = \sum_{k=1}^{\infty} \left[a_k \cos\left(\sqrt{\lambda_k} t\right) + \frac{b_k}{\sqrt{\lambda_k}} \operatorname{sen}\left(\sqrt{\lambda_k} t\right) \right] \varphi_k(x), \tag{4.1}$$

siendo $\{\varphi_k\}_{k \geqslant 1}$ y $\{\lambda_k\}_{k \geqslant 1}$ las autofunciones y autovalores del Laplaciano. Como vimos, es conveniente elegir $\{\varphi_k\}_{k \geqslant 1}$ de modo que constituyan una base ortonormal de $L^2(\Omega)$.

Vimos asimismo que la energía

$$E(t) = \frac{1}{2} \int_{\Omega} \left[\mid \nabla u(x,t) \mid^2 + \mid u_t(x,t) \mid^2 \right] dx \tag{4.2}$$

se conserva a lo largo de las trayectorias.

La energía inicial de las soluciones viene dada por

$$E(0) = \frac{1}{2} \sum_{k=1}^{\infty} \left[\mid \lambda_k a_k \mid^2 + \mid b_k \mid^2 \right]. \tag{4.3}$$

Así, la hipótesis de que los datos iniciales sean de energía finita

$$(u_0, u_1) \in H_0^1(\Omega) \times L^2(\Omega) \tag{4.4}$$

es equivalente a que las sucesiones $\left\{a_k \sqrt{\lambda_k}\right\}_{k \geqslant 1}$, $\{b_k\}$ pertenezcan al espacio de las sucesiones de cuadrado sumable ℓ^2.

En vista del desarrollo en serie (4.1) de las soluciones, parece natural construir un método numérico en el que la aproximación venga dada, simplemente, por las sumas parciales de la serie:

$$u_N(x,t) = \sum_{k=1}^{N} \left[a_k \cos\left(\sqrt{\lambda_k} t\right) + \frac{b_k}{\sqrt{\lambda_k}} \operatorname{sen}\left(\sqrt{\lambda_k} t\right) \right] \varphi_k(x). \tag{4.5}$$

La suma finita de u_N en (4.5) proporciona, efectivamente, una aproximación de la solución u representada en la serie de Fourier (4.1). Para comprobarlo consideremos el resto

$$\varepsilon_N = u - u_N = \sum_{k \geqslant N+1} \left[a_k \cos\left(\sqrt{\lambda_k} t\right) + \frac{b_k}{\sqrt{\lambda_k}} \operatorname{sen}\left(\sqrt{\lambda_k} t\right) \right] \varphi_k(x). \tag{4.6}$$

Teniendo en cuenta que

$$\int_\Omega \nabla \varphi_k \cdot \nabla \varphi_j dx = \begin{cases} 0, & \text{si} \quad k \neq j \\ \lambda_k, & \text{si} \quad k = j, \end{cases}$$

es fácil comprobar que

$$\begin{aligned} \left|\left| \nabla \varepsilon_N(t) \right|\right|^2_{L^2(\Omega)} &= \sum_{k \geqslant N+1} \lambda_k \left[a_k \cos\left(\sqrt{\lambda_k} t\right) + \frac{b_k}{\sqrt{\lambda_k}} \operatorname{sen}\left(\sqrt{\lambda_k} t\right) \right]^2 \\ &\leqslant 2 \sum_{k \geqslant N+1} \left[\lambda_k \mid a_k \mid^2 + \mid b_k \mid^2 \right]. \end{aligned} \tag{4.7}$$

Como la serie (4.3) de la energía inicial es convergente, en virtud de (4.7) deducimos que

$$u_N(t) \to u(t) \text{ en } C\left([0, \infty);\, H_0^1(\Omega)\right), \tag{4.8}$$

cuando $N \to \infty$.

El mismo argumento permite probar que

$$u_{N,t} \to u_t(t) \text{ en } C\left([0, \infty);\, L^2(\Omega)\right). \tag{4.9}$$

De (4.8)-(4.9) deducimos que, cuando los datos iniciales están en el espacio de la energía $H_0^1(\Omega) \times L^2(\Omega)$, las sumas parciales (4.5) proporcionan una aproximación eficaz de la solución en dicho espacio, uniformemente en tiempo $t \geqslant 0$.

Cabe por tanto preguntarse sobre la tasa o velocidad de la convergencia. El argumento anterior no proporciona ninguna información en este sentido puesto que la mera convergencia de la serie (4.3) no permite decir nada sobre la velocidad de convergencia de sus sumas parciales.

Con el objeto de obtener tasas de convergencia es necesario hacer hipótesis adicionales sobre los datos iniciales. Supongamos por ejemplo que

$$(u_0,\, u_1) \in \left[H^2 \cap H_0^1(\Omega)\right] \times H_0^1(\Omega). \tag{4.10}$$

En este caso tenemos

$$\sum_{k \geqslant 1} \left[\lambda_k^2 \mid a_k \mid^2 + \lambda_k \mid b_k \mid^2 \right] < \infty. \tag{4.11}$$

En efecto, tal y como veíamos anteriormente en el caso de u_0, el que $u_1 \in H_0^1(\Omega)$ se caracteriza porque sus coeficientes de Fourier $(b_k)_{k \geqslant 1}$ satisfacen

$$\sum_{k \geqslant 1}^{\infty} \lambda_k \mid b_k \mid^2 < \infty. \tag{4.12}$$

Por otra parte, $\| \Delta\varphi \|_{L^2(\Omega)}$ define una norma equivalente a la inducida por $H^2(\Omega)$ en el subespacio $H^2 \cap H_0^1(\Omega)$[3]. Por otra parte, se tiene

$$\int_\Omega \Delta\varphi_k \Delta\varphi_j dx = \begin{cases} 0 & \text{si} \quad k \neq j \\ \lambda_k^2 & \text{si} \quad k = j. \end{cases} \tag{4.13}$$

Deducimos por tanto que

$$\left|\left|\Delta u_0\right|\right|^2_{L^2(\Omega)} = \sum_{k \geqslant 1} \lambda_k^2 \left|a_k\right|^2 \tag{4.14}$$

y, de este modo, observamos que, efectivamente, la serie (4.11) converge.

La información adicional que (4.11) proporciona sobre los coeficientes de Fourier permite obtener tasas de convergencia de u_N hacia u en el espacio de la energía. Por ejemplo, volviendo a (4.7) tenemos

$$\begin{aligned} \left|\left|\nabla \varepsilon_N(t)\right|\right|^2_{L^2(\Omega)} &\leqslant 2 \sum_{k \geqslant N+1} \left[\lambda_k \left|a_k\right|^2 + \left|b_k\right|^2\right] \\ &\leqslant 2 \sum_{k \geqslant N+1} \frac{1}{\lambda_k} \left[\lambda_k^2 \left|a_k\right|^2 + \lambda_k \left|b_k\right|^2\right] \\ &\leqslant \frac{2}{\lambda_{N+1}} \sum_{k \geqslant N+1} \left[\lambda_k^2 \left|a_k\right|^2 + \lambda_k \left|b_k\right|^2\right] \\ &\leqslant \frac{C}{\lambda_{N+1}} \left|\left| (u_0, u_1) \right|\right|^2_{H^2 \cap H_0^1(\Omega) \times H_0^1(\Omega)}. \end{aligned}$$

El mismo argumento puede ser utilizado para estimar la norma de $\varepsilon_{N,t}$ en $L^2(\Omega)$. De este modo deducimos que

$$\| u - u_N \|_{L^\infty\left(0,\infty;\, H_0^1(\Omega)\right) \cap W^{1,\infty}(0,\infty;\, L^2(\Omega))} \leq \frac{C}{\sqrt{\lambda_{N+1}}} \left|\left| (u_0,\, u_1) \right|\right|_{H^2 \cap H_0^1(\Omega) \times H_0^1(\Omega)}. \tag{4.15}$$

Esta desigualdad proporciona estimaciones explícitas sobre la velocidad de convergencia. En efecto, el clásico Teorema de Weyl sobre la distribución asintótica de los autovalores del Laplaciano asegura que

$$\lambda_N \sim c(\Omega) N^{2/n},\ N \to \infty \tag{4.16}$$

donde $c(\Omega)$ es una constante positiva que depende del dominio y n es la dimensión espacial[4].

Combinando (4.15) y (4.16) obtenemos que u_N converge a u en el espacio de la energía, uniformemente en tiempo $t \geqslant 0$, con un orden de $O\left(N^{-1/n}\right)$.

[3] En este punto ultilizamos el resultado clásico de regularidad de las soluciones del problema de Dirichlet para el Laplaciano que garantiza que, si el dominio es de clase C^2 y el segundo miembro está en $L^2(\Omega)$, entonces la solución pertenece a $H^2 \cap H_0^1(\Omega)$

[4] Es obvio que, por ejemplo, en una dimensión espacial $n = 1$, la expresión asintótica en (4.16) coincide con lo que se observa en la expresión explícita del espectro. En efecto, recordemos que, cuando $\Omega = (0, \pi)$, $\lambda_k = k^2$.

La hipótesis $(u_0, u_1) \in H^2 \cap H_0^1(\Omega) \times H_0^1(\Omega)$ realizada sobre los datos iniciales es sólo una de las posibles. De manera general puede decirse que, cuando los datos iniciales son más regulares que lo que el espacio de la energía exige y verifican las condiciones de compatibilidad adecuadas en relación a las condiciones de contorno, entonces, se puede establecer una estimación sobre la velocidad de convergencia de la aproximación que las sumas parciales del desarrollo en serie de Fourier proporcionan a la solución de la ecuación de ondas.

Este método de aproximación lo denominaremos *método de Fourier*. Se trata de un método de aproximación sumamente útil en una dimensión espacial puesto que, al disponer de la expresión explícita de las autofunciones φ_k y autovalores de λ_k, la aproximación u_N puede calcularse de manera totalmente explícita. Bastaría para ello con utilizar una fórmula de cuadratura para aproximar el valor (3.3) de los coeficientes de Fourier.

El método de Fourier es sin embargo mucho menos eficaz en varias dimensiones espaciales. En efecto, en ese caso no disponemos de la expresión explícita de las autofunciones y autovalores y su aproximación numérica es un problema tan complejo como el de la propia aproximación de la ecuación de ondas.

Otro de los inconvenientes del método de Fourier es que, cuando la ecuación es no-lineal o tiene coeficientes que depende de (x, t), ya no se puede obtener una expresión explícita de la solución en serie de Fourier y por tanto tampoco de sus aproximaciones.

Es por eso que el método de Fourier tiene una utilidad limitada y por tanto precisamos de métodos más sistemáticos y robustos que funcionen no sólo en casos particulares sino para amplias clases de ecuaciones. En este marco destacan los métodos de diferencias finitas que serán el objeto central de estas notas.

5. La ecuación de transporte lineal

Las ecuaciones que modelizan fenómenos de propagación de ondas y vibraciones son típicamente Ecuaciones en Derivadas Parciales (EDP) de orden dos. Sin embargo en todas ellas subyacen las ecuaciones de transporte de orden uno que analizamos en esta sección.

El modelo más sencillo es

$$u_t + u_x = 0. \tag{5.1}$$

Es fácil comprobar que $u = u(x, t)$ es solución de esta ecuación si y sólo si es constante a lo largo de las *líneas características*

$$x + t = cte. \tag{5.2}$$

De este modo deducimos que las soluciones de (5.1) son de la forma

$$u = f(x - t), \tag{5.3}$$

donde f es el perfil de la solución en el instante inicial $t = 0$, i.e.

$$u(x, 0) = f(x). \tag{5.4}$$

La solución (5.3) es entonces una simple onda de transporte pura en la que el perfil f se transporta (avanza) en el eje real a velocidad constante uno[5] .

Al invertir el sentido del tiempo (i.e. haciendo el cambio de variable $t \to -t$) la ecuación (5.1) se transforma en

$$u_t - u_x = 0 \tag{5.5}$$

cuyas soluciones son ahora de la forma

$$u = g(x+t), \tag{5.6}$$

tratándose de ondas viajeras que se propagan en dirección opuesta a velocidad uno.

Vemos por tanto que las soluciones de la ecuación de transporte pueden calcularse de manera explícita y que en ellas se observa un sencillo fenómeno de transporte lineal sin deformación.

Esta ecuación es por tanto un excelente laboratorio para experimentar algunas de las ideas más sencillas del análisis numérico.

Consideremos pues un paso de discretización $h > 0$ en la variable espacial e introduzcamos el mallado $\{x_j\}_{j\in\mathbf{Z}}$, $x_j = jh$.

Buscamos una semi-discretización (continua en tiempo y discreta en espacio) que reduzca la EDP (5.1) a un sistema de ecuaciones diferenciales cuya solución proporcione una aproximación $u_j(t)$ de la solución $u = u(x,t)$ de (5.1) en el punto $x = x_j$.

La manera más sencilla de construir esta semi-discretización es utilizar el desarrollo de Taylor para introducir una aproximación de la derivación parcial en la variable espacial. Son varias las posibilidades:

$$u_x(x_j,t) \sim \frac{u(x_{j+1},t) - u(x_j,t)}{h} \sim \frac{u_{j+1}(t) - u_j(t)}{h}, \tag{5.7}$$

$$u_x(x_j,t) \sim \frac{u(x_j,t) - u(x_{j-1},t)}{h} \sim \frac{u_j(t) - u_{j-1}(t)}{h} \tag{5.8}$$

$$u_x(x_j,t) \sim \frac{u(x_{j+1},t) - u(x_{j-1},t)}{2h} \sim \frac{u_{j+1}(t) - u_{j-1}(t)}{2h}. \tag{5.9}$$

Cada una de estas elecciones corresponde a un determinado sentido de avance a lo largo del eje x. En efecto (5.7) y (5.8) y (5.9) corresponden a diferencias progresivas, regresivas y centradas respectivamente.

[5] Si bien en este caso la ecuación puede resolverse explícitamente, el problema (5.1) entra en el marco de la Teoría de Semigrupos. En efecto, basta considerar el espacio de Hilbert $H = L^2(\mathbf{R})$ y el operador $A = -\partial_x$ con dominio $D(A) = H^1(\mathbf{R})$ para que el problema (5.1) entre en el marco abstracto del Teorema de Hille-Yosida. En efecto, el operador A así definido es maximal disipativo. Para ver que es maximal basta con observar que $< Au, u >_{L^2(\mathbf{R})} = -\int_{\mathbf{R}} \partial_x u u dx = -\frac{1}{2}\int_{\mathbf{R}} \partial_x(u^2) dx = 0$. Además A es maximal. En efecto, dado $f \in L^2(\mathbf{R})$, existe una única solución $u \in H^1(\mathbf{R})$ de $u + \partial_x u = f$. Esta solución puede calcularse explícitamente y se obtiene: $u(x) = \int_{-\infty}^{x} f(s)e^{s-x}ds = \int_{-\infty}^{0} f(z+x)e^z dz$. Tomando normas en $L^2(\mathbf{R})$ y aplicando la desigualdad de Minkowski se deduce fácilmente que $||u||_{L^2(\mathbf{R})} \leq \int_{-\infty}^{0} ||f||_{L^2(\mathbf{R})} e^z dz = ||f||_{L^2(\mathbf{R})}$. Como $u_x = f - u$ vemos inmediatamente que, efectivamente, u pertenece a $H^1(\mathbf{R})$.

Cada una de estas elecciones proporciona un sistema semi-discreto diferente de aproximación de la EDP (5.1) en diferencias finitas:

- *Esquema progresivo:*

$$u_j'(t) + \frac{u_{j+1}(t) - u_j(t)}{h} = 0,\ j \in \mathbf{Z},\ t > 0, \tag{5.10}$$

- *Esquema regresivo:*

$$u_j'(t) + \frac{u_j(t) - u_{j-1}(t)}{h} = 0,\ j \in \mathbf{Z},\ t > 0, \tag{5.11}$$

- *Esquema centrado:*

$$u_j'(t) + \frac{u_{j+1}(t) - u_{j-1}(t)}{2h} = 0,\ j \in \mathbf{Z},\ t > 0. \tag{5.12}$$

Estos sistemas constituyen un conjunto numerable de ecuaciones diferenciales de orden uno lineales acopladas. Al tratarse de sistema infinitos su resolución no entra en el marco de la teoría clásica de EDO. Sin embargo, es fácil verificar que su solución existe y es única sin necesidad de utilizar la Teoría de Semigrupos. Para ello basta considerar el espacio de Hilbert $H = \ell^2$ de las sucesiones de cuadrado sumables. La solución de cualquiera de estas ecuaciones semi-discretas puede entonces verse como un elemento de este espacio: $\vec{u} = \{u_j\}_{j\in\mathbf{Z}} \in \ell^2$. Estos sistemas pueden escribirse entonces en forma abstracta

$$\frac{d}{dt}\vec{u} = A_h\vec{u}. \tag{5.13}$$

Es fácil comprobar que en cada uno de los casos anteriores el operador A_h involucrado puede representarse a través de una matriz infinita, tridiagonal y acotada con norma $1/h$. Se trata pues de ecuaciones de evolución en espacios de Hilbert de dimensión infinita pero en las que el generador A_h está acotado. Esto nos permite calcular el semigrupo $e^{A_h t}$ mediante la representación en desarrollo de serie de potencias de la exponencial. Obtenemos así que estas ecuaciones generan semigrupos en $H = \ell^2$. De este modo deducimos que para cada dato inicial dado en ℓ^2 cada una de estas ecuaciones admite una única solución $C^\infty(\mathbf{R}, \ell^2)$ que toma ese dato en el instante $t = 0$. Las soluciones dependen en realidad de manera analítica con respecto a la variable temporal.

Todos estos esquemas son consistentes con la ecuación de transporte. Es decir, al llevar a estos esquemas una solución regular de la ecuación de transporte continua vemos que se produce un error que tiende a cero a medida que $h \to 0$.

La mejor manera de analizar la estabilidad es a través del método de von Neumann. Así, introduciendo

$$\breve{u}(\theta, t) = \sum_{j\in\mathbf{Z}} u_j(t)e^{i\theta j} \tag{5.14}$$

obtenemos que $\check{u}$, en cada uno de los casos, satisface

$$\check{u}'(\theta,t)+\left(\frac{e^{-i\theta}-1}{h}\right)\check{u}(\theta,t)=0,\ t>0, \tag{5.15}$$

$$\check{u}'(\theta,t)+\left(\frac{1-e^{i\theta}}{h}\right)\check{u}(\theta,t)=0,\ t>0, \tag{5.16}$$

$$\check{u}'(\theta,t)+\left(\frac{e^{-i\theta}-e^{i\theta}}{2h}\right)\check{u}(\theta,t)=0,\ t>0. \tag{5.17}$$

La transformada discreta de Fourier no sólo tiene la virtud de transformar los sistemas de ecuaciones semi-discretas (5.10)-(5.12) en ecuaciones diferenciales con paramétro θ (5.15)-(5.17) que son inmediatas de resolver, sino que define también una isometríca de ℓ^2 a valores en $L^2(0,2\pi)$. En efecto, la fórmula (5.14) puede invertirse fácilmente. De hecho tenemos

$$u_j(t)=\frac{1}{2\pi}\int_0^{2\pi}\check{u}(\theta,t)e^{-ij\theta}d\theta. \tag{5.18}$$

Además

$$\frac{1}{2\pi}\int_0^{2\pi}|\check{u}(\theta,t)|^2d\theta=\sum_{j\in\mathbf{Z}}|u_j(t)|^2. \tag{5.19}$$

Obtenemos por tanto

$$\check{u}(\theta,t)=e^{a_h(\theta)t}\check{u}(\theta,0) \tag{5.20}$$

donde $a_h(\theta)$ varía de un caso a otro. De manera más precisa se tiene

$$a(\theta)=\begin{cases}\dfrac{1-e^{-i\theta}}{h}, & \text{(esquema progresivo)}\\[2mm] \dfrac{e^{i\theta}-1}{h}, & \text{(esquema regresivo)}\\[2mm] \dfrac{e^{i\theta}-e^{-i\theta}}{2h}, & \text{(esquema centrado).}\end{cases} \tag{5.21}$$

Como es bien sabido, la convergencia de un método numérico exige su estabilidad[6] y ésta pasa por que $\operatorname{Re} a_h(\theta)$ permanezca acotada superiormente cuando $h\to 0$ uniformemente en $\theta\in[0,2\pi)$. Verifiquemos si esta propiedad se cumple en cada uno de los casos:

[6] En este punto estamos haciendo uso del clásico Teorema de Lax que dice que la convergencia de un esquema es equivalente a su estabilidad más consistencia. En el caso más sencillo de la resolución de un sistema lineal $Ax=b$, podemos interpretar este resultado del siguiente modo. Aproximemos este problema por otro de características semejantes $A_\varepsilon x_\varepsilon=b_\varepsilon$. Suponemos que $b_\varepsilon\to b$ cuando $\varepsilon\to 0$. Deseamos probar que $x_\varepsilon\to x$. Para ello hacemos las dos siguientes hipótesis: a) $A_\varepsilon y\to Ay$ para todo y (*consistencia*) y b) Las matrices inversas $(A_\varepsilon)^{-1}$ están uniformemente acotadas (*estabilidad*). Deducimos entonces la convergencia de las soluciones: $x_\varepsilon\to x$ cuando $\varepsilon\to 0$. En efecto, tenemos $A_\varepsilon(x_\varepsilon-x)=b_\varepsilon-b+(A-A_\varepsilon)x=r_\varepsilon$. Por las hipótesis realizadas sobre la aproximación deducimos que $r_\varepsilon\to 0$. La hipótesis de estabilidad garantiza entonces que $x_\varepsilon-x\to 0$. El Teorema de Lax generaliza este resultado al caso de las EDP y sus aproximaciones numéricas. La ecuación $Ax=b$ del ejemplo anterior juega el papel de la EDP, la ecuación cuya solución deseamos aproximar. La ecuación aproximada $A_\varepsilon x_\varepsilon=b_\varepsilon$ juega el papel de la aproximación numérica, y ε es el parámetro destinado a tender a cero, lo mismo que hace h en las aproximaciones numéricas.

- *Esquema progresivo:* Tenemos
$$a_h(\theta) = \frac{1-e^{i\theta}}{h} = \frac{1-\cos(\theta)}{h} - \frac{i\,\mathrm{sen}(\theta)}{h}. \tag{5.22}$$
Por tanto
$$\mathrm{Re},\, a_h(\theta) = \frac{1-\cos(\theta)}{h}.$$
Obviamente,
$$\mathrm{Re},\, a_h(\theta) \nearrow \infty,\ h \to 0,\ \forall 0 < \theta < 2\pi, \tag{5.23}$$
lo cual demuestra la falta de estabilidad y por tanto de convergencia de este esquema.

- *Esquema regresivo:* En este caso
$$a_h(\theta) = \frac{e^{-i\theta}-1}{h} = \frac{\cos(\theta)-1}{h} - \frac{i\,\mathrm{sen}\,\theta}{h} \tag{5.24}$$
de modo que
$$\mathrm{Re},\, a_h(\theta) = \frac{\cos(\theta)-1}{h} \leq 0,\ \forall \theta \in [0, 2\pi). \tag{5.25}$$
La estabilidad del esquema está por tanto garantizada. Esto demuestra que el esquema es también convergente, propiedad que analizaremos más adelante.

- *Esquema centrado:* En este caso
$$a_h(\theta) = \frac{e^{-i\theta}-e^{i\theta}}{2h} = -\frac{i\,\mathrm{sen}\,\theta}{h}. \tag{5.26}$$
Obviamente,
$$\mathrm{Re},\, a_h(\theta) = 0 \tag{5.27}$$
por lo que este esquema es también estable y convergente.

En realidad bastaría verificar las propiedades geométricas más elementales asociadas a la evolución temporal que la ecuación continua y semi-discreta generan para ver que el esquema progresivo no puede de ningún modo ser convergente y que, sin embargo, los otros dos esquemas pueden perfectamente serlo.

En efecto, en virtud de la expresión explícita (5.3) de la solución de la ecuación de transporte (5.1), observamos que el dominio de dependencia de la solución en el punto (x, t) se reduce al punto $x-t$ en el instante inicial. Veamos ahora cuáles son los dominio de dependencia en los esquemas discretos.

En el esquema progresivo, fijado un punto $x = x_j$, vemos que la ecuación que gobierna la dinámica de $u_j(t)$ depende de $u_{j+1}(t)$, la aproximación de la solución en el nodo x_{j+1} inmediatamente a la derecha de x_j, que a su vez depende de $u_{j+1}(t)$, etc. Vemos pues que, en este caso, el sistema semi-discreto depende del valor del dato inicial a la derecha de x_j

mientras que el único valor relevante para la solución real es el punto $x-t$ que está al lado opuesto, a la izquierda de x.

Por lo tanto el esquema semi-discreto progresivo viola la condición indispensable para la convergencia de un esquema numérico según la cual *el dominio de dependencia del esquema numérico ha de contener el dominio de dependencia de la ecuación original*[7].

Sin embargo, los otros dos esquemas si que verifican esta propiedad geométrica, lo cual garantiza su convergencia. El esquema progresivo para la ecuación de transporte que consideramos suele normalmente denominarse "upwind", que viene a significar algo asi como "a favor de la corriente". Con este término se pone de manifiesto que en los problemas en los que está presente el fenómeno de transporte, el sentido y orientación del mismo ha de ser tenido en cuenta a la hora de diseñar métodos numéricos convergentes.

El análisis que acabamos de realizar indica que:

∗ Las ondas continuas se propagan en el espacio-tiempo con una velocidad y dirección determinadas.

∗ Los esquemas numéricos, a pesar de estar basados en un mecanismo aparentemente coherentes de discretización, pueden generar ondas que se propagan con velocidades y direcciones distintas y no converger a medida que el paso del mallado tiende a cero.

Los tres esquemas que hemos analizado son en principio coherentes. En realidad en la terminología del Análisis Numérico se dice que son esquemas consistentes. De manera más precisa, mientras que el esquema progresivo y regresivo son consistentes de orden 1, el esquema centrado es consistente de orden 2. En efecto, supongamos que u es una solución suficientemente regular de la ecuación de transporte (5.1) (basta con que u tenga una derivada continua en la variable tiempo y tres en la variable espacial).

Sea entonces

$$\underline{u}_j(t) = u(x_j, t), \tag{5.28}$$

la restricción de (5.1) a los puntos del mallado.

Para analizar la consistencia de los esquemas numéricos introducidos consideramos $\left(\, \underline{u}_j\right)_{j\in\mathbf{Z}}$ como una solución aproximada de dicho esquema[8].

[7] Se trata efectivamente de una condición necesaria para la convergencia de un método numérico. Cuando no se cumple, hay puntos del dominio de dependencia del problema continuo que no pertenecen al del problema discreto. En estas circunstancias, modificando los datos iniciales en esos puntos, podemos conseguir alterar la solución del problema continuo sin que la del problema discreto sufra ningún cambio. Esto excluye cualquier posibilidad de convergencia del método numérico.

[8] Conviene subrayar que, a la hora de comprobar la consistencia de un método numérico, lo que comunmente se hace es considerar la solución del problema continuo como una solución aproximada del esquema discreto y no al revés, como podría esperarse en la medida en que el esquema numérico tiene como objeto aproximar la ecuación continua.

Tenemos entonces, en el caso de esquema progresivo

$$\begin{aligned}\underline{u}'_j + \frac{\underline{u}_{j+1} - \underline{u}_j}{h} &= u_t(x_j,t) + \frac{u(x_{j+1},t) - u(x_j,t)}{h} \\ &= u_t(x_j,t) + u(x_j,t) + \frac{h\, u_x(x_j,t) + O\left(h^2\right) - u(x_j,t)}{h} \\ &= u_t(x_j,t) + u_x(x_j,t) + O(h) = O(h),\end{aligned} \tag{5.29}$$

lo cual indica que se trata efectivamente de un esquema consistente de orden 1.

Por último, el esquema centrado es consistente de orden 2:

$$\begin{aligned}\underline{u}'_j + \frac{\underline{u}_{j+1} - \underline{u}_{j-1}}{2h} &= u_t(x_j,t) + \frac{u(x_{j+1},t) - u(x_{j-1},t)}{2h} \\ &= u_t(x_j,t) + \Bigg[u(x_j,t) + hu_x(x_j,t) + \frac{h^2}{2}u_{xx}(x_j,t) + O(h^3) \\ &\qquad -u(x_j,t) + hu_x(x_j,t) + \frac{h^2}{2}u_{xx}(x_j,t) + O\left(h^3\right)\Bigg]\Bigg/ h, \\ &= u_t(x_j,t) + u_x(x_j,t) + O\left(h^2\right) = O\left(h^2\right).\end{aligned} \tag{5.30}$$

En virtud del Teorema de equivalencia de P. Lax que garantiza que la convergencia equivale a la consistencia más la estabilidad cabe entonces esperar que el esquema regresivo sea convergente de orden 1 y que el centrado sea convergente de orden 2.

Comprobémoslo. Consideremos en primer lugar el esquema regresivo y analicemos el error

$$\varepsilon_j(t) = \underline{u}_j(t) - u_j(t) = u(x_j,t) - u_j(t), \tag{5.31}$$

es decir la diferencia entre la solución real y la numérica sobre los puntos del mallado. Para simplificar la presentación suponemos que el dato inicial es continuo[9], lo cual permite tomar datos iniciales exactos en el esquema semi-discreto:

$$u_j(0) = f(x_j),\ j \in \mathbf{Z}. \tag{5.32}$$

En virtud del análisis de consistencia anterior, sustrayendo la ecuación verificada por $\underline{u}_j$ y u_j deducimos que

$$\begin{cases} \varepsilon'_j + \frac{\varepsilon_j - \varepsilon_{j-1}}{h} = O_j(h),\ j \in \mathbf{Z},\ t > 0 \\ \varepsilon_j(0) = 0,\ j \in \mathbf{Z}. \end{cases} \tag{5.33}$$

Multiplicando en (5.33) por ε_j y sumando en $j \in \mathbf{Z}$ obtenemos

$$\frac{1}{2}\frac{d}{dt}\left[\sum_{j\in\mathbf{Z}} |\varepsilon_j(t)|^2\right] + \frac{1}{h}\sum_{j\in\mathbf{Z}}\left(\varepsilon_j^2 - \varepsilon_{j-1}\varepsilon_j\right) = \sum_{j\in\mathbf{Z}} O_j(h)\varepsilon_j.$$

[9] Si el dato inicial no fuese continuo sino solamente localmente integrable, por ejemplo, tomaríamos como dato inicial para el problema discreto una media del dato inicial $f = f(x)$ en torno a los puntos del mallado. Por ejemplo, $u_j(0) = \frac{1}{h}\int_{x_j-h/2}^{x_j+h/2} f(s)ds$.

En este punto conviene observar que

$$\begin{aligned}\sum_{j\in\mathbf{Z}}\left(\varepsilon_j^2-\varepsilon_{j-1}\varepsilon_j\right) &= \frac{1}{2}\sum_{j\in\mathbf{Z}}\left(\varepsilon_j^2+\varepsilon_{j-1}^2-2\varepsilon_{j-1}\varepsilon_j\right)\\ &= \frac{1}{2}\sum_{j\in\mathbf{Z}}\left(\varepsilon_j-\varepsilon_{j-1}\right)^2\geq 0.\end{aligned}$$

Por tanto la identidad de energía anterior puede reescribirse del siguiente modo

$$\frac{1}{2}\frac{d}{dt}\left[\sum_{j\in\mathbf{Z}}|\varepsilon_j(t)|^2\right]+\frac{1}{2h}\sum_{j\in\mathbf{Z}}\left(\varepsilon_j(t)-\varepsilon_{j-1}(t)\right)^2=\sum_{j\in\mathbf{Z}}O_j(h)\varepsilon_j(t). \tag{5.34}$$

En este punto introducimos la norma en ℓ^2, el espacio de las sucesiones de cuadrado sumable a escala h:

$$|(\varepsilon_j)|_h=\left[h\sum_{j\in\mathbf{Z}}|\varepsilon_j|^2\right]^{1/2}. \tag{5.35}$$

En lo sucesivo utilizaremos la notación vectorial $\vec{\varepsilon}$ para denotar el vector infinito numerable de componentes $(\varepsilon_j)_{j\in\mathbf{Z}}$.

Conviene observar que (5.35) es una aproximación discreta de la norma continua en $L^2(\mathbf{R})$ en el mallado de paso h.

Con esta notación, y denotando mediante $\tau_{-1}\vec{\varepsilon}$ la sucesión trasladada de una unidad con componentes $(\varepsilon_{j-1})_{j\in\mathbf{Z}}$, la identidad (5.34) puede reescribirse del modo siguiente:

$$\frac{1}{2}\frac{d}{dt}\left|\vec{\varepsilon}(t)\right|_h^2+\frac{1}{2h}\left|\vec{\varepsilon}(t)-\tau_{-1}\vec{\varepsilon}(t)\right|_h^2=h\sum_{j\in\mathbf{Z}}O_j(h)\varepsilon_j(t)\leq\left|\vec{O}(h)\right|_h\left|\varepsilon_j(t)\right|_h. \tag{5.36}$$

De esta desigualdad se deduce que

$$\frac{d}{dt}\left|\vec{\varepsilon}(t)\right|_h\leq\left|\vec{O}(h)\right|_h$$

de donde se sigue que

$$\left|\vec{\varepsilon}(t)\right|_h\leq\int_0^t\left|\vec{O}(h)\right|_h ds \tag{5.37}$$

puesto que $\vec{\varepsilon}(0)=0$.

En este punto tenemos que analizar el error de truncatura $\vec{O}(h)$. En vista del análisis de la consistencia realizado previamente se observa que cada componente $O_j(h)$ del error es de la forma

$$O_j(h)=\frac{h}{2}u_{xx}\left(\xi_j,t\right)$$

donde ξ_j es un punto en el intervalo $[x_{j-1},x_j]$.

Con el objeto de concluir la prueba de la convergencia suponemos que el dato inicial $f = f(x)$ es de clase C^2 y de soporte compacto: $f \in C_c^2(\mathbf{R})$. Entonces, la solución u, cuya forma explícita fue derivada en (5.3), tiene la misma propiedad para todo $t > 0$ y además:

$$\max_{x\in\mathbf{R},\, t\geq 0} |u_{xx}(x,t)| = C < \infty,$$

de donde, habida cuenta que el soporte de u_{xx} está contenido en una traslación del soporte compacto de f, se sigue que

$$\left|\vec{O}(h)\right|_h \leq Ch,\ \forall t \geq 0,\ \forall h > 0. \tag{5.38}$$

Combinando (5.37)-(5.38) se concluye que

$$|\vec{\varepsilon}(t)|_h \leq Cth,\ \forall t \geq 0,\ \forall h > 0, \tag{5.39}$$

lo cual concluye la demostración de que el método semi-discreto de diferencias finitas regresivas es convergente de orden uno.

El método empleado en la prueba de la convergencia es el denominado método de la energía y está basado en la siguiente ley de energía que las soluciones del problema semi-discreto verifican

$$\frac{1}{2}\frac{d}{dt}\sum_{j\in\mathbf{Z}} |u_j(t)|^2 + h\sum_{j\in\mathbf{Z}} \left\{\frac{u_j(t) - u_{j-1}(t)}{h}\right\}^2 = 0$$

y que, con las notaciones anteriores, puede reescribirse como

$$\frac{1}{2}\frac{d}{dt}\,|\vec{u}(t)|_h^2 + h\,|\vec{u}(t)|_{1,h}^2 = 0. \tag{5.40}$$

Aquí y en lo sucesivo $\|\cdot\|_{1,h}$ denota la versión discreta de la semi-norma $\left(\int_{\mathbf{R}} u_x^2 dx\right)^{1/2}$, i.e.

$$|\vec{u}|_{1,h} = \left[h\sum_{j\in\mathbf{Z}}\left|\frac{u_j - u_{j-1}}{h}\right|^2\right]^{1/2}. \tag{5.41}$$

Conviene comparar (5.40) con la ley de conservación de la energía para la ecuación de transporte continua (5.1) donde, multiplicando por u e integrando con respecto a x se deduce que

$$\frac{d}{dt}\parallel u(t)\parallel_{L^2(\mathbf{R})}^2 = 0. \tag{5.42}$$

Obviamente, la ley de conservación de energía (5.42) para el problema continuo (5.1) es perfectamente coherente con la forma explícita (5.3) de la solución (5.1).

Sin embargo, es de señalar que, en contraste con la ley de conservación de energía (5.42) de la ecuación continua (5.1), la identidad (5.40) establece el carácter disipativo del esquema semi-discreto regresivo. Este carácter disipativo no está reñido con la convergencia del esquema cuando $h \to 0$, esencialmente por dos razones:

$*$ La tasa de disipación del esquema semi-discreto decrece a medida que $h \to 0$, tal y como se observa con claridad en (5.40).

$*$ El carácter disipativo del equema numérico contribuye a su estabilidad.

Verifiquemos la ley de energía de los otros dos esquemas considerados.

En el esquema progresivo tenemos

$$\frac{1}{2}\frac{d}{dt}\sum_{j\in\mathbf{Z}}|u_j(t)|^2+\sum_{j\in\mathbf{Z}}\frac{(u_{j+1}-u_j)}{h}u_j=\frac{1}{2}\frac{d}{dt}\sum_{j\in\mathbf{Z}}|u_j(t)|^2-\frac{1}{2h}\sum_{j\in\mathbf{Z}}|u_{j+1}-u_j|^2=0. \quad (5.43)$$

En esta identidad queda claramente de manifiesto el carácter anti-disipativo del método progresivo, causante de su inestabilidad.

En el caso del esquema centrado tenemos sin embargo

$$\frac{d}{dt}\sum_{j\in\mathbf{Z}}|u_j(t)|^2=0, \quad (5.44)$$

identidad que garantiza su carácter puramente conservativo y su estabilidad.

Las propiedades disipativas, anti-disipativas y conservativas de los esquemas regresivo, progresivo y centrado pueden interpretarse fácilmente de la siguiente manera.

Consideremos por ejemplo el esquema regresivo en el que hemos adoptado la siguiente aproximación de la derivada espacial

$$u_x(x,t)\sim\frac{u(x,t)-u(x-h,t)}{h}.$$

Un análisis más cuidadoso indica que, en realidad,

$$\frac{u(x,t)-u(x-h,t)}{h}=u_x(x,t)-\frac{h}{2}u_{xx}(x,t)+O(h^2).$$

Por lo tanto, el esquema regresivo es en realidad una aproximación de orden dos de la ecuación de transporte perturbada

$$u_t+u_x-\frac{h}{2}u_{xx}=0. \quad (5.45)$$

La ecuación (5.45) es una aproximación parabólica o viscosa de la ecuación de transporte puro [10] (5.1). Multiplicando en (5.45) por u e integrando en x deducimos que

$$\frac{1}{2}\frac{d}{dt}\int_{\mathbf{R}}u^2(x,t)dx+\frac{h}{2}\int_{\mathbf{R}}u_x^2(x,t)dx=0, \quad (5.46)$$

[10] No es difícil comprobar que la ecuación (5.45) genera un semigrupo de contracciones en $L^2(\mathbf{R})$ para cada $h>0$ y que, dado un dato inicial $f\in L^2(\mathbf{R})$, la solución $u_h=u_h(x,t)$ de (5.45) converge a la solución de la ecuación de transporte puro $u(x,t)=f(x-t)$, cuando $h\to 0$ en $L^2(\mathbf{R})$ para cada $t>0$. Para ello basta observar que $v_h(x,t)=u_h(x+t,t)$ es solución de la ecuación del calor $v_t-hv_{xx}=0$ que, tras el cambio de variables $w_h(x,t)=v_h(x,t/h)$, se convierte en una solución de la ecuación del calor $w_t-w_{xx}=0$. Así, vemos que $v_h(x,t)=[G_h(t)*f](x)$ siendo G_h el núcleo del calor reescalado: $G_h(x,t)=(4\pi ht)^{-1/2}\exp(-x^2/4ht)$. De esta expresión se deduce fácilmente que $v_h(x,t)\to f(x)$ en $L^2(\mathbf{R})$, para cada $t>0$, o, lo que es lo mismo, $u_h(x,t)\to f(x-t)$.

lo cual refleja el carácter disipativo del término de regularización $-hu_{xx}/2$ añadido en la ecuación (5.45) y supone, claramente, la versión continua de la ley de disipación de energía (5.40) del esquema regresivo.

El mismo argumento permite detectar el carácter inestable de la aproximación progresiva puesto que

$$\frac{u(x+h,t)-u(x,t)}{h} = u_x(x,t) + \frac{1}{2}u_{xx}(x,t) + O(h^2). \tag{5.47}$$

En este caso, el esquema progresivo resulta ser una aproximación de orden dos de la EDP de segundo orden

$$u_t + u_x + \frac{h}{2}u_{xx} = 0. \tag{5.48}$$

En esta ocasión (5.48) es una ecuación parabólica retrógrada de carácter inestable[11] tal y como queda de manifiesto en la ley de amplificación de la energía que las soluciones de (5.48) satisfacen

$$\frac{1}{2}\frac{d}{dt}\int_{\mathbf{R}} u^2(x,t)dx = \frac{h}{2}\int_{\mathbf{R}} u_x^2(x,t)dx. \tag{5.49}$$

Sin embargo, este argumento permite confirmar el carácter puramente conservativo de la aproximación centrada. En efecto:

$$\frac{u(x+h,t)-u(x-h,t)}{2h} = u_x(x,t) + \frac{h^2}{3!}\partial_x^3 u(x,t) + \cdots + \frac{h^{2\ell}}{(2\ell+1)!}\partial_x^{2\ell+1}u(x,t) + \cdots . \tag{5.50}$$

Es fácil comprobar, en efecto, que cualquiera de las aproximaciones de la ecuación de transporte (5.1) obtenidas truncando el desarrollo en serie de potencias (5.50) de la forma

$$u_t + \sum_{\ell=0}^{L} \frac{h^{2\ell}}{(2\ell+1)!}\partial_x^{2\ell+1} = 0 \tag{5.51}$$

Tiene un carácter puramente conservativo.

Las ecuaciones (5.51) tienen sin embargo un carácter dispersivo que analizaremos más adelante.

En relación a la ecuación de transporte (5.5) en la que el sentido de progresión de las ondas ha sido invertido, como es de esperar, se tiene que el esquema regresivo es inestable y no converge mientras que el progresivo y centrado son convergentes de orden 1 y 2, respectivamente.

[11] La inestabilidad de esta ecuación a medida que $h \to 0$ se pone claramente de manifiesto a través del cambio de variable $v_h(x,t) = u_h(x+t,t)$. En este caso, se trata de una solución de la ecuación del calor retrógrada $v_t + hv_{xx} = 0$ que, tras el cambio de variables $w_h(x,t) = v_h(x,t/h)$, se convierte en una solución de la ecuación del calor retrógrada normalizada $w_t + w_{xx} = 0$. Así, vemos que $v_h(x,t) = [G_h(\tau - t) * v_h(\cdot,\tau)](x)$, para cada par de instantes de tiempo $0 < t < \tau$, siendo G_h el núcleo del calor reescalado: $G_h(x,t) = (4\pi ht)^{-1/2}\exp(-x^2/4ht)$. De esta expresión, aplicada con $t = 0$ de modo que $v_h(x,t) = f(x)$, se deduce fácilmente que $v_h(x,t)$ no está acotada en $L^\infty(0,T;L^2(\mathbf{R}))$, para ningún $T > 0$.

Para concluir esta sección consideremos el siguiente esquema completamente discreto para la aproximación de (5.1):

$$\frac{u_j^{k+1}-u_j^{k-1}}{2\Delta t}+\frac{u_{j+1}^k-u_{j-1}^k}{2\Delta x}=0. \tag{5.52}$$

Aquí y en lo sucesivo utilizamos las notaciones habituales de modo que Δt y Δx denotan los pasos del mallado en la dirección temporal y espacial respectivamente. Por otra parte, u_j^k denota la aproximación de la solución continua $u=u(x,t)$ de (5.1) en el punto $(x,t)=(x_j,t_k)=(j\Delta x,k\Delta t)$.

El esquema (5.52) está perfectamente centrado tanto en la variable espacial como temporal y se denomina esquema “leap-frog”.

Se trata de un esquema consistente de orden 2 y puede ser escrito en la forma

$$u_j^{k+1}=u_j^{k-1}+\mu\left[u_{j-1}^k-u_{j+1}^k\right] \tag{5.53}$$

donde μ es el número de Courant:

$$\mu=\Delta t/\Delta x. \tag{5.54}$$

El método de von Neumann permite analizar fácilmente la estabilidad del esquema. En este caso, la transformada de Fourier $\breve{u}^k(\theta)$ de la solución de (5.53) satisface

$$\breve{u}^{k+1}(\theta)=\breve{u}^{k-1}(\theta)+\mu\left[e^{-i\theta}-e^{i\theta}\right]\breve{u}^k(\theta)=\breve{u}^{k-1}(\theta)-2i\mu\operatorname{sen}(\theta)\breve{u}^k(\theta),$$

es decir,

$$\breve{u}^{k+1}(\theta)+2i\mu\operatorname{sen}(\theta)\breve{u}^k(\theta)-\breve{u}^{k-1}(\theta)=0. \tag{5.55}$$

En (5.55) vemos que cada componente de Fourier $\breve{u}^k(\theta)$ satisface un esquema de evolución discreto de dos pasos cuyos coeficientes dependen de $\theta\in[0,2\pi)$. Basta por tanto verificar si se satisface el criterio de la raiz. En este caso los ceros del polinomio característico del esquema (5.55) son

$$\lambda_\pm(\theta)=\frac{-2i\mu\operatorname{sen}(\theta)\pm\sqrt{-4\mu^2\operatorname{sen}^2(\theta)+4}}{2}. \tag{5.56}$$

Conviene entonces distinguir los dos siguientes casos:

- *Caso 1:* $\mu\leq 1$.

  En este caso

  $$|\lambda_\pm|^2=\frac{1}{4}\left[4\mu^2\operatorname{sen}^2\theta+4-\mu^2\operatorname{sen}^2\theta\right]=1$$

  con lo cual la estabilidad queda garantizada al ser las raices $\lambda_\pm$ simples.

- *Caso 2:* $\mu>1$.

  En este caso, cuando $\theta\sim\pi/2$ tenemos que

  $$-4\mu^2\operatorname{sen}^2(\theta)+4<0$$

y por lo tanto los ceros son de la forma

$$\lambda_\pm(\theta) = -i\left[\mu \operatorname{sen}\theta \mp \sqrt{\mu^2 \operatorname{sen}^2\theta - 1}\right].$$

La raiz de mayor módulo es la que corresponde al signo negativo. En este caso tenemos

$$|\lambda_-(\theta)| = \mu \operatorname{sen}\theta + \sqrt{\mu^2 \operatorname{sen}^2\theta - 1} > 1$$

puesto que $\mu \operatorname{sen}\theta > 1$.

El método es por tanto inestable en este caso.

De este análisis deducimos que el método completamente discreto de leap-frog es convergente de orden dos si y sólo si $\mu \leq 1$.

Es fácil comprobar también que $\mu \leq 1$ es precisamente la condición que garantiza que el dominio de dependencia del esquema discreto contiene el de la ecuación continua. Señalemos por último que el método consistente en sustituir el esquema numérico por una aproximación semejante escrita en términos de EDP puede también aplicarse en este caso. Obtendríamos ahora aproximaciones conservativas pero dispersivas de la ecuación de transporte de la forma

$$\sum_{m=0}^{M} \frac{(\Delta t)^{2m}}{(2m+1)!}\partial_t^{2m+1} + \sum_{\ell=0}^{L} \frac{(\Delta x)^{2\ell}}{(2\ell+1)!}\partial_x^{2\ell+1} = 0 \tag{5.57}$$

Tomando por ejemplo $L = M = 1$ obtenemos la ecuación:

$$\partial_t u + \partial_x u + \frac{(\Delta t)^2}{6}\partial_t^3 u + \frac{(\Delta x)^2}{6}\partial_x^3 u = 0.$$

Ahora bien, la ecuación de transporte indica que $\partial_t u = -\partial_x u$ y por tanto $\partial_t^2 = \partial_x^2$, de modo que la ecuación anterior puede escribirse del modo siguiente:

$$\partial_t[u + \frac{(\Delta t)^2}{6}\partial_x^2 u] + \partial_x[u + \frac{(\Delta x)^2}{6}\partial_x^2 u] = 0.$$

En esta última expresión es fácil comprobar el carácter conservativo de estas aproximaciones. En efecto, multiplicando en la ecuación por u e integrando en $\mathbb{R}$ obtenemos:

$$\int_{\mathbb{R}} \partial_t \Big(u + \frac{(\Delta t)^2}{6}\partial_x^2 u\Big) u dx + \int_{\mathbb{R}} \partial_x \Big(u + \frac{(\Delta x)^2}{6}\partial_x^2 u\Big) u dx + 0.$$

Ahora bien, tenemos,

$$\int_{\mathbb{R}} \partial_t u u dx = \frac{1}{2}\frac{d}{dt}\int_{\mathbb{R}} u^2 dx; \quad \int_{\mathbb{R}} \partial_x^3 u\, u dx = -\int_{\mathbb{R}} \partial_x^2 u\, \partial_x u\, dx = 0.$$

$$\int_{\mathbb{R}} \partial_t \partial_x^2 u\, u dx = -\int_{\mathbb{R}} \partial_t \partial_x u \partial_x u dx = -\frac{1}{2}\frac{d}{dt}\int_{\mathbb{R}} \mid \partial_x u \mid^2 dx; \quad \int_{\mathbb{R}} \partial_x u\, u dx = 0.$$

Obtenemos así la ley de conservación de la energía:

$$\frac{d}{dt}\Big[\frac{1}{2}\int_{\mathbb{R}}\Big[u^2-\frac{(\Delta t)^2}{6}\mid u_x\mid^2\Big]dx\Big]=0$$

Vemos sin embargo que el efecto dispersivo introducido por el esquema numérico hace que no sea la norma de u en $L^2(\mathbb{R})$ la que se conserve en tiempo sino la cantidad:

$$\int_{\mathbb{R}}\Big(u^2-\frac{(\Delta t)^2}{6}\mid u_x\mid^2\Big)dx$$

que, incluso puede ser negativa si la función u oscila rápidamente. Conviene sin embargo no olvidar que en las soluciones numéricas su máxima oscilación está limitada por el paso del mallado, por lo que esta cantidad nunca se puede hacer negativa en ellas. Son muchos los esquemas completamente discretos que surgen de manera natural en la aproximación de la ecuación de transporte (5.1), además del esquema "leap-frog" ya estudiado. La mayoría de ellos aparecen al realizar una aproximación discreta en tiempo de un esquema semidiscreto, pero no siempre es así. Obviamente, en caso de proceder a la obtención del esquema completamente discreto mediante la discretización temporal de un esquema semi-discreto, elegiremos uno que sea convergente puesto que si el esquema semi-discreto de partida fuese divergente, el esquema completamente discreto obtenido tampoco convergería. En vista de este hecho, conviene excluir inmediatamente los esquemas completamente discretos derivados del esquema semi-discreto progresivo (5.10) puesto que ya vimos que es inestable y por tanto divergente. Sin embargo, como el esquema regresivo (5.11) es convergente parece natural introducir el esquema de Euler regresivo

$$\frac{u_j^{k+1}-u_j^k}{\Delta t}+\frac{u_j^k-u_{j-1}^k}{\Delta x}=0 \tag{5.58}$$

o su versión implícita

$$\frac{u^{k+1}-u_j^k}{\Delta t}+\frac{u_j^{k+1}-u_{j-1}^{k+1}}{\Delta x}=0 \tag{5.59}$$

Nos referiremos a estos esquemas con ER (Euler regresivo) y ERI (Euler regresivo implícito), respectivamente. Ambos esquemas son de un paso temporal y consistentes de orden uno. Comprobemos pues su estabilidad. El esquema ER puede reescribirse como

$$u_j^{k+1}=u_j^k+\mu(u_{j-1}^k-u_j^k). \tag{5.60}$$

El análisis de von Neumann conduce al esquema discreto

$$\check{u}^{k+1}(\theta)=\check{u}^k(\theta)+\mu(e^{i\theta}\check{u}^k(\theta)-\check{u}^k(\theta))=\Big[1+\mu(e^{-i\theta}-1)\Big]\check{u}^k(\theta) \tag{5.61}$$

Para su estabilidad basta entonces comprobar si $\mid 1+\mu(e^{-i\theta}-1)\mid\leqslant 1$. Como

$$1+\mu(e^{-i\theta}-1) \;\;=\;\; 1+\mu[\cos\theta-i\operatorname{sen}\theta-1]=1+\mu(\cos\theta-1)-i\mu\operatorname{sen}\theta$$

tenemos que

$$\begin{aligned}
\mid 1+\mu(e^{-i\theta}-1)\mid^2 &= (1+\mu(\cos\theta-1))^2+\mu^2\operatorname{sen}^2\theta\\
&= 1+\mu^2(\cos\theta-1)^2+2\mu(\cos\theta-1)+\mu^2\operatorname{sen}^2\theta^2\\
&= 1+\mu^2(\cos^2\theta+1-2\cos\theta)+2\mu(\cos\theta-1)+\mu^2\operatorname{sen}^2\theta=1-2\mu
\end{aligned}$$

de donde deducimos que es estable, y por tanto convergente de orden uno, si y sólo si

$$\mid 1-2\mu \mid\leqslant 1 \Leftrightarrow \mu\leqslant 1. \tag{5.62}$$

Es fácil comprobar que esta condición de estabilidad es precisamente la que se obtiene al imponer que el dominio de dependencia del esquema discreto contenga al de la ecuación de transporte continua. En el caso del ERI tenemos

$$u_j^{k+1}=u_j^k-\mu(u_j^{k+1}-u_{j-1}^{k+1})$$

que al aplicar la transformada de Fourier, se convierte en

$$\breve{u}^{k+1}(\theta)=\breve{u}^k(\theta)-\mu(\breve{u}^{k+1}(\theta)-e^{i\theta}\breve{u}^{k+1}(\theta)), \tag{5.63}$$

es decir,

$$\Big[1+\mu(1-e^{-i\theta})\Big]\breve{u}^{k+1}(\theta)=\breve{u}^k(\theta),$$

o

$$\breve{u}^{k+1}(\theta)=[1+\mu(1-e^{-i\theta})]^{-1}\breve{u}^k(\theta). \tag{5.64}$$

La condición de estabilidad es entonces en este caso $\mid 1+\mu(1-e^{-1\theta})\mid\geqslant 1$. Como

$$1+\mu(1-e^{-i\theta})=1+\mu(1-\cos\theta)+i\mu\operatorname{sen}\theta$$

tenemos que

$$\begin{aligned}
\mid 1+\mu(1-e^{-i\theta})\mid^2 &= (1+\mu(1-\cos\theta))^2+\mu^2\operatorname{sen}^2\theta\\
&= 1+\mu^2(1+\cos^2\theta-2\cos\theta)+2\mu(1-\cos\theta)=\mu^2\operatorname{sen}^2\theta\\
&= 1+2\mu(1-\cos\theta)+2\mu^2(1-\cos\theta)\geqslant 1
\end{aligned}$$

y por tanto el método es incondicionalmente estable. A primera vista puede resultar sorprendente que el método ERI sea convergente para cualquier valor del número de Courant pues cabría preguntarse si la condición de inclusión de los dominios de dependencia se cumple con independencia del valor de μ. Esto es efectivamente así puesto que en el esquema discreto (5.59) el cálculo de u_j^{k+1} involucra a u_{j-1}^{k+1}, cuyo valor a su vez involucra a $u_{j-2}^{k+1},\cdots$. Vemos pues que el dominio de dependencia de ERI es el conjunto de todos los nodos del mallado, con independencia del valor de μ. De hecho cabe preguntarse sobre cual es el modo de resolver el sistema (5.59). Este sistema, con la notación vectorial habitual puede escribirse en la forma

$$B_\mu \overrightarrow{u}^{k+1} = \overrightarrow{u}^k$$

donde B_μ es una matriz infinita con valores $1+\mu$ en la subdiagonal. Se trata por tanto de una matriz infinita "triangular inferior" que define un operador acotado de ℓ^2 en ℓ^2. Pero, ¿se puede invertir el operador B_μ? Para comprobar que esto es efectivamente así, conviene utilizar el análisis de von Neumann. En efecto, el sistema equivalente (5.63) se resuelve inmediatamente y tiene como solución (5.64). Además, tal y como hemos visto en el análisis de la estabilidad del esquema

$$\mid \check{u}^{k+1}(\theta) \mid \leqslant \mid \check{u}^k(\theta) \mid, \quad \forall \theta \in [0, 2\pi).$$

Deducimos por tanto que

$$\begin{aligned} \parallel (u_j^{k+1})_{j\in\mathbb{Z}} \parallel_{\ell^2}^2 = \sum_{j\in\mathbb{Z}} \mid u_j^{k+1} \mid^2 = \frac{1}{2\pi}\int_0^{2\pi} \mid \check{u}^{k+1}(\theta) \mid^2 d\theta \\ \leqslant \tfrac{1}{2\pi}\int_0^{2\pi} \mid \check{u}^{k}(\theta) \mid^2 d\theta = \sum_{j\in\mathbb{Z}} \mid u_j^{k} \mid^2 = \parallel (u_j^{k})_{j\in\mathbb{Z}} \parallel_{\ell^2}^2 \end{aligned}$$

de modo que B_μ^{-1} está bien definido y es un operador acotado de ℓ^2 en ℓ^2 con norma no superior a uno. Vemos pues que la transformada discreta de Fourier permite probar la resolubilidad del sistema algebráico (5.59) que el método ERI plantea. Evidentemente hay muchos otros esquemas que pueden considerarse. Por ejemplo, el esquema de Crank-Nicolson (CN) inspirado en la regla del trapecio para la resolución de ecuaciones diferenciales y en la diferencia finita centrada para la aproximación de la derivada espacial:

$$\frac{u_j^{k+1} - u_j^k}{\Delta t} = -\frac{1}{2}\Big[\frac{u_{j+1}^k - u_{j-1}^k}{2\Delta x} + \frac{u_{j+1}^{k+1} - u_{j-1}^{k+1}}{2\Delta x}\Big]. \tag{5.65}$$

El método CN es convergente de orden dos para cualquier valor del parámetro de Courant μ. Vemos pues que CN preserva la propiedad que el método ERI de converger para todo valor de μ, pero tiene además la propiedad de ser de orden dos. El orden dos proviene de la combinación de los dos hechos siguientes: a) La utilización de diferencias centradas en la aproximación de la derivada espacial, lo cual da, efectivamente, una aproximación de orden dos de la derivada espacial; b) La utilización del método del trapecio en la aproximación de la derivada temporal, que es también un método de orden dos, aunque esta vez en tiempo. Nuevamente (5.65) es un sistema implícito. Pero se puede ver que es resoluble utilizando el argumento que hemos usado para el método ERI, mediante la transformada discreta de Fourier. Existen otros muchos métodos que proporcionan aproximaciones convergentes de la ecuación de transporte. Tenemos por ejemplo de "leap-frog" de orden cuatro (LF4):

$$\frac{u_j^{k+1} - u_j^{k-1}}{2\Delta t} = -\Big[\frac{4}{3}\Big[\frac{u_{j+1}^k - u_{j-1}^k}{2\Delta x}\Big] - \frac{1}{3}\Big[\frac{u_{j+2}^k - u_{j-2}^k}{4\Delta x}\Big]\Big] \tag{5.66}$$

que es de orden dos en tiempo y orden cuatro en espacio. En todos los métodos descritos hasta ahora puede observarse que han sido derivados en dos pasos, discretizando primero la variable espacial y después la temporal. Sin ir más lejos es obvio que (5.66) proviene de una semi-discretización de la forma

$$u_j'(t) = -\Big[\frac{4}{3}\Big[\frac{u_{j+1}(t)-u_{j-1}(t)}{2\Delta x}\Big] - \frac{1}{3}\Big[\frac{u_{j+2}(t)-u_{j-2}(t)}{4\Delta x}\Big]\Big] \tag{5.67}$$

que es efectivamente consistente con la ecuación de transporte. El paso de (5.67) a (5.66) es claro. Basta con utilizar el esquema de dos pasos

$$\frac{y^{k+1}-y^{k-1}}{2\Delta t} = f(y^k)$$

para la resolución de la ecuación diferencial

$$y'(t) = f(y(t)).$$

Pero, como decíamos, no todos los métodos discretos provienen de discretizar en tiempo una semi-discretización. Por ejemplo el método de la derivada oblicua es de la forma

$$u_j^{k+2} = (1-2\mu)\Big(u_j^{k+1} - u_{j-1}^{k+1}\Big) + u_{j-1}^k. \tag{5.68}$$

Se trata de un método de orden dos. Para ver que, efectivamente, es un método consistente con la ecuación de transporte lo escribimos como

$$\frac{u_j^{k+2} - u_{j-1}^k - u_j^{k+1} + u_{j-1}^{k+1}}{\Delta t} = -2\frac{(u_j^{k+1} - u_{j-1}^{k+1})}{\Delta x},$$

o, de manera más clara aún,

$$\frac{1}{2}\Big[\frac{u_j^{k+2}-u_j^{k+1}}{\Delta t} + \frac{u_{j-1}^{k+1}-u_{j-1}^k}{\Delta t}\Big] + \frac{u_j^{k+1}-u_{j-1}^{k+1}}{\Delta x} = 0, \tag{5.69}$$

expresión en la que queda claramente de manifiesto la analogía del esquema discreto con la ecuación de transporte continua. Citemos por último los esquemas de Lax-Wendroff

$$u_j^{k+1} = \frac{1}{2}\mu(1+\mu)u_{j-1}^k + (1-\mu^2)u_j^k - \frac{1}{2}\mu(1-\mu)u_{j+1}^k \tag{5.70}$$

y el esquema de Lax-Friedrichs

$$u_j^{k+1} = \frac{1}{2}(1-\mu)u_{j-1}^k + \frac{1}{2}(1+\mu)u_{j+1}^k. \tag{5.71}$$

El esquema de Lax-Friedrichs puede escribirse en la forma

$$\frac{u_j^{k+1} - \frac{1}{2}(u_{j-1}^k + u_{j+1}^k)}{\Delta t} = -\frac{u_{j+1}^k - u_{j-1}^k}{2\Delta x} \tag{5.72}$$

en la que queda claramente de manifiesto la consistencia con la ecuación de transporte. En esta expresión se observa que este esquema se obtiene a partir de la semi-discretización centrada realizando una discretización explícita de la derivada temporal en la que el valor u_j^k de la discretización más típica de u_t, (i.e. $(u_j^{k+1} - u_j^k)/\Delta t)$ ha sido sustituido por la media de los valores u_{j-1}^k y u_{j+1}^k. Es fácil comprobar que (5.72) es una aproximación difusiva de la ecuación de transporte. En efecto, basta aplicar formalmente en (5.72) el desarrollo de Taylor para observar que (5.72) da lugar en realidad a la siguiente corrección difusiva de la ecuación de transporte:

$$u_t - \frac{(\Delta x)^2}{2\Delta t} u_{xx} + u_x = 0,$$

que es análoga a la que obtuvimos para el esquema semi-discreto regresivo. Obviamente, se trata de un esquema discreto. Es de orden uno y tiene la propiedad de conservar la masa de la solución. En efecto, definimos la masa de la solución discreta como

$$m^k = \sum_{j \in \mathbb{Z}} u_j^k.$$

Basta entonces sumar con respecto a $j \in \mathbb{Z}$ en (5.71) para obtener que las soluciones del esquema de Lax-Friedrichs satisfacen $m^{k+1} = m^k$. Esto es una versión discreta de la propiedad de conservación de la masa que las soluciones $u(x,t) = f(x-t)$ de (5.1) satisfacen, i.e.

$$\int_{\mathbb{R}} u(x,t)dx = \int_{\mathbb{R}} f(x)dx, \quad \forall t \in \mathbb{R}.$$

Esta propiedad de conservación de la masa juega un papel relevante en la aproximación numérica de las ecuaciones de transporte no-lineales, como la ecuación de Burgers, y los esquemas que la verifican se dicen *conservativos*. Es fácil comprobar que el esquema de Lax-Friedrichs es estable (y, por tanto, convergente de orden uno) si y sólo

$$\mu = \Delta t / \Delta x \leqslant 1.$$

El esquema de Lax-Wendroff es consistente de orden dos y es fácil comprobar que es también un esquema conservativo y es convergente bajo la misma condición $\mu \leqslant 1$.

6. Dispersión numérica y velocidad de grupo

En el apartado anterior hemos estudiado la convergencia de diversos esquemas semi-discretos y completamente discretos de aproximación de la ecuación continua (5.1). Hemos comprobado que esquemas numéricos convergentes pueden introducir efectos disipativos o anti-disipativos que pueden ser respectivamente la causa de su estabilidad o inestabilidad o, por el contrario, ser puramente conservativos.

Sin embargo con independencia de su carácter convergente o divergente la mayoría de los esquemas numéricos tienen un carácter dispersivo. Por dispersión entendemos la propiedad de un sistema dinámico continuo o discreto en tiempo de propagar a diferentes velocidades las diversas componentes de la solución.

La ecuación de transporte (5.1) es precisamente un ejemplo claro de sistema no dispersivo pues, como vimos en la sección 5, todas sus soluciones son ondas de transporte puras que se propagan en el espacio a velocidad uno. Este hecho, obvio de la expresión explicita de la solución (5.3), puede también comprobarse a través del análisis de Fourier.

En efecto, consideremos soluciones u de (5.1) de la forma

$$u = e^{i\omega t} e^{i\xi x}, \tag{6.1}$$

es decir soluciones sinusoidales en variables separadas de frecuencia temporal ω y longitud de onda espacial $2\pi/\xi$, i.e. número de onda ξ.

Es fácil comprobar que u de la forma (6.1) es solución de (5.1) si y sólo si

$$\omega = -\xi. \tag{6.2}$$

En este caso la solución (6.1) adquiere la forma

$$u(x,t) = e^{i\omega(t-x)} \tag{6.3}$$

y se confirma lo observado en (5.3) en el sentido que las soluciones de (5.1) son meras ondas de transporte progresivas con velocidad uno.

La relación (6.2) es la que se denomina *relación de dispersión* para la ecuación de transporte (5.1).

Analicemos ahora, por ejemplo, el esquema semi-discreto regresivo que, como vimos en la sección anterior, es convergente de orden uno:

$$u_j' + \frac{u_j - u_{j-1}}{h} = 0. \tag{6.4}$$

Buscamos ahora soluciones de la forma

$$u_j(t) = e^{i\omega t} e^{i\xi x_j}. \tag{6.5}$$

Llevando la expresión (6.5) a la ecuación (6.4) obtenemos la ecuación

$$i\omega + \frac{1 - e^{-i\xi h}}{h} = 0,$$

es decir

$$\omega = \frac{i}{h}\left[1 - e^{-i\xi h}\right]. \tag{6.6}$$

La ecuación (6.6) es la relación de dispersión para el esquema semi-discreto (6.4).

Un simple desarrollo de Taylor permite comprobar que, en una primera aproximación, (6.6) coincide con la relación de dispersión (6.2) de la ecuación de transporte continua. En efecto,

$$\frac{i}{h}\left[1 - e^{-i\xi h}\right] = \frac{i}{h}\left[1 - \left[1 - i\xi h - \frac{\xi^2 h^2}{2} + O(h^3)\right]\right] = -\xi + \frac{i\xi^2 h}{2} + O(h^2). \tag{6.7}$$

En virtud de (6.6) la solución (6.5) de (6.4) puede escribirse en la forma

$$u_j(t) = e^{i\xi(x_j + \frac{\omega}{\xi}t)},$$

de donde vemos que la solución semi-discreta es una onda de transporte progresiva que avanza a una velocidad

$$c_h(\xi) = -\frac{\omega_h(\xi)}{\xi} = -\frac{i}{h\xi}(1 - e^{-i\xi h}) = 1 - \frac{i\xi h}{2} + O(h^2), \tag{6.8}$$

denominada *velocidad de fase.*

En la expresión (6.8) queda claramente de manifiesto el carácter dispersivo de la ecuación semi-discreta, en la medida en que la velocidad de propagación de la onda depende de la longitud de la misma.

Pero, cabría argumentar que la expresión (6.8) es un número complejo, por lo que no representa realmente una velocidad de transporte en el espacio fisico. Esto es debido al efecto disipativo que el esquema (6.4) introduce y que quedó claramente de manifiesto en su análogo continuo (5.45).

Consideremos ahora el esquema centrado

$$u_j' + \frac{u_{j+1} - u_{j-1}}{2h} = 0 \tag{6.9}$$

que, como vimos en la sección 5, es convergente de orden 2 y puramente conservativo.

En este caso se obtiene la relación de dispersión

$$i\omega + \frac{e^{i\xi h} - e^{-i\xi h}}{2h} = 0.$$

Es decir,

$$i\omega + \frac{i\,\mathrm{sen}(\xi h)}{h} = 0$$

o, equivalentemente,

$$\omega = -\frac{\mathrm{sen}(h\xi)}{h}. \tag{6.10}$$

Nuevamente observamos que (6.10) es una aproximación de la relación de dispersión (6.2) de la ecuación de transporte continua. De (6.10) se deduce que la velocidad de propagación de las ondas semi-discretas es en este caso

$$c_h(\xi) = \frac{\mathrm{sen}(\xi h)}{\xi h} = 1 - \frac{\xi^2 h^2}{3!} + O(h^4). \tag{6.11}$$

Comprobamos por lo tanto que las ondas en el medio semi-discreto se propagan más lentamente que en el medio continuo si bien, fijada la longitud de onda espacial, la velocidad de propagación $c_h(\xi)$, cuando $h \to 0$, converge a la velocidad de propagación en el caso continuo $c \equiv 1$. Obviamente, la convergencia de las velocidades de propagación está motivada por el hecho que el esquema sea convergente. En efecto, el esquema numérico no podría ser convergente si para algunas longitudes de onda las velocidades de propagación no convergiesen cuando el paso del mallado tiende a cero.

Consideremos por último el esquema “leap-frog” completamente discreto (5.52). En este caso buscamos ondas discretas de la forma

$$u_j^k = e^{i\omega\Delta t_k} e^{i\xi x_j} = e^{i\omega k\Delta t} e^{i\xi j\Delta x}. \tag{6.12}$$

Obtenemos entonces la relación de dispersión:

$$\frac{e^{i\omega\Delta t} - e^{-i\omega\Delta t}}{2\Delta t} + \frac{e^{i\xi\Delta x} - e^{-i\xi\Delta x}}{2\Delta x} = 0$$

que, en función del número de Courant $\mu = \Delta t/\Delta x$, puede reeescribirse como

$$\mathrm{sen}(\omega\Delta t) = -\mu\,\mathrm{sen}(\xi\Delta x),$$

o, de otro modo,

$$\omega = -\frac{1}{\Delta t}\mathrm{arcsen}\left[\mu\,\mathrm{sen}(\xi\Delta x)\right]. \tag{6.13}$$

Nuevamente es evidente que a medida que $\Delta x \to 0$, $\Delta t \to 0$ la relación de dispersión (6.13) se aproxima a la de la ecuación de transporte continua.

El caso

$$\mu = 1 \tag{6.14}$$

es particularmente interesante puesto que la relación de dispersión (6.13) se reduce a

$$\omega = -\xi \tag{6.15}$$

que es precisamente la correspondiente a la ecuación de transporte continua. En este caso las ondas discretas se propagan a velocidad constante idénticamente igual a uno, como lo hacen en el caso continuo.

Con el objeto de entender esta coincidencia de las velocidades de propagación continua y discretas conviene reescribir el esquema discreto con $\mu = 1$. Se obtiene en este caso

$$u_j^{k+1} - u_j^{k-1} + u_{j+1}^k - u_{j-1}^k = 0.$$

Es decir

$$u_j^{k+1} + u_{j+1}^k = u_j^{k-1} + u_{j-1}^k. \tag{6.16}$$

Habida cuenta que las soluciones de la ecuación de transporte continua son de la forma $u = f(x - t)$, se comprueba que, en este caso, son también soluciones exactas del esquema discreto (6.16) con $\mu = 1$. El esquema numérico es por tanto en este caso de orden infinito y reproduce de manera exacta las soluciones de la ecuación de transporte continua sobre los puntos del mallado.

Pero esto ocurre sólo cuando $\mu = 1$. Cuando $0 < \mu < 1$ el esquema es convergente pero también dispersivo. En efecto, en este caso la velocidad de propagación es

$$c_h(\xi) = -\frac{1}{\Delta t \xi} \operatorname{arcsen}\left[\mu \operatorname{sen}(\xi \Delta x)\right]. \tag{6.17}$$

Nuevamente observamos que, para cualquier valor del número de Courant $0 < \mu < 1$:

- $c_h(\xi) \to -1,\ h \to 0,\ \forall \xi;$

- $\mid c_h(\xi) \mid < 1,\ \forall h > 0,\ \forall \xi.$

La velocidad $c_h(\xi)$ describe de manera adecuada la propagación de las ondas semi-discretas o discretas que involucran un solo modo de Fourier. Son las que llamaremos ondas monocromáticas. Pero, cuando se superponen dos ondas con velocidades de propagación semejantes pero no idénticas surgen paquetes de ondas que pueden propagarse a velocidades distintas. Con el objeto de entender este fenómeno es conveniente introducir la noción de *velocidad de grupo.*

Para introducir esta noción consideremos cualquiera de los anteriores esquemas semi-discretos que admite soluciones de la forma

$$u_j(t) = e^{i\omega_h(\xi)t} e^{i\xi x_j}. \tag{6.18}$$

Superponiendo dos soluciones de esta forma con longitudes de ondas ξ y $\xi + \Delta\xi$ respectivamente obtenemos una nueva solución

$$u_{\Delta\xi, j}(t) = \frac{e^{i\omega_h(\xi)t} e^{i\xi x_j} - e^{i\omega_h(\xi+\Delta\xi)t} e^{i(\xi+\Delta\xi)x_j}}{\Delta\xi}$$

cuyo límite, cuando $\Delta\xi \to 0$, viene dado por

$$w_j(t) = -i[\omega_h'(\xi)t + x_j] e^{i\omega_h(\xi)t} e^{i\xi x_j}.$$

El resultado es un nuevo tipo de onda, producto de la solución (6.18) que se propaga a la velocidad de fase habitual $c_h(\xi)$ con la onda $g(x,t) = -i[\omega_h'(\xi)t + x_j]$ que se propaga a velocidad $\omega_h'(\xi)$ que se denomina *velocidad de grupo.* La velocidad de grupo es la que determina la propagación de paquetes de ondas conteniendo varias ondas de números de onda semejantes. Para comprobar este hecho basta con considerar la solución que se obtendría a partir de un dato inicial $f = f(x)$ con transformada de Fourier $F(\xi)$. La solución tendría entonces la expresión [12]:

$$u(x,t) = \int_{-\infty}^{+\infty} F(\xi) e^{i(\omega_h(\xi)t + \xi x)} d\xi = \int_{-\infty}^{+\infty} F(\xi) e^{it(\omega_h(\xi) + \xi x/t)} d\xi. \tag{6.19}$$

[12] Evitamos aquí las constantes multiplicativas de la transformada y antitransformada de Fourier que en nada afectan al fenómeno cualitativo que pretendemos ilustrar.

Supongamos ahora que fijamos el valor de x/t, lo cual corresponde a mover el origen de referencia a velocidad $x/t = cte$. Evidentemente, cuando $t \to \infty$ la exponencial del integrando oscila más y más con respecto a la variable ξ y tiende a cero en un sentido débil haciendo que la integral tienda a anularse. Esta cancelación ocurre efectivamente para todos los valores de ξ salvo para aquéllos en los que

$$\frac{d}{d\xi}(\omega_h(\xi) + \xi x/t) = 0. \tag{6.20}$$

Este hecho puede probarse de manera rigurosa mediante el Teorema de la Fase Estacionaria (TFE) (véase [6]).

La ecuación (6.20) puede también escribirse del modo siguiente:

$$\omega_h'(\xi) = -x/t. \tag{6.21}$$

Esta relación indica que, a medida que nos trasladamos en el espacio a velocidad x/t, sólo podemos ver las componentes cuyo número de onda ξ satisfaga la relación (6.20), o, dicho de otro modo, la energía asociada al número de onda ξ se propaga a una *velocidad de grupo*

$$C_h(\xi) = -\omega_h'(\xi). \tag{6.22}$$

Conviene en este punto señalar que la velocidad de fase $c_h(\xi)$ y la velocidad de grupo $C_h(\xi)$, en general, no coinciden. Analicemos este hecho en los ejemplos que hemos introducido más arriba. En el caso de la ecuación de transporte continua teníamos que $w(\xi) = -\xi$ para todo ξ. En este caso, obviamente $c_h(\xi) \equiv C_h(\xi) \equiv 1$, lo cual indica que todas las ondas se propagan a velociad uno en este modelo. Sin embargo en el esquema semi-discreto regresivo teníamos que

$$\omega = \frac{i}{h}\left[1 - e^{-i\xi h}\right]; \quad c_h(\xi) = -\frac{\omega_h(\xi)}{\xi} = 1 - \frac{i\xi h}{2} + O(h^2), \tag{6.23}$$

mientras la velocidad de grupo viene dada por la expresión

$$C_h(\xi) = -\omega_h'(\xi) = e^{-i\xi h} = 1 - i\xi h + O(h^2), \tag{6.24}$$

Se observa efectivamente una sutil diferencia entre las expresiones obtenidas en (6.23) y (6.24). Consideramos ahora el esquema centrado en el que, como veíamos anteriormente,

$$\omega = -\frac{\operatorname{sen}(h\xi)}{h}; \quad c_h(\xi) = \frac{\operatorname{sen}(\xi h)}{\xi h} = 1 - \frac{\xi^2 h^2}{6} + O(h^4). \tag{6.25}$$

En este caso la velocidad de grupo es

$$C_h(\xi) = \cos(\xi h) = 1 - \frac{\xi^2 h^2}{2} + O(h^4). \tag{6.26}$$

Nuevamente se observa una ligera diferencia en las expresiones de velocidad de fase y de grupo. Consideremos por último el esquema completamente discreto de "leap-frog". En aquél caso veíamos que

$$\omega_h(\xi) = \frac{1}{\Delta t}\text{arcsen}\left[\mu\,\text{sen}(\xi\Delta x)\right]; \quad c_h(\xi) = \frac{1}{\Delta t\xi}\text{arcsen}\left[\mu\,\text{sen}(\xi\Delta x)\right]. \tag{6.27}$$

Sin embargo, la velocidad de grupo viene dada por la expresión:

$$C_h(\xi) = \frac{\Delta x\mu\text{cos}(\xi\Delta x)}{\Delta t\sqrt{1-\mu^2\text{sen}^2(\xi\Delta x)}}, \tag{6.28}$$

que, nuevamente, difiere de la velocidad de fase, salvo en el caso $\mu = 1$ en el que $C_h(\xi) \equiv 1$. Estas, aparentemente, pequeñas diferencias entre la velocidad de fase y de grupo pueden sin embargo ser la causa de comportamientos inesperados de las soluciones de los esquemas numéricos.

7. Transformada discreta de Fourier a escala h

En la sección 5 hemos introducido y utilizado el método de von Neumann para el análisis de la estabilidad de un esquema numérico que está basado en la utilización de una transformada discreta de Fourier que permite:

∗ Definir una isometría entre ℓ^2 y $L^2(0, 2\pi)$;

∗ Transformar un esquema en diferencias en una ecuación diferencial dependiente de un parámetro $\theta \in [0,\, 2\pi)$.

La transformación de Fourier que introducimos en su momento, sin embargo, no tiene en cuenta el paso h del mallado puesto que se aplica meramente sobre sucesiones en ℓ^2, sin tener en cuenta el mallado al que están asociadas. Con el objeto de analizar el comportamiento de las soluciones cuando $h \to 0$ es conveniente introducir una *transformada de Fourier a escala h,* cuyo límite cuando $h \to 0$ sea la clásica transformada de Fourier, de modo que recuperemos en el límite la ecuación en derivadas parciales.

Recordemos en primer lugar la definición clásica de la transformada de Fourier continua

$$\widehat{f}(\xi) = \int_{\mathbb{R}} f(x)e^{-i\xi x}dx = \mathcal{F}(f). \tag{7.1}$$

Es bien sabido que la transformada de Fourier define una isometría de $L^2(\mathbb{R})$ en sí mismo. La transformada inversa de Fourier viene dada por

$$\mathcal{F}^{-1}(g)(x) = \frac{1}{2\pi}\int_{\mathbb{R}} g(\xi)e^{i\xi x}d\xi. \tag{7.2}$$

Una de las mayores utilidades de la transformada continua de Fourier es su posible utilización para la resolución de EDP con coeficientes constantes. En esto la siguiente propiedad juega un papel fundamental[13]

$$\widehat{\partial_x f}(\xi) = i\xi \widehat{f}(\xi). \tag{7.3}$$

Por ejemplo, gracias a la propiedad (7.3), la ecuación de transporte

$$u_t + u_x = 0, \tag{7.4}$$

mediante la aplicación de la transformación de Fourier en la variable x, se convierte en

$$\widehat{u}_t + i\xi \widehat{u} = 0 \tag{7.5}$$

de donde deducimos que

$$\widehat{u}(\xi, t) = e^{-i\xi t}\widehat{f}(\xi). \tag{7.6}$$

La expresión (7.6) ya nos confirma el carácter conservativo de la ecuación de transporte (7.4) puesto que proporciona la identidad

$$\mid \widehat{u}(\xi, t) \mid = \mid \widehat{f}(\xi) \mid, \quad \forall \xi \in \mathbb{R}, \quad \forall t > 0 \tag{7.7}$$

que, tras integración en $\xi \in \mathbb{R}$, asegura que

$$\parallel \widehat{u}(t) \parallel_{L^2(\mathbb{R})} = \parallel \widehat{f} \parallel_{L^2(\mathbb{R})}, \, \forall t > 0 \tag{7.8}$$

lo cual, a su vez, por el carácter isométrico de la transformada de Fourier, garantiza que

$$\parallel u(t) \parallel_{L^2(\mathbb{R})} = \parallel f \parallel_{L^2(\mathbb{R})}, \, \forall t > 0. \tag{7.9}$$

Introduzcamos pues ahora la transformada discreta de Fourier a escala h, una de cuyas propiedades más relevantes será que, en el límite cuando $h \to 0$, recuperaremos la transformada continua de Fourier que acabamos de definir.

Dada la sucesión $(f_j)_{j\in\mathbb{Z}}$ proveniente de un mallado espacial de paso h (i.e. de modo que $f_j \sim f(x_j)$ con $x_j = jh$), definimos la transformada discreta de Fourier a escala h como

$$\overset{\sqcap}{f}(\xi) = h \sum_{j\in\mathbb{Z}} f_j e^{-i\xi h j}, \quad -\frac{\pi}{h} \leqslant \xi \leqslant \frac{\pi}{h}. \tag{7.10}$$

Denotamos la transformada discreta de Fourier mediante el símbolo $\overset{\sqcap}{\cdot}$ para distinguirla de la transformada continua. A pesar de que la transformación (7.10) depende del parámetro h, no lo expresamos explícitamente en la notación para aligerarla.

[13] El lector interesado en un estudio de las propiedades básicas de la Transformada de Fourier y su aplicación a las EDP puede consultar los textos de F. John [12] y J. Rauch [16].

Vemos que la imagen mediante la transformada discreta de Fourier de una sucesión de paso h es una función continua con soporte en el intervalo $[-\pi/h,\ \pi/h]$. Obviamente, a medida que $h \to 0$, este soporte converge a toda la recta real. Este hecho refleja una de las propiedades fundamentales de la transformada de Fourier, a medida que el carácter oscilante de la función en el espacio físico aumenta, su transformada de Fourier se amplifica para las altas frecuencias. La sucesión discreta de paso h puede verse como una función que oscila a escala h (basta para ello extender la sucesión discreta de valores a una función constante o lineal a trozos definida en toda la recta real). El soporte de su transformada de Fourier aumenta, consecuentemente.

La transformación de Fourier discreta puede invertirse con facilidad. Tenemos

$$f_j = \frac{1}{2\pi} \int_{-\pi/h}^{\pi/h} \overset{\sqcap}{f}(\xi) e^{i\xi hj} d\xi \tag{7.11}$$

La analogía entre las fórmulas (7.1) y (7.2) de la transformada continua de Fourier y (7.10)-(7.11) de la transforma discreta a escala h son evidentes. Mientras que (7.10) se asemeja a una suma de Riemann de la integral (7.1) que define la transformada continua de Fourier sobre la partición $x_j = jh$, $j \in \mathbb{Z}$, la transformada discreta inversa (7.11) es simplemente una versión truncada de la integral (7.2) que define la transformada inversa de Fourier.

Es fácil también comprobar que la transformada discreta define una isometría:

$$\| \vec{f} \|_h^2 = h \sum_{j \in \mathbb{Z}} |f_j|^2 = \frac{1}{2\pi} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} |\ \overset{\sqcap}{f}(\xi)\ |^2\ d\xi = \frac{1}{2\pi} \left|\left| \overset{\sqcap}{f} \right|\right|^2_{L^2(-\frac{\pi}{h}, \frac{\pi}{h})}. \tag{7.12}$$

Esto, evidentemente, no es más que la versión discreta de la identidad de Parseval para la transformada de Fourier

$$\left|\left| f \right|\right|^2_{L^2(\mathbb{R})} = \frac{1}{2\pi} \int_{\mathbb{R}} |\ \widehat{f}(\xi)\ |^2\ d\xi = \frac{1}{2\pi} \left|\left| \widehat{f} \right|\right|^2_{L^2(\mathbb{R})}. \tag{7.13}$$

La relación entre transformada continua y discreta se hace más clara aún si utilizamos la *función cardinal* (también denominada función *cardinal de Whittaker* of *función de Shannon*, por su papel relevante en teoría de la comunicación):

$$\psi_0(x) = \frac{\operatorname{sen}(\pi x/h)}{\pi x/h}. \tag{7.14}$$

Denotamos mediante ψ_j su trasladada al punto $x_j = jh$, i.e.

$$\psi_j(x) = \frac{\operatorname{sen}(\pi(x - x_j)/h)}{\pi(x - x_j)/h}. \tag{7.15}$$

Dada una función discreta $(f_j)_{j \in \mathbb{Z}}$ de paso h definimos entonces la función continua

$$f^*(x) = \sum_{j \in \mathbb{Z}} f_j \psi_j(x). \tag{7.16}$$

Es fácil comprobar que la función continua f^* interpola la sucesión $(f_j)_{j\in\mathbb{Z}}$. En efecto,

$$f^*(x_j) = f_j,\ \forall j \in \mathbb{Z}. \tag{7.17}$$

Esto es simplemente debido a que

$$\psi_j(x_k) = \delta_{jk},\ \forall j, k \in \mathbb{Z}. \tag{7.18}$$

La función cardinal ψ_0 tiene además la interesante propiedad que[14]

$$\widehat{\psi}_0(\xi) = h1_{(-\pi/h,\,\pi/h)}(\xi). \tag{7.19}$$

Su transformada de Fourier es por tanto, módulo un factor multiplicativo h, la función característica del intervalo $\Big(-\dfrac{\pi}{h}, \dfrac{\pi}{h}\Big)$.

Es fácil probar que, como ψ_j se obtiene de ψ_0 mediante una nueva traslación, entonces

$$\widehat{\psi}_j(\xi) = e^{-i\xi jh}\widehat{\psi}_0(\xi). \tag{7.20}$$

Por otra parte, utilizando la identidad de Plancherel obtenemos que

$$\int_{\mathbb{R}} \psi_j(x)\psi_k(x)dx = \frac{1}{2\pi}\int_{\mathbb{R}} \widehat{\psi}_j(\xi)\overline{\widehat{\psi}_k}(\xi)d\xi = \frac{h^2}{2\pi}\int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} e^{i\xi h(k-j)}d\xi = h\delta_{jk}. \tag{7.21}$$

Vemos por tanto que las funciones $\{\psi_j(x)\}_{j\in\mathbb{Z}}$ son ortogonales. De esta propiedad de ortogonalidad deducimos fácilmente que

$$\Big|\Big|f^*\Big|\Big|^2_{L^2(\mathbb{R})} = h\sum_{j\in\mathbb{Z}} \mid f_j \mid^2 . \tag{7.22}$$

Por tanto, la extensión continua f^* de sucesiones de paso h define en realidad una isometría de ℓ^2 en un subespacio de $L^2(\mathbb{R})$.

Por otra parte, la transformada continua de Fourier de la función f^* está íntimamente ligada a la transformada discreta de la sucesión $(f_j)_{j\in\mathbb{Z}}$. En efecto,

$$\widehat{f^*}(\xi) = \sum_{j\in\mathbb{Z}} f_j\widehat{\psi_j}(\xi) = \sum_{j\in\mathbb{Z}} f_j e^{-i\xi jh}\widehat{\psi_0}(\xi) = h\sum_{j\in\mathbb{Z}} f_j e^{-i\xi jh}1_{(-\pi/h,\,\pi/h)} = \overset{\sqcap}{f}(\xi).$$

Vemos pues que *la transformada discreta de Fourier no es más que la transformada continua aplicada a la interpolación de la sucesión mediante la función cardinal.*

Estos resultados, probados por Whitakker en 1915 y utilizados en 1949 por Shannon, contribuyendo de manera decisiva a la teoría de la comunicación, indican que una función de *banda limitada* (cuya transformada de Fourier se anula fuera del intervalo $\xi \in [-B, B]$), puede ser reconstruida a través de la interpolación mediante la función cardinal a partir del muestreo de sus valores en los puntos $x_j = jh$, siempre que $h \leqslant \pi/B$. Retomaremos esta cuestión en la siguiente sección.

[14] Para comprobarlo observamos que $\partial_\xi 1_{[-A,A]} = \delta_A - \delta_{-A}$. Además $\mathcal{F}^{-1}(\partial_\xi 1_{[-A,A]}) = -i\xi\mathcal{F}^{-1}(1_{[-A,A]})$ y, por otra parte, $\mathcal{F}^{-1}(\delta_A - \delta_{-A}) = \frac{1}{2\pi}(e^{ixA} - e^{-ixA}) = \frac{i}{\pi}\,\mathrm{sen}(xA)$. De estas identidades deducimos (7.19) fácilmente, utilizando el hecho que la transformada de Fourier es un isomorfismo.

8. Revisión de la ecuación de transporte y sus aproximaciones a través de la transformada discreta de Fourier

Consideremos el problema de Cauchy para la ecuación de transporte

$$u_t + u_x = 0,\ x \in \mathbb{R},\ t > 0;\ u(x,0) = f(x),\ x \in \mathbb{R}. \tag{8.1}$$

Sabemos que la solución es una onda de transporte pura

$$u(x,t) = f(x-t). \tag{8.2}$$

Esta expresión puede también obtenerse mediante la transformación de Fourier. En efecto, como veíamos en (7.5),

$$\widehat{u}_t + i\xi\widehat{u} = 0,\ \xi \in \mathbb{R},\ t > 0, \widehat{u}(\xi,0) = \widehat{f}(\xi),\ \xi \in \mathbb{R}, \tag{8.3}$$

de donde deducimos que

$$\widehat{u}(\xi,t) = e^{-i\xi t}\widehat{f}(\xi). \tag{8.4}$$

Aplicando la transformada inversa obtenemos

$$u(x,t) = \mathcal{F}^{-1}(e^{-i\xi t}\widehat{f}(\xi)) = f(x-t). \tag{8.5}$$

En este último punto hemos usado el hecho de que la transformada de Fourier de la masa de Dirac es la constante unidad ($\widehat{\delta_0} \equiv 1$) o, equivalentemente, $\widehat{\delta_{x_0}} \equiv e^{-i\xi x_0}$.

Retomemos ahora el problema de la aproximación numérica de la solución.

Suponiendo que el dato inicial f es continuo, es natural tomar los datos discretos

$$f_j = f(x_j) = f(jh),\ j \in \mathbb{Z}, \tag{8.6}$$

lo cual supone realizar un muestreo de la función f.

Gracias a la fórmula de sumación de Poisson[15] es fácil comprobar que:

$$\overset{\sqcap}{f}(\xi) = \sum_{k\in\mathbb{Z}} \widehat{f}(\xi + k\omega_0),\ \forall -\pi/h \leqslant \xi \leqslant \pi/h, \tag{8.7}$$

donde

$$\omega_0 = 2\pi/h. \tag{8.8}$$

[15] La fórmula de sumación de Poisson asegura que $\sum_{j\in\mathbf{Z}} f(j) = \sum_{k\in\mathbf{Z}} \widehat{f}(2\pi k)$. Para comprobar esta fórmula basta considerar la función $g(x) = \sum_{j\in\mathbf{Z}} f(x+j)$, observar que es periódica de período uno y aplicar su desarrollo en series de Fourier. Los términos del sumando de la derecha son precisamente sus coeficientes de Fourier en la base $e^{i2\pi kx}$. Al aplicar este desarrollo en $x = 0$ obtenemos esta fórmula de sumación de Poisson. Al aplicar esta identidad a escala h a la función $f(x)e^{-i\xi x}$ obtenemos la identidad (8.7).

Si el dato inicial f es de banda limitada o, más precisamente, si $\widehat{f}(\xi) = 0$ para todo ξ tal que $\mid \xi \mid > \pi/h$, tenemos entonces

$$\overset{\sqcap}{f}(\xi) = \widehat{f}(\xi) \tag{8.9}$$

y por tanto el muestreo del dato inicial sobre los puntos del mallado no introduce error alguno.

Sin embargo, cuando f no es de banda limitada, en virtud de (8.7), las componentes de $\widehat{f}$ de altas frecuencias se superponen con las de la banda principal $[-\pi/h, \, \pi/h]$ dando lugar a lo que se conoce como fenómeno de *aliasing*. En este caso, la tranformada discreta de Fourier de la sucesión obtenida al muestrar f a lo largo de la sucesión $x_j = jh$ no permite recuperar la tranformada de Fourier de f y por tanto no permite codificar todas las características de la función f. De este análisis deducimos que una función f es de banda limitada si y sólo si se obtiene como una función de la forma f^* a través de las funciones de Shannon a partir de su muestreo a lo largo de la sucesión $x_j = jh$.

En la práctica es por tanto recomendable aproximar en primer lugar la función $f(x)$ para una familia de funciones de banda limitada

$$f_h(x) = \mathcal{F}^{-1}\Big(\widehat{f}(\xi) 1_{(-\pi/h, \, \pi/h)}(\xi)\Big) \tag{8.10}$$

que tienen la virtud de converger a f en $L^2(\mathbb{R})$ cuando $h \to 0$ y de forma que su muestreo no introduzca ningún error.[16]

Pero, dejando de lado los errores introducidos por la aproximación de los datos iniciales, consideremos el generado por los esquemas numéricos. Consideramos por tanto el esquema semi-discreto regresivo y progresivo (5.10) y (5.11) que, como vimos, son convergentes y divergentes respectivamente.

Revisión del esquema semi-discreto regresivo.

Consideremos en primer lugar el esquema

$$u_j'(t) + \frac{u_j(t) - u_{j-1}(t)}{h} = 0, \, j \in \mathbb{Z}, \, t > 0. \tag{8.11}$$

Aplicando la transformada discreta de Fourier a escala h obtenemos

$$\frac{d}{dt}\overset{\sqcap}{u}(\xi, t) + \frac{1}{h}(1 - e^{-i\xi h})\overset{\sqcap}{u}(\xi, t) = 0, \, t > 0, \, \xi \in [-\pi/h, \, \pi/h]. \tag{8.12}$$

En este punto hemos utilizado la siguiente propiedad fundamental de la transformada discreta de Fourier:

$$\overset{\sqcap}{\tau_{-1} f}(\xi) = h \sum_{j \in \mathbb{Z}} f_{j-1} e^{-i\xi jh} = e^{-i\xi h} h \sum_{j \in \mathbb{Z}} f_j e^{-i\xi jh} = e^{-i\xi h} \overset{\sqcap}{f}(\xi). \tag{8.13}$$

[16] La prueba de la convergencia de f_h a f en $L^2(\mathbb{R})$ se realiza combinando el Teorema de la Convergencia Dominada con el hecho de que $\mathcal{F}$ sea una isometría en $L^2(\mathbb{R})$.

La proximidad entre la ecuación de transporte continua (8.1) y la aproximación semi-discreta regresiva (8.11) es evidente. El coeficiente

$$\omega_h(\xi) = \frac{1}{h}(1 - e^{-i\xi h}) \tag{8.14}$$

que interviene en la ecuación diferencial (8.12) converge, cuando $h \to 0$, de manera evidente, al coeficiente

$$\omega(\xi) = i\xi \tag{8.15}$$

correspondiente a la ecuación de transporte continua.

De hecho, mediante el desarrollo de Taylor se observa que

$$\omega_h(\xi) = i\xi + \frac{\xi^2 h}{2} + \cdots . \tag{8.16}$$

En la expresión (8.16) se observa que, efectivamente, para cada $\xi \in \mathbb{R}$,

$$\omega_h(\xi) \to C(\xi) \text{ cuando } h \to 0. \tag{8.17}$$

Además en (8.16) volvemos a constatar el carácter difusivo de la aproximación regresiva. En efecto, esto queda de manifiesto en que el primer término corrector en (8.16) $(\xi^2 h/2)$ sea real y positivo.

Conviene sin embargo observar que la convergencia (8.17) es sólamente uniforme en conjuntos R_h en los que

$$\max_{\omega \in R_h} \xi^2 h \to 0,\ h \to 0. \tag{8.18}$$

En otras palabras, la convergencia de los *símbolos* (8.17) sólo se produce en regiones en las que

$$\mid \xi \mid = O(h^{-1/2}),\ h \to 0. \tag{8.19}$$

Sin embargo, conviene señalar que la convergencia (8.17) interesa para cualquier $\xi \in \mathbb{R}$. En efecto, en el límite cuando $h \to 0$, la banda de frecuencias del dato inicial continuo $f = f(x)$ de la ecuación de transporte en toda la recta real. Por otra parte, a medida que $h \to 0$, la banda de frecuencias de los datos iniciales del problema discreto $[-\pi/h, \pi/h]$ aumenta hasta cubrir toda la recta real.

En virtud de que la convergencia (8.17) es uniforme en conjuntos de la forma (8.19) es fácil ver que las soluciones del problema discreto convergen a las del continuo para datos con una banda de frecuencias limitada, independiente de h. La estabilidad del esquema permite después extender esta convergencia a un dato inicial cualquiera $f \in L^2(\mathbb{R})$.

En efecto, en virtud de (8.12) tenemos

$$\overset{\sqcap}{u}(\xi, t) = e^{-\omega_h(\xi)t}\overset{\sqcap}{f}(\xi) = e^{-\frac{1}{h}(1-e^{-i\xi h})t}\overset{\sqcap}{f}(\xi) = e^{-\frac{1}{h}(1-\cos(\xi h)t)}e^{-i\,\mathrm{sen}(\xi h)t/h}\overset{\sqcap}{f}(\xi).$$

Aplicando la anti-transformada discreta de Fourier tenemos

$$u_j(t) = \frac{1}{2\pi} \int_{-\pi/h}^{\pi/h} e^{-\frac{1}{h}(1-\cos(\xi h)t)} e^{i\xi t(jh-\mathrm{sen}(\xi h)/\xi h)} \overset{\sqcap}{f}(\xi) d\xi. \tag{8.20}$$

En virtud de (8.9) sabemos que, si f es de banda acotada, $\overset{\sqcap}{f} \equiv \widehat{f}$, para h suficientemente pequeño. Bajo estas hipótesis es por tanto evidente que la solución del problema numérico puede reescribirse como

$$u_j(t) = \frac{1}{2\pi} \int_{-B}^{B} e^{-\frac{1}{h}(1-\cos(\xi h)t)} e^{i\xi t(jh-\mathrm{sen}(\xi h)/\xi h)t} \widehat{f}(\xi) d\xi, \tag{8.21}$$

donde $B > 0$ es tal que $\mathrm{sop}(\widehat{f}) \subset [-B,\, B]$.

Elegimos ahora $j \in \mathbb{Z}$ de modo que $jh = x_0$, siendo $x_0 \in \mathbb{R}$ un punto fijado. Evidentemente, esto supone elegir $j = x_0/h$ que, para que $j \in \mathbb{Z}$, exige a su vez tomar una sucesión determinada de $h \to 0$. Bajo esta condición ($x_0 = jh$), deberíamos ser capaces de ver que la expresión (8.21) converge, cuando $h \to 0$, al valor de la solución continua $u(x_0, t) = f(x_0 - t)$. Veámos que esto es efectivamente así. Cuando $h \to 0$, el integrando de (8.21) converge a $e^{i\xi(x_0-1)t}\widehat{f}(\xi)$. La aplicación del Teorema de la convergencia dominada permite entonces ver que el límite de (8.21) es

$$u(x_0, t) = \frac{1}{2\pi} \int_{-B}^{B} e^{i\xi(x_0-1)t} \widehat{f}(\xi) d\xi = \frac{1}{2\pi} \int_{-B}^{B} e^{i\xi x_0} e^{-i\xi t} \widehat{f}(\xi) d\xi = f(x_0 - t), \tag{8.22}$$

que coincide con la solución del problema continuo, gracias a la hipótesis de que f sea de banda limitada.

En realidad, bajo estas hipótesis, se puede probar que la convergencia de la solución discreta a la continua tiene lugar en la norma L^2. Se puede ver esto de dos maneras. Tomando normas discretas de diferencias en ℓ^2 o bien tomando normas continuas en $L^2(\mathbb{R})$ observando que la expresión (8.21) de la solución del esquema numérico puede extenderse a una función continua con respecto a la variable espacial x, dependiente del parámetro h:

$$u_h(x, t) = \frac{1}{2\pi} \int_{-B}^{B} e^{-\frac{1}{h}(1-\cos(\xi h)t)} e^{i\xi(x-\mathrm{sen}(\xi h)/\xi h)t} \widehat{f}(\xi) d\xi. \tag{8.23}$$

Combinando (8.22) y (8.23) vemos que, tanto la solución continua como la discreta pueden ser escritas de un modo semejante mediante la transformada inversa de Fourier

$$u_h(x, t) = \mathcal{F}^{-1}\Big[e^{-\frac{1}{h}(1-\cos(\xi h)t)} e^{-i\,\mathrm{sen}(\xi h)t/h} \widehat{f}\Big](x) \tag{8.24}$$

y

$$u(x, t) = \mathcal{F}^{-1}\Big[e^{-i\xi t} \widehat{f}\Big](x). \tag{8.25}$$

Para comprobar que $u_h(t) \to u(t)$ en $L^2(\mathbb{R})$ cuando $h \to 0$ basta entonces ver que

$$e^{-\frac{1}{h}(1-\cos(\xi h)t)} e^{i\,\mathrm{sen}(\xi h)t/h} \widehat{f}(\xi) \to e^{-i\xi t} \widehat{f}(\xi) \text{ en } L^2(\mathbb{R})$$

cuando $h \to 0$. Teniendo en cuenta que f es, por hipótesis, de banda acotada, vemos que esto es equivalente a que

$$\int_{-B}^{B} \left| e^{-\frac{1}{h}(1-\cos(\xi h)t)} e^{-i\,\mathrm{sen}(\xi h)t/h} - e^{-i\xi t} \right|^2 \mid \widehat{f}(\xi) \mid^2 d\xi \to 0$$

y esto, efectivamente, ocurre en virtud del Teorema de la convergencia dominada.

Esto confirma la convergencia del esquema regresivo para datos iniciales con banda acotada. Para considerar el caso general, dado $f \in L^2(\mathbb{R})$ basta introducir su aproximación

$$f_B(x) = \mathcal{F}^{-1}\Big(\widehat{f}(\xi) 1_{(-B,B)}(\xi)\Big)$$

que es, por definición, una función de banda acotada tal que

$$f_B \to f \text{ en } L^2(\mathbb{R}) \text{ cuando } B \to \infty.$$

Denotamos mediante u y u_B la solución de la ecuación de transporte con datos f y f_B respectivamente. Como

$$\| u(t) - u_B(t) \|_{L^2(\mathbb{R})} = \| f - f_B \|_{L^2(\mathbb{R})}$$

vemos que

$$u_B \to u \text{ en } L^\infty(0,\infty;\, L^2(\mathbb{R})),\ B \to \infty.$$

Por tanto, dado $\varepsilon > 0$ existe $B_0 > 0$ suficientemente grande tal que

$$\| u - u_{B_0} \|_{L^\infty(0,\infty;L^2(\mathbb{R}))} \leqslant \varepsilon/2.$$

Fijado este valor de B_0 y resolviendo la ecuación semi-discreta con dato inicial f_{B_0} muestreado sobre el mallado tenemos que

$$u_{h,B_0}(t) \to u_{B_0}(t),\ h \to 0, \text{ en } L^2(\mathbb{R})$$

para cada $t > 0$. Por tanto, para h suficientemente pequeño

$$\| u(t) - u_{h,B_0}(t) \|_{L^2(\mathbb{R})} \leqslant \| u(t) - u_{B_0}(t) \|_{L^2(\mathbb{R})} + \| u_{B_0}(t) - u_{h,B_0}(t) \|_{L^2(\mathbb{R})} \leqslant \varepsilon/2 + \varepsilon/2 = \varepsilon.$$

Esto demuestra la convergencia para datos iniciales obtenidos muestreando una aproximación del dato inicial de banda acotada. Evidentemente, gracias a la estabilidad del esquema numérico, se puede obtener el mismo resultado de convergencia para cualquier elección de la aproximación de los datos iniciales que converja al dato inicial del problema continuo.

Revisión del esquema semi-discreto progresivo.

Consideramos ahora el caso progresivo que, como vimos, es inestable y divergente. Analicémoslo pues con la herramienta que las transformadas de Fourier proporcionan.

En este caso el esquema es de la forma

$$u_j'(t) + \frac{u_{j+1} - u_j(t)}{h} = 0,\, j \in \mathbb{Z},\, t > 0. \tag{8.26}$$

Aplicando la transformada discreta de Fourier obtenemos

$$\frac{d}{dt}\overset{\sqcap}{u}(\xi, t) + \frac{1}{h}(e^{i\xi h} - 1)\overset{\sqcap}{u}(\xi, t) = 0,\, t > 0,\, \xi \in [-\pi/h,\, \pi/h], \tag{8.27}$$

de modo que

$$\overset{\sqcap}{u}(\xi, t) = e^{-\frac{1}{h}(e^{i\xi h}-1)t}\overset{\sqcap}{f}(\xi) = e^{\frac{1}{h}(1-\cos(\xi h))t}e^{-i\,\mathrm{sen}(\xi h)t/h}\overset{\sqcap}{f}(\xi). \tag{8.28}$$

Pretendemos ahora ilustrar de manera aún más explícita la ausencia de convergencia de este método. Para ello tomamos un dato inicial f de banda acotada de modo que $\overset{\sqcap}{f} \equiv \widehat{f}$ para h suficientemente pequeño.

Obtenemos así

$$\overset{\sqcap}{u}(\xi, t) = e^{\frac{1}{h}(1-\cos(\xi h))t}e^{-i\,\mathrm{sen}(\xi h)t/h}\widehat{f}(\xi), \tag{8.29}$$

de modo que

$$\mid \overset{\sqcap}{u}(\xi, t) \mid = e^{\frac{1}{h}(1-\cos(\xi h))t} \mid \widehat{f}(\xi) \mid, \tag{8.30}$$

y entonces

$$\parallel \vec{u}_h \parallel_h^2 = \frac{1}{2\pi}\int_{-B}^{B} e^{\frac{2}{h}(1-\cos(\xi h))t} \mid \widehat{f}(\xi) \mid^2 d\xi. \tag{8.31}$$

Habida cuenta que $\xi \in [-B,\, B]$, tenemos que

$$1 - \cos(\xi h) \geqslant c\xi^2 h^2 \tag{8.32}$$

con $c > 0$ para todo $\xi \in [-B\, B]$, a condición que h sea suficientemente pequeño.

Combinando (8.31) y (8.32) vemos que

$$\parallel \vec{u}_h(t) \parallel_h^2 \geqslant \frac{1}{2\pi}\int_{-B}^{B} e^{ch\xi^2 t} \mid \widehat{f}(\xi) \mid^2 d\xi. \tag{8.33}$$

Pero esta estimación es claramente insuficiente para concluir la divergencia del método puesto que la integral a la derecha de (8.33) permanece acotada cuando $h \to 0$.

Para ilustrar la divergencia hemos considerar datos iniciales de banda más ancha. Dado $f \in L^2(\mathbb{R})$ tal que el soporte de su transformada de Fourier sea toda la recta real (por ejemplo la Gaussiana[17]), introducimos el dato inicial del esquema discreto $f_h(x)$ truncando la transformada de Fourier de f a la banda admisible $[-\pi/h,\, \pi/h]$, i.e.

$$f_h(x) = \mathcal{F}^{-1}(\widehat{f}1_{(-\pi/h,\, \pi/h)}(\xi)). \tag{8.34}$$

[17] Es bien sabido que la transformada de Fourier de la función $f(x) = e^{-x^2/2}$ es la Gaussiana $\widehat{f}(\xi) = \sqrt{2\pi}e^{-\xi^2/2}$.

En este caso la norma de la aproximación discreta viene dada por

$$\| \vec{u}_h(t) \|_h^2 = \frac{1}{2\pi} \int_{-\pi/h}^{\pi/h} e^{\frac{2}{h}(1-\cos(\xi h))t} \mid \widehat{f}(\xi) \mid^2 d\xi. \tag{8.35}$$

Utilizando ahora el hecho que

$$1 - \cos(\eta) \geqslant c\eta^2, \quad \forall \eta \in [-\pi, \pi], \tag{8.36}$$

vemos que

$$\| \vec{u}_h(t) \|_h^2 \geqslant \frac{1}{2\pi} \int_{-\pi/h}^{\pi/h} e^{ch\xi^2 t} \mid \widehat{f}(\xi) \mid^2 d\xi. \tag{8.37}$$

En esta ocasión la integral a la derecha de (8.37) puede diverger puesto que en la banda $\xi \in [-\pi/h, \pi/h]$ hay zonas donde $h\xi^2 \to \infty$. Para comprobar la divergencia de esta integral con más detalle consideremos el dato inicial f de modo que

$$\widehat{f}(\xi) = \sum_{k \in \mathbb{Z}} \alpha_k 1_{I_k}(\xi) \tag{8.38}$$

donde $(I_k)_{k\in\mathbb{Z}}$ son intervalos disjuntos de $\mathbb{R}$. Para que $f = \mathcal{F}^{-1}(\widehat{f}) \in L^2(\mathbb{R})$ basta entonces con que

$$\sum_{k \in \mathbb{Z}} \alpha_k^2 \mid I_k \mid < \infty, \tag{8.39}$$

donde $\mid I_k \mid$ denota la longitud del intervalo I_k.

En este caso la integral a la derecha (8.37) puede reescribirse como

$$\frac{1}{2\pi} \int_{-\pi/h}^{\pi/h} e^{ch\xi^2 t} \mid \widehat{f}(\xi) \mid^2 d\xi = \frac{1}{2\pi} \sum_{k \in \mathbb{Z}} \alpha_k^2 \int_{I_k \cap [-\pi/h, \, \pi/h]} e^{ch\xi^2 t} d\xi. \tag{8.40}$$

Si elegimos los intervalos $I_k = (k, \, k+1)$ observamos que el último sumatorio puede acotarse inferiormente por

$$\frac{1}{2\pi} \alpha_{k_0}^2 e^{ch(\frac{\pi}{h}-1)^2 t} \tag{8.41}$$

con $k_0 = \frac{\pi}{h} - 1$.

En este caso, además, la condición (8.39) puede simplemente reescribirse como

$$\sum_{k \in \mathbb{Z}} \alpha_k^2 < \infty. \tag{8.42}$$

Es evidente que es perfectamente posible elegir una sucesión α_k de la forma $\alpha_k = 1/k$, de modo que (8.42) se cumpla y que, sin embargo, la cota inferior (8.41) de la norma ℓ^2 de la solución discreta correspondiente diverja cuando $h \to 0$ con un orden $e^{ct/h}$.

Vemos por tanto que la utilización de la transformada de Fourier a escala h permite ilustrar de manera mucho más cuantitativa la divergencia del método semi-discreto progresivo que habíamos predicho mediante el análisis de von Neumann. En el caso en que el esquema es convergente puede también aplicarse para ilustrar de un modo más claro la convergencia hacia la solución del problema continuo.

A pesar de que en esta sección sólo hemos analizado las aproximaciones semi-discretas progresiva y regresiva las ideas que hemos desarrollado son completamente generales y pueden ser aplicadas al estudio de cualquier otro esquema, en particular para los completamente discretos. *Revisión del comportamiento de las velocidades de fase y grupo.* Ahora que sabemos

que el rango de frecuencias relevantes para una aproximación numérica es $-\pi/h \leq \xi \leq \pi/h$, conviene revisar los conceptos de velocidades de fase y grupo. Consideremos en primer lugar el esquema centrado semi-discreto. En este caso la velocidad de fase viene dada por

$$c_h(\xi) = \frac{\text{sen}(\xi h)}{\xi h}. \tag{8.43}$$

Vemos entonces que la velocidad de fase se anula cuando $\xi h = \pm\pi$. Se trata evidentemente de un fenómeno nuevo con respecto a la ecuación de transporte continua donde todas las componentes de Fourier de las soluciones se transportan a velocidad constante uno. En virtud de este hecho, para cada $h > 0$ fijo, existen soluciones del problema numérica que apenas se tranportan. Esto no es incompatible con la convergencia de orden dos del esquema numérico centrado que ya comprobamos. En efecto, en el problema clásico de la convergencia, el dato inicial se supone fijo, lo cual, en la práctica, gracias a la propiedad de estabilidad del esquema, permite filtrar las altas frecuencias del dato inicial y considerar únicamente datos cuya transformada de Fourier tiene soporte compacto. El hecho de que la velocidad de propagación se anule cuando $|\xi| \sim \pi/h$, no tiene entonces efectos a nivel de la converegncia. Pero, insistimos, si lo que nos interesa es la dinámica de las soluciones para h pequeño pero fijo, este hecho tiene un gran impacto puesto que surgen soluciones que nada tienen que ver con el comportamiento de la ecuación de transporte continua. Se trata del mismo fenómeno que surge al estudiar la estabilidad absoluta de los sistemas stiff de ecuaciones diferenciales ordinarias (véase por ejemplo [11]). En el caso del sistema centrado este hecho especialmente grave puesto que el esquema es puramente conservativo y por tanto estas soluciones a altas frecuencias en absoluto se disipan. Diremos que se trata de *soluciones espúreas*, en el sentido que son ficticias puesto que no corresponden a la ecuación de transporte continua y sólo surgen como soluciones del esquema numérico. Por otra parte, la velocidad de grupo en este caso toma el valor

$$C_h(\xi) = \cos(\xi h). \tag{8.44}$$

Vemos que la situación es aún peor puesto que se anula cuando $\xi h = \pi/2$ y tiene signo negativo para todo $\xi h \in (\pi/2, \pi]$. En este caso por tanto tendremos incluso soluciones que se transportan en la dirección opuesta a la de la ecuación de transporte continua. Se trata de un fenómeno de soluciones numéricas espúreas que no es incompatible con la convergencia del esquema numérico. Consideremos ahora el esquema numérico regresivo que, como vimos, tiene un carácter disipativo. En este caso la velocidad de fase viene dada por:

$$c_h(\xi) = \frac{i(e^{-i\xi h} - 1)}{\xi h} = \frac{\operatorname{sen}(\xi h)}{\xi h} + i\frac{\cos(\xi h) - 1}{\xi h}. \tag{8.45}$$

Vemos que la parte real de la velocidad de fase se comporta como en el caso de la aproximación centrada de modo que ésta se anula cuando $\xi h = \pm\pi$. Sin embargo, vemos también que para estos valores de frecuencias la parte imaginaria de la velocidad de grupo es estrictamente negativa, lo cual asegura que estas componentes de Fourier de la solución decaen exponencialmente en tiempo. Vemos pues que la aproximación regresiva, a pesar de introducir soluciones numéricas espúreas, las disipa. Es a causa de este hecho que la soluciones del esquema regresivo para $h > 0$ pequeño y fijo se comportan de manera mucho más semejante a las de la ecuación de transporte continua que las del esquema centrado. Lo mismo ocurre con la velocidad de grupo. De este análisis concluimos que más allá de las propiedades de convergencia clásicas de un esquema numérico, con el objeto de garantizar que para $h > 0$ pequeño la dinámica del esquema discreto se asemeja a la del continuo es preciso tener en cuenta el comportamiento de las velocidades de fase y de grupo en frecuencias $|\xi|$ del orden de c/h con $0 < c < \pi$.

Referencias

[1] Bender, C.M. and Orszag, S.A. (1978). *Advanced Mathematical Methods for Scientists and Engineers*, McGraw-Hill.

[2] Brezis, H. (1983). *Analyse Fonctionnelle, Théorie et Applications*, Masson, Paris.

[3] Cazenave, T. and Haraux A. (1989), *Introduction aux problèmes d'évolution semilinéaires*, Mathématiques & Applications, Ellipses, Paris.

[4] Cohen, G. (2001). *Higher-order numerical methods for transient wave equations.* Scientific Computation, Springer.

[5] Eastham, M.S.P. (1973). *The Spectral Theory of Periodic Differential Equations*, Scottish Academic Press, Edinburgh.

[6] Evans, L. C. (1998). *Partial Differential Equations*, Graduate Studies in Mathematics, Vol.19, AMS.

[7] Glowinski, R. (1992). "Ensuring well-posedness by analogy; Stokes problem and boundary control of the wave equation". *J. Compt. Phys.*, **103**(2), 189–221.

[8] Glowinski, R., Li, C. H. and Lions, J.-L. (1990). "A numerical approach to the exact boundary controllability of the wave equation (I). Dirichlet controls: Description of the numerical methods". *Japan J. Appl. Math.*, **7**, 1–76.

[9] Infante, J.A. and Zuazua, E. (1999). "Boundary observability for the space-discretizations of the one-dimensional wave equation", *Mathematical Modelling and Numerical Analysis,* **33**, 407–438.

[10] Isaacson, E. and Keller, H.B. (1966). *Analysis of Numerical Methods*, John Wiley & Sons.

[11] Iserles, A. (1996) *A First Course in the Numerical Analysis of Differential Equations*, Cambridge Texts in Applied Mathematics, Cambridge University Press.

[12] John, F. (1982) *Partial differential Equations,* (4. ed), Springer.

[13] LeVeque, R. J. (1992). *Numerical methods for conservation laws.* Second edition. Lectures in Mathematics ETH Zürich. Birkhäuser Verlag, Basel.

[14] Negreanu, M. and Zuazua, E. (2003). "Uniform boundary controllability of a discrete 1D wave equation. *Systems and Control Letters*, **48** (3-4) , 261-280.

[15] Quarteroni A. y Valli, A. (1998). *Numerical approximation of Partial differential Equations*, Springer, Springer Series in Computational Mathematics, 23.

[16] Rauch, J. (1991) *Partial Differential Equations*, Graduate Texts in Mathematics, Springer Verlag.

[17] Sanz-Serna, J. (1985). "Stability and convergence in numerical analysis. I: linear problems—a simple, comprehensive account". *Res. Notes in Math.*, **132**, Pitman, Boston, MA, pp. 64–113.

[18] Trefethen, L. N. (1982). "Group velocity in finite difference schemes", *SIAM Rev.,* **24** (2), pp. 113–136.

[19] Vichnevetsky, R. and Bowles, J.B. (1982). *Fourier Analysis of Numerical Approximations of Hyperbolic Equations.* SIAM Studies in Applied Mathematics, **5**, SIAM, Philadelphia.

[20] Young, R. M. (1980). *An Introduction to Nonharmonic Fourier Series*, Academic Press, New York.

[21] Zuazua, E. (1999). "Boundary observability for the finite-difference space semi-discretizations of the 2D wave equation in the square", *J. Math. Pures et Appliquées*, **78**, 523–563.

[22] Zuazua, E. (2002). “Controllability of Partial Differential Equations and its Semi-Discrete Approximations”. *Discrete and Continuous Dynamical Systems*, **8** (2), 469–513.

[23] Zuazua, E. (1999). “Observability of 1D waves in heterogeneous and semi-discrete media”. *Advances in Structural Control.* J. Rodellar et al., eds., CIMNE, Barcelona, pp. 1–30.

[24] Zuazua, E. (2003). “Propagation, Observation, Control and Numerical Approximation of Waves”, preprint (http://www.uam.es/enrique.zuazua).

Some applications of combinatorial optimization in telecommunications

Mauricio G. C. Resende[*]

Abstract

Combinatorial optimization problems are abundant in the telecommunications industry. In this paper, we present three real-world telecommunications applications where combinatorial optimization plays a major role. The first problem concerns the optimal location of modem pools for an internet service provider. The second problem deals with the optimal routing of permanent virtual circuits for a frame relay service. The last problem comes up when routing packets on the Internet.

Keywords: Telecommunications, combinatorial optimization, linear programming, GRASP, local search, multi-commodity flows, genetic algorithms, OSPF routing.

1 Introduction

Combinatorial optimization problems are abundant in the telecommunications industry. In this paper, we present three real-world telecommunications applications where combinatorial optimization plays a major role.

In Section 2, we consider the PoP (point-of-presence) placement problem, an optimization problem confronted by internet access providers. The most common, and potentially least expensive, way for a customer to access the internet is with a modem by making a phone call to a PoP of the provider. It has been conjectured that potential customers are more likely to subscribe to internet access service if they can make a local (free unmetered) phone call to access at least one of the internet provider's PoPs. Given that the number of PoPs that can be deployed is limited by a number of constraints, such as budget and office capacity, one would like to place (or locate) the PoPs in a configuration that maximizes the number of customers than can make local calls to at least one PoP. We call this number of customers the *coverage*. A greedy randomized adaptive search procedure (GRASP) is used to find solutions to this location problem that, in real-world situations, are shown to be near-optimal. We follow closely the presentation given in Resende [1].

[*] ATT, USA. E-mail:mgcr@research.att.com

A Frame Relay (FR) service offers virtual private networks to customers by provisioning a set of permanent (long-term) virtual circuits (PVCs) between customer endpoints on a large backbone network. During the provisioning of a PVC, routing decisions are made either automatically by the FR switch or by the network designer, through the use of preferred routing assignments, without any knowledge of future requests. Over time, these decisions usually cause inefficiencies in the network and occasional rerouting of the PVCs is needed. The new PVC routing scheme is then implemented on the network through preferred routing assignments. Given a preferred routing assignment, the FR switch will move the PVC from its current route to the new preferred route as soon as that move becomes feasible. Section 3, deals with a GRASP for optimal routing of permanent virtual circuits for a frame relay service. We follow closely the presentation given in Resende and Ribeiro [2].

Intra-domain traffic engineering aims to make more efficient use of network resources within an autonomous system. Interior Gateway Protocols such as OSPF (Open Shortest Path First) and IS-IS (Intermediate System-Intermediate System) are commonly used to select the paths along which traffic is routed within an autonomous system. These routing protocols direct traffic based on link weights assigned by the network operator. Each router in the autonomous system computes shortest paths and creates destination tables used to direct each packet to the next router on the path to its final destination. Given a set of traffic demands between origin-destination pairs, the *OSPF weight setting problem* consists in determining weights to be assigned to the links so as to optimize a cost function, typically associated with a network congestion measure. In Section 4, we present a genetic algorithm with a local improvement procedure for the OSPF weight setting problem. The local improvement procedure makes use of an efficient dynamic shortest path algorithm to recompute shortest paths after the modification of link weights. We test the algorithm on a set of real and synthetic test problems and show that it produces near-optimal solutions. We compare the hybrid algorithm with other algorithms for this problem illustrating its efficiency and robustness. We follow closely the presentation given in Buriol, Resende, Ribeiro, and Thorup [3].

2 Pop placement for an internet service provider

In this section, we consider the PoP (point-of-presence) placement problem, an optimization problem confronted by internet access providers. The most common, and potentially least expensive, way for a customer to access the internet is with a modem by making a phone call to a PoP of the provider. It has been conjectured that potential customers are more likely to subscribe to internet access service if they can make a local (free unmetered) phone call to access at least one of the internet provider's PoPs. Given that the number of PoPs that can be deployed is limited by a number of constraints, such as budget and office capacity, one would like to place (or locate) the PoPs in a configuration that maximizes the number of customers than can make local calls to at least one PoP. We call this number of customers the *coverage*.

A formal statement of the problem is given next. Let $J = \{1, 2, \ldots, n\}$ denote the set of n potential PoP locations. Define n finite sets $P_1, P_2, \ldots, P_n$, each corresponding to a potential PoP location, such that $I = \cup_{j \in J} P_j = \{1, 2, \ldots, m\}$ is the set of the m exchanges that can be covered by the n potential PoPs. With each exchange $i \in I$, we associate a weight $w_i \geq 0$, denoting for example, the number of lines served by exchange i. A *cover* $J^* \subseteq J$ covers the exchanges in set $I^* = \cup_{j \in J^*} P_j$ and has an associated weight $w(J^*) = \sum_{i \in I^*} w_i$. Given the number $p > 0$ of PoPs to be placed, we wish to find the set $J^* \subseteq J$ that maximizes $w(J^*)$, subject to the constraint that $|J^*| = p$.

This problem, also known as the maximum covering problem (MCP) [4], has been applied to numerous location problems, including rural health centers [5], emergency vehicles [6], and commercial bank branches [7], as well as other applications [8, 9, 10]. It has an compact integer programming formulation, first described by Church and ReVelle [11]. For $i = 1, \ldots, m$ and $j = 1, \ldots, n$, let x_j and y_i be $(0, 1)$ variables such that

$$x_j = \begin{cases} 1 & \text{if } j \in J^* \\ 0 & \text{otherwise} \end{cases}$$

and

$$y_i = \begin{cases} 1 & \text{if } i \in I^* \\ 0 & \text{otherwise.} \end{cases}$$

Define

$$a_{ij} = \begin{cases} 1 & \text{if } i \in P_j \\ 0 & \text{otherwise.} \end{cases}$$

The following is an integer programming formulation for the maximum covering problem:

$$\max \sum_{i=1}^{m} w_i y_i$$

subject to:

$$\sum_{j=1}^{n} a_{ij} x_j \geq y_i, \quad i = 1, \ldots, m,$$

$$\sum_{j=1}^{n} x_j = p,$$

$$x_j = (0, 1), \quad j = 1, \ldots, n$$

$$y_i = (0, 1), \quad i = 1, \ldots, m.$$

The solution to the linear programming relaxation of the above integer program produces as its optimal objective function value, an upper bound on the maximum coverage. We shall call this bound, the LP upper bound, denoted by

$$\text{UB} = \max\{w^\top y \mid Ax \geq y,\ e^\top x = p,\ 0 \leq x \leq 1,\ 0 \leq y \leq 1\},$$

where $w = (w_1, w_2, \ldots, w_m)$, $y = (y_1, y_2, \ldots, y_m)$, $A = [a_{\cdot 1}, a_{\cdot 2}, \ldots, a_{\cdot n}]$, $x = (x_1, x_2, \ldots, x_n)$, and $e = (1, 1, \ldots, 1)$ of dimension n.

In this section, we describe a greedy randomized adaptive search procedure (GRASP) for PoP placement that finds approximate, i.e. good though not necessarily optimum, placement configurations. GRASP [12] is a metaheuristic that has been applied to a wide range of combinatorial optimization problems, including set covering [13], maximum satisfiability [14], and p-hub location [15], all three of which have some similarities with the PoP placement problem. GRASP is an iterative process, with a feasible solution constructed at each independent GRASP iteration. Each GRASP iteration consists of two phases, a construction phase and a local search phase. The best overall solution is kept as the result.

In the construction phase, a feasible solution is iteratively constructed, one element at a time. At each construction iteration, the choice of the next element to be added is determined by ordering all elements in a candidate list with respect to a greedy function. This function measures the (myopic) benefit of selecting each element. The heuristic is adaptive because the benefits associated with every element are updated at each iteration of the construction phase to reflect the changes brought on by the selection of the previous element. The probabilistic component of a GRASP is characterized by randomly choosing one of the best candidates in the list, but not necessarily the top candidate. This choice technique allows for different solutions to be obtained at each GRASP iteration, but does not necessarily compromise the power of the adaptive greedy component of the method.

As is the case for many deterministic methods, the solutions generated by a GRASP construction are not guaranteed to be locally optimal with respect to simple neighborhood definitions. Hence, it is usually beneficial to apply a local search to attempt to improve each constructed solution. While such local optimization procedures can require exponential time from an arbitrary starting point, empirically their efficiency significantly improves as the initial solutions improve. Through the use of customized data structures and careful implementation, an efficient construction phase can be created which produces good initial solutions for efficient local search. The result is that often many GRASP solutions are generated in the same amount of time required for the local optimization procedure to converge from a single random start. Furthermore, the best of these GRASP solutions is generally significantly better than the solution obtained from a random starting point.

An especially appealing characteristic of GRASP is the ease with which it can be implemented. Few parameters need to be set and tuned (candidate list size and number of GRASP

```
procedure grasp($\alpha$,MaxIter,RandomSeed)
    BestSolutionFound = $\emptyset$;
    do $k = 1, \ldots,$ MaxIter $\rightarrow$
        ConstructGreedyRandomizedSoln($\alpha$,RandomSeed,$p$,$J^*$);
        LocalSearch($J^*$);
        if $w(J^*) > w($BestSolutionFound$)$ $\rightarrow$ BestSolutionFound = $J^*$;
    od;
    return(BestSolutionFound)
end grasp;
```

Figure 1: A generic GRASP pseudo-code

```
procedure ConstructGreedyRandomizedSoln($\alpha$,RandomSeed,$p$,$J^*$)
    $J^* = \emptyset$;
    do $k = 1, \ldots, p$ $\rightarrow$
        RCL = MakeRCL($\alpha, J, J^*, \gamma$);
        $s$ = SelectPoP(RCL,RandomSeed,$J^*$);
        $J^* = J^* \cup \{s\}$;
        AdaptGreedyFunction($s, J, J^*, \Gamma, \Gamma^{-1}, \gamma$);
    od;
end ConstructGreedyRandomizedSoln;
```

Figure 2: GRASP construction phase pseudo-code

iterations) and therefore development can focus on implementing efficient data structures to assure quick GRASP iterations. Finally, GRASP can be trivially implemented on a parallel processor in an MIMD environment. For example, each processor can be initialized with its own copy of the procedure, the instance data, and an independent random number sequence. The GRASP iterations are then performed in parallel with only a single global variable required to store the best solution found over all processors.

The TM is organized as follows. In Subection 2.1, we describe the GRASP. In Subsection 2.2, we show how the GRASP solution is better than the pure random or pure greedy alternatives. On a large instance arising from a real-world application, we show how the GRASP solution is near optimal. Parallelization of GRASP is also illustrated.

```
**procedure** MakeRCL($\alpha, J, J^*, \gamma$)
    RCL $= \emptyset$;
    $\gamma^* = \max\{\gamma_j \mid j \in J \setminus J^*\}$;
    **do** $s \in J \setminus J^* \rightarrow$
        **if** $\gamma_s \geq \alpha \times \gamma^* \rightarrow$
            RCL $=$ RCL $\cup \{s\}$;
        **fi**;
    **od**;
    **return**(RCL);
**end** MakeRCL;
```

Figure 3: MakeRCL pseudo-code

```
**procedure** AdaptGreedyFunction($s, J, J^*, \Gamma, \Gamma^{-1}, \gamma$)
    **do** $i \in \Gamma_s \rightarrow$
        **do** $j \in \Gamma_i^{-1} \cap \{J \setminus J^*\}$ $(j \neq i) \rightarrow$
            $\Gamma_j = \Gamma_j - \{i\}$;
            $\gamma_j = \gamma_j - w_i$;
        **od**;
    **od**;
**end** AdaptGreedyFunction;
```

Figure 4: AdaptGreedyFunction pseudo-code

2.1 GRASP for PoP placement

As outlined in Section 3.2, a GRASP possesses four basic components: a greedy function, an adaptive search strategy, a probabilistic selection procedure, and a local search technique. These components are interlinked, forming an iterative method that, at each iteration, constructs a feasible solution, one element at a time, guided by an adaptive greedy function, and then searches the neighborhood of the constructed solution for a locally optimal solution. Figure 1 shows a GRASP in pseudo-code. The best solution found so far (`BestSolution-Found`) is initialized in line 1. The GRASP iterations are carried out in lines 2 through 6. Each GRASP iteration has a construction phase (line 3) and a local search phase (line 4). If necessary, the solution is updated in line 5. The GRASP returns the best solution found.

In the remainder of this subsection, we describe in detail the ingredients of the GRASP for the PoP placement problem, i.e. the GRASP construction and local search phases. To describe the construction phase, one needs to provide a candidate definition (for the restricted candidate list) and an adaptive greedy function, and specify the candidate restriction mechanism. For the local search phase, one must define the neighborhood and specify a local search algorithm.

2.1.1 Construction phase

The construction phase of a GRASP builds a solution, around whose neighborhood a local search is carried out in the local phase, producing a locally optimal solution. This construction phase solution is built, one element at a time, guided by a greedy function and randomization. Figure 2 describes in pseudo-code a GRASP construction phase. Since in the PoP placement problem there are p PoP locations to be chosen, each construction phase consists of p iterations, with one location chosen per iteration. In `MakeRCL` the restricted candidate list of PoP locations is set up. The index of the next PoP location to be chosen is determined in `SelectPoP`. The PoP location selected is added to the set J^* of chosen PoP locations in line 5 of the pseudo-code. In `AdaptGreedyFunction` the greedy function that guides the construction phase is changed to reflect the choice just made. As before, let $J = \{1, 2, \ldots, n\}$ be set of indices of the sets of potential PoP locations. Solutions are constructed by selecting one PoP location at a time to be in the set J^* of chosen PoP locations. To define a restricted candidate list, we must rank the yet unchosen PoP locations according to an adaptive greedy function.

The greedy function used in this algorithm is the total weight of yet-uncovered exchanges that become covered after the selection in each construction phase iteration. Let J^* denote the set (initially empty) of chosen PoP locations being built in the construction phase. At any construction phase iteration, let Γ_j be the set of additional uncovered exchanges that would become covered if PoP location j (for $j \in J \setminus J^*$) were to be added to J^*. Define the *greedy function*

$$\gamma_j = \sum_{i \in \Gamma_j} w_i$$

to be the incremental weight covered by the choice of PoP location $j \in J \setminus J^*$. The greedy choice is to select the PoP location k having the largest γ_k value. Note that with every selection made, the sets Γ_j, for all yet unchosen PoP location indices $j \in J \setminus J^*$, change to reflect the new selection. This consequently changes the values of the greedy function γ_j, characterizing the adaptive component of the heuristic.

We describe next the restriction mechanism for the restricted candidate list (RCL) used in this GRASP. The RCL is set up in `MakeRCL` of the pseudo-code of Figure 3. A value restriction mechanism is used. Value restriction imposes a parameter based *achievement level*, that a candidate has to satisfy to be included in the RCL. Let

$$\gamma^* = \max\{\gamma_j \mid \text{PoP location } j \text{ is yet unselected, i.e. } j \in J \setminus J^*\}$$

and α be the restricted candidate parameter ($0 \leq \alpha \leq 1$). We say a PoP location j is a *potential candidate*, and is added to the RCL, if $\gamma_j \geq \alpha \times \gamma^*$. `MakeRCL` returns the set `RCL` with the indices of all potential PoP locations that have greedy function values within $\alpha \times 100\%$ of the value of the greedy choice. Note that by varying the parameter α the heuristic can be made to construct a set of p random PoP locations ($\alpha = 0$) or act as a greedy algorithm ($\alpha = 1$).

Once the RCL is set up, a candidate from the list must be selected and made part of the solution being constructed. `SelectPoP` selects, at random, the PoP location index s from the RCL. In line 5 of `ConstructGreedyRandomizedSoln`, the choice made in `SelectPoP` is added to the set of PoP locations J^*.

The greedy function γ_j is changed in `AdaptGreedyFunction` to reflect the choice made in `SelectPoP`. This requires that some of the sets Γ_j as well as the values γ_j be updated. Let Γ_i^{-1} denote the set of PoP locations to which a caller in exchange i can make a local call to. Let s be the newly added PoP location. The potential PoP locations j whose elements Γ_j need to be updated are those not yet in the PoP location set J^* for which exchanges in P_s are covered by PoP location j.

2.1.2 Local search phase

Given a solution neighborhood structure $N(\cdot)$ and a weight function $w(\cdot)$, a local search algorithm takes an initial solution J^0 and seeks a locally optimal solution with respect to $N(\cdot)$. For a maximization problem, such as the PoP placement problem, a local optimum is a solution J^* having weight $w(J^*)$ greater than or equal to the weight $w(J^+)$ for any $J^+ \in N(J^*)$. The local search algorithm examines a sequence of solutions $J^0, J^1, \ldots, J^k = J^*$, where $J^{i+1} \in N(J^i)$, i.e. immediately after examining solution J^i, it can only examine a

```
procedure LocalSearch($J^0, N(\cdot), w(\cdot), J^*$)
1    $J^* = J^0$;
2    do  $\exists\ J^+ \in N(J^*) \ni w(J^+) > w(J^*)$  →
3        $J^* = J^+$;
4    od;
end LocalSearch;
```

Figure 5: A generic local search algorithm

```
procedure LocalSearch($J^*$)
1    do local maximum not found →
2        do $s \in J^*$  →
3            do $t \in J \setminus J^*$  →
4                if WeightGain($J^*, t$) > WeightLoss($J^*, s$)  →
5                    $J^* = J^* \cup \{t\} \setminus \{s\}$;
6                fi;
7            od;
8        od;
9    od;
end LocalSearch;
```

Figure 6: The local search procedure in pseudo-code

solution J^{i+1} that is a neighbor of J^i. Figure 5 illustrates a generic local search algorithm that finds a local maximum of the function $w(\cdot)$. If in line 2 there exists a solution J^+ in the neighborhood of the current solution J^* with a weight greater than that of the current solution, then in line 3 the improved solution is made the current solution. The loop from line 2 to 4 is repeated until no local improvement is possible.

A combinatorial optimization problem can have many different neighborhood structures. For the PoP placement problem, a simple structure is 2-exchange. Two solutions (sets of PoP locations) J^1 and J^2 are said to be neighbors in the 2-exchange neighborhood if they differ by exactly one element, i.e. $\mid J^1 \cap \Delta J \mid \; = \; \mid J^2 \cap \Delta J \mid \; = 1$, where $\Delta J = (J^1 \cup J^2) \setminus (J^1 \cap J^2)$. The local search starts with a set J^* of p PoP locations, and at each iteration attempts to find a pair of locations $s \in J^*$ and $t \in J \setminus J^*$ such that $w(J^* \setminus \{s\} \cup \{t\}) > w(J^*)$. If such a pair exists, then location s is replaced by location t in J^*. A solution is locally optimal with respect to this neighborhood if there exists no pairwise exchange that increases the total weight of J^*. This local search algorithm is described in the pseudo-code in Figure 6. Though it is not the objective of this TM to delve into implementation details, it is interesting to observe that the total weight of the neighborhood solutions need not be computed from scratch, Rather, in line 4 of the pseudo-code, procedures `WeightGain` and `WeightLoss` compute, respectively, the weight gained by J^* with the inclusion of PoP location j and the weight loss by J^* with the removal of PoP location i from J^*. The weight gained can be computed by adding the weights of all exchanges not covered by any PoP location in J^* that is covered by j, while the weight loss can be computed by adding up the weights of the exchanges covered by PoP location i and no other PoP location in J^*.

The GRASP construction phase described in Subsection 2.1.1 computes a feasible set of chosen PoP locations that is not necessarily locally optimal with respect the 2-exchange neighborhood structure. Consequently, local search can be applied with the objective of finding a locally optimal solution that may be better than the constructed solution. In fact, the main purpose of the construction phase is to produce a good initial solution for the local search. It is empirically known that simple local search techniques perform better if they start with a good initial solution. This will be illustrated in the computational results subsection, where experiments indicate that local search applied to a solution generated by the construction phase, rather than random generation, produces better overall solutions, and GRASP converges faster to an approximate solution.

2.2 Computing PoP placements with GRASP

In this subsection, we illustrate the use of GRASP on a large PoP placement problem. We consider a problem with $m = 18,419$ calling areas and $n = 27,521$ potential PoP location. The sum of the number of lines over the calling areas is 27,197,601. We compare an implementation of the GRASP described in Subsection 2.1 with implementations of an algorithm having a purely greedy construction phase and one having purely random construction. All three algorithms use the same local search procedure, described in Subsection 2.1.2. Furthermore, since pure greedy and pure random are special cases of GRASP construction, all three algorithms are implemented using the same code, simply by setting the RCL parameter value α to appropriate values. For GRASP, $\alpha = 0.85$, while for the purely greedy algorithm, $\alpha = 1$, and for the purely random algorithm, $\alpha = 0$. All runs were carried out on a Silicon Graphics Challenge computer (196MHz IPS R10000 processor). The GRASP code is written in Fortran and was compiled with the SGI Fortran compiler `f77` using compiler flags `-O3 -r4 -64`.

In this experiment the number of PoPs to be place is fixed at $p = 146$ and the three implementations are compared. Each code is run on 10 processors, each using a different random number generator seed for 500 iterations of the build–local search cycle, thus each totaling 5000 iterations. Because of the long processing times associated with the random algorithm, the random algorithm processes were interrupted before completing the full 500 iterations on each processor. They did 422, 419, 418, 420, 415, 420, 420, 412, 411, and 410 iterations on each corresponding processor, totaling 4167 iterations.

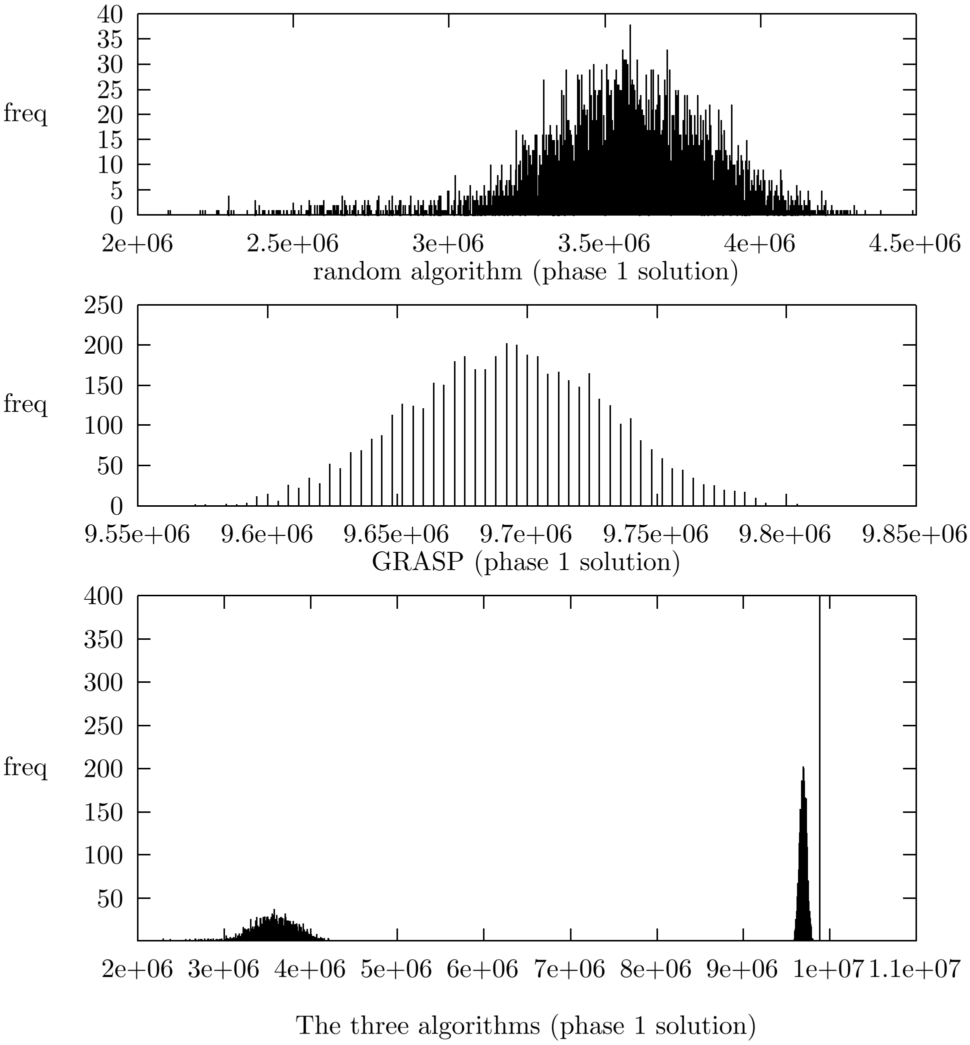


Figure 7: Phase 1 solution distribution for random algorithm (RCL parameter $\alpha = 0$), GRASP ($\alpha = 0.85$), and greedy algorithm ($\alpha = 1$)

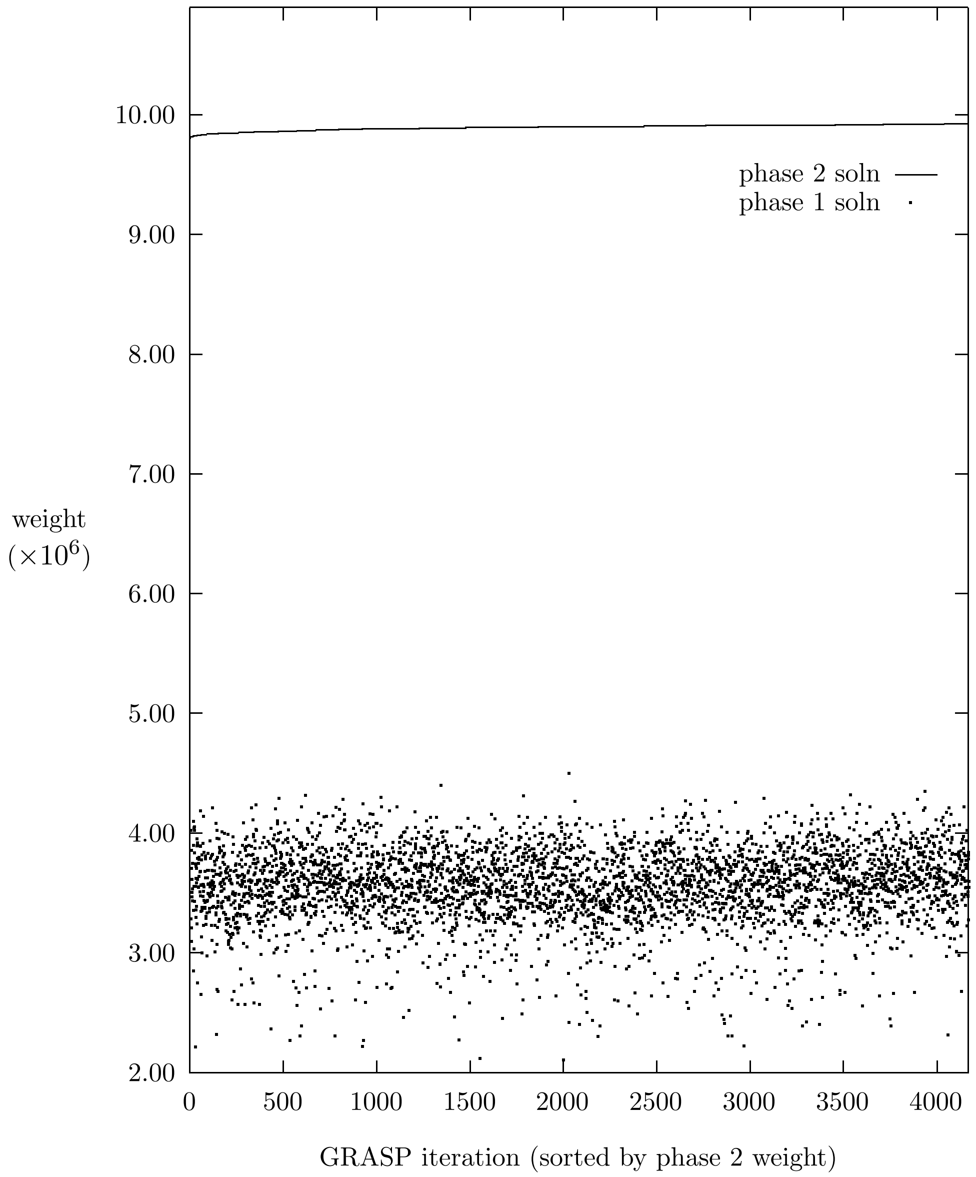


Figure 8: GRASP phase 1 and phase 2 solutions, sorted by phase 2, then phase 1 solutions. RCL parameter $\alpha = 0.0$ (purely random construction)

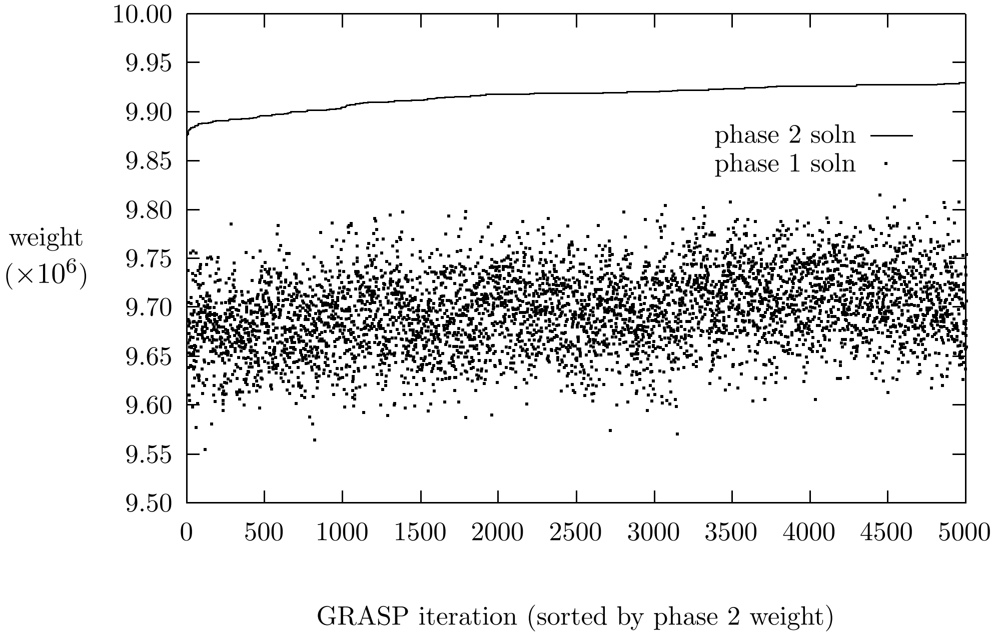


Figure 9: GRASP phase 1 and phase 2 solutions, sorted by phase 2, then phase 1 solutions. RCL parameter $\alpha = 0.85$

Figure 7 illustrates the relative behavior of the three algorithms. The top and middle plots in Figure 7 show the frequency of the solution values generated by the purely random construction and GRASP construction respectively. The plot on the bottom of Figure 7 compares the constructed solutions of the three algorithms. As can be observed, the purely greedy algorithm constructs the best quality solution, followed by the GRASP, and then by the purely random algorithm. On the other hand, the purely random algorithm produces the largest amount of variance in the constructed solutions, followed by the GRASP and then the purely greedy algorithm, which generated the same solution on all 5000 repetitions. High quality solutions as well as large variances are desirable characteristics of constructed solutions. Of the three algorithms, GRASP captures these two characteristics in its phase 1 solutions. As we will see next, the tradeoff between solution quality and variance plays an important role in designing a GRASP.

The solutions generated by the purely random algorithm and the GRASP are shown in Figures 8 and 9, respectively. The solution values on these plots are sorted according to local search phase solution value. As one can see, the differences between the values of the construction phase solutions and the local search phase solutions are much smaller for the GRASP than for the purely random algorithm. This suggests that the purely random algorithm requires greater effort in the local search phase than does GRASP. This indeed is observed and will be shown next. Figures 10 and 11 illustrate how the three algorithms

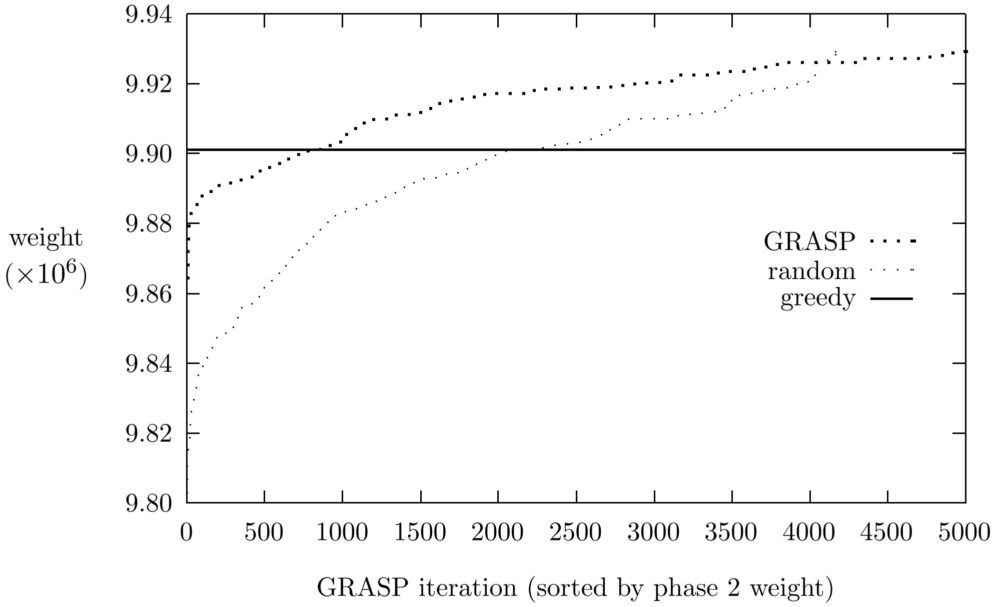


Figure 10: Phase 2 solutions, sorted by phase 2 for random, GRASP, and greedy algorithms

compare in terms of best solution found so far, as a function of algorithms iteration and running time. Figure 10 shows local search phase solution for each algorithm, sorted by increasing value for each algorithm. The solution produced by applying local search to the solution constructed with the purely greedy algorithm is constant. Its value is only better than the worst 849 GRASP solutions and the worst 2086 purely random solutions. This figure illustrates well the effect of the tradeoff between greediness and randomness in terms of solution quality as a function of the number of iterations that the algorithm is repeated.

3 GRASP with path-relinking for PVC routing

A frame relay service offers virtual private networks to customers by provisioning a set of permanent (long-term) private virtual circuits (PVCs) between endpoints on a large backbone network. During the provisioning of a PVC, routing decisions are made either automatically by the frame relay switch or by the network designer, through the use of preferred routing assignments and without any knowledge of future requests. Over time, these decisions usually cause inefficiencies in the network and occasional rerouting of the PVCs is needed. The new routing scheme is then implemented on the network through preferred routing assignments. Given a preferred routing assignment, the switch will move the PVC from its current route to the new preferred route as soon as this move becomes feasible.

One possible way to create the preferred routing assignments is to appropriately order the set of PVCs currently in the network and apply an algorithm that mimics the routing

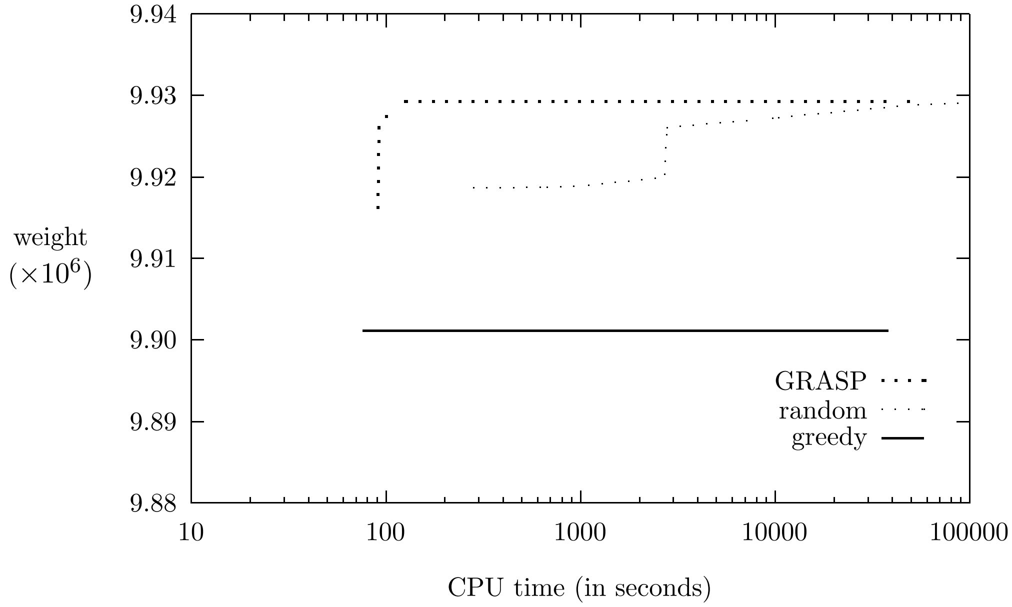


Figure 11: Incumbent phase 2 solution of random algorithm ($\alpha = 0$), GRASP ($\alpha = 0.85$), and greedy algorithm ($\alpha = 1$) as a function of CPU time (in seconds), running 10 processes in parallel.

algorithm used by the frame relay switch to each PVC in that order. However, more elaborate routing algorithms, that take into account factors not considered by the switch, could further improve the efficiency of network resource utilization.

Typically, the routing scheme used by the frame relay switch to automatically provision PVCs is also used to reroute them in the case of trunk or card failures. Therefore, this routing algorithm should be efficient in terms of running time, a requirement that can be traded off for improved network resource utilization when building preferred routing assignments offline.

In this section, we propose variants of a GRASP (greedy randomized adaptive search procedure) with path-relinking algorithm for the problem of routing offline a set of PVC demands over a backbone network, such that a combination of the delays due to propagation and congestion is minimized. This problem and its variants are also known in the literature as bandwidth packing problems. The set of PVCs to be routed can include all or a subset of the PVCs currently in the network, and/or a set of forecast PVCs. The explicit handling of propagation delays, as opposed to just handling the number of hops (as in the routing algorithm implemented in Cisco switches) is particularly important in international networks, where distances between backbone nodes vary considerably. The minimization of network congestion is important for providing the maximum flexibility to handle the following situations:

- overbooking, which is typically used by network designers to account for non-coincidence of traffic;

- PVC rerouting, due to link or card failures; and
- bursting above the committed rate, which is not only allowed but sold to customers as one of the attractive features of frame relay.

In Subsection 3.1, we formulate the offline PVC routing problem as an integer multicommodity flow problem with additional constraints and a hybrid objective function, which takes into account delays due to propagation as well as delays due to network congestion. Minimum cost multicommodity network flow problems are characterized by a set of commodities flowing through an underlying network, each commodity having an associated integral demand which must flow from its source to its destination. The flows are simultaneous and the commodities share network resources. If the cost function in each edge is convex, then this problem can be solved in polynomial time [16]. The problem is NP-hard if the flows are required to be integral [17] or if each commodity is required to follow a single path from its source to its destination [18]. In Subsection 3.2, we propose variants of a GRASP with path-relinking heuristic for this problem. Experimental results, reported in Subsection 3.3, show that the proposed heuristics are able to improve the solutions found with standard routing techniques on realistic-size problems. Concluding remarks are made in Subsection 3.4.

Though we motivate the algorithm with a frame relay routing application, we note that the algorithm can be applied to routing problems that arise in other connection-switched protocols, such as in asynchronous transfer mode (ATM).

3.1 Problem formulation

Let $G = (V, E)$ be an undirected graph representing the frame relay network. We denote by $V = \{1, \ldots, n\}$ the set of backbone nodes where switches reside, while E is set of trunks (or edges) that connect the backbone nodes, with $|E| = m$. Parallel trunks are allowed. Since G is an undirected graph, flows through each trunk $(i, j) \in E$ have two components to be summed up, one in each direction. However, for modeling purposes, costs and capacities will always be associated only with the ordered pair (i, j) satisfying $i < j$. For each trunk $(i, j) \in E$, we denote by b_{ij} its maximum allowed bandwidth (in kbits/second), while c_{ij} denotes the maximum number of PVCs that can be routed through it and d_{ij} is the propagation, or hopping, delay associated with the trunk. Each commodity $k \in K = \{1, \ldots, p\}$ is a PVC to be routed, associated with an origin-destination pair and with a bandwidth requirement (or demand, also known as its effective bandwidth) r_k. It takes into account the actual bandwidth required by the customer in the forward and reverse directions, as well as an overbooking factor.

The ultimate objective of the offline PVC routing problem is to minimize propagation delays and/or network congestion, subject to several technological constraints. Queueing delays are often associated with network congestion and in some networks account for a large part of the total delay. In other networks, distances may be long and loads low, causing propagation delay to account for a large part of the total delay. For a discussion of delay in data networks, see [19]. Two common measures of network congestion are the load on the most utilized trunk, and the average delay in a network of independent $M/M/1$ queues, as in [20]. Another measure, which we use in this section, is a cost function that penalizes heavily loaded trunks. This function resembles the average delay function, except that it allows loads to exceed trunk capacities. Routing assignments with minimum propagation delays may not achieve the least network congestion. Likewise, routing assignments having the least congestion may not minimize propagation delays. A compromising objective is to route the PVCs such that a desired point in the tradeoff curve between propagation delays and network congestion is achieved.

The upper bound on the number of PVCs allowed on a trunk depends on the port card used to implement it. A set of routing assignments is feasible if and only if for every trunk $(i, j) \in E$ the total PVC effective bandwidth requirements routed through it does not exceed its maximum bandwidth b_{ij} and the number of PVCs routed through it is not greater than c_{ij}.

Let x^k_{ij} be a 0-1 variable such that $x^k_{ij} = 1$ if and only if trunk $(i, j) \in E$ is used to route commodity $k \in K$ from node i to node j. The following linear integer program models the problem:

$$\min \phi(x) = \sum_{(i,j)\in E, i<j} \phi_{ij}(x^1_{ij}, \cdots, x^p_{ij}, x^1_{ji}, \cdots, x^p_{ji}) \tag{1}$$

subject to

$$\sum_{k\in K} r_k(x^k_{ij} + x^k_{ji}) \leq b_{ij}, \ \ \forall (i, j) \in E, i < j, \tag{2}$$

$$\sum_{k\in K} (x^k_{ij} + x^k_{ji}) \leq c_{ij}, \ \ \forall (i, j) \in E, i < j, \tag{3}$$

$$\sum_{(i,j)\in E} x^k_{ij} - \sum_{(i,j)\in E} x^k_{ji} = a^k_i, \ \ \forall i \in V, \forall k \in K, \tag{4}$$

$$x^k_{ij} \in \{0, 1\}, \ \ \forall (i, j) \in E, \forall k \in K. \tag{5}$$

Constraints of type (2) limit the total flow on each trunk to at most its capacity. Constraints of type (3) enforce the limit on the number of PVCs routed through each trunk. Constraints of type (4) are flow conservation equations, which together with (5), state that the flow associated with each PVC cannot be split, where $a^k_i = 1$ if node i is the source for commodity k, $a^k_i = -1$ if node i is the destination for commodity k, and $a^k_i = 0$ otherwise.

The cost function $\phi_{ij}(x^1_{ij}, \cdots, x^p_{ij}, x^1_{ji}, \cdots, x^p_{ji})$ associated with each trunk $(i, j) \in E$ with $i < j$ is the linear combination of a trunk propagation delay component and a trunk congestion component. The propagation delay component is defined as

$$\phi^d_{ij}(x^1_{ij}, \cdots, x^p_{ij}, x^1_{ji}, \cdots, x^p_{ji}) = d_{ij} \cdot \sum_{k \in K} \rho_k (x^k_{ij} + x^k_{ji}), \tag{6}$$

where coefficients ρ_k are used to model two plausible delay functions:

- If $\rho_k = 1$, then this component leads to the minimization of the number of hops weighted by the propagation delay on each trunk.
- If $\rho_k = r_k$, then the minimization takes into account the effective bandwidth routed through each trunk weighted by its propagation delay.

Let $y_{ij} = \sum_{k \in K} r_k(x^k_{ij} + x^k_{ji})$ be the total flow through trunk $(i, j) \in E$ with $i < j$. The trunk congestion component depends on the utilization rates $u_{ij} = y_{ij}/b_{ij}$ of each trunk $(i, j) \in E$ with $i < j$. It is taken as the piecewise linear function proposed by Fortz and Thorup [21] and depicted in Figure 12, which increasingly penalizes flows approaching or violating the capacity limits:

$$\phi^b_{ij}(x^1_{ij}, \cdots, x^p_{ij}, x^1_{ji}, \cdots, x^p_{ji}) = b_{ij} \cdot \begin{cases} u_{ij}, & u_{ij} \in [0, 1/3) \\ 3 \cdot u_{ij} - 2/3, & u_{ij} \in [1/3, 2/3), \\ 10 \cdot u_{ij} - 16/3, & u_{ij} \in [2/3, 9/10), \\ 70 \cdot u_{ij} - 178/3, & u_{ij} \in [9/10, 1), \\ 500 \cdot u_{ij} - 1468/3, & u_{ij} \in [1, 11/10), \\ 5000 \cdot u_{ij} - 16318/3, & u_{ij} \in [11/10, \infty). \end{cases} \tag{7}$$

The value $\Omega = \max_{(i,j) \in E, i<j}\{u_{ij}\}$ gives a global measure of the maximum congestion in the network.

In this section, we use the cost function

$$\begin{aligned} &\phi_{ij}(x^1_{ij}, \cdots, x^p_{ij}, x^1_{ji}, \cdots, x^p_{ji}) = \\ &\quad = (1 - \delta) \cdot \phi^d_{ij}(x^1_{ij}, \cdots, x^p_{ij}, x^1_{ji}, \cdots, x^p_{ji}) + \delta \cdot \phi^b_{ij}(x^1_{ij}, \cdots, x^p_{ij}, x^1_{ji}, \cdots, x^p_{ji}) \end{aligned} \tag{8}$$

associated with each trunk $(i, j) \in E$ with $i < j$, where weights $(1 - \delta)$ and δ correspond respectively to the propagation delay and the network congestion components, with $\delta \in [0, 1]$. Note that if $\delta > 0$, then the network congestion component is present in the objective function, which allows us to relax capacity constraints (2). This is assumed in the algorithms we propose in Subsection 3.2. We show in Subsection 3.3 that small values, such as $\delta = 0.1$, lead to feasible solutions minimizing the overall propagation delays (measured in terms of either hops or propagation) and with balanced loads on the trunks, characterized by reduced values of the maximum congestion index Ω.

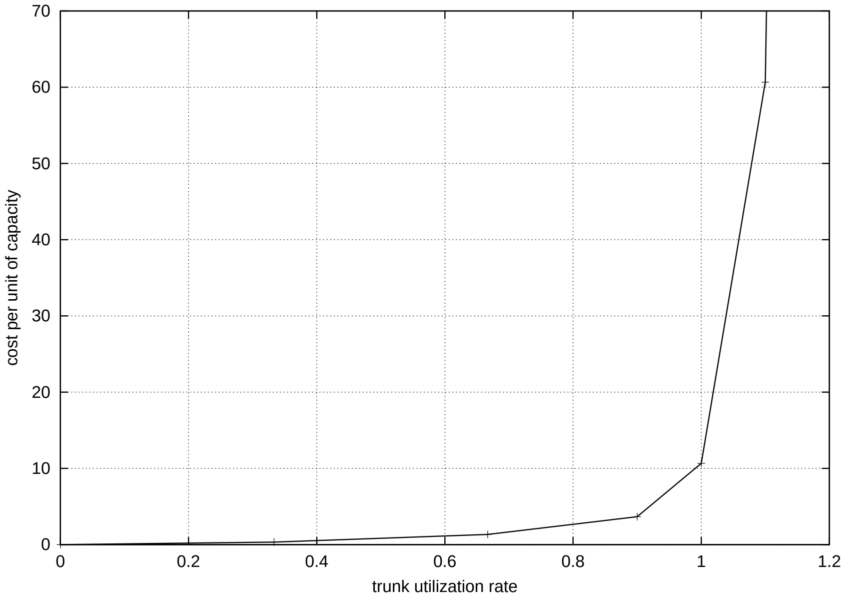


Figure 12: Piecewise linear load balance cost component associated with each trunk.

Several heuristics have been proposed for different variants of the bandwidth packing problem. One of the first algorithms for routing virtual circuits in communication networks was proposed by Yee and Lin [22]. Their problem is formulated as a nonlinear multicommodity flow problem with integer decision variables. Their heuristic applies Lagrangean relaxation and a multiplier adjustment procedure to solve a sequence of restricted problems. Computational results are illustrated for problems in three different networks. Their largest problem had 61 nodes and 148 links. Sung and Park [23] have also developed a Lagrangean heuristic for a similar variant of this problem. They limited its application to six small networks, the largest of which had 20 nodes and 52 links. Laguna and Glover [24] considered a bandwidth packing problem in which they want to assign calls to paths in a capacitated graph, such that capacities are not violated and some measure of the total profit is maximized. They develop a tabu search algorithm which makes use of an efficient implementation of the k-shortest path algorithm. Computational results for small problems involving up to 31 nodes and 50 calls are reported. Amiri et al. [25] proposed another formulation for the bandwidth packing problem. They consider both revenue losses and costs associated with communication delays as part of the objective. A heuristic procedure based on Lagrangean relaxation is applied for finding bounds and solutions. Computational results are reported for problems with up to 50 nodes, with the number of calls ranging from 50 to 90% of the maximum number of all possible calls. Resende and Resende [26] proposed a GRASP for offline PVC rerouting in which a different objective function is considered. Their construction and local search procedures are different than those proposed in this section. In particular, the construction procedure often encountered difficulties in finding feasible solutions for tightly constrained instances. Also, their local search procedure is much more time-consuming, limiting its application to small instances. Shyur and Wen [27] proposed a tabu search algorithm for optimizing the system of virtual paths. The objective function consists in minimizing the maximum link load, by requiring that each route visits the minimum number of hubs. The load of a link is defined as the sum of the virtual path capacities, summed over the virtual paths that traverse the link. Computation results for problems with up to 64 nodes, 112 links, and 2048 demand pairs are given.

A number of exact approaches for solving variants of the bandwidth packing problem have also appeared in the literature. Parker and Ryan [28] described a branch and bound procedure for optimally solving a bandwidth packing problem. Their objective is to allocate bandwidth so as to maximize the total revenue. The linear relaxation of the associated integer programming problem is solved using column generation. Computational results for 14 different networks with up to 29 nodes, 61 links, and 93 calls are presented. LeBlanc et al. [29] addressed packet switched telecommunication networks, considering restrictions on paths and flows: hop limits, node and link capacity constraints, and high- and low-priority flows. They minimize the expected queueing time and do not impose integrality constraints on the flows. Dahl et al. [30] studied a network configuration problem in telecommunications, searching for paths in a capacitated network to accommodate a given traffic demand matrix. Their model also involves an intermediate pipe layer. The problem is formulated as an integer linear program, where the 0-1 variables represent different paths. An associated integral polytope is studied and different classes of facets are described. These are embedded in a cutting plane algorithm. Computational results for realistic-size problems, with up to 62 nodes, 81 links, and 33 origin-destination pairs are presented. Barnhart et al. [31] proposed a branch-and-cut-and-price algorithm for origin-destination integer multicommodity flow problems. This problem is a constrained version of the linear multicommodity network flow problem, in which each flow may use only path from its origin to its destination. Because this model contains one variable for each origin-destination path, for every commodity, the linear programming relaxations are solved using column generation. New branching rules allow columns to be efficiently generated at each node of the branch and bound tree. New cuts can also be generated at each node of the branch and bound tree, helping to strengthen the linear programming relaxation. Implementation details, together with computational results for problems with at most 50 nodes, 130 edges, and 585 commodities, are reported.

The model (1)–(5) proposed in this section has two distinctive features with respect to other formulations. First, it takes into account a two component objective function which is able to handle both delays and load balance. Second, it enforces constraints that limit the maximum number of PVCs that can be routed through any trunk. An approximate algorithm for its solution is described in the next section.

3.2 Approximate algorithm for PVC routing

A GRASP is a multistart or iterative process, in which each GRASP iteration consists of two phases, a construction phase, in which a feasible solution is produced, and a local search phase, in which a local optimum in the neighborhood of the constructed solution is sought [32]. The best overall solution is kept as the result. The pseudo-code in Figure 13 illustrates a general GRASP procedure for the minimization of an objective function $f(x)$ under constraints $x \in X$, in which `Max_Iterations` GRASP iterations are done.

```
procedure GRASP
    $f^* \leftarrow \infty$;
    for $k = 1, \ldots,$ Max_Iterations do
        Construct a greedy randomized solution $x \in X$;
        Find $y$ by applying local search to $x$;
        if $f(y) < f^*$ do
            $x^* \leftarrow y$;
            $f^* \leftarrow f(x^*)$;
        end if;
    end for;
    return $x^*$;
end GRASP;
```

Figure 13: Pseudo-code of a general GRASP procedure.

A feasible solution is iteratively constructed in the first phase, one element at a time. At each construction iteration, the choice of the next element to be added is determined by ordering all candidate elements (i.e. those that can be added to the solution) in a candidate list with respect to its contribution to the objective function. The list of best candidates is called the *restricted candidate list* (RCL). The random selection of an element from the RCL allows for different solutions to be obtained at each GRASP iteration.

Another construction mechanism, called *heuristic-biased stochastic sampling*, was introduced by Bresina [33]. In the construction procedure of the basic GRASP, the next element to be introduced in the solution is chosen at random from the candidates in the RCL. The elements of the RCL are assigned equal probabilities of being chosen. However, any probability distribution can be used to bias the selection toward some particular candidates. Bresina [33] introduced a family of such probability distributions. In Bresina's selection procedure, the candidates are ranked according to the greedy function. Binato et al. [34] use Bresina's selection procedure, but restricted to elements of the RCL.

Since solutions generated by a GRASP construction are not guaranteed to be locally optimal, it is almost always beneficial to apply a local search to attempt to improve each constructed solution.

In the remainder of this section, we customize a GRASP for the offline PVC routing problem. We describe construction and local search procedures, as well as a path-relinking intensification strategy.

3.2.1 Construction phase

In the construction phase, the routes are determined, one at a time. A new PVC is selected to be routed in each iteration. To reduce the computation times, we used a combination of the strategies usually employed by GRASP and heuristic-biased stochastic sampling. We create a restricted candidate list with a fixed number of elements n_c. At each iteration, it is formed by the n_c unrouted PVCs pairs with the largest demands. An element ℓ is selected at random from this list with probability $\pi(\ell) = r_\ell / \sum_{k \in \mathrm{RCL}} r_k$.

Once a PVC $\ell \in K$ is selected, it is routed on a shortest path from its origin to its destination. The capacity constraints (2) are relaxed and handled via the penalty function introduced by the load balance component (7) of the edge weights. The constraints of type (3) are explicitly taken into account by forbidding routing through trunks already using its maximum number of PVCs. The weight $\Delta\phi_{ij}$ of each edge $(i,j) \in E$ is given by the increment of the cost function value $\phi_{ij}(x^1_{ij}, \cdots, x^p_{ij}, x^1_{ji}, \cdots, x^p_{ji})$, associated with routing r_ℓ additional units of demand through edge (i,j).

More precisely, let $\underline{K} \subseteq K$ be the set of previously routed PVCs and $\underline{K}_{ij} \subseteq \underline{K}$ be the subset of PVCs that are routed through trunk $(i,j) \in E$. Likewise, let $\overline{K} = \underline{K} \cup \{\ell\} \subseteq K$ be the new set of routed PVCs and $\overline{K}_{ij} = \underline{K}_{ij} \cup \{\ell\} \subseteq \overline{K}$ be the new subset of PVCs that are routed through trunk (i,j). Then, we define $\underline{x}^\ell_{ij} = 1$ if PVC $\ell \in \underline{K}$ is routed through trunk $(i,j) \in E$ from i to j, $\underline{x}^\ell_{ij} = 0$ otherwise. Similarly, we define $\overline{x}^\ell_{ij} = 1$ if PVC $\ell \in \overline{K}$ is routed through trunk $(i,j) \in E$ from i to j, $\overline{x}^\ell_{ij} = 0$ otherwise. According with (8), the cost associated with each edge $(i,j) \in E$ in the current solution is given by $\phi_{ij}(\underline{x}^1_{ij}, \cdots, \underline{x}^p_{ij}, \underline{x}^1_{ji}, \cdots, \underline{x}^p_{ji})$. In the same manner, the cost associated with each edge $(i,j) \in E$ after routing PVC ℓ will be $\phi_{ij}(\overline{x}^1_{ij}, \cdots, \overline{x}^p_{ij}, \overline{x}^1_{ji}, \cdots, \overline{x}^p_{ji})$. Then, the incremental edge weight $\Delta\phi_{ij}$ associated with routing PVC $\ell \in K$ through edge $(i,j) \in E$, used in the shortest path computations, is given by

$$\Delta\phi_{ij} = \phi_{ij}(\overline{x}^1_{ij}, \cdots, \overline{x}^p_{ij}, \overline{x}^1_{ji}, \cdots, \overline{x}^p_{ji}) - \phi_{ij}(\underline{x}^1_{ij}, \cdots, \underline{x}^p_{ij}, \underline{x}^1_{ji}, \cdots, \underline{x}^p_{ji}). \tag{9}$$

The enforcement of type (3) constraints may lead to unroutable demand pairs. In this case, the current solution is discarded and a new construction phase starts.

3.2.2 Local search

Each solution built in the first phase may be viewed as a set of routes, one for each PVC. Our local search procedure seeks to improve each route in the current solution. For each PVC

$k \in K$, we start by removing r_k units of flow from each edge in its current route. Next, we compute incremental edge weights $\Delta\phi_{ij}$ associated with routing this demand through each trunk $(i, j) \in E$ according to (9), as described in Subsection 3.2.1. A tentative new shortest path route is computed using the incremental edge weights. If the new route improves the solution, it replaces the current route of PVC k. This is continued until no improving route can be found.

3.2.3 Path-relinking

Path-relinking was originally proposed by Glover [35] as an intensification strategy exploring trajectories connecting elite solutions obtained by tabu search or scatter search [36, 37, 38]. Starting from one or more elite solutions, paths in the solution space leading toward other elite solutions are generated and explored in the search for better solutions. This is accomplished by selecting moves that introduce attributes contained in the guiding solutions. Path-relinking may be viewed as a strategy that seeks to incorporate attributes of high quality solutions, by favoring these attributes in the selected moves.

The use of path-relinking within a GRASP procedure as an intensification strategy applied to each locally optimal solution was first proposed by Laguna and Martí [39], being followed by several extensions, improvements, and successful applications [40, 41, 42].

In this context, path-relinking is applied to pairs $\{x_1, x_2\}$ of solutions, where x_1 is the locally optimal solution obtained after local search and x_2 is one of a few elite solutions randomly chosen from a pool with a limited number `Max_Elite` of elite solutions found along the search. The pool is originally empty. Each locally optimal solution obtained by local search is considered as a candidate to be inserted into the pool if it is different (by at least one trunk in one route, in the case of the bandwidth packing problem) from every other solution currently in the pool. If the pool already has `Max_Elite` solutions and the candidate is better than the worst of them, then the former replaces the latter. If the pool is not full, the candidate is simply inserted.

The algorithm starts by computing the symmetric difference $\Delta(x_1, x_2)$ between x_1 and x_2, resulting in a set of moves which should be applied to one of them (the initial solution) to reach the other (the guiding solution). Starting from the initial solution, the best move still not performed is applied to the current solution, until the guiding one is attained. The best solution found along this trajectory is also considered as a candidate for insertion in the pool and the incumbent is updated. Several alternatives have been considered and combined in recent implementations to explore trajectories connecting x_1 and x_2:

- do not apply path-relinking at every GRASP iteration, but only periodically;
- explore two different trajectories, using first x_1, then x_2 as the initial solution;
- explore only one trajectory, starting from either x_1 or x_2; and

- do not follow the full trajectory, but instead only part of it.

All these alternatives involve the trade-offs between computation time and solution quality. Ribeiro et al. [42] observed that exploring two different trajectories for each pair $x_1 - x_2$ takes approximately twice the time needed to explore only one of them, with very marginal improvements in solution quality. They have also observed that if only one trajectory is to be investigated, better solutions are found when path-relinking starts from the best among x_1 and x_2. Since the neighborhood of the initial solution is much more carefully explored than that of the guiding solution, starting from the best of them gives to the algorithm a better chance to investigate with more details the neighborhood of the most promising solution. For the same reason, the best solutions are usually found closer to the initial solution than to the guiding one, allowing pruning the relinking trajectory before the latter is reached.

Computational results illustrating a trade-off between these strategies for the bandwidth packing problem are reported later in Subsection 3.3. In this case, the set of moves corresponding to the symmetric difference $\Delta(x_1, x_2)$ between any pair $\{x_1, x_2\}$ of solutions is the subset $K_{x_1,x_2} \subseteq K$ of PVCs routed through different routes in x_1 and x_2. Without loss of generality, let us suppose that path-relinking starts from any elite solution z in the pool and uses the locally optimal solution y as the guiding solution.

The best solution $\overline{y}$ along the new path to be constructed is initialized with z. For each PVC $k \in K_{y,z}$, the same shortest path computations described in Subsections 3.2.1 and 3.2.2 are used to evaluate the cost of the new solution obtained by rerouting the demand associated with PVC k through the route used in the guiding solution y instead of that used in the current solution originated from z. The best move is selected and removed from $K_{y,z}$. The new solution obtained by rerouting the above selected PVC is computed, the incumbent $\overline{y}$ is updated, and a new iteration resumes. These steps are repeated, until the guiding solution y is reached. The incumbent $\overline{y}$ is returned as the best solution found by path-relinking and inserted into the pool if it satisfies the membership conditions.

The pseudo-code with the complete description of the procedure `GRASP+PR_BPP` for the bandwidth packing problem arising in the context of offline PVC rerouting is given in Figure 14. This description incorporates the construction, local search, and path-relinking phases.

3.3 Computational experiments

The experiments were performed on an SGI Challenge computer (28 196-MHz MIPS R10000 processors) with 7.6 Gb of memory. Each run used a single processor. The algorithms were coded in Fortran and were compiled with the SGI MIPSpro F77 compiler using flags `-O3 -64 -static`. CPU times were measured with the system function `etime`.

The experiments were run on two groups of test instances. The first one is formed by some of the test problems from three of the classes used by Fortz and Thorup [21]. The first class is the *AT&T Worldnet backbone* with projected demands, a real-world network with 90 nodes

```
procedure GRASP+PR_BPP;
$\phi^* \leftarrow \infty$;
Pool $\leftarrow \emptyset$;
for $k = 1, \dots,$ Max_Iterations do
    Construct a greedy randomized solution $x$;
    Find $y$ by applying local search to $x$;
    if $y$ satisfies the membership conditions then insert $y$ into Pool;
    Randomly select an elite solution $z \in$ Pool with uniform probability;
    Compute $K_{y,z}$;
    Let $\overline{y}$ be the best solution found by applying path-relinking to $y - z$;
    if $\overline{y}$ satisfies the membership conditions then insert $\overline{y}$ into Pool;
    if $\phi(\overline{y}) < \phi^*$ do
        $x^* \leftarrow \overline{y}$;
        $\phi^* \leftarrow \phi(x^*)$;
    end if;
end for;
return $x^*$;
end GRASP+PR_BPP;
```

Figure 14: Pseudo-code of the GRASP with path-relinking procedure for the bandwidth packing problem

and 274 links. The other two classes are formed by synthetic networks. More specifically, *2-level hierarchical graphs* are generated using the GT-ITM generator [43], based on a model of Calvert et al. [44] and Zegura et al. [45]. Edges are of two types: local access trunks and long distance trunks. The capacities of edges of the same type are equal. Local access trunks have lower capacities than long distance trunks. On *Waxman graphs*, the nodes are uniformly distributed points in the unit square. The probability of having an edge between two nodes u and v is given by $\eta e^{-\delta(u,v)/2\theta}$, where η is a parameter used to control the density of the graph, $\delta(u,v)$ is the Euclidean distance between u and v, and θ is the maximum distance between any two nodes [46]. All trunk capacities are equal. The demands are such that different nodes have different levels of activity, modeling hot spots on the network. They are relatively larger between closer pairs of nodes. We have used another problem generator [26] to create the second group of test instances, with characteristics more similar to those of a frame-relay network. This problem generator and the test instances are available from the authors.

Five problems were selected from each of these groups, whose characteristics are summarized in Table 1. These ten instances are among the largest, to date, to appear in the literature. They are available for download [1] from the authors. The table shows, for each instance, its name, network type, number of nodes, number of trunks, number of demand pairs, and the value Φ_{uncap}, which is the same normalizing scaling factor used by Fortz and Thorup [21]. This normalization allows to compare costs across different network sizes and topologies. This uncapacitated measure is defined as

$$\Phi_{uncap} = \sum_{k \in K} r_k \cdot h_k,$$

where r_k is the bandwidth requirement associated with pair $k \in K$ and h_k is the minimum distance measured with unit weights (hop count) between the origin and destination nodes of demand pair k.

3.3.1 Algorithm variants

In the first set of experiments, we considered four variants of the GRASP and path-relinking schemes proposed in Subsection 3.2.3:

- `G`: This variant is a pure GRASP with no path-relinking.
- `GPRf`: This variant adds to `G` a one-way path-relinking starting from a locally optimal solution and using a randomly selected elite solution as the guiding solution.
- `GPRb`: This variant adds to `G` a one way path-relinking starting from a randomly selected elite solution and using a locally optimal solution as the guiding solution.

[1] `http://www.research.att.com/~mgcr/data/pvc-routing.tar.gz`

Table 1: Problem characteristics.

Instance	Network type	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert K\rvert$	Φ_{uncap}
att	AT&T Worldnet backbone	90	274	272	92,607
hier50a	2-level hierarchical	50	148	2450	113,976,500
hier100a	2-level hierarchical	100	360	9900	435,618,300
wax50a	Waxman	50	476	9900	47,719,429
wax100a	Waxman	100	230	2220	198,827,455
fr250	Frame-relay	60	344	250	173,194
fr500	Frame-relay	60	453	500	288,086
fr750	Frame-relay	60	498	750	448,220
fr1000	Frame-relay	60	518	1000	603,362
fr1250	Frame-relay	60	535	1250	955,568

- GPRfb: This variant combines GPRf and GPRb, performing path-relinking in both directions.

We evaluate the effectiveness of the above variants in terms of the tradeoffs between computational time and solution quality. The parameter δ was set to 1 in the objective function, i.e. only the load balancing component is used.

To study the effect of path-relinking on GRASP, we compared the four variants on two instances. The first is instance att from Table 1. The second is instance fr750a, derived from instance fr750 from Table 1 by scaling all demands by a factor of $1/1.3 = 0.76923$. Two hundred independent runs for each variant were done for each problem. Execution was terminated when a solution of value less than or equal to look4 was found. We used look4 values of 129400 and 479000 for att and fr750a, respectively. These are sub-optimal values chosen such that the slowest variant could terminate in a reasonable amount of computation time. Empirical probability distributions for time to target solution are plotted in Figures 15 and 16. To plot the empirical distribution for each algorithm and each instance, we follow the procedure described in [47]. We associate with the i-th smallest running time t_i a probability $p_i = (i - \frac{1}{2})/200$, and plot the points $z_i = (t_i, p_i)$, for $i = 1, \ldots, 200$. Due to the time taken by the pure GRASP procedure, we limited its plot in Figure 16 to 60 points.

These plots show a similar relative behavior of the four variants on the two instances. Since instance fr750a is harder for all variants and computation times are longer, its plot is more discerning. For a given computation time, the probability of finding a solution at least as good as the target value increase from G to GPRf, from GPRf to GPRfb, and from GPRfb to GPRb. For example, there is 9.25% probability for GPRfb to find a target solution in less than 100 seconds, while this probability increases to 28.75% for GPRb. For G, there is a 8.33%

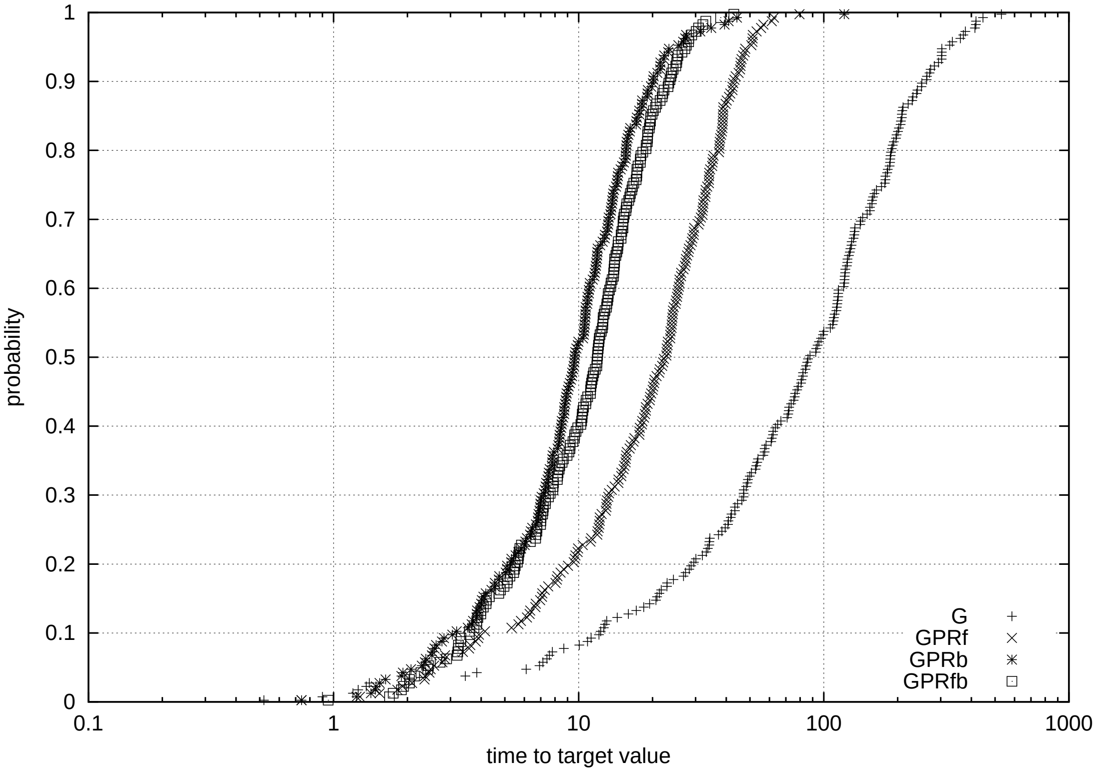


Figure 15: Empirical distributions of time to target solution for GRASP, GRASP with forward path-relinking, GRASP with backward path-relinking, and GRASP with back and forward path-relinking for instance `att`.

probability of finding a target solution within 2000 seconds, while for `GPRf` this probability increases to 65.25%. `GPRb` finds a target solution in at most 129 seconds with 50% probability. For the same probability, this time increases to 172, 1727, and 10933 seconds, respectively, for variants `GPRfb`, `GPRf`, and `G`.

In accordance with these results, variant `GPRb`, which does path-relinking backwards from an elite solution to a locally optimal solution, is the most effective. Because of this, we limit ourselves to only this GRASP with path-relinking variant in the remaining experiments.

3.3.2 Comparison with other heuristics

We now compare `GPRb` using a relatively small number of iterations, fixed at 200, with other simpler heuristics, one of them (heuristic `H1` described below) used in traffic engineering by network planners:

- Heuristic `H1` starts by sorting the pairwise demands in decreasing order and sequentially routes each pair in this order. Each pair is assigned to a minimum hop path (a path minimizing the number of links between the origin and destination nodes).

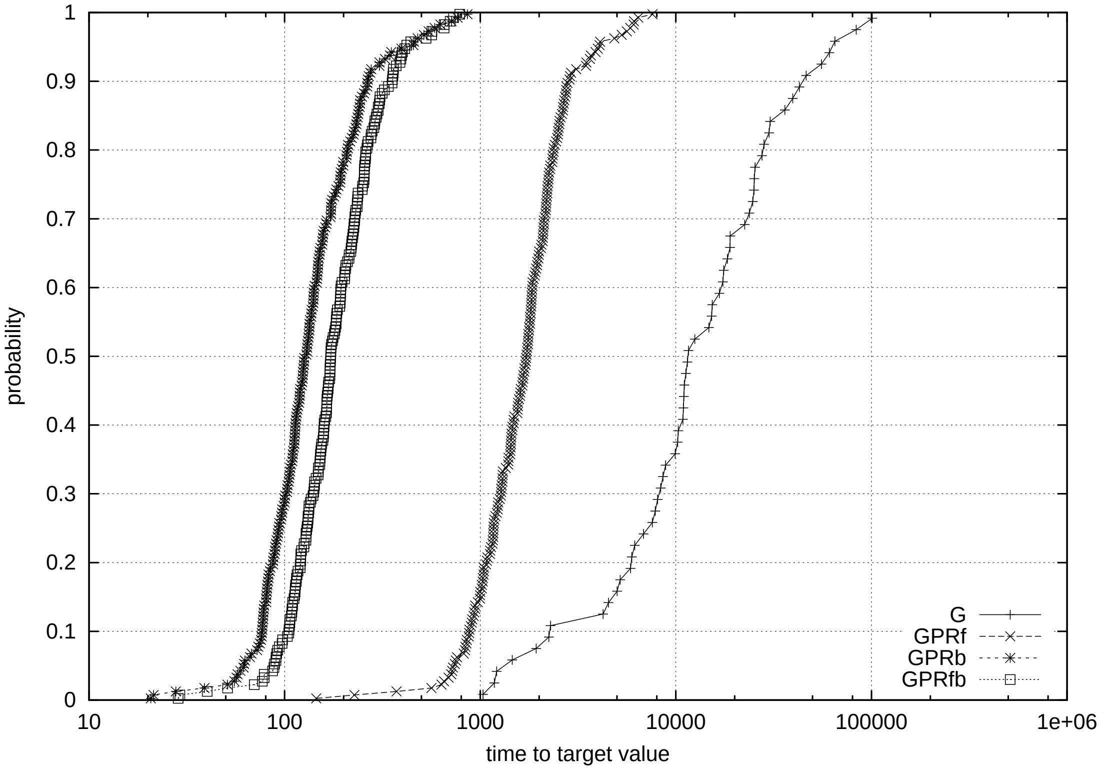


Figure 16: Empirical distributions of time to target solution for GRASP, GRASP with forward path-relinking, GRASP with backward path-relinking, and GRASP with back and forward path-relinking for instance `fr750a`.

- Heuristic `H2` also starts by sorting the pairwise demands in decreasing order and sequentially routes each pair in this order. Each pair is assigned to a route minimizing the same cost function ϕ used in `GPRb`.
- Heuristic `H3` adds to `H2` the same local search procedure used in `GPRb`.

The heuristics above have been implemented in Fortran using the same components used to implement the GRASP with path-relinking variants.

We considered the test problems listed in Table 1. We also wanted to compare our heuristics on other test problems, and with other algorithms, described in [31, 27]. Unfortunately, data for these problems were not available from the authors.

Table 2 summarizes the numerical results. For each algorithm and for each instance, we give the normalized value $\Phi^* = \Phi/\Phi_{uncap}$ (where Φ is the cost function value of the best solution found), the corresponding maximum edge utilization rate Ω, and the distribution of the number of edges that have flow in each of the intervals defining each function ϕ^b_{ij}. The distribution is represented by a sequence of integers, separated by slashes. Right trailing zeroes are omitted. For example, 0/88/416/14 (results obtained by `GPRb` for instance `fr1000`) cor-

Table 2: Numerical results for short runs.

	H1			H2		
Instance	Φ^*	Ω	distribution	Φ^*	Ω	distribution
att	615.34	6.079	228/17/13/1/2/13	1.4995	0.700	216/52/6
hier50a	312.03	2.355	109/21/6/4/0/8	1.2586	0.669	86/60/2
hier100a	460.48	3.151	231/54/25/12/6/39	1.4211	0.900	141/177/42
wax50a	49.048	1.290	109/79/24/3/6/9	1.7204	0.810	5/212/13
wax100a	101.01	2.026	198/218/42/4/5/9	3.8827	1.125	13/447/11/2/0/3
fr250	670.57	3.245	227/62/9/6/9/31	4.0348	1.026	62/223/44/13/2
fr500	959.05	4.119	264/68/37/13/9/62	3.6743	1.006	0/307/125/20/1
fr750	1118.5	4.749	235/98/44/12/13/96	4.3025	1.012	0/43/429/25/1
fr1000	1254.1	4.087	224/96/40/17/10/131	4.9545	0.935	0/1/454/63
fr1250	1909.5	6.420	185/77/49/20/21/183	2472.6	3.278	0/0/0/0/1/534

	H3			GPRb		
Instance	Φ^*	Ω	distribution	Φ^*	Ω	distribution
att	1.3682	0.689	225/46/3	1.3578	0.689	230/40/4
hier50a	1.2295	0.668	92/54/2	1.2141	0.667	103/44/1
hier100a	1.3363	0.898	202/136/22	1.3195	0.875	220/124/16
wax50a	1.4938	0.773	29/197/4	1.4467	0.772	29/197/4
wax100a	2.0175	1.087	18/451/4/0/3	1.9791	1.097	20/449/4/0/3
fr250	4.0042	1.026	69/217/43/13/2	3.3590	1.008	104/194/34/11/1
fr500	3.6270	1.006	2/315/115/20/1	3.1477	1.006	48/304/82/18/1
fr750	4.0242	1.012	1/152/316/28/1	3.5415	1.012	6/206/268/17/1
fr1000	4.6429	0.935	0/19/467/32	3.8461	0.990	0/88/416/14
fr1250	414.56	3.794	0/0/0/1/168/366	345.89	4.867	0/0/0/3/229/303

responds to a solution in which 88 edges have their utilization rates in the interval $[1/3, 2/3)$, 416 edges have their utilization rates in the interval $[2/3, 9/10)$, and 14 edges have their utilization rates in the interval $[9/10, 1)$. Better solutions, in general, will be characterized by smaller cost values, smaller maximum utilization rates, and distributions skewed to the left.

Heuristic H1 does not take into account the cost function ϕ. As expected, for the other heuristics, H3 systematically finds solutions with smaller costs than those found by H2, while GPRb further improves upon H3. Though none of the heuristics considers explicitly the minimization of the maximum utilization rate, this rate is systematically reduced by going from H1 to H2, to H3, and to GPRb.

In general, both the local search in H3 and, more strongly GPRb, contribute to improve the distribution of edges and to rerouting them on less loaded edges. As a result, the skewness to the left is accentuated and the maximum utilization rate is reduced. Figure 17 shows plots illustrating this for all ten test instances. Each plot has two parts. In the left, we represent the difference between the distributions found by heuristics H2 and H3. For each interval, we give the increase in the corresponding number of edges in the solution found by H3 with respect to those in the solution found by H2. Similar results are depicted in the right side of each plot, regarding the solutions obtained by H2 and GPRb. In each case, the areas above and below the horizontal axis are equal.

We note from these plots that both heuristics improve the greedy solution, by reducing the number of overloaded edges and increasing the number of underutilized edges. The plots also illustrate the general performance of GPRb with respect to H3. GPRb tends to improve trunk utilization with respect to H3 by more strongly shifting flow from overloaded to underutilized edges. As a consequence, it obtains solutions characterized by smaller costs and smaller utilization rates.

Table 3: Numerical results for long runs.

	computation time							
Instance	25s	125s	625s	3125s	15625s	78125s	Ω	distribution
att	1.3607	1.3578	1.3577	1.3577	1.3562	1.3562	0.689	231/39/4
hier50a	1.2219	1.2189	1.2146	1.2136	1.2129	1.2113	0.667	98/50
wax50a	1.4849	1.4716	1.4533	1.4462	1.4412	1.4384	0.758	21/206/3
fr250	3.7091	3.3590	3.2796	3.2727	3.2497	3.2496	1.008	135/163/34/11/1
fr500	3.5496	3.3340	3.1466	3.1317	3.1306	3.0912	1.006	39/326/69/18/1

Heuristics H1, H2, and H3, which are not multi-start heuristics, are much faster than the multi-start GPRb. However, there is a clear tradeoff in terms of solution quality when extra time is taken by GPRb. We ran GPRb on five of the ten instances in the experiment for about one CPU-day. Table 3 lists objective function values as a function of the running time for these instances, as well as the maximum utilization rate and the distribution of the number of edges that have flow in each of the intervals defining each cost function ϕ_{ij}^b, for the best solution found. We notice that in most of the cases GPRb continues to improve the solution as the running time increases. Even if the maximum utilization rate does not change, the distribution of the number of edges does. For example, on instance hier50a, the improved solution shifted several edges into the lowest range with respect to the solution in Table 2.

3.3.3 Variation of the hybrid objective function parameter

In this last experiment, we investigate the behavior of the hybrid objective function ϕ with the variation of the parameter δ, used to weigh the network congestion and propagation delay components. We ran GPRb for 2000 iterations on instance att using 41 different values of δ, ranging from 0 to 1. For each value of δ (with the exception of $\delta = 0$ which cannot be plotted on a log scale), the plot in Figure 18 shows the delay and the maximum utilization rate of the best solution found. We note that although the maximum utilization rate is not explicitly considered in the objective function, it appears to be inversely correlated with the parameter

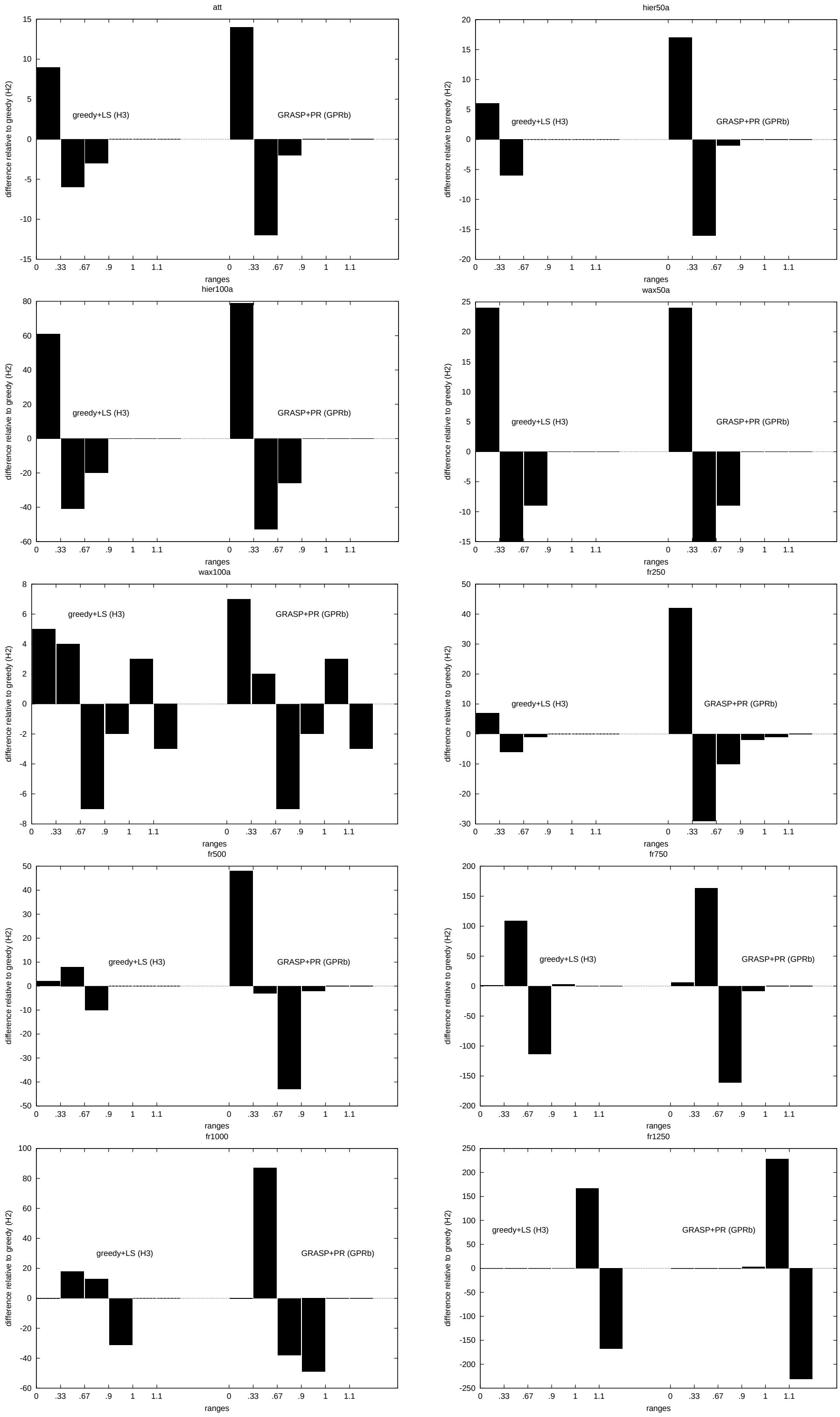


Figure 17: Number of arcs with loads in intervals $[0, .33)$, $[.33, .67)$, $[.67, .9)$, $[.9, 1)$, $[1, 1.1)$, and $[1.1, \infty)$ of trunk capacity, for heuristics H3 (greedy with local search) and GPRb (GRASP with backward path-relinking). Plots show difference with respect to solution found with heuristic H2 (pure greedy).

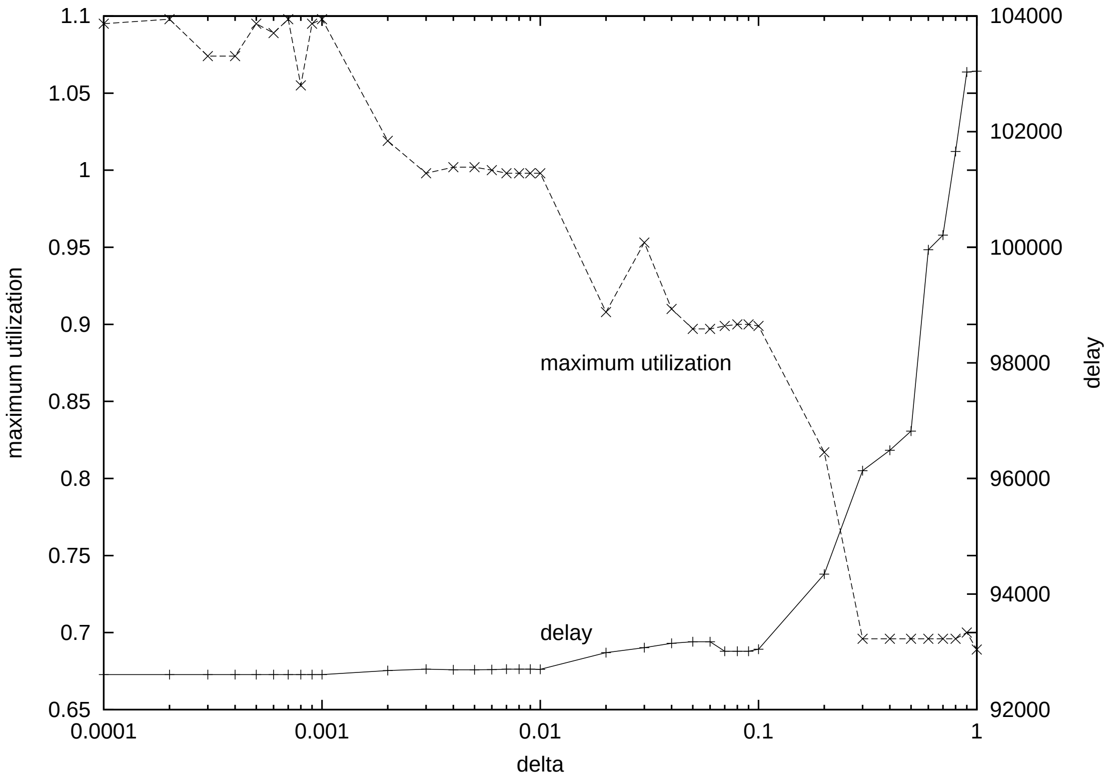


Figure 18: Delay and maximum utilization as a function of objective function parameter δ on instance `att` with unit edge delays.

δ of the objective function. Delays were computed using $\rho_k = r_k$ (see Subsection 3.1) and using unit delays, i.e. $d_{ij} = 1$, for every $(i, j) \in E$.

We first notice from this figure that there is a range of small values of δ, for which the delay is kept at a low value without serious overload. When the value of δ approaches 1, the maximum utilization rate is strongly reduced at the cost of larger delays. Likewise, when the value of δ approaches 0, the delay is strongly reduced at the expense of higher utilization rates. The extreme case, where $\delta = 0$, corresponds to using the purely greedy heuristic `H1`. In this case, the utilization rate is 6.08, and the delay is 92607, which is a lower bound on the value of optimal solution of the capacity constrained delay minimization problem. Since the resulting utilization rate is high, this is an indication that one should use a strictly positive value of δ.

We also observe that, as the value of δ increases from 0 to 1, the maximum utilization rate decreases, following approximately a step function taking values equal to those appearing in the definition of the functions ϕ^b_{ij}, i.e. 1.1, 1.0, 0.9, and 0.67. As the value of δ increases, the minimization of the maximum utilization rate dominates the objective function. As a consequence, the algorithm attempts to reduce the flow on edges with higher loads. To balance this reduction, flows on less loaded edges are increased up to the next breakpoint in its cost function. Therefore, the flows have a tendency to concentrate around breakpoint levels. This characteristic provides a useful strategy for setting the appropriate value of parameter δ of the objective function, to achieve some quality of service (QoS) level defined by a desired balance between propagation delay and delay due to network congestion.

3.4 Concluding remarks

In this section, we presented a new formulation for the bandwidth packing problem arising in the context of offline PVC routing. This formulation uses an objective function that simultaneously takes into account propagation delays and network congestion. Emphasis on either component is controlled by a single parameter. We proposed a family of heuristics for finding approximate solutions to this problem, ranging from a simple greedy algorithm (`H2`) and its improved version using local search (`H3`), to an elaborate combination of GRASP and path-relinking.

Experimental results on realistic-size test problems show that even the simplest greedy heuristic (`H2`) is able to improve on a heuristic used in traffic engineering by network planners (`H1`). The two new simple heuristics (`H2` and `H3`) are fast and find good approximate solutions. The GRASP with path-relinking variants are able to significantly improve upon these simple heuristics, at the expense of additional computation time. GRASP with path-relinking has been shown to be efficiently implemented in parallel with approximate linear speedups in the number of processors [40] and such a strategy could be applied to accelerate `GPRb` and its variants.

The structure of the objective function proposed in this section is such that as the weight of its network congestion component increases, the maximum utilization rate decreases, following approximately a step function. As a consequence, this structure provides a useful strategy for setting the appropriate value of the weight parameter of the objective function, to achieve some quality of service (QoS) level defined by a desired balance between propagation delay and delay due to network congestion.

4 A hybrid GA for OSPF routing

The Internet is divided into many routing domains, called autonomous systems (ASes). These ASes interact to control and deliver IP traffic. They typically fall under the administration of a single institution, such as a company, a university, or a service provider. Neighboring ASes use the Border Gateway Protocol (BGP) to route traffic [48].

The goal of intra-domain traffic engineering [49] consists in improving user performance and making more efficient use of network resources within an AS. Interior Gateway Protocols (IGPs) such as OSPF (Open Shortest Path First) and IS-IS (Intermediate System-

Intermediate System) are commonly used to select the paths along which traffic is routed within an AS.

These routing protocols direct traffic based on link weights assigned by the network operator. Each router in the AS computes shortest paths and creates destination tables used to direct each IP packet to the next router on the path to its final destination. OSPF calculates routes as follows. To each link is assigned an integer weight ranging from 1 to 65535 $(= 2^{16} - 1)$. The weight of a path is the sum of the link weights on the path. OSPF mandates that each router computes a graph of shortest paths with itself as the root [50]. This graph gives the least weight routes (including multiple routes in case of ties) to all destinations in the AS. In the case of multiple shortest paths originating at a router, OSPF is usually implemented so that it will accomplish load balancing by splitting the traffic flow over all shortest paths leaving from each router [51]. In this section, we consider that traffic is split evenly between all outgoing links on the shortest paths to the destination IP address. OSPF requires routers to exchange routing information with all the other routers in the AS. Complete network topology knowledge is required for the computation of the shortest paths.

Given a set of traffic demands between origin-destination pairs [52], the *OSPF weight setting problem* consists in determining weights to be assigned to the links so as to optimize a cost function, typically associated with a network congestion measure.

The NP-hardness of the OSPF weight setting problem was established in [21]. Previous work on optimizing OSPF weights have either chosen weights so as to avoid multiple shortest paths from source to destination or applied a protocol for breaking ties, thus selecting a unique shortest path for each source-destination pair [53, 54, 55]. Fortz and Thorup [21] were the first to consider even traffic splitting in OSPF weight setting. They proposed a local search heuristic and tested it on a realistic AT&T backbone network and on synthetic networks. Ericsson, Resende, and Pardalos [56] proposed a genetic algorithm and used the set of test problems considered in [21]. Sridharan, Guérin, and Diot [57] developed another heuristic for a slightly different version of the problem, in which flow is split among a subset of the outgoing links on the shortest paths to the destination IP address.

In this section, we propose a hybrid genetic algorithm incorporating a local improvement procedure to the crossover operator of the genetic algorithm proposed in [56]. The local improvement procedure makes use of an efficient dynamic shortest path algorithm to recompute shortest paths after the modification of link weights. We compare the hybrid algorithm with the genetic algorithm as well as with the local search procedure in [21].

In the next subsection, we give the mathematical formulation of the OSPF weight setting problem. The hybrid genetic algorithm is described in Subsection 4.2 and the local improvement procedure in Subsection 4.3. Subsection 4.4 describes efficient algorithms for solution update used in the local improvement procedure. Computational results are reported in Subsection 4.5. Concluding remarks are made in the last subsection.

4.1 Problem formulation

In a data communication network, nodes and arcs represent routers and transmission links, respectively. Let N and A denote, respectively, the sets of nodes and arcs. Data packets are routed along links, which have fixed capacities. Consider a directed network graph $G = (N, A)$ with a capacity c_a for each $a \in A$, and a demand matrix D that, for each pair $(s,t) \in N \times N$, gives the demand d_{st} in traffic flow from node s to node t. Then, the OSPF weight setting problem consists in assigning positive integer weights $w_a \in [1, w_{\max}]$ to each arc $a \in A$, such that a measure of routing cost is optimized when the demands are routed according to the rules of the OSPF protocol. The OSPF protocol allows for $w_{\max} \leq 65535$.

For each pair (s,t) and each arc a, let $f_a^{(st)}$ indicate how much of the traffic flow from s to t goes over arc a. Let l_a be the total load on arc a, i.e. the sum of the flows going over a, and let the trunk utilization rate $u_a = l_a/c_a$. The routing cost in each arc $a \in A$ is taken as the piecewise linear function $\Phi_a(l_a)$, proposed by Fortz and Thorup [21] and depicted in Figure 19, which increasingly penalizes flows approaching or violating the capacity limits:

$$\Phi_a(l_a) = \begin{cases} u_a, & u_a \in [0, 1/3) \\ 3 \cdot u_a - 2/3, & u_a \in [1/3, 2/3), \\ 10 \cdot u_a - 16/3, & u_a \in [2/3, 9/10), \\ 70 \cdot u_a - 178/3, & u_a \in [9/10, 1), \\ 500 \cdot u_a - 1468/3, & u_a \in [1, 11/10), \\ 5000 \cdot u_a - 16318/3, & u_a \in [11/10, \infty). \end{cases} \tag{10}$$

Given a weight assignment w and the loads $l_a^{OSPF(w)}$ associated with each arc $a \in A$ corresponding to the routes obtained with OSPF, we denote its routing cost by $\Phi_{OSPF(w)} = \sum_{a \in A} \Phi_a(l_a^{OSPF(w)})$. The OSPF weight setting problem is then equivalent to finding arc weights $w^* \in [1, w_{\max}]$ such that $\Phi_{OSPF(w)}$ is minimized.

The general routing problem can be formulated as the following linear programming problem with a piecewise linear objective function:

$$\Phi_{OPT} = \min \Phi = \sum_{a \in A} \Phi_a(l_a) \tag{11}$$

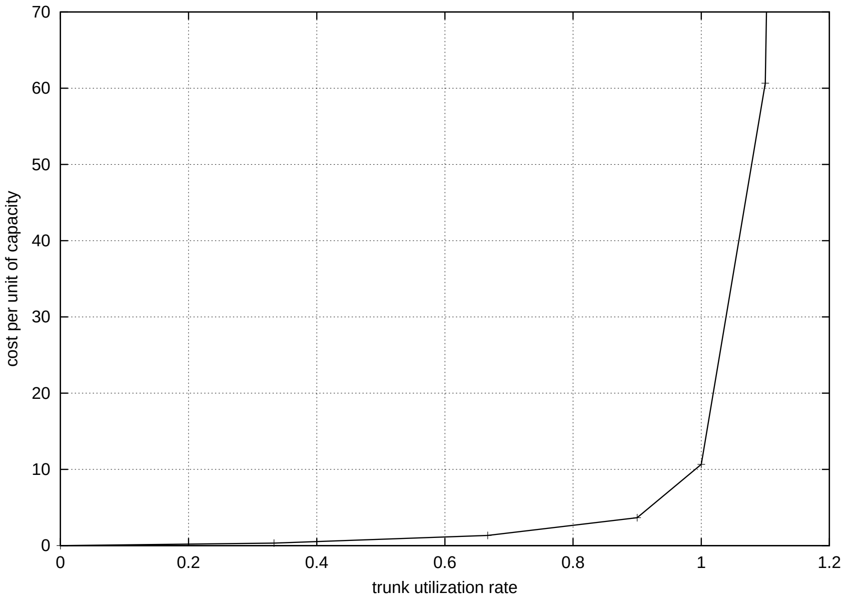


Figure 19: Piecewise linear function $\Phi_a(l_a)$.

subject to

$$\sum_{u:(u,v)\in A} f^{(st)}_{(u,v)} - \sum_{u:(v,u)\in A} f^{(st)}_{(v,u)} = \begin{cases} -d_{st} & \text{if } v = s, \\ d_{st} & \text{if } v = t, \\ 0 & \text{otherwise,} \end{cases} \qquad v, s, t \in N, \tag{12}$$

$$l_a = \sum_{(s,t)\in N\times N} f^{(st)}_a, \qquad a \in A, \tag{13}$$

$$\Phi_a(l_a) \geq l_a, \qquad a \in A, \tag{14}$$

$$\Phi_a(l_a) \geq 3l_a - 2/3c_a, \qquad a \in A, \tag{15}$$

$$\Phi_a(l_a) \geq 10l_a - 16/3c_a, \qquad a \in A, \tag{16}$$

$$\Phi_a(l_a) \geq 70l_a - 178/3c_a, \qquad a \in A, \tag{17}$$

$$\Phi_a(l_a) \geq 500l_a - 1468/3c_a, \qquad a \in A, \tag{18}$$

$$\Phi_a(l_a) \geq 5000l_a - 16318/3c_a, \qquad a \in A, \tag{19}$$

$$f^{(st)}_a \geq 0, \quad a \in A; s, t \in N. \tag{20}$$

Constraints (12) are flow conservation constraints that ensure routing of the desired traffic. Constraints (13) define the load on each arc a and constraints (14–19) define the cost on each arc a according to the cost function $\Phi_a(l_a)$.

The above is a relaxation of OSPF routing, as it allows for arbitrary routing of traffic. Then, Φ_{OPT} is a lower bound on the optimal OSPF routing cost $\Phi_{OSPF(w^*)}$. Also, if $\Phi_{OSPF(1)}$ denotes the optimal OSPF routing cost when unit weights are used, then $\Phi_{OSPF(w^*)} \leq \Phi_{OSPF(1)}$.

Fortz and Thorup [21] proposed a normalizing scaling factor for the routing cost which makes possible comparisons across different network sizes and topologies:

$$\Phi_{UNCAP} = \sum_{(s,t)\in N\times N} d_{st}h_{st},$$

where h_{st} is the minimum hop count between nodes s and t. For any routing cost Φ, the scaled routing cost is defined as

$$\Phi^* = \Phi/\Phi_{UNCAP}.$$

Using this notation, the following results hold:

- The optimal routing costs satisfy

$$1 \leq \Phi^*_{OPT} \leq \Phi^*_{OSPF(w^*)} \leq \Phi^*_{OSPF(1)} \leq 5000.$$

- Given any solution to (11-20) with normalized routing cost Φ^*, then $\Phi^* = 1$ if and only if all arc loads are below 1/3 of their capacities and all demands are routed on minimum hop routes.

- Given any solution to (11-20) where all arcs are at their maximum capacity, then the normalized routing cost $\Phi^* = 10\frac{2}{3}$. We say that a routing *congests* a network if $\Phi^* \geq 10\frac{2}{3}$.

4.2 Hybrid genetic algorithm for OSPF weight setting

In this section, we summarize the detailed description of the genetic algorithm given in [56] and propose a hybrid genetic algorithm by adding a local improvement procedure after the crossover.

A genetic algorithm is a population-based metaheuristic for combinatorial optimization. In this context, a population is simply a set of feasible solutions. Solutions in a population are combined (through crossover) and perturbed (by mutation) to produce a new generation of solutions. When solutions are combined, attributes of higher-quality solutions have a greater probability to be passed down to the next generation. This process is repeated over many generations as long as the quality of the solutions in the new population improves over time. We next show how this idea can be explored for weight setting in OSPF routing.

Each solution is represented by an array of integer weights, where each component corresponds to the weight of an arc of the network. Each individual weight belongs to the interval $[1, w_{\max}]$. Each solution w is associated with a fitness value defined by the OSPF routing cost $\Phi_{OSPF(w)}$. The initial population is randomly generated, with arc weights selected from a uniform distribution in the interval $[1, w_{\max}/3]$. The population is partitioned into three sets $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$. The best solutions are kept in $\mathcal{A}$, while the worst ones are in $\mathcal{C}$. All solutions in $\mathcal{A}$ are promoted to the next generation. Solutions in $\mathcal{B}$ are replaced by crossover of one parent from $\mathcal{A}$ with another from $\mathcal{B} \cup \mathcal{C}$ using the *random keys* crossover scheme of Bean [58]. All solutions in $\mathcal{C}$ are replaced by new randomly generated solutions with arc weights selected in the interval $[1, w_{\max}]$.

In the random keys scheme, crossover is carried out on a selected pair of parent solutions to produce an offspring solution. Each selected pair consists of an elite parent and a non-elite parent. The elite parent is selected, at random, uniformly from solutions in set $\mathcal{A}$, while the non-elite parent is selected, at random, uniformly from solutions in set $\mathcal{B} \cup \mathcal{C}$. Each weight in the offspring solution is either inherited from one of its parents or is reset by mutation. With mutation probability p_m, the weight is reset to a value selected at random in the interval $[1, w_{\max}]$. If mutation does not occur, then the child inherits the weight from its elite parent with a given probability $p_{\mathcal{A}} > 1/2$. Otherwise, it inherits the weight from its non-elite parent.

The hybrid genetic algorithm proposed in this subsection, applies a local improvement procedure to each offspring solution obtained by crossover. This local improvement procedure is described in the next subsection.

4.3 Local Improvement Procedure

In this subsection, we describe the local improvement procedure. Starting from an initial solution, the local improvement procedure analyzes solutions in the neighborhood of a current solution w in the search for a solution having a smaller routing cost. If such a solution exists, then it replaces the current solution. Otherwise, the current solution is returned as a local minimum.

The local improvement procedure is incorporated in the genetic algorithm, described in Subsection 4.2, to enhance its ability to find better-quality solutions with less computational effort. Local improvement is applied to each solution generated by the crossover operator. Besides being computationally demanding, the use of large neighborhoods in a hybrid genetic algorithm can lead to loss of population diversity, and consequently premature convergence to low-quality local minima. We next describe the local improvement procedure using a reduced neighborhood.

As before, let l_a denote the total load on arc $a \in A$ in the solution defined by the current weight settings w. We recall that $\Phi_a(l_a)$ denotes the routing cost on this arc. The local improvement procedure examines the effect of increasing the weights of a subset of the arcs. These candidate arcs are selected among those with the highest routing costs and whose weight is smaller than $w_{\max}$. To reduce the routing cost of a candidate arc, the procedure attempts to increase its weight to induce a reduction on its load. If this leads to a reduction in the overall routing cost, the change is accepted and the procedure is restarted. The procedure stops at a local minimum when no improvement results from changing the weights of the candidate arcs. The pseudo-code in Figure 20 describes the local improvement procedure in detail.

```
procedure LocalImprovement($q, w$)
    $dontlook_a \leftarrow 0, \forall a \in A$;
    $i \leftarrow 1$;
    while $i \leq q$ do
        Renumber the arc indices such that
        $\Phi_a(l_a) \geq \Phi_{a+1}(l_{a+1}), \forall a = 1, \ldots, |A| - 1$;
        $a' \leftarrow 0$;
        for $a = 1, \ldots, |A|$ while $a' = 0$ do
            if $dontlook_a = 1$ then $dontlook_a \leftarrow 0$;
            else if $w_a < w_{\max}$ then $a' \leftarrow a$;
        end for;
        if $a' = 0$ then return;
        $dontlook_{a'} \leftarrow 1$;
        for $\hat{w} = w_{a'} + 1, \ldots, w_{a'} + \lceil (w_{\max} - w_{a'})/4 \rceil$ do
            $w'_a \leftarrow w_a, \forall a \in A, a \neq a'$;
            $w'_{a'} \leftarrow \hat{w}$;
            if $\Phi_{OSPF(w')} < \Phi_{OSPF(w)}$ then
                $w \leftarrow w'$;
                $dontlook_{a'} \leftarrow 0$;
                $i \leftarrow 0$;
            end if
        end for
        $i \leftarrow i + 1$;
    end while
end LocalImprovement.
```

Figure 20: Pseudo-code of procedure LocalImprovement.

The procedure `LocalImprovement` takes as input parameters the current solution defined by the weights w and a parameter q which specifies the maximum number of candidate arcs to be examined at each local improvement iteration. To speed up the search, we disallow the weight increase of arcs for which no weight increase leads to an improvement in the routing cost in the previous iteration. To implement this strategy, we make use of a *don't look* bit for each arc.

The *don't look* bits are set unmarked in line 1 and the counter of candidate arcs is initialized in line 2. The loop in lines 3 to 22 investigates at most q selected candidate arcs for weight increase in the current solution. The arc indices are renumbered in line 4 such that the arcs are considered in non-increasing order of routing cost. The loop in lines 6 to 9 searches for an unmarked arc with weight less than $w_{\max}$. Marked arcs which cannot be selected at the current iteration are unmarked in line 7 for future investigation. Arc a' is selected in line 8. If no arc satisfying these conditions is found, the procedure stops in line 10 returning the current weights w as the local minimum. In line 11, arc a' is temporarily marked to disallow its investigation in the next iteration, unless a weight change in $w_{a'}$ results in a better solution.

The loop in lines 12 to 20 examines all possible weight changes for arc a' in the range $[w_{a'} + 1, w_{a'} + \lceil (w_{\max} - w_{a'})/4 \rceil]$. A neighbor solution w', keeping all arc weights unchanged except for arc a', is built in lines 13 and 14. If the new solution w' has a smaller routing cost than the current solution (test in line 15), then the current solution is updated in line 16, arc a' is unmarked in line 17, and the arc counter i is reset in line 18. In line 21, we increment the candidate arc counter i.

The routing cost $\Phi_{OSPF}(w')$ associated with the neighbor solution w' must be evaluated in line 15. Instead of computing it from scratch, we use fast update procedures for recomputing the shortest path graphs as well as the arc loads. These procedures are considered in the next subsection. Once the new arc loads are known, the total routing cost is computed as the sum of the individual arc routing costs.

4.4 Fast updates of arc loads and routing costs

In this subsection, we describe the procedures used for fast update of the cost (line 15 of procedure `LocalImprovement`) and arc loads. We are in the situation where l are the loads associated with the current weight settings w and the weight of a unique arc a' is increased by exactly a unit.

Let T be the set of destination nodes and denote by $g^t = (N, A^t)$ the shortest path graph associated with each destination node $t \in T$. One or more of these shortest path graphs will be affected by the change of the weight of arc a' from $w_{a'}$ to $w_{a'} + 1$. Consequently, the loads of some of the arcs in each graph will change.

The procedures described in this subsection use a set of data structures that work like a memory for the solution. With a weight change, the shortest path graph and the loads can change, and the memories are updated instead of recomputed from scratch.

Each shortest path graph with node t as destination has an $|A|$-vector A^t indicating the arcs in g^t. If arc a is in the shortest path graph, then $A^t_a = 1$. Otherwise, $A^t_a = 0$. Another $|A|$-vector, l^t, associated with the arcs, stores the partial loads flowing to t traversing each arc $a \in A$. The total load from each arc is represented in the $|A|$-vector l_a which stores the total load traversing each arc $a \in A$. The $|N|$-vectors π^t and δ^t are associated with the nodes. The distance from each node to the destination t is stored in π^t, while δ^t keeps the number of arcs outgoing from each node in g^t. All these structures are populated in the beginning and set free at the end of procedure *LocalImprovement*. For simplicity, they were omitted from this procedure and from the parameter list of the procedures described in this subsection.

```
procedure UpdateCost($a', d, l$)
1    forall $a \in A$ do $l_a \leftarrow 0$;
2    forall $t \in T$ do
3        UpdateShortestPaths($a', w, t$);
4        UpdateLoads($d, t$);
5        forall $a \in A^t$ do $l_a \leftarrow l_a + l^t_a$;
6    end forall
7    $\Phi_{OSPF(w')} \leftarrow \sum_{a \in A} \Phi_a(l_a)$;
end UpdateCost.
```

Figure 21: Pseudo-code of procedure `UpdateCost`.

The pseudo-code in Figure 21 summarizes the main steps of the update procedure. The new load l_a on each arc $a \in A$ is set to zero in line 1. The loop in lines 2 to 6 considers each destination node $t \in T$. For each one of them, the shortest path graph g^t is updated in line 3 and the partial arc loads are updated in line 4. The arc loads are updated in line 5. Lines 1 and 5 are removed from this procedure in case the arc loads be updated inside procedure `UpdateLoads`. Finally, the cost $\Phi_{OSPF(w')}$ of the new solution is computed in line 7. In the remainder of this subsection we describe the procedures `UpdateShortestPaths` and `UpdateLoads` used in lines 3 and 4.

4.4.1 Dynamic reverse shortest path algorithm

We denote by $g^t = (N, A^t)$ the shortest paths graph associated with each destination node $t \in T$. Since the weight of a unique arc a' was changed, the graph g^t does not have to be recomputed from scratch. Instead, we update the part of it which is affected by the weight change. Ramalingam and Reps [59] and Frigioni et al. [60] proposed efficient algorithms for these dynamic computations in Dijkstra's algorithm. These two algorithms are compared experimentally in [61]. Although the algorithm of Frigioni et al. is theoretically better, the algorithm of Ramalingam and Reps usually runs faster in practice. Due to the nature of the OSPF weight setting problem, we use the reversed version of Dijkstra's shortest path algorithm.

The pseudo-code of the specialized dynamic shortest path algorithm for unit weight increases is given in Figure 22. Buriol et al. [62] showed empirically that this algorithm is faster than the general dynamic reverse shortest path algorithm of Ramalingam and Reps [59]. First, it identifies the set Q of nodes whose distance labels change due to the increased weight. Next, the shortest paths graph is updated by deleting and adding arcs for which at least one of its extremities belongs to Q.

Algorithm `UpdateShortestPaths` takes as parameters the arc $a' = (\overrightarrow{u,v})$ whose weight changed, the current setting of weights w, and a destination node $t \in T$. The algorithm checks in line 1 if arc a' does not belong to the shortest paths graph g^t, in which case the weight change does not affect the latter and the procedure returns. Arc a' is eliminated from the shortest paths graph g^t in line 2. In line 3 the tail node of arc a' is inserted in a heap containing all nodes for which the load of its outgoing links might change. This heap will be used later by the load update procedure. The distances to the destination node t are used as priority keys and the root contains the node with maximum distance. The outdegree δ_u^+ of the tail node of arc a' is updated in line 4. In line 5, we check if there is an alternative path to the destination starting from u. If this is the case, the procedure returns in line 5, since no further change is needed. The set Q of nodes affected by the weight change in arc a' is initialized with node u in line 6. The loop in lines 7 to 15 builds the set Q. For each node identified in this set (line 7), its distance $\pi^t(v)$ to the destination node is increased by 1 in line 8. The loop in lines 16 to 24 updates the shortest paths. Each node u in set Q is investigated one-by-one in line 16 and each outgoing arc a is scanned in line 17. We check in line 18 if arc a belongs to the new shortest path to the destination. If so, arc a is inserted in the shortest paths graph in line 19, its tail u is inserted in the heap in line 20, and its outdegree $\delta^+(u)$ is updated in line 21.

4.4.2 Dynamic load update

We first recall that procedure `UpdateShortestPaths` built a heap H containing all nodes for which the set of outgoing arcs was modified in the shortest paths graph.

The pseudo-code in Figure 23 summarizes the main steps of the load update procedure. We denote by l_a^t the load on arc $a \in A$ associated with the destination node $t \in T$. Procedure `UpdateLoads` takes as parameters the demands d and a destination node t. The loop in lines 1 to 9 removes nodes from the heap until the heap becomes empty. The node u with maximum distance to the destination node is removed in line 2. The total load flowing through node u is equal $d_{ut} + \sum_{a=(v,u)\in A^t} l_a$. The load in each arc leaving node u is computed in line 3. The

```
procedure UpdateShortestPaths($a' = (\overrightarrow{u,v}), w, t$)
    if $a' \notin A^t$ return;
    $A^t \leftarrow A^t \setminus \{a'\}$;
    HeapInsertMax($H, u, \pi^t(u)$);
    $\delta^+_u \leftarrow \delta^+_u - 1$;
    if $\delta^+_u > 0$ then return
    $Q = \{u\}$;
    forall $v \in Q$ do
        $\pi^t(v) \leftarrow \pi^t(v) + 1$;
        forall $a = (u, v) \in$ IN($v$) $\cap A^t$ do
            $A^t \leftarrow A^t \setminus \{a\}$;
            HeapInsertMax($H, u, \pi^t(u)$);
            $\delta^+(u) \leftarrow \delta^+(u) - 1$;
            if $\delta^+(u) = 0$ then $Q \leftarrow Q \cup \{u\}$;
        end forall
    end forall
    forall $u \in Q$ do
        forall $a = (u, v) \in$ OUT($u$) do
            if $\pi^t(u) = w_a + \pi^t(v)$ then
                $A^t \leftarrow A^t \cup \{a\}$;
                HeapInsertMax($H, u, \pi^t(u)$);
                $\delta^+(u) \leftarrow \delta^+(u) + 1$;
            end if
        end forall
    end forall
end UpdateShortestPaths.
```

Figure 22: Pseudo-code of procedure UpdateShortestPaths.

```
procedure UpdateLoads($H, d, t$)
1   while HeapSize($H$) $> 0$ do
2       $u \leftarrow$ HeapExtractMax($H$);
3       $load \leftarrow (d_{ut} + \sum_{a=(v,u) \in A^t} l_a^t)/\delta^+(u)$;
4       forall $a = (u, v) \in A^t : l_a^t \neq load$ do
5           $l_a^t \leftarrow load$;
6           HeapInsertMax($H, v, \pi(v)$);
7       end forall
8   end while
end UpdateLoads.
```

Figure 23: Pseudo-code of procedure `UpdateLoads`.

loop in lines 4 to 7 scans all arcs leaving node u in the current shortest paths graph for which the partial load l_a^t has to be updated. The new partial load l_a^t is set in line 5 and the head v of arc a is inserted in the heap H in line 6.

4.5 Computational Results

In this subsection, we describe the experimental results using the hybrid genetic algorithm introduced in this subsection. We describe the computer environment, list the values of the algorithm parameters, present the test problems, and outline the experimental setup.

In the experiments, we compare the hybrid genetic algorithm with the lower bound associated with the linear program (11–20) and other heuristics.

4.5.1 The setup

The experiments were done on an SGI Challenge computer (28 196-MHz MIPS R10000 processors) with 7.6 Gb of memory. Each run used a single processor.

The algorithms were implemented in C and compiled with the MIPSpro `cc` compiler, version 7.30, using flag `-O3`. Running times were measured with the `getrusage` function. Random numbers were generated in the hybrid genetic algorithm as well as in the pure genetic algorithm using Matsumoto and Nishimura's *Mersenne Twister* [63].

The following parameters were set in both the pure and the hybrid genetic algorithms:

- Population size: 50.
- Weight range: $[1, w_{\max} = 20]$.

- Population partitioning placed the top 25% of the solutions (rounded up to 13) in set $\mathcal{A}$, the bottom 5% of the solutions (rounded up to 3) in set $\mathcal{C}$, and the remaining solutions in set $\mathcal{B}$.

- Probability that mutation occurs: $p_m = 0.01$.

- Probability that an offspring inherits the weight from the elite parent during crossover: $p_{\mathcal{A}} = 0.7$.

- The number of generations varied according to the type of experiment.

In addition, the maximum number of candidate arcs is set to $q = 5$ in the local improvement procedure in the hybrid genetic algorithm.

The experiments were done on a real-world network from proposed by Fortz and Thorup [21] and also used in [56]. The *AT&T Worldnet backbone* is a proposed real-world network of 90 routers and 274 links with 17 destination nodes and 272 origin-destination pairs.

Twelve distinct demand matrices $D^1, D^2, \dots, D^{12}$ are generated. Starting from demand matrix D^1, the other demand matrices are generated by repeatedly multiplying D^1 by a scaling factor: $D^k = \rho^{k-1} D^1, \forall\, k = 1, \dots, 12$.

4.5.2 Fixed time comparison

In this subsection, we compare the hybrid genetic algorithm with three heuristics:

- InvCap: weights are set proportional to the inverse of the link capacity, i.e. $w_a = \lceil c_{\max}/c_a \rceil$, where $c_{\max}$ is the maximum link capacity;

- GA: the basic genetic algorithm without the local search used by the hybrid genetic algorithm;

- LS: the local search algorithm of Fortz and Thorup [21];

as well as with LPLB, the linear programming lower bound Φ_{OPT}. InvCap is used in Cisco IOS 10.3 and later by default [64, 65]. GA is derived from the genetic algorithm in [56]. LS is the implementation used in [21].

Twelve increasingly loaded traffic demand matrices are considered. InvCap and LPLB were run a single time. Ten one-hour runs were done with GA, HGA, and LS on each instance and average routing costs computed.

Table 4 and Figure 26 summarize these results. For each demand level, the table list the normalized costs for InvCap and LPLB as well as the average normalized costs over ten one-hour runs for GA, HGA, and LS. The last row in the table lists the sum of the normalized average costs for each algorithm. Normalized cost values less than $10\frac{2}{3}$ (i.e., recall that when the routing cost exceeds $10\frac{2}{3}$ we say that the routing congests the network; see Subsection 4.1) are separated from those for which the network is congested by a line segment in the tables. The distribution of the costs can be seen in the figure, where all ten cost values for each algorithm and each demand point are plotted together with the average costs.

We make the following remarks about the computational results. The pure genetic algorithm (GA) consistently found better solutions than InvCap. Solution differences increased with traffic intensity.

HGA found solutions at least as good as GA for all demand levels. Solution differences increased with traffic intensity. HGA not only found better-quality solutions, but did so in less CPU time (see Figures 24 and 25, which compare one run of HGA and GA each on the `att` network with demand $D = 45134.146$). Figure 24 shows the value of the best-quality solution in the population as a function of CPU time, while Figure 25 shows the value of the best-quality solution in the population as a function of the generation of the algorithm. These figures illustrate how close to the LP lower bound the HGA comes and how much faster HGA is to converge compared to GA.

4.5.3 Distribution of time-to-target-solution-value

To study and compare the computation times, we used the methodology proposed by Aiex et al.[66] and further explored by Resende and Ribeiro e.g. in [67].

Without loss of generality, we considered network `att` with the demand equal to 37611.7 to illustrate the general behavior observed for most instances. We performed one hundred independent runs with different seeds of each algorithm GA, HGA, and LS, considering a given parameter value `look4`. Each execution was terminated when a solution of value less than or equal to the target value `look4` was found or when the time limit of one hour was reached. Three different values (corresponding to easy, medium, and difficult cases) of `look4` were investigated: 2.89, 2.77 and 2.64. Different target values were used since the networks and demands were different. Empirical probability distributions for the time-to-target-solution-value are plotted in Figure 27. Runs which failed to find a solution of value less than or equal to the target value `look4` within the one-hour time limit were discarded in these plots. To plot the empirical distribution for each algorithm and each instance, we associate with the i-th smallest running time t_i a probability $p_i = (i - \frac{1}{2})/100$, and plot the points $z_i = (t_i, p_i)$, for $i = 1, \dots, n_r$, where $n_r \leq 100$ is the number of runs which found a solution of value less than or equal to the target value `look4` within the one-hour time limit.

HGA and LS found solutions with value less than or equal to the target in all runs associated with network `att` (Figures 27 and 28). GA failed to find solutions at least as good as the target value on eight runs with the easiest target, 19 runs with the medium target, and 59 runs with the hardest target. HGA is not only much faster than GA, but also the computation times of the former are more predictable than those of the latter. As we can see from Figure 27, in many runs of GA the computation times are several orders of magnitude

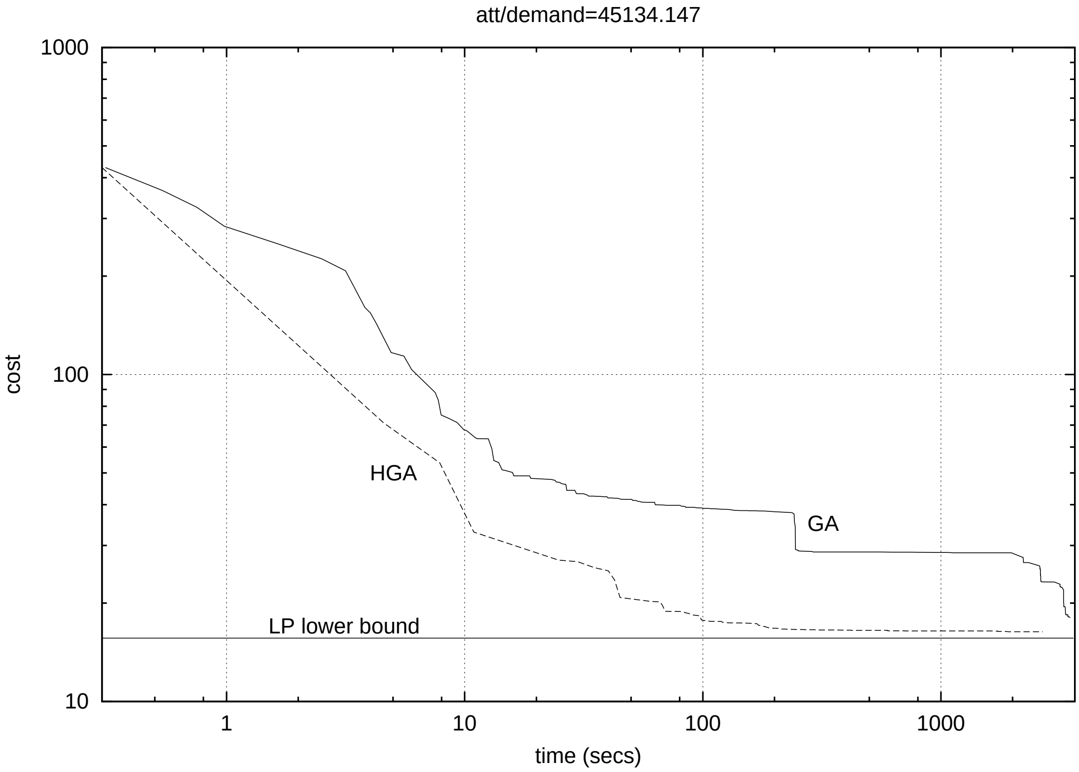


Figure 24: Cost as a function of time on 1-hour run: HGA versus GA on `att` with demand 45134.146

larger than those of HGA. Considering Figure 28, we notice that the computation times of HGA are more predictable than those of LS. The latter are several times larger than the former in many runs. In more than 30% of the runs LS encounters difficulties to converge and requires very long computation times. The figure seems to suggest that this is related to the fact that LS frequently gets stuck at a local minimum and the escape mechanism often needs to be applied repeatedly to succeed.

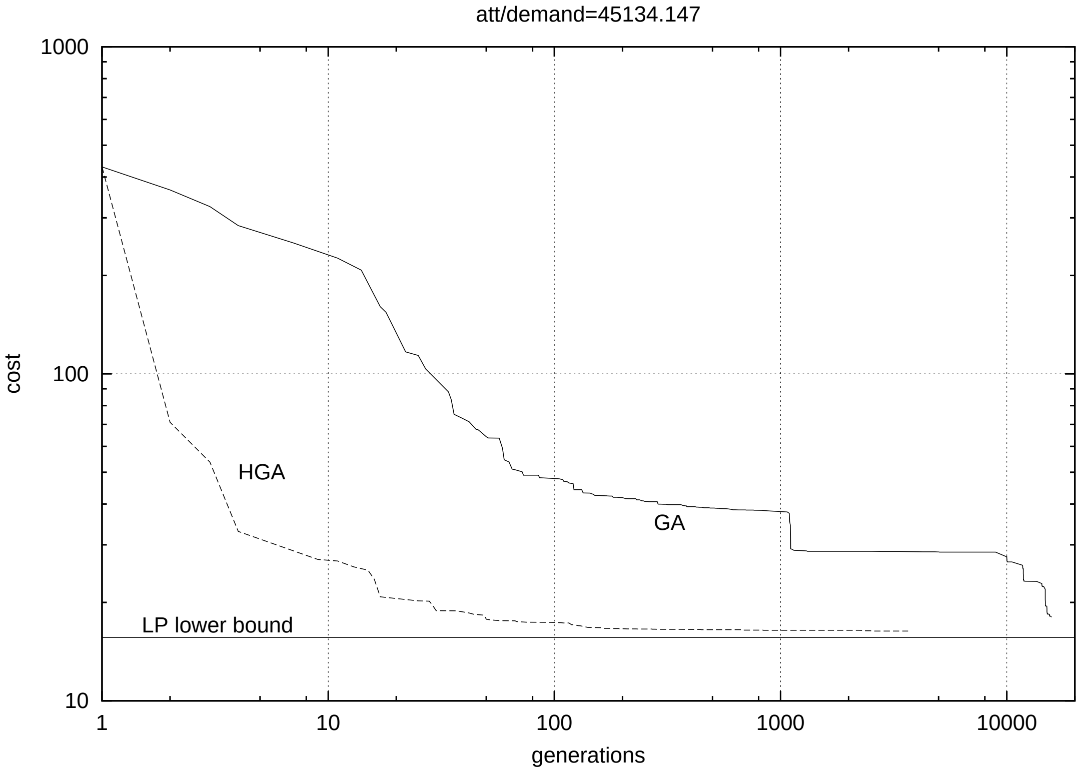


Figure 25: Cost as a function of generations on 1-hour run: HGA versus GA on `att` with demand 45134.146

Table 4: Routing costs for `att` with scaled projected demands. Solutions are averaged over ten one-hour runs.

Demand	InvCap	GA	HGA	LS	LPLB
3761.179	1.013	1.000	1.000	1.000	1.00
7522.358	1.013	1.000	1.000	1.000	1.00
11283.536	1.052	1.010	1.008	1.008	1.01
15044.715	1.152	1.057	1.050	1.050	1.05
18805.894	1.356	1.173	1.168	1.168	1.15
22567.073	1.663	1.340	1.332	1.331	1.31
26328.252	2.940	1.520	1.504	1.506	1.48
30089.431	21.051	1.731	1.689	1.691	1.65
33850.609	60.827	2.089	2.007	2.004	1.93
37611.788	116.690	2.663	2.520	2.520	2.40
41372.967	185.671	5.194	4.382	4.377	3.97
45134.146	258.263	20.983	16.433	16.667	15.62
Total	652.691	40.760	35.093	35.322	33.57

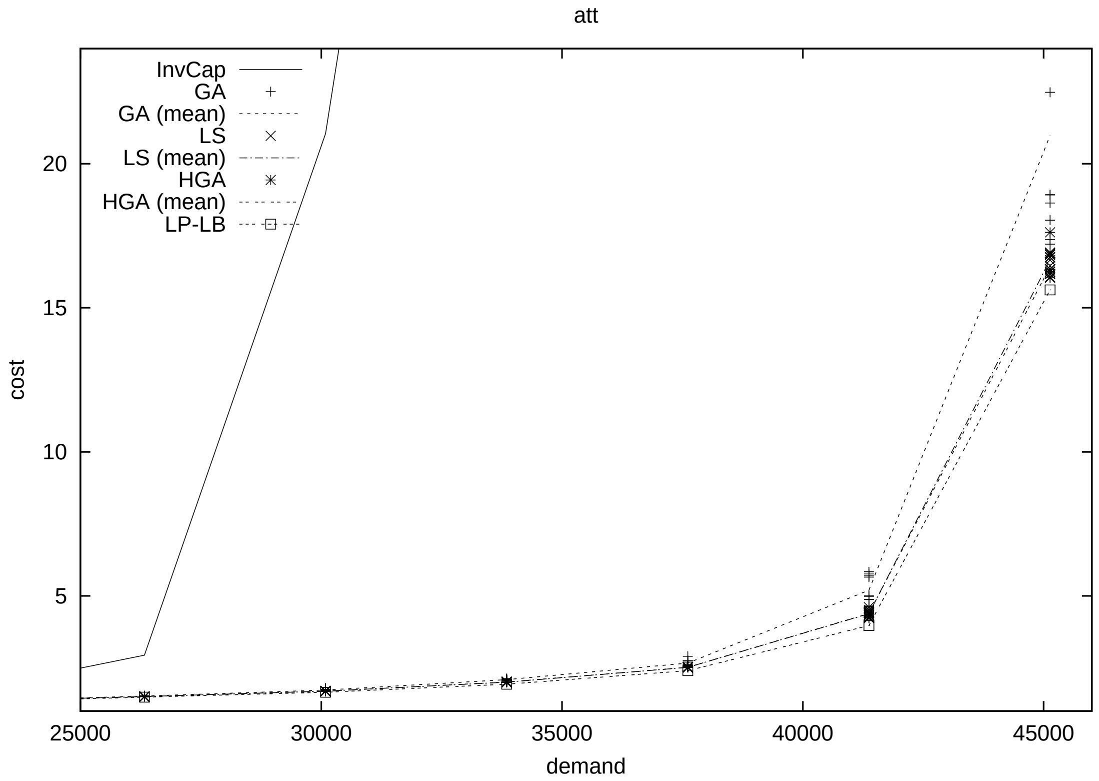


Figure 26: InvCap, GA, HGA, LS, and LP lower bound on `att`.

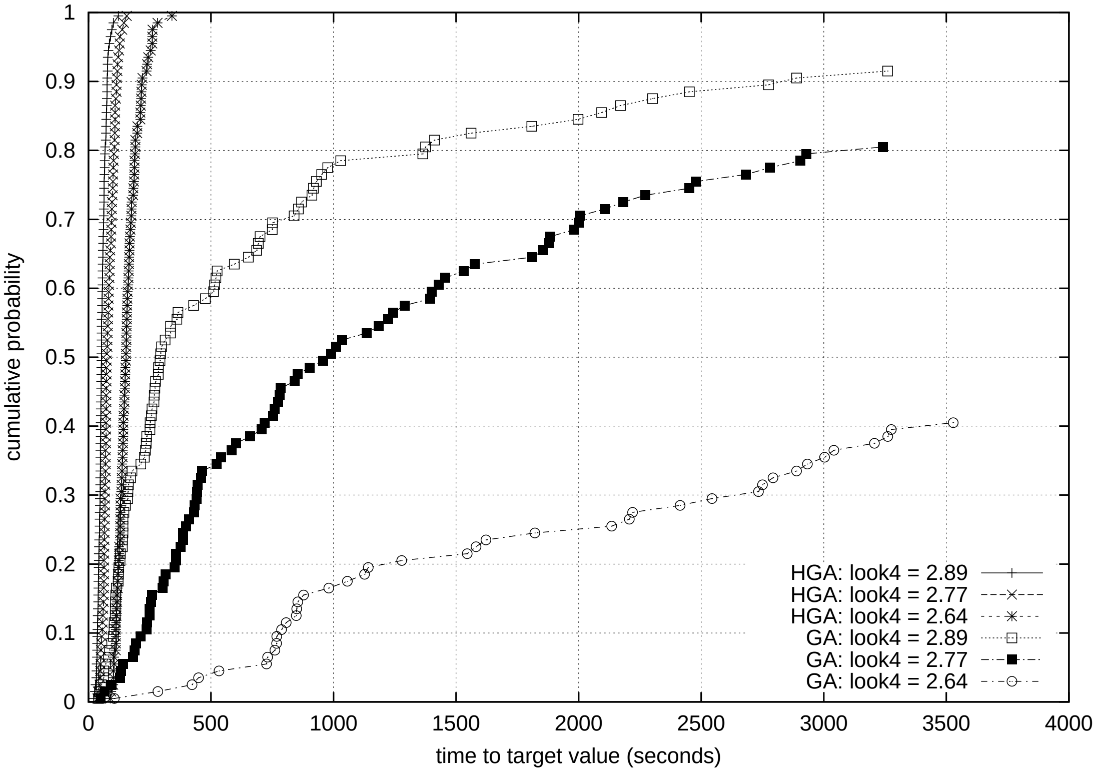


Figure 27: Time to target solution value: HGA versus GA on network `att` with demand 37611.788

4.6 Concluding remarks

We presented a new hybrid genetic algorithm (HGA) for solving the OSPF weight setting problem, combining the traditional genetic algorithm (GA) strategy with a local search procedure to improve the solutions obtained by crossover.

The local search procedure uses small neighborhoods and is based on the fast computation of dynamic shortest paths. Since it considers only unit weight increments with respect to the weights in the current solution, our implementation of `UpdateShortestPaths` is 2 to 3 times faster than its original implementation. This specialization accounts significantly to speedup the implementation of the hybrid genetic algorithm.

The new heuristic performs systematically better than the genetic algorithm without local search (GA and GA^0). HGA finds better solutions in substantially less computation times. The experimental results also showed that it is also more robust, in the sense that it rarely gets stuck in suboptimal local minima, while the genetic algorithm often does so.

We also compared the new algorithm with a local search heuristic (LS). Once again, HGA is more robust than LS. Algorithms HGA and LS are competitive in terms of solution quality and time. HGA is better than LS for some classes of test problems, while LS is better for

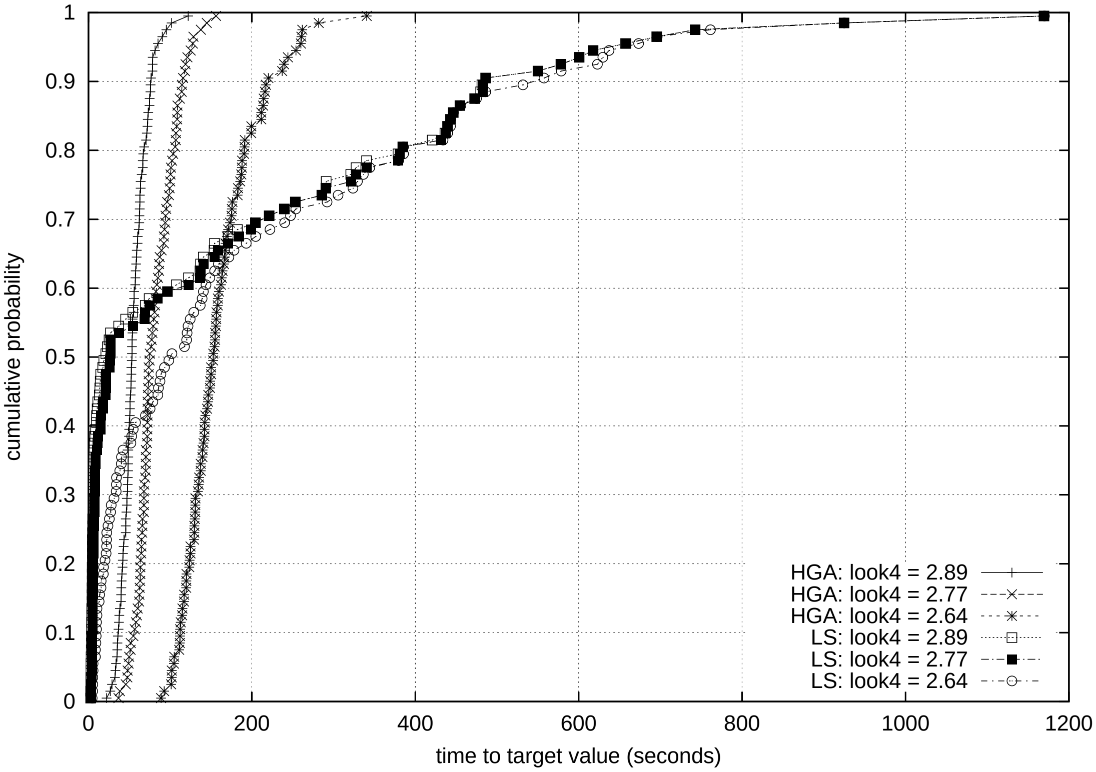


Figure 28: Time to target solution value: HGA versus LS on network `att` with demand 37611.788

others. Moreover, the implementation of LS is based on limited-size hashing tables which limits the number of iterations it can perform and, consequently, the solution quality that can be obtained for larger problems.

Acknowledgement

The author would like to acknowledge the following people who worked with him on the projects described in this paper. Celso Ribeiro collaborated with the author on routing (PVC and OSPF). David Johnson, Anja Feldman, and Chris Sorensen suggested local search strategies and provided data for the PoP placement study. Lucia Resende introduced the author to the private virtual circuit routing problem and modeled and implemented an early version of the routing tool. The author wishes to thank her for comments and suggestions that greatly improved the section. The research on PVC routing described in this paper was done while C. C. Ribeiro was visiting AT&T Labs Research during his sabbatical leave. His work was sponsored by FAPERJ research grant 150966/99 and by CNPq research grants 302281/85-1, 202005/89-5, and 910062/99-4. The work on OSPF routing described in this paper was done during the period Luciana Buriol was a visiting scholar at AT&T Labs Research in 2001/2002. Her work was partially sponsored by a CAPES grant. Finally, Mikkel Thorup collaborated with the author on the OSPF routing research and suggested many of the ideas that made the hybrid genetic algorithm efficient.

References

[1] M. Resende, "Computing approximate solutions of the maximum covering problem using GRASP," *Journal of Heuristics*, vol. 4, pp. 161–171, 1998.

[2] M. Resende and C. Ribeiro, "A GRASP with path-relinking for private virtual circuit routing," *Networks*, vol. 41, no. 3, pp. 104–114, 2003.

[3] L. Buriol, M. Resende, C. Ribeiro, and M. Thorup, "A hybrid genetic algorithm for the weight setting problem in ospf/is-is routing," tech. rep., AT&T Labs Research, 180 Park Avenue, Florham Park, NJ 07932 USA, 2003.

[4] D. Schilling, V. Jayaraman, and R. Barkhi, "A review of covering problems in facility location," *Location Science*, vol. 1, pp. 25–55, 1993.

[5] V. L. Bennett, D. J. Eaton, and R. L. Church, "Selecting sites for rural health workers," *Social Science Medicine*, vol. 16, pp. 63–72, 1982.

[6] D. J. Eaton, M. S. Daskin, D. Simmons, B. Bulloch, and G. Jansma, "Determining medical service vehicle deployment in Austin, Texas," *Interfaces*, vol. 15, pp. 96–108, 1985.

[7] D. J. Sweeney, R. L. Mairose, and R. K. Martin, "Strategic planning in bank location," in *AIDS Proceedings*, November 1979.

[8] C. H. Chung, "Recent applications of the maximal covering location planning (M. C. L. P.) model," *Journal of the European Operational Research Society*, vol. 37, pp. 735–746, 1986.

[9] M. S. Daskin, P. C. Jones, and T. J. Lowe, "Rationalizing tool selection in a flexble manufacturing system for sheet metal products," *Operations Research*, vol. 38, pp. 1104–1115, 1990.

[10] F. P. Dwyer and J. R. Evans, "A branch and bound algorithm for the list selection problem in direct mail advertising," *Management Science*, vol. 27, pp. 658–667, 1981.

[11] R. Church and C. ReVelle, "The maximal covering location problem," *Papers of the Regional Science Association*, vol. 32, pp. 101–118, 1974.

[12] T. A. Feo and M. G. C. Resende, "Greedy randomized adaptive search procedures," *Journal of Global Optimization*, vol. 6, pp. 109–133, 1995.

[13] T. Feo and M. Resende, "A probabilistic heuristic for a computationally difficult set covering problem," *Operations Research Letters*, vol. 8, pp. 67–71, 1989.

[14] M. G. C. Resende, L. S. Pitsoulis, and P. M. Pardalos, "Approximate solution of weighted MAX-SAT problems using GRASP," in *Satisfiability Problem: Theory and Applications* (D.-Z. Du, J. Gu, and P. M. Pardalos, eds.), DIMACS Series on Discrete Mathematics and Theoretical Computer Science, Providence, R.I.: American Mathematical Society, 1996.

[15] J. G. Klincewicz, "Avoiding local optima in the p-hub location problem using tabu search and GRASP," *Annals of Operations Research*, vol. 40, pp. 283–302, 1992.

[16] A. Ouorou, P. Mahey, and J. Vial, "A survey of algorithms for convex multicommodity flow problems," *Management Science*, vol. 46, pp. 126–147, 2000.

[17] S. Even, A. Itai, and A. Shamir, "On the complexity of timetable and multicommodity flow problems," *SIAM Journal on Computing*, vol. 5, pp. 691–703, 1976.

[18] I. Chlamtac, A. Faragó, and T. Zhang, "Optimizing the system of virtual paths," *IEEE/ACM Trans. on Networking*, vol. 2, pp. 581–587, 1994.

[19] D. Bertsekas and R. Gallager, *Data networks.* Prentice-Hall, Inc., 2nd ed., 1992.

[20] M. Gerla, J. Monteiro, and R. Pazos, "Topology design and bandwidth allocation in ATM nets," *IEEE J. on Selected Areas in Communications*, vol. 7, pp. 1253–1262, 1989.

[21] B. Fortz and M. Thorup, "Increasing internet capacity using local search," tech. rep., AT&T Labs Research, Florham Park, NJ 07932, 2000. Preliminary short version of this paper published as "Internet Traffic Engineering by Optimizing OSPF weights" in *Proc. IEEE INFOCOM 2000 – The Conference on Computer Communications*, pp. 519–528.

[22] J. Yee and F. Lin, "A routing algorithm for virtual circuit data networks with multiple sessions per O-D pair," *Networks*, vol. 22, pp. 185–208, 1992.

[23] C. Sung and S. Park, "An algorithm for configuring embedded networks in reconfigurable telecommunication networks," *Telecommunication Systems*, vol. 4, pp. 241–271, 1995.

[24] M. Laguna and F. Glover, "Bandwidth packing: A tabu search approach," *Management Science*, vol. 39, pp. 492–500, 1993.

[25] A. Amiri, E. Rolland, and R. Barkhi, "Bandwidth packing with queueing delay costs: Bounding and heuristic solution procedures," *European Journal of Operational Research*, vol. 112, pp. 635–645, 1999.

[26] L. Resende and M. Resende, "A GRASP for frame relay permanent virtual circuit routing," in *Extended Abstracts of the III Metaheuristics International Conference (MIC'99)* (C. Ribeiro and P. Hansen, eds.), pp. 397–401, 1999.

[27] C.-C. Shyur and U.-E. Wen, "Optimizing the system of virtual paths by tabu search," *European Journal of Operational Research*, vol. 129, pp. 650–662, 2001.

[28] M. Parker and J. Ryan, "A column generation algorithm for bandwidth packing," *Telecommunication Systems*, vol. 2, pp. 185–195, 1994.

[29] L. LeBlanc, J. Chifflet, and P. Mahey, "Packet routing in telecommunication networks with path and flow restrictions," *INFORMS Journal on Computing*, vol. 11, pp. 188–197, 1999.

[30] G. Dahl, A. Martin, and M. Stoer, "Routing through virtual paths in layered telecommunication networks," *Operations Research*, vol. 47, pp. 693–702, 1999.

[31] C. Barnhart, C. Hane, and P. Vance, "Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems," *Operations Research*, vol. 48, pp. 318–326, 2000.

[32] T. Feo and M. Resende, “Greedy randomized adaptive search procedures,” *Journal of Global Optimization*, vol. 6, pp. 109–133, 1995.

[33] J. Bresina, “Heuristic-biased stochastic sampling,” in *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, (Portland), pp. 271–278, 1996.

[34] S. Binato, W. Hery, D. Loewenstern, and M. Resende, “A GRASP for job shop scheduling,” in *Essays and Surveys in Metaheuristics* (C. Ribeiro and P. Hansen, eds.), pp. 59–79, Kluwer Academic Publishers, 2002.

[35] F. Glover, “Tabu search and adaptive memory programing – Advances, applications and challenges,” in *Interfaces in Computer Science and Operations Research* (R. Barr, R. Helgason, and J. Kennington, eds.), pp. 1–75, Kluwer, 1996.

[36] F. Glover, “Multi-start and strategic oscillation methods – Principles to exploit adaptive memory,” in *Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research* (M. Laguna and J. Gonzáles-Velarde, eds.), pp. 1–24, Kluwer, 2000.

[37] F. Glover and M. Laguna, *Tabu Search.* Kluwer, 1997.

[38] F. Glover, M. Laguna, and R. Martí, “Fundamentals of scatter search and path relinking,” *Control and Cybernetics*, vol. 39, pp. 653–684, 2000.

[39] M. Laguna and R. Martí, “GRASP and path relinking for 2-layer straight line crossing minimization,” *INFORMS Journal on Computing*, vol. 11, pp. 44–52, 1999.

[40] R. Aiex, M. Resende, P. Pardalos, and G. Toraldo, “GRASP with path-relinking for the three-index assignment problem,” tech. rep., AT&T Labs-Research, 2000.

[41] S. Canuto, M. Resende, and C. Ribeiro, “Local search with perturbations for the prize-collecting Steiner tree problem in graphs,” *Networks*, vol. 38, pp. 50–58, 2001.

[42] C. Ribeiro, E. Uchoa, and R. Werneck, “A hybrid GRASP with perturbations for the Steiner problem in graphs,” *INFORMS Journal on Computing*, vol. 14, pp. 228–246, 2002.

[43] E. Zegura, “GT-ITM: Georgia Tech internetwork topology models (software),” tech. rep., Georgia Institute of Technology, 1996. (Online document at `http://www.cc.gatech.edu/fac/Ellen.Zegura/gt-itm/gt-itm.tar.gz`).

[44] K. Calvert and M. D. E. Zegura, “Modeling Internet topology,” *IEEE Communications Magazine*, vol. 35, pp. 160–163, 1997.

[45] E. Zegura, K. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Proceedings of 15th IEEE Conf. on Computer Communications (INFOCOM)*, pp. 594–602, 1996.

[46] B. Waxman, "Routing of multipoint connections," *IEEE Journal Selected Areas in Communications*, vol. 6, pp. 1617–1622, 1998.

[47] R. Aiex, M. Resende, and C. Ribeiro, "Probability distribution of solution time in GRASP: An experimental investigation," *Journal of Heuristics*, vol. 8, pp. 343–373, 2002.

[48] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz, "Characterizing the Internet hierarchy from multiple vantage points," in *Proc. 21st IEEE Conf. on Computer Communications (INFOCOM 2002)*, vol. 2, pp. 618–627, 2002.

[49] B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional IP routing protocols," *IEEE Communications Magazine*, pp. 118–124, October 2002.

[50] Internet Engineering Task Force, "Ospf version 2," Tech. Rep. RFC 1583, Network Working Group, 1994.

[51] J. Moy, *OSPF, Anatomy of an Internet Routing Protocol.* Addison-Wesley, 1998.

[52] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True, "Deriving traffic demands for operational IP networks: Methodology and experience," *IEEE/ACM Transactions on Networking*, vol. 9, pp. 265–279, 2001.

[53] M. Rodrigues and K. Ramakrishnan, "Optimal routing in data networks," 1994. Presentation at International Telecommunication Symposium (ITS).

[54] F. Lin and J. Wang, "Minimax open shortest path first routing algorithms in networks supporting the smds services," in *Proc. IEEE International Conference on Communications (ICC)*, vol. 2, pp. 666–670, 1993.

[55] A. Bley, M. Grötchel, and R. Wessläy, "Design of broadband virtual private networks: Model and heuristics for the B-WiN," Tech. Rep. SC 98-13, Konrad-Zuse-Zentrum fur Informationstecknik Berlin, 1998. To appear in *Proc. DIMACS Workshop on Robust Communication Network and Survivability, AMS-DIMACS Series.*

[56] M. Ericsson, M. Resende, and P. Pardalos, "A genetic algorithm for the weight setting problem in OSPF routing," *Journal of Combinatorial Optimization*, vol. 6, pp. 299–333, 2002.

[57] A. Sridharan, R. Guérin, and C. Diot, "Achieving Near-Optimal Traffic Engineering Solutions for Current OSPF/IS-IS Networks," Sprint ATL Technical Report TR02-ATL-022037, Sprint Labs, Feb. 2002.

[58] J. C. Bean, "Genetic algorithms and random keys for sequencing and optimization," *ORSA J. on Comp.*, vol. 6, pp. 154–160, 1994.

[59] G. Ramalingam and T. Reps, "An incremental algorithm for a generalization of the shortest-path problem," *J. of Algorithms*, vol. 21, pp. 267–305, 1996.

[60] D. Frigioni, A. Marchetti-Spaccamela, and U. Nanni, "Fully dynamic output-bounded single source shortest-paths problem," in *Proc. ACM-SIAM Symp. Discrete Algorithms*, pp. 212–221, 1996.

[61] D. Frigioni, M. Ioffreda, U. Nanni, and G. Pasqualone, "Experimental analysis of dynamic algorithms for the single source shortest path problem," *ACM J. of Exp. Alg.*, vol. 3, 1998. article 5.

[62] L. Buriol, M. Resende, and M. Thorup, "Speeding up dynamic shortest path algorithms," tech. rep., AT&T Labs Research, 180 Park Avenue, Florham Park, NJ 07932 USA, 2003.

[63] M. Matsumoto and T. Nishimura, "Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator," *ACM Transactions on Modeling and Computer Simulation*, vol. 8, no. 1, pp. 3–30, 1998.

[64] Cisco, *Configuring OSPF.* Cisco Press, 1997.

[65] T. Thomas II, *OSPF Network Design Solutions.* Cisco Press, 1998.

[66] R. Aiex, M. Resende, and C. Ribeiro, "Probability distribution of solution time in GRASP: An experimental investigation," *Journal of Heuristics*, vol. 8, pp. 343–373, 2002.

[67] M. Resende and C. Ribeiro, "Greedy randomized adaptive search procedures," in *Handbook of Metaheuristics* (F. Glover and G. Kochenberger, eds.), pp. 219–249, Kluwer Academic Publishers, 2003.

Evolutionary Algorithms, Multiobjective Optimization, and Applications

Eckart Zitzler[*]

Abstract

This paper gives an introduction into evolutionary computation, in particular in the light of multiobjective optimization, and demonstrates how evolutionary algorithms can be used to tackle a highly demanding application in telecommunications, namely the design of a network processor.

1 Introductory Example

To illustrate the basic principles of multiobjective optimization and evolutionary algorithms, consider the following example: given is a set of items together with a profit and a weight associated with each item; the goal is to determine a subset of items such that the overall profit, i.e., the sum of the profits of the selected items, is maximum, while the overall weight, i.e., the sum of the weights of the selected items, is minimal. This problem is generally denoted as knapsack problem.

Now assume that four items are available: a camera (weight = 750, profit = 5), a thermos flask (weight = 1500, profit = 8), a pocket knife (weight = 300, profit = 7), and a book (weight = 1000, profit = 3). The set of all possible selections contains 16 possible subsets (cf. Fig. 1), in general 2^n where n is the number of items. In this context, two observations can be made: there is no single optimal selection of items, and some subsets are better than others. With respect to the first issue, the empty subset minimizes the overall weight, while the set containing all items maximizes the overall profit. We say the two optimization criteria are conflicting. Nevertheless, subsets for which there exists another subset that is better in at least one criterion, while not being worse in the other criterion, can be neglected. As a consequence, a set of optimal trade-offs emerges as shown in Fig. 1 at the bottom. At the end, though, we are interested in a single solution, and therefore a decision making process is necessary: which of the optimal trade-offs represents the best compromise for our needs?

In practice, the search for optimal solutions by using an appropriate optimization algorithm and the decision making process can be integrated in different ways. One possibility

[*] TIK Swiss Federal Instutite of Technology. E-mail:zitzler@tik.ee.ethz.ch

is to aggregate the multiple optimization criteria into a single one. That means the decision is made before the search. In our example, one could transform the second objective into

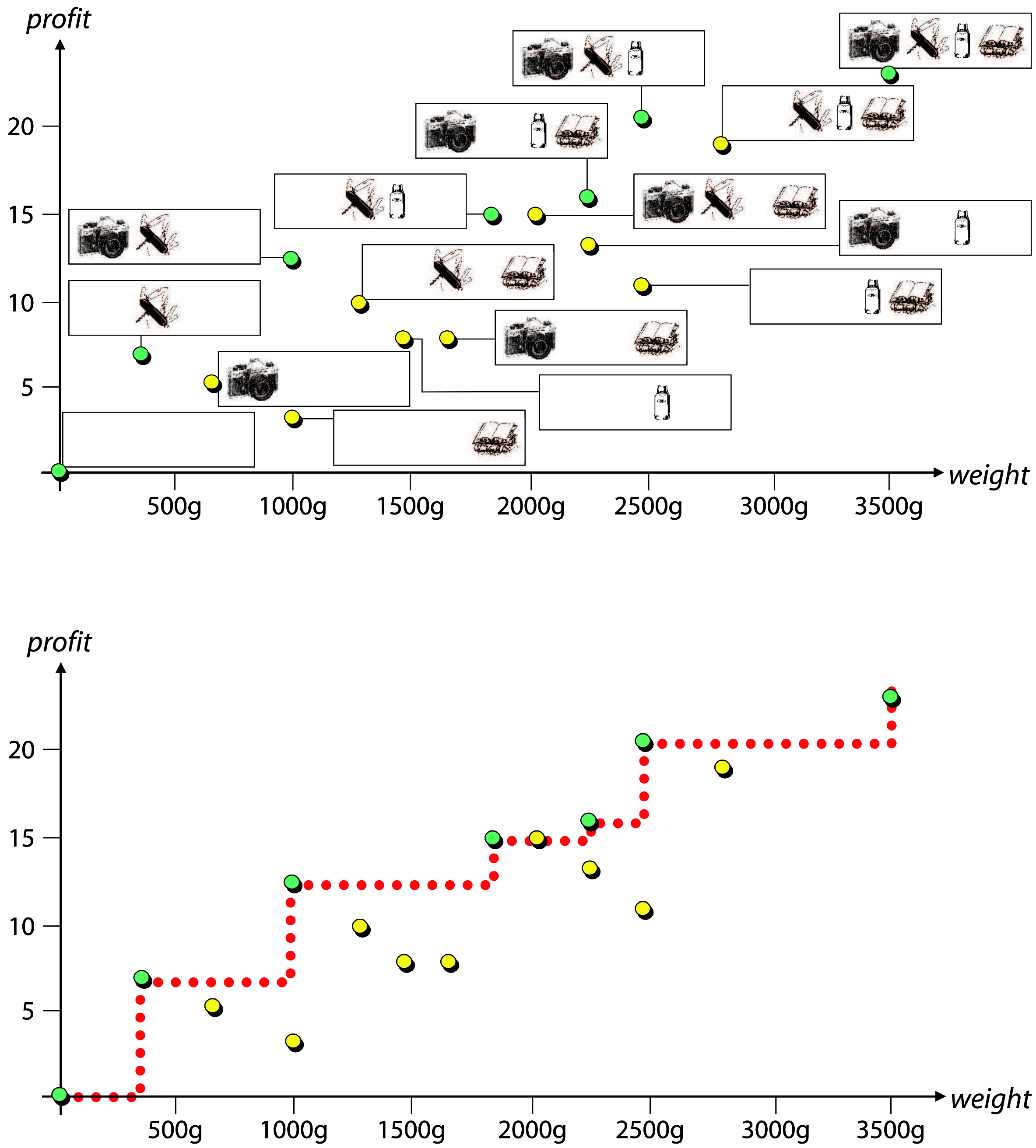


Figure 1: Illustration of the search space for a simple knapsack problem instance. At the bottom, the dark-shaeded solutions connected by the dotted line represent the optimal trade-offs.

a constraint and look for the selection with maximum profit that does not exceed a given weight bound. Alternatively, we can first search for all optimal trade-offs and then choose one solution out of them. In this case, decision making is done after the search, which is especially useful if little is known about the underlying problem.

Although being simple regarding the problem formulation, the above knapsack problem reflects two problem difficulties that arise in many real-world applications: i) the set of possible solutions is large, and ii) multiple, competing optimization criteria are involved. Thus, efficient search strategies are required that are able to deal with both difficulties.

Evolutionary algorithms possess several characteristics that are desirable in this context. The term evolutionary algorithm (EA) stands for a class of randomized search strategies that simulate the process of natural evolution. The origins of EAs can be traced back to the late 1950s, and since the 1970s several evolutionary methodologies have been proposed, mainly genetic algorithms, evolutionary programming, and evolution strategies [1]. All of these approaches operate on a set of candidate solutions. Using strong simplifications, this set is subsequently modified by two basic principles: selection and variation. While selection mimics the competition for reproduction and resources among living beings, the other principle, variation, imitates the natural capability of creating "new" living beings by means of recombination and mutation. Although the underlying mechanisms are simple, these algorithms have proven themselves as a general, robust and powerful search mechanism [1].

In this lecture, it will be discussed how evolutionary algorithms work, how they can be tailored to a problem at hand, and how they can be used to tackle a complex application in telecommunications, namely the design of a network processor.

2 Optimization and Randomized Search Algorithms

2.1 Basic Terms

The scenario considered in this paper involves an arbitrary optimization problem with k objectives, which are, without loss of generality, all to be maximized and all equally important, i.e., no additional knowledge about the problem is available. We assume that a solution to this problem can be described in terms of a *decision vector* $(x_1, x_2, \ldots, x_n)$ in the *decision space* $\boldsymbol{X}$. A function $\boldsymbol{f} : \boldsymbol{X} \rightarrow \boldsymbol{Y}$ evaluates the quality of a specific solution by assigning it an *objective vector* $(y_1, y_2, \ldots, y_k)$ in the *objective space* $\boldsymbol{Y}$ (cf. Fig. 2).

Now, let us suppose that the objective space is a subset of the real numbers, i.e., $\boldsymbol{Y} \subseteq \mathbb{R}$, and that the goal of the optimization is to maximize the single objective. In such a single-objective optimization problem, a solution $\boldsymbol{x}^1 \in \boldsymbol{X}$ is better than another solution $\boldsymbol{x}^2 \in \boldsymbol{X}$ if $\boldsymbol{y}^1 > \boldsymbol{y}^2$ where $\boldsymbol{y}^1 = \boldsymbol{f}(\boldsymbol{x}^1)$ and $\boldsymbol{y}^2 = \boldsymbol{f}(\boldsymbol{x}^2)$. Although several optimal solutions may exist in decision space, they are all mapped to the same objective vector, i.e., there exists only a single optimum in objective space.

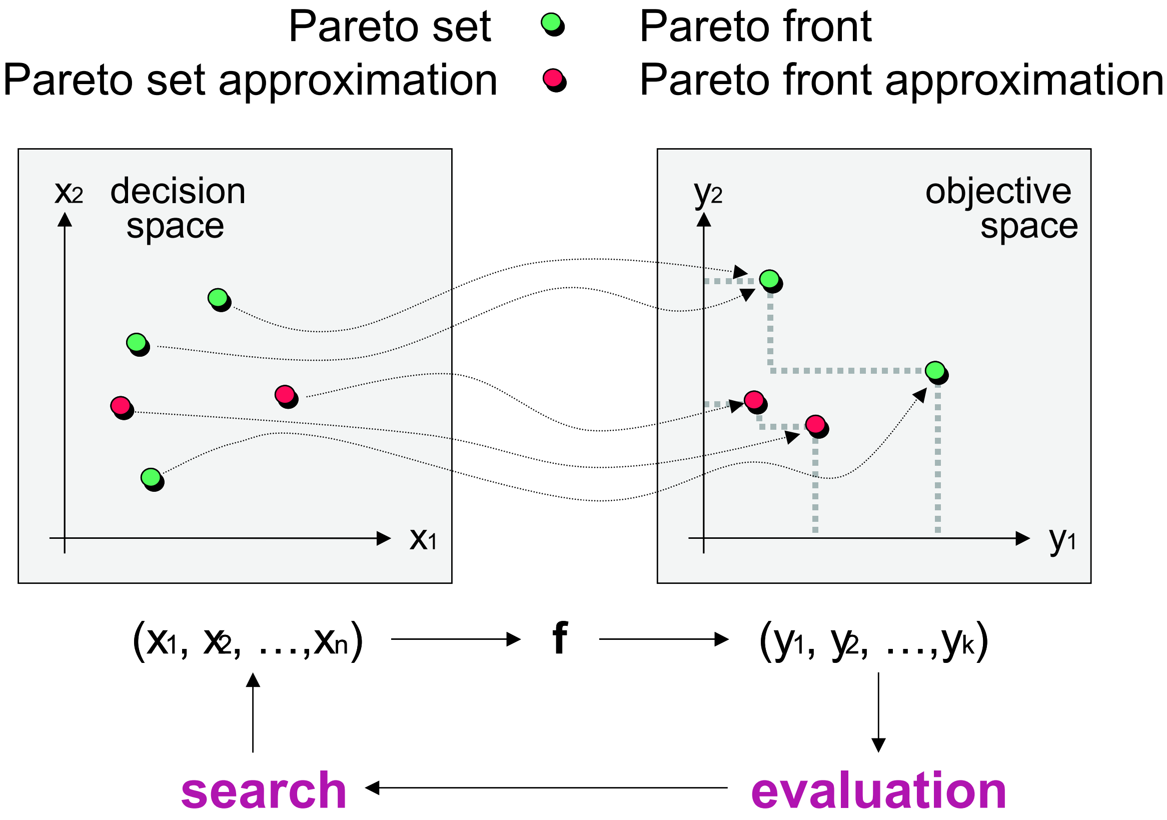


Figure 2: Illustration of a general (multiobjective) optimization problem

In the case of a vector-valued evaluation function $\boldsymbol{f}$ with $\boldsymbol{Y} \subseteq \mathbb{R}^k$ and $k > 1$, the situation of comparing two solutions $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$ is more complex. Following the well-known concept of Pareto dominance, an objective vector $\boldsymbol{y}^1$ is said to *dominate* another objective vectors $\boldsymbol{y}^2$ ($\boldsymbol{y}^1 \succ \boldsymbol{y}^2$) if no component of $\boldsymbol{y}^1$ is smaller than the corresponding component of $\boldsymbol{y}^2$ and at least one component is greater. Accordingly, we can say that a solution $\boldsymbol{x}^1$ is better than another solution $\boldsymbol{x}^2$, i.e., $\boldsymbol{x}^1$ *dominates* $\boldsymbol{x}^2$ ($\boldsymbol{x}^1 \succ \boldsymbol{x}^2$), if $\boldsymbol{f}(\boldsymbol{x}^1)$ dominates $\boldsymbol{f}(\boldsymbol{x}^2)$. Here, optimal solutions, i.e., solutions not dominated by any other solution, may be mapped to different objective vectors. In other words: there may exist several optimal objective vectors representing different trade-offs between the objectives.

The set of optimal solutions in the decision space $\boldsymbol{X}$ is in general denoted as the *Pareto set* $\boldsymbol{X}^* \subseteq \boldsymbol{X}$, and we will denote its image in objective space as *Pareto front* $\boldsymbol{Y}^* = \boldsymbol{f}(\boldsymbol{X}^*) \subseteq \boldsymbol{Y}$. With many multiobjective optimization problems, knowledge about this set helps the decision maker in choosing the best compromise solution. For instance, when designing telecommunication systems, engineers often perform a so-called design space exploration to learn more about the Pareto set. Thereby, the design space is reduced to the set of optimal trade-offs: a first step in selecting an appropriate system implementation.

In the following, we will assume that the goal of the optimization process is to find or approximate the Pareto set (in the case of a single objective, the Pareto front consists of a single objective vector only). Therefore, the outcome of an algorithm is considered to be a set of mutually nondominated solutions, or *Pareto set approximation* for short.

2.2 Blackbox Optimization

Randomized search algorithms form a class of heuristics that aim at finding good solutions to the optimization problem at hand without investigating all solutions. Different types of randomized search algorithms have been proposed such as evolutionary algorithms and simulated annealing, and they are characterized by the fact that minimal requirements with respect to the objective functions are made, i.e., they have been designed for so-called blackbox optimization scenarios. A blackbox optimization scenario assumes that nothing is known about the optimization criteria. Each objective function is considered as a black box, and the only way to obtain information about it is by asking for the objective vector to which a particular decision vector is mapped to.

In general, a randomized search algorithm works as follows. First, a decision vector $\boldsymbol{x}^1$ is chosen at random, and the corresponding objective vector $\boldsymbol{y}^1 = \boldsymbol{f}(\boldsymbol{x}^1)$ is determined by using the black boxes for the objective functions. In the next step, another solution $\boldsymbol{x}^2$ is selected randomly on the basis of the information given by $\boldsymbol{x}^1$ and $\boldsymbol{y}^1$. This process is repeated many times, where in iteration t all the previously investigated solutions $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^{t-1}$ and its objective function values $\boldsymbol{y}^1, \ldots, \boldsymbol{y}^{t-1}$ can be used to select the next decision vector $\boldsymbol{x}^t$. The algorithm terminates until a certain termination criterion is fulfilled. In practice, though, randomized search algorithms do not store all previously considered solutions but rather only keep the best ones. Local search strategies, the Metropolis algorithm, or simulated annealing, for instance, only store one solution, while evolutionary algorithms usually work with a population of solutions. Furthermore, the different variants of randomized search algorithms used in practice distinguish themselves by the way new solutions are generated and by the criteria on the basis of which the solutions to be kept in the memory are selected.

In this context, we may ask why to use randomized search algorithms and which variant of randomized search algorithms to use. As to the first question, we often do not have sufficient time resources or insight into the problem to design a problem-specific algorithm, or the problem is too complex to be solved by exact methods. Since randomized search algorithms have minimum requirements with respect to the objective functions, they can be useful tools to tackle hard optimization problems. The second question is more difficult to answer. The No-Free-Lunch (NFL) theorem states that over all possible problems all search algorithms have the same average performance [2]. This makes clear that we cannot expect to find the perfect search method that outperforms all other techniques. One rather tries to identify classes of problems for which particular algorithms are well suited for. Evolutionary algorithms, e.g., are for practical reasons a good choice in a multiobjective scenario as the Pareto set can be approximated in a single optimization run.

3 Design Issues in Evolutionary Computation

In contrast to other randomized search algorithms, an evolutionary algorithm is characterized by three features:

1. a set of solution candidates is maintained,

2. a selection process is performed on this set which determines which solutions are considered to generate new solutions, and

3. several solutions may be combined in terms of recombination to generate new solutions.

By analogy to natural evolution, the solution candidates are called *individuals* and the set of solution candidates is called the *population*. Each individual represents a possible solution, i.e., a decision vector, to the problem at hand; however, an individual *is not* a decision vector but rather encodes it based on an appropriate representation.

At the beginning, the population is filled with a certain number of randomly chosen individuals. Each of these individuals is then evaluated on the basis of the objective functions and is assigned a scalar value, the fitness value, which reflects its quality. Afterwards, a selection process is performed in which high-quality individuals are chosen for the generation of new individuals. Variation, the process of creating new solutions, is usually implemented on the basis of two operators: recombination and mutation. While recombination assembles a new solution by combining two or several individuals, mutation creates new individuals by slightly modifying single individuals. Finally, there is another selection procedure, environmental selection, which determines which of the old individuals and newly generated ones are kept in memory, i.e., in the new population. The steps fitness evaluation, mating selection, recombination, mutation, and environmental selection form one iteration of the algorithm, which is called *generation*. The number of iterations to be executed can be defined beforehand or may depend on on other conditions, e.g., stagnation in the population or existence of an individual with sufficient quality. The general flow of an EA is shown in Fig. 3.

In the following, we will briefly discuss the different issues arising when tailoring an EA to a specific problem, namely representation of the solution space, fitness assignment, selection, and variation.

3.1 Representation

With many optimization problems, the decision space has a straight-forward representation on a computer, e.g., for the knapsack problem a subset can be encoded by a binary bitstring

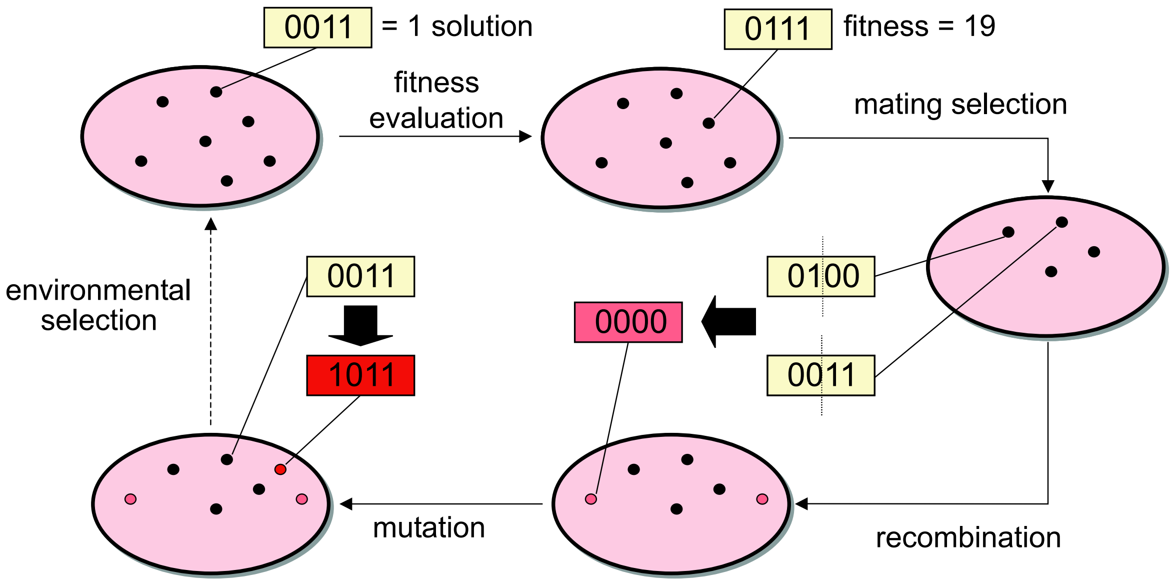


Figure 3: Outline of a general evolutionary algorithm for a problem with four binary decision variables

where each position is associated with a particular item. For other problems, though, an appropriate encoding has to be defined, e.g., for graph problems (network topologies), scheduling problems, symbolic regression, etc. The choice of the representation is often underestimated, although it influences the performance of the algorithm; this holds for randomized search algorithms in general.

Most commonly used are vector representations (binary, integer, real), where the elements represent atomic units that are modified as a whole in the variation process. While vectors are usually of fixed length, tress can be used to encode solutions of variable length such as symbolic expressions and programs. Genetic programming, a subbranch of evolutionary computation, is devoted to EAs using tree representations. Many other representations such as matrices are possible, in particular mixed encoding can be often found with many real-world applications.

What the optimal choice of an encoding for a given problem is also depends on the variation operators; however, in general and as a rule of thumb, a representation should be

- *complete*, i.e., for each potential solution a corresponding encoding exists,
- *one-to-one*, i.e., each solution has a unique encoding, and if not at least
- *uniform*, i.e., each solution is represented by the same number of possible encodings,
- *feasible*, i.e., each possible encoding is mapped to a feasible solution, and

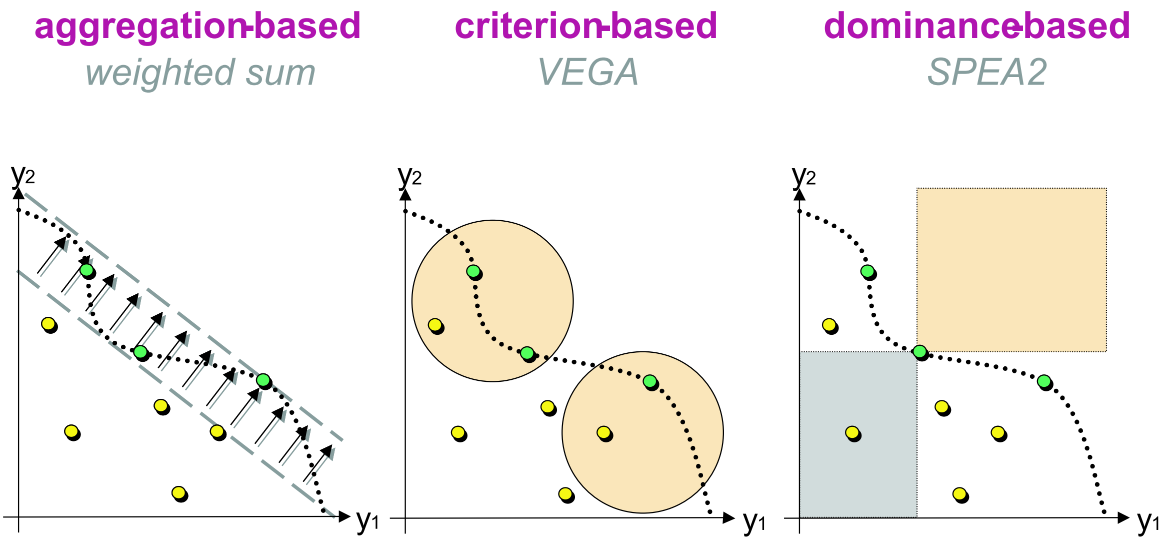


Figure 4: Different fitness assignment strategies

- *locality preserving*, i.e., the distance between two encoded solutions is the same as between the two decoded solutions with respect to appropriate metrics.

These criteria only can serve as guidelines and often not all of them can be fulfilled.

3.2 Fitness Assignment

The fitness of an individual describes its quality with regard to the optimization task under consideration on the basis of a real number (there may be exceptions, though). In the case of a single objective, often the objective function value is taken as the fitness value. However, there are different situations which require more complex fitness assignment strategies:

- multiple objectives are involved and the aim is to approximate the Pareto set,
- multiple optima are sought,
- constraints divide the decision space into feasible and infeasible solutions.

In the following, we will deal with each of these aspects separately.

3.2.1 Multiple Objectives

In the presence of multiple optimization criteria, the question is how to assign scalar fitness values such that the population is guided towards the Pareto set. There are different approaches, the major three are aggregation-based, criterion-based, and Pareto-based fitness assignment strategies, cf. Fig 7.

One approach which is built on the traditional techniques for generating trade-off surfaces is to aggregate the objectives into a single parameterized objective function. The parameters of this function are systematically varied during the optimization run in order to find a set of nondominated solutions instead of a single trade-off solution. For instance, some EA implementations use weighted-sum aggregation, where the weights represent the parameters which are changed during the evolution process [3, 4].

Criterion-based methods switch between the objectives during the selection phase. Here, the fitness of an individual is identical to the objective vector, i.e., it is not a scalar value. Each time an individual is chosen for reproduction, potentially a different objective will decide which member of the population will be selected for variation. For example, Schaffer [5] proposed to divide the mating selection phase into k phases, where at phase i the individuals are chosen according to objective i; at each phase, the same number of individuals is selected. In contrast, Kursawe [6] suggested assigning a probability to each objective which determines whether the objective will be the sorting criterion in the next selection step—the probabilities can be user-defined or chosen randomly over time.

The idea of calculating an individual's fitness on the basis of Pareto dominance goes back to Goldberg [7], and different ways of exploiting the partial order on the population have been proposed. Some approaches use the dominance rank, i.e., the number of individuals by which an individual is dominated, to determine the fitness values [8]. Others make use of the dominance depth; here, the population is divided into several fronts and the depth reflects to which front an individual belongs to [9, 10]. Alternatively, also the dominance count, i.e., the number of individuals dominated by a certain individual, can be taken into account. For instance, SPEA [11] and SPEA2 [12] assign fitness values on the basis of both dominance rank and count. Independent of the technique used, the fitness is related to the whole population in contrast to aggregation-based methods which calculate an individual's raw fitness value independently of other individuals.

3.2.2 Multiple Optima and Diversity Preservation

In the presence of multiple optimization criteria, we are often interested in finding the Pareto-optimal solutions. With many applications, though, this goal cannot achieved, and instead the aim is to generate a well-distributed subset of the Pareto set. A similar situation may arise in single-objective optimization, if we would like to find multiple different optima that all have the same objective function value. The problem, however, is a phenomenon known in Biology as *genetic drift*. Genetic drift denotes random changes in allele frequencies (alleles are the different "values" a gene can take) due to sampling errors in finite and particularly small populations. Here, we mean the tendency of small populations to converge to a single optimal solution.

If no specific means are incorporated, the diversity within the population may be lost due to genetic drift. One way to circumvent this problem is to incorporate density information into the fitness such that an individual's chance of being selected is decreased the greater the density of individuals in its neighborhood. This issue is closely related to the estimation of probability density functions in statistics, and the methods used in EAs can be classified according to the categories for techniques in statistical density estimation [13].

Kernel methods [13] define the neighborhood of a point in terms of a so-called Kernel function K which takes the distance to another point as an argument. In practice, for each individual the distances $d_{\boldsymbol{i}}$ to all other individuals $\boldsymbol{i}$ are calculated and after applying K the resulting values $K(d_{\boldsymbol{i}})$ are summed up. The sum of the K function values represents the density estimate for the individual. Fitness sharing is the most popular technique of this type within the field of evolutionary computation, which is used, e.g., in MOGA [8], NSGA [9], and NPGA [14].

Nearest neighbor techniques [13] take the distance of a given point to its kth nearest neighbor into account in order to estimate the density in its neighborhood. Usually, the estimator is a function of the inverse of this distance. SPEA2 [12], for instance, makes use of this density estimation technique as will be discussed in Section 4.

Histograms [13] define a third category of density estimators that use a hypergrid to define neighborhoods within the space. The density around an individual is simply estimated by the number of individuals in the same box of the grid. The hypergrid can be fixed, though usually it is adapted with regard to the current population as, e.g., in PAES [15].

Each of the three approaches is visualized in Fig. 5. However, due to space-limitations, a discussion of strengths and weaknesses of the various methods cannot be provided here—the interested reader is referred to Silverman's book [13]. Furthermore, note that all of the above methods require a distance measure which can be defined on the encoded decision vectors, on the decoded decision vectors, or on the objective vectors. Most approaches consider the distance between two individuals as the distance between the two corresponding objective vectors.

3.2.3 Constraint Handling

With many applications, constraints restrict the set of admissible solutions, and we are interested in the solutions that meet these constraints *and* are optimal with respect to all other feasible solutions.

Assume that we have a set of inequality constraints $\boldsymbol{g}_1 \geq 0, \boldsymbol{g}_2 \geq 0, \ldots, \boldsymbol{g}_l \geq 0$ (other constraints can easily be expressed in this manner). The question we consider in the following is how to handle constraints within an EA such that the search focuses on the feasible solutions. In principle, there are three different ways:

- Firstly, the representation can be chosen such that the decoder function maps each individual to a feasible solutions.

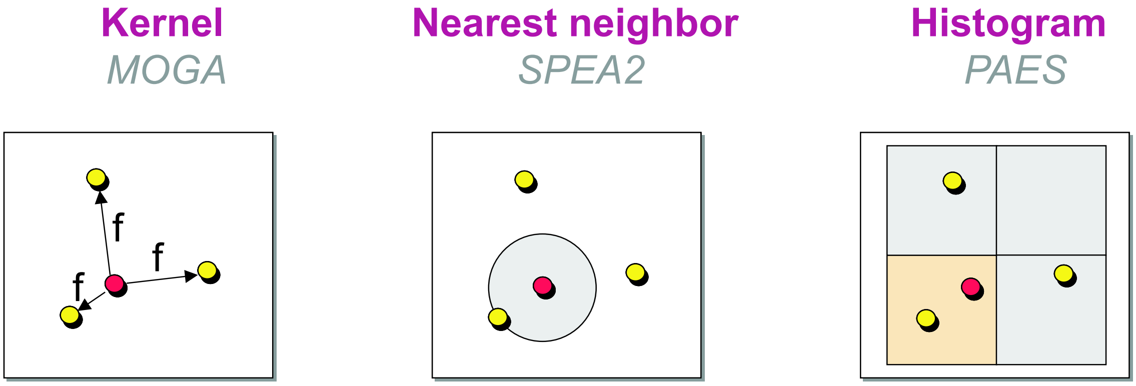


Figure 5: Illustration of diversity preservation techniques

- Secondly, we can make sure that only feasible solutions are generated. To this end, the initialization procedure of the population as well as the variation operators need to be designed accordingly.

- Most general is the penalty approach where the fitness of individuals violating any constraints is diminished. Usually, an overall constraint violation $C_{\boldsymbol{i}} \in \mathbb{R}$ is computed as

  $$C_{\boldsymbol{i}} = \sum_{p=0}^{l} |\min\{\boldsymbol{g}_p(\boldsymbol{i}), 0\}|$$

  Provided that fitness is to be minimized, the fitness $F_{\boldsymbol{i}}$ of an individual can be considered as the sum of the original fitness $F'_{\boldsymbol{i}}$ and the overall constraint violation: $F_{\boldsymbol{i}} = F'_{\boldsymbol{i}} + C_{\boldsymbol{i}}$. Another possibility is to calculate the fitness in the following way, if we know the worst fitness value $F_{\max}$ possible for a feasible solution:

  $$F_{\boldsymbol{i}} = \begin{cases} F'_{\boldsymbol{i}} & \text{if } C_{\boldsymbol{i}} = 0 \\ F_{\max} + C_{\boldsymbol{i}} & \text{else} \end{cases}$$

  The latter method ensures that feasible solutions are alway preferred over infeasible solutions.

Actually, only the last approach is directly related to fitness assignment, the other two are just mentioned here for reasons of completeness. Moreover, further possibilities emerge in multiobjective optimization, e.g., some authors have suggested a modified definition of Pareto dominance that takes constraints into account [16, 17].

3.3 Selection

Selection, which can be divided into mating and environmental selection, decides which individuals are considered for variation and which individuals survive, i.e., are kept in memory. It serves two conflicting goals: exploitation and exploration. The first term means we are trying to generate better solutions by modifying the current best solutions; in this sense, selection should favor the best individuals in the population. On the other hand, keeping diversity in the population is advantageous in avoiding to get stuck in local optima; in this respect, selection should focus on the diversity of the chosen individuals.

3.3.1 Mating Selection

Mostly, mating selection is implemented in terms of a randomized selection procedure. In this context, one has to distinguish two phases: sampling rate assignment and sampling. In the first phase, each individual is assigned a probability of being selected. The second phase realizes the actual selection, i.e., a predefined number of individuals is chosen on the basis of the sampling rates.

In the literature, different sampling rate assignment schemes have been proposed. Evolution strategies, e.g., assign each individual the same probability, while genetic algorithms traditionally used a fitness proportionate scheme. Fitness proportionate means the sampling rate is set to the ratio of an individual's fitness divided by the sum of the fitness values of all individuals in the population. The disadvantage of this scheme is that adding a constant to all fitness values results in different sampling rates; the larger the constant, the more likely it is that all individuals have similar sampling rates. Rank-based schemes avoid this problem by first sorting the individual according to the fitness values, and afterwards assigning the sampling rates in dependence of the position within the resulting order.

As to sampling, there are two main methods. Both methods can be best illustrated by thinking of a roulette wheel which is divided into N parts where N is the number of individuals in the population. The size of the slot associated with individual $\boldsymbol{i}$ is in proportion to its sampling rate $C_{\boldsymbol{i}}$. The first technique, known as roulette wheel reproduction, simply spins the roulette wheel as many times as individuals need to be selected; each time that individual that is associated with the slot under the pointer is selected. In contrast, with stochastic universal sampling (SUS) the roulette wheel is spun only once; instead N pointers are distributed evenly spaced around the roulette wheel. Each pointer determines one individual for selection. The advantage of SUS over roulette wheel reproduction is the lower variance with respect to the selected individuals.

Finally, tournament selection integrates sampling rate assignment and sampling in the same procedure. A certain number of individuals is chosen uniformly from the population, and the individual with the best fitness within this group is selected. This process is iterated until the predefined number of individuals has been selected.

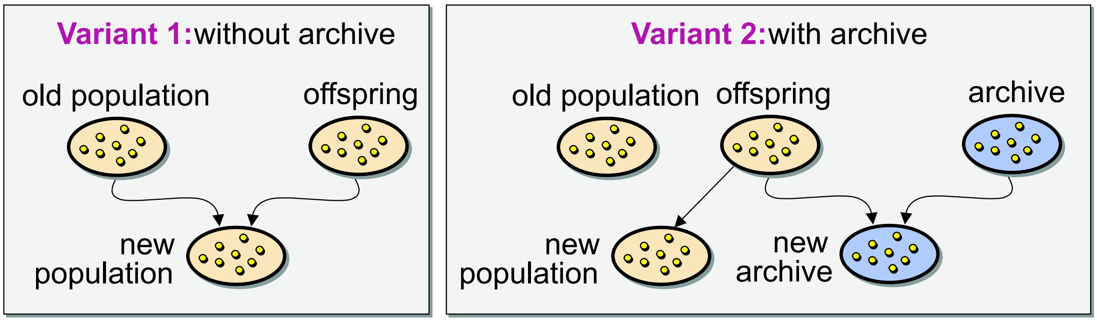


Figure 6: Two possible ways to implement environmental selection in a multiobjective EA

3.3.2 Environmental Selection

Environmental selection determines which of the individuals are kept in the population and is usually realized via a deterministic algorithm. One strategy is to replace the old population by the set of individuals that have been generated using mating selection and variation. Alternatively, parents and offspring can be combined and afterwards the best N individuals from the union form the next population, where N is the population size. Especially in the presence of multiple objectives, it is important to choose an appropriate environmental selection scheme as we would like to prevent nondominated individuals from being lost due to random effects. Therefore, the discussion will focus on multiobjective optimization in the following.

The two fundamental approaches used in multiobjective EAs are depicted in Fig. 6. The first one corresponds to the aforementioned strategy of combining parent and offspring population and to take the best N individuals. Alternatively, a secondary population, the so-called archive, can be maintained to which promising solutions in the population are copied at each generation. The archive may just be used as an external storage separate from the optimization engine or may be integrated into the EA by including archive members in the selection process.

As the memory resources are usually restricted, with both variants criteria have to be defined on this basis of which the solutions to be kept are selected. The dominance criterion is most commonly used. If an archive is maintained, the archive comprises only the current approximation of the Pareto set, i.e., dominated archive members are removed. Otherwise, special care is taken to ensure that nondominated solutions are preferred to dominated ones. However, the dominance criterion is in general not sufficient (e.g., for continuous problems the Pareto set may contain an infinite number of solutions); therefore, additional information is taken into account to reduce the number of stored solutions further. Examples are density information [11, 15] and the time that has passed since the individual entered the archive [18].

Most multiobjective EAs make use of a combination of dominance and density to choose the individuals that will be kept in the archive at every generation. However, these approaches may suffer from the problem of deterioration, i.e., solutions contained in the archive at generation t may be dominated by solutions that were members of the archive at any generation $t' < t$ and were discarded later. Recently, Laumanns et al. [19] presented an archiving strategy which avoids this problem and guarantees to maintain a diverse set of Pareto-optimal solutions (provided that the optimization algorithm is able to generate the Pareto-optimal solutions).

3.4 Variation

Variation aims at generating new individuals on the basis of those individuals that were chosen during the mating selection phase. While mutation creates a new solution by modifying a given one, the recombination operator takes two or more individuals, combines them in a randomized fashion, and outputs one or more offspring. Since the choice of the operators is strongly problem-dependent and many different variation procedures have been suggested, we will only sketch the underlying ideas assuming a bitvector representation.

With binary vectors, mutation is usually implemented by flipping each bit independently with a predefined mutation probability p_m; a standard setting is $p_m = 1/n$, where n is the number of bits, such that in average one bit is flipped per individual. As to recombination, a popular operator is one-point or, in general, N-point crossover. The two parents are cut at randomly chosen positions into $N + 1$ parts, and afterwards children a created by alternately choosing parts from the first and from the second parent. As with mutation, there is a crossover probability associated with this operator. With probability p_c, the two parents are recombined and the two resulting children are returned; otherwise, two copies of the parents are returned.

4 An Example Evolutionary Algorithm: SPEA2

As an illustrative example, we here present a generic implementation of a multiobjective EA, namely SPEA2 [12], that has been used to tackle the network processor application discussed in the lecture. The overall algorithm is as follows:

Algorithm 1 (SPEA2 Main Loop)

Input:	N	*(population size)*
	$\overline{N}$	*(archive size)*
	T	*(maximum number of generations)*
Output:	$\boldsymbol{A}$	*(nondominated set)*

Step 1: **Initialization**: *Generate an initial population* $\boldsymbol{P}_0$ *and create the empty archive (external set)* $\overline{\boldsymbol{P}}_0 = \emptyset$. *Set* $t = 0$.

Step 2: **Fitness assignment**: *Calculate fitness values of individuals in* $\boldsymbol{P}_t$ *and* $\overline{\boldsymbol{P}}_t$ *(cf. Section 4.1).*

Step 3: **Environmental selection**: *Copy all nondominated individuals in* $\boldsymbol{P}_t$ *and* $\overline{\boldsymbol{P}}_t$ *to* $\overline{\boldsymbol{P}}_{t+1}$. *If size of* $\overline{\boldsymbol{P}}_{t+1}$ *exceeds* $\overline{N}$ *then reduce* $\overline{\boldsymbol{P}}_{t+1}$ *by means of the truncation operator, otherwise if size of* $\overline{\boldsymbol{P}}_{t+1}$ *is less than* $\overline{N}$ *then fill* $\overline{\boldsymbol{P}}_{t+1}$ *with dominated individuals in* $\boldsymbol{P}_t$ *and* $\overline{\boldsymbol{P}}_t$ *(cf. Section 4.2).*

Step 4: **Termination**: *If* $t \geq T$ *or another stopping criterion is satisfied then set* $\boldsymbol{A}$ *to the set of decision vectors represented by the nondominated individuals in* $\overline{\boldsymbol{P}}_{t+1}$. *Stop.*

Step 5: **Mating selection**: *Perform binary tournament selection with replacement on* $\overline{\boldsymbol{P}}_{t+1}$ *in order to fill the mating pool.*

Step 6: **Variation**: *Apply recombination and mutation operators to the mating pool and set* $\boldsymbol{P}_{t+1}$ *to the resulting population. Increment generation counter* ($t = t + 1$) *and go to Step 2.*

SPEA2 uses a fine-grained fitness assignment strategy which incorporates density information as will be described in Section 4.1. Furthermore, an archive of fixed size is maintained that contains a representation of the current nondominated front. Whenever the number of nondominated individuals is less than the predefined archive size, the archive is filled up by dominated individuals; otherwise, a truncation method is applied which is described in Section 4.2.

4.1 Fitness Assignment

To avoid the situation that individuals dominated by the same archive members have identical fitness values, with SPEA2 for each individual both dominating and dominated solutions are taken into account. In detail, each individual $\boldsymbol{i}$ in the archive $\overline{\boldsymbol{P}}_t$ *and* the population $\boldsymbol{P}_t$ is assigned a strength value $S_{\boldsymbol{i}}$, representing the number of solutions it dominates:

$$S_{\boldsymbol{i}} = |\{\boldsymbol{j} \mid \boldsymbol{j} \in \boldsymbol{P}_t + \overline{\boldsymbol{P}}_t \wedge \boldsymbol{i} \succ \boldsymbol{j}\}|$$

where $|\cdot|$ denotes the cardinality of a set, $+$ stands for multiset union and the symbol $\succ$ corresponds to the Pareto dominance relation. On the basis of the S values, the raw fitness $R_{\boldsymbol{i}}$ of an individual $\boldsymbol{i}$ is calculated:

$$R_{\boldsymbol{i}} = \sum_{\boldsymbol{j} \in \boldsymbol{P}_t + \overline{\boldsymbol{P}}_t, \boldsymbol{j} \succ \boldsymbol{i}} S_{\boldsymbol{j}}$$

That is the raw fitness is determined by the strengths of its dominators in both archive and population. It is important to note that fitness is to be minimized here, i.e., $R_{\boldsymbol{i}} = 0$

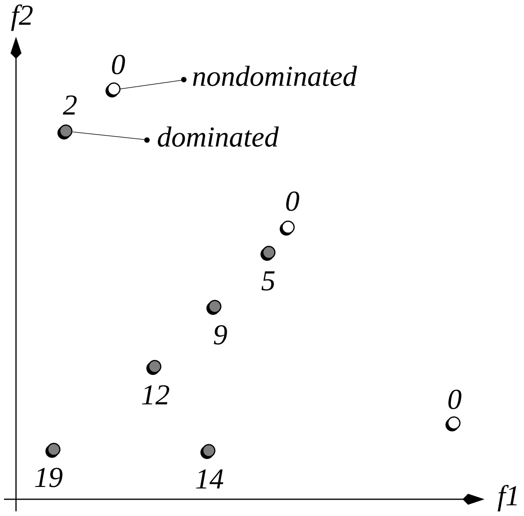


Figure 7: Illustration of the fitness assignment scheme used SPEA2 for a maximization problem with two objectives f_1 and f_2; the numbers give the raw fitness values of the corresponding individuals.

corresponds to a nondominated individual, while a high $R_{\boldsymbol{i}}$ value means that $\boldsymbol{i}$ is dominated by many individuals (which in turn dominate many individuals). This scheme is illustrated in Figure 7.

Although the raw fitness assignment provides a sort of niching mechanism based on the concept of Pareto dominance, it may fail when most individuals do not dominate each other. Therefore, additional density information is incorporated to discriminate between individuals having identical raw fitness values. The density estimation technique used in SPEA2 is an adaptation of the k-th nearest neighbor method [13], where the density at any point is a (decreasing) function of the distance to the k-th nearest data point. Here, we simply take the inverse of the distance to the k-th nearest neighbor as the density estimate. To be more precise, for each individual $\boldsymbol{i}$ the distances (in objective space) to all individuals $\boldsymbol{j}$ in archive and population are calculated and stored in a list. After sorting the list in increasing order, the k-th element gives the distance sought, denoted as $\sigma_{\boldsymbol{i}}^k$. As a common setting, we use k equal to the square root of the sample size [13], thus, $k = \sqrt{N + \overline{N}}$. Afterwards, the density $D_{\boldsymbol{i}}$ corresponding to $\boldsymbol{i}$ is defined by

$$D_{\boldsymbol{i}} = \frac{1}{\sigma_{\boldsymbol{i}}^k + 2}$$

In the denominator, two is added to ensure that its value is greater than zero and that $D_{\boldsymbol{i}} < 1$. Finally, adding $D_{\boldsymbol{i}}$ to the raw fitness value $R_{\boldsymbol{i}}$ of an individual $\boldsymbol{i}$ yields its fitness $F_{\boldsymbol{i}}$:

$$F_{\boldsymbol{i}} = R_{\boldsymbol{i}} + D_{\boldsymbol{i}}$$

4.2 Environmental Selection

The archive update operation (Step 3 in Algorithm 1) in SPEA2 was designed in such a way that i) the number of individuals contained in the archive is constant over time, and ii) boundary solutions are not lost.

During environmental selection, the first step is to copy all nondominated individuals, i.e., those which have a fitness lower than one, from archive and population to the archive of the next generation:

$$\overline{\boldsymbol{P}}_{t+1} = \{\boldsymbol{i} \mid \boldsymbol{i} \in \boldsymbol{P}_t + \overline{\boldsymbol{P}}_t \wedge F_{\boldsymbol{i}} < 1\}$$

If the nondominated front fits exactly into the archive ($|\overline{\boldsymbol{P}}_{t+1}| = \overline{N}$) the environmental selection step is completed. Otherwise, there can be two situations: Either the archive is too small ($|\overline{\boldsymbol{P}}_{t+1}| < \overline{N}$) or too large ($|\overline{\boldsymbol{P}}_{t+1}| > \overline{N}$). In the first case, the best $\overline{N} - |\overline{\boldsymbol{P}}_{t+1}|$ dominated individuals in the previous archive and population are copied to the new archive. This can be implemented by sorting the multiset $\boldsymbol{P}_t + \overline{\boldsymbol{P}}_t$ according to the fitness values and copy the first $\overline{N} - |\overline{\boldsymbol{P}}_{t+1}|$ individuals $\boldsymbol{i}$ with $F_{\boldsymbol{i}} \geq 1$ from the resulting ordered list to $\overline{\boldsymbol{P}}_{t+1}$. In the second case, when the size of the current nondominated (multi)set exceeds $\overline{N}$, an archive truncation procedure is invoked which iteratively removes individuals from $\overline{\boldsymbol{P}}_{t+1}$ until $|\overline{\boldsymbol{P}}_{t+1}| = \overline{N}$. Here, at each iteration that individual $\boldsymbol{i}$ is chosen for removal for which $\boldsymbol{i} \leq_d \boldsymbol{j}$ for all $\boldsymbol{j} \in \overline{\boldsymbol{P}}_{t+1}$ with

$$\begin{aligned}\boldsymbol{i} \leq_d \boldsymbol{j} \quad :\Leftrightarrow \quad & \forall\, 0 < k < |\overline{\boldsymbol{P}}_{t+1}| \;:\; \sigma_{\boldsymbol{i}}^k = \sigma_{\boldsymbol{j}}^k \quad \vee \\ & \exists\, 0 < k < |\overline{\boldsymbol{P}}_{t+1}| \;: \\ & \left[\left(\forall\, 0 < l < k \;:\; \sigma_{\boldsymbol{i}}^l = \sigma_{\boldsymbol{j}}^l\right) \wedge\; \sigma_{\boldsymbol{i}}^k < \sigma_{\boldsymbol{j}}^k\right]\end{aligned}$$

where $\sigma_{\boldsymbol{i}}^k$ denotes the distance of $\boldsymbol{i}$ to its k-th nearest neighbor in $\overline{\boldsymbol{P}}_{t+1}$. In other words, the individual which has the minimum distance to another individual is chosen at each stage; if there are several individuals with minimum distance the tie is broken by considering the second smallest distances and so forth. How this truncation technique works is illustrated in Figure 8.

5 Applications

There are numerous applications of EAs in the area of telecommunications ranging from network design to routing problems. An overview of multicriteria studies in this context can be found in [20].

In the lecture, we will present a network processor design application that involves several objectives. Due to space restrictions, the problem is not discussed here; instead, the interested reader is referred to the original paper [21].

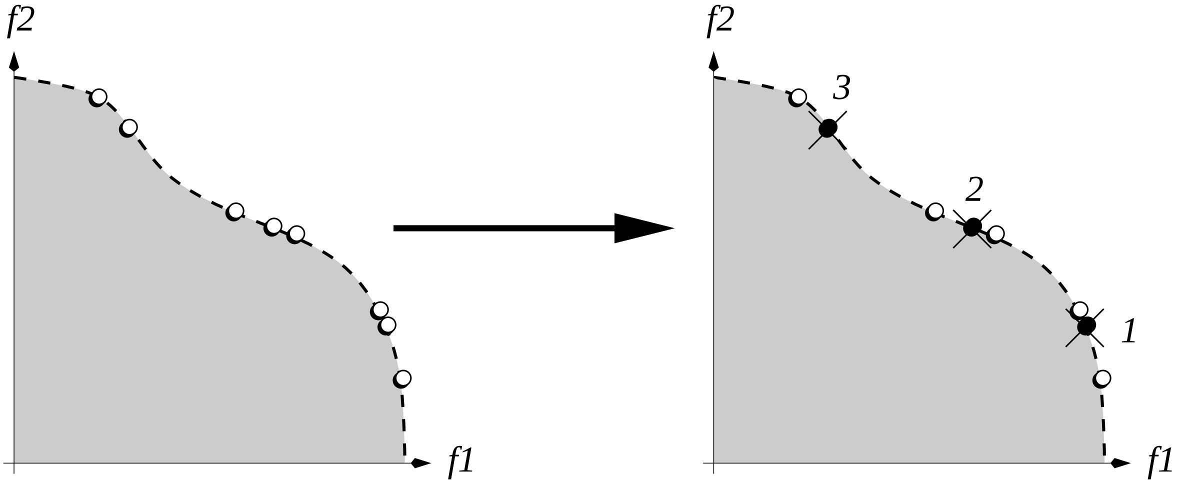


Figure 8: Illustration of the archive truncation method used in SPEA2. On the right, a nondominated set is shown. On the left, it is depicted which solutions are removed in which order by the truncate operator (assuming that $\overline{N} = 5$).

References

[1] T. Bäck, U. Hammel, and H.-P. Schwefel, "Evolutionary computation: Comments on the history and current state," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 3–17, 1997.

[2] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 67–82, April 1997.

[3] P. Hajela and C.-Y. Lin, "Genetic search strategies in multicriterion optimal design," *Structural Optimization*, vol. 4, pp. 99–107, 1992.

[4] H. Ishibuchi and T. Murata, "Multi-objective genetic local search algorithm," in *Proceedings of 1996 IEEE International Conference on Evolutionary Computation (ICEC'96)*, (Piscataway, NJ), pp. 119–124, IEEE Press, 1996.

[5] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of an International Conference on Genetic Algorithms and Their Applications* (J. J. Grefenstette, ed.), (Pittsburgh, PA), pp. 93–100, 1985. sponsored by Texas Instruments and U.S. Navy Center for Applied Research in Artificial Intelligence (NCARAI).

[6] F. Kursawe, "A variant of evolution strategies for vector optimization," in *Parallel Problem Solving from Nature* (H.-P. Schwefel and R. Männer, eds.), (Berlin), pp. 193–197, Springer, 1991.

[7] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning.* Reading, Massachusetts: Addison-Wesley, 1989.

[8] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," in *Proceedings of the Fifth International Conference on Genetic Algorithms* (S. Forrest, ed.), (San Mateo, California), pp. 416–423, Morgan Kaufmann, 1993.

[9] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.

[10] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Parallel Problem Solving from Nature – PPSN VI* (M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, eds.), (Berlin), pp. 849–858, Springer, 2000.

[11] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[12] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems. Proceedings of the EUROGEN2001 Conference, Athens, Greece, September 19-21, 2001* (K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papaliliou, and T. Fogarty, eds.), (Barcelona, Spain), pp. 95–100, International Center for Numerical Methos in Engineering (CIMNE), 2002.

[13] B. W. Silverman, *Density estimation for statistics and data analysis.* London: Chapman and Hall, 1986.

[14] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched pareto genetic algorithm for multiobjective optimization," in *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Computation*, vol. 1, (Piscataway, NJ), pp. 82–87, IEEE Press, 1994.

[15] J. D. Knowles and D. W. Corne, "The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation," in *Congress on Evolutionary Computation (CEC99)*, vol. 1, (Piscataway, NJ), pp. 98–105, IEEE Press, 1999.

[16] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms—part ii: Application example," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 28, no. 1, pp. 38–47, 1998.

[17] K. Deb, *Multi-objective optimization using evolutionary algorithms.* Chichester, UK: Wiley, 2001.

[18] G. Rudolph and A. Agapie, "Convergence properties of some multi-objective evolutionary algorithms," in *Congress on Evolutionary Computation (CEC 2000)*, vol. 2, (Piscataway, NJ), pp. 1010–1016, IEEE Press, 2000.

[19] M. Laumanns, L. Thiele, E. Zitzler, and K. Deb, "Archiving with guaranteed convergence and diversity in multi-objective optimization," in *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference* (W. B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, and N. Jonoska, eds.), (New York), pp. 439–447, Morgan Kaufmann Publishers, 9-13 July 2002.

[20] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems.* New York: Kluwer, 2002.

[21] L. Thiele, S. Chakraborty, M. Gries, and S. Künzli, *Network Processor Design: Issues and Practices, Volume 1*, ch. Exploration of Network Processor Architectures, pp. 55–89. Morgan Kaufmann, 2002.

Efficient and accurate numerical solution of stochastic partial differential equations: formulating a problem

Adolfo V. T. Cartaxo[*]   José A. P. Morgado[†]

Abstract

Accurate numerical computation of a set of stochastic partial differential equations (SPDEs) presents considerable difficulties. A technique to deal with this problem in a time efficient way is sought. Tracks to find a statistically correct numerical solution of the set of SPDEs are outlined. A technique used to solve a set of stochastic differential equations corresponding to a set of SPDEs with just one independent variable is presented, discussed and validated with several examples.

Keywords: stochastic differential equation, numerical solution, partial differential equation, nonlinear differential equation, accuracy, time efficiency, single-mode laser rate equations, relative intensity noise, frequency noise, periodogram, sample process, estimator, Fast Fourier Transform.

1 Introduction

In nowadays communication systems, as traffic increases rapidly in the Internet, wavelength division multiplexing (WDM) optical networks become the paradigm for fixed telecommunication networks. Semiconductor optical amplifiers (SOAs), and semiconductor distributed feedback (DFB) lasers are key components to implement these WDM optical networks.

Numerical simulation has been used as a powerful tool concerning the performance assessment and optimization of these systems [1, 2, 3]. Nevertheless, the accuracy and purposes of the results achieved by simulation depend strongly on the correctness and efficiency of the methods used to solve numerically the differential equations that govern the dynamics of the majority of the components comprising these systems, especially if noise features are taken into account.

The dynamics of the components indicated so far, including noise features, are usually described by a set of stochastic partial differential equations (SPDEs), see for instances [2, ch.

[*] Instituto de Telecomunicações. E-mail:adolfo.cartaxo@lx.it.pt

[†] Instituto de Telecomunicações. E-mail:j.morgado@lx.it.pt

11], [4, ch. 4 and ch. 5]. The accurate noise characterisation of these devices is of particular interest from a telecommunications point of view or, by other words, the accurate statistical characterisation of the stochastic process at the device output.

In the general case, each equation of the set of SPDEs is nonlinear. Therefore, the rigorous characterisation of noise at the device output can only be obtained by accurately solving, in a numerical way, the set of SPDEs. The accuracy of a numerical scheme for integrating a set of SPDEs is judged on the basis of its ability to provide samples of the stochastic process from which accurate estimates of some statistical parameters can be computed. Often, we are mainly interested in the estimation of moments, probabilities or other functional as the noise power spectral density from samples of the solution of the set of SPDEs. Furthermore, it is also particularly important to obtain accurate estimates in a time saving form because these estimates are often used in a more general process of system optimisation. So, a time efficient and accurate technique of numerically solving a set of SPDEs is very desirable and, to the authors knowledge, the development of such technique still remains.

Techniques based on first order approximation in the step size of each independent variable [5] should be avoided because of the high computation time required to assess each sample. Furthermore, the accuracy, as described above, of the techniques presented in reference [5] is questionable. Techniques like those presented in [5] seem to be particularly devoted to deal with another problem of the numerical solution of a set of SPDEs, namely its robustness.

The paper is structured as follows. In section 2, the formulation of the problem is presented in a general form. In section 3, a particular case of the formulated problem, which has been already successfully solved, is presented. Results related to its use on the simulation of the dynamics of single-mode bulk lasers are presented and discussed. In section 4, the main conclusions are drawn and suggestions to solve the formulated problem are given.

2 Formulations of the problem

The equations that govern the dynamics of SOAs and DFB lasers (propagation equations of the fields inside the device), including noise features, are formally equivalent to [2, 4]

$$\frac{\partial A_i(z,t)}{\partial z} + k_i \frac{\partial A_i(z,t)}{\partial t} = f_i(A_1, \dots, A_n, z, t) + \eta_i(z,t);\ 0 \le z \le t;\ t \ge 0;\ 1 \le i \le n \qquad (1)$$

where the index i refers to the different low–pass equivalent complex fields with a total count of n, t and z are the independent variables corresponding to time and space coordinates, respectively, L is the maximum space coordinate of interest (device length), $A_i(z,t)$ is the i-th low–pass equivalent complex field, k_i is a real constant, $f_i(A_1, \dots, A_n, z, t)$ is a complex nonlinear function of the complex fields representing the nonlinear field evolution, $A_i(z = 0, t)$ is the known time waveform of the i-th low-pass equivalent complex field at the device input ($z = 0$), and $\eta_i(z,t)$ is a complex Gaussian-distributed stochastic field. The two Gaussian components, corresponding to the real and imaginary parts of the complex fields $\eta_i(z,t)$, are zero mean, statistically independent stochastic processes. Besides, the stochastic fields $\eta_i(z,t)$ are generally uncorrelated in t and z, so that they satisfy the following property:

$$\left\langle \eta_i(z,t) \cdot \eta_i^*(z',t') \right\rangle = \xi_i \cdot \delta(t-t') \cdot \delta(z-z') \qquad 1 \leq i \leq n \tag{2}$$

where $\langle x \rangle$ means expected value of x, $\eta_i^*(z,t)$ stands for complex conjugate of $\eta_i(z,t)$, i is a constant, and $\delta(x)$ is the delta Dirac function.

The problem may be formulated as the derivation of a procedure for accurate numerical integration of the set of SPDEs presented in (1). The procedure should generate representative values of $A_i(z,t)$ at discrete times t_j for the specific space coordinate of $z = L$ by direct solution of the SPDEs. Then, these values of $A_i(L,t)$ are used to estimate accurately the statistical parameters of interest, as those above mentioned. The procedure should produce results that are statistically correct to a given order in the time and space step. Higher order approximations than the first order one, if they exist, seem desirable because of shorter computation time.

3 A particular case

In this section, a particular case of the formulated problem will be considered, assuming no space dependence in eqs. (1). This is a very useful case to simulate the dynamics of single-mode bulk lasers which, differently from the DFB lasers, can be modelled by stochastic differential equations (SDEs) with time as the only independent variable and, therefore, no space dependence exists [4, 6].

3.1 The Greenside–Helfand technique to solve SDEs with uncorrelated noise sources

A very interesting and useful technique presented in [7], to solve the set of SDEs resulting from (1) assuming no space dependence has been presented in [7]. In the following, the technique will be called Greenside–Helfand technique (GHT). The GHT is an extension of the Runge–Kutta technique used in the numerical solution of deterministic differential equations. The main idea of the GHT is to evaluate the nonlinear functions $fi(A_1, \ldots, A_n, t)$ at stochastically selected points, so that all moments of the extrapolated estimate after a time step are correct to some order in the step size [7]. The set of SDEs to be solved can be generically written as

$$\begin{cases} \dfrac{dx_1(t)}{dt} & = & h_1(x_1, x_2, t) + r_1(t) \\[2ex] \dfrac{dx_2(t)}{dt} & = & h_2(x_1, x_2, t) + r_2(t) \end{cases} \tag{3}$$

where $h_1(x_1, x_2, t)$ and $h_2(x_1, x_2, t)$ are real nonlinear functions representing the nonlinear time evolution of real processes $x_1(t)$ and $x_2(t)$, respectively, and $r_1(t)$ and $r_2(t)$ are real Gaussian-distributed noise sources with zero mean and autocorrelation functions given by $\langle r_1(t) \cdot r_1(t') \rangle = \xi_1 \cdot \delta(t - t')$ and $\langle r_2(t) \cdot r_2(t') \rangle = \xi_2 \cdot \delta(t - t')$, respectively, where ξ_1 and ξ_2 are constants representing the white power spectral densities of $r_1(t)$ and $r_2(t)$, respectively. Furthermore, the stochastic processes $r_1(t)$ and $r_2(t)$ are assumed uncorrelated, i. e., $\langle r_1(t) \cdot r_2(t') \rangle = 0$. A set of two equations is considered, but the GHT is easily generalised to an arbitrary number of equations. The samples of $x_1(t)$ and $x_2(t)$ after the time step, T_S, are given by [7]

$$\left\{ \begin{array}{rcl} x_1(T_S) & = & x_1(0) + T_S \cdot [A_1 g_{11} + A_2 g_{21} + A_3 g_{31} + A_4 g_{41}] + T_S^{1/2} \xi_1^{1/2} Y_{01} \\ \\ x_2(T_S) & = & x_2(0) + T_S \cdot [A_1 g_{12} + A_2 g_{22} + A_3 g_{32} + A_4 g_{42}] + T_S^{1/2} \xi_2^{1/2} Y_{02} \end{array} \right. \tag{4}$$

where A_i are constants given by

$$A_1 = 0.0; \qquad A_1 = 0.644468; \qquad A_3 = 0.194450; \qquad A_4 = 0.161082 \tag{5}$$

and g_{ij} are given by

$$\begin{aligned} g_{11} &= h_1\left(\left[x_1(0) + T_S^{1/2}\xi_1^{1/2}Y_{11}\right], \left[x_2(0) + T_S^{1/2}\xi_2^{1/2}Y_{12}\right]\right); \\ g_{12} &= h_2\left(\left[x_1(0) + T_S^{1/2}\xi_1^{1/2}Y_{11}\right], \left[x_2(0) + T_S^{1/2}\xi_2^{1/2}Y_{12}\right]\right); \\ g_{21} &= h_1\left(\left[x_1(0) + T_S\beta_{21}g_{11} + T_S^{1/2}\xi_1^{1/2}Y_{21}\right], \left[x_2(0) + T_S\beta_{21}g_{12} + T_S^{1/2}\xi_2^{1/2}Y_{22}\right]\right); \\ g_{22} &= h_2\left(\left[x_1(0) + T_S\beta_{21}g_{11} + T_S^{1/2}\xi_1^{1/2}Y_{21}\right], \left[x_2(0) + T_S\beta_{21}g_{12} + T_S^{1/2}\xi_2^{1/2}Y_{22}\right]\right); \\ g_{31} &= h_1\left(\left[x_1(0) + T_S\beta_{31}g_{11} + T_S\beta_{32}g_{21} + T_S^{1/2}\xi_1^{1/2}Y_{31}\right], \right. \\ &\qquad \left. \left[x_2(0) + T_S\beta_{31}g_{12} + T_S\beta_{32}g_{22} + T_S^{1/2}\xi_2^{1/2}Y_{32}\right]\right); \\ g_{32} &= h_2\left(\left[x_1(0) + T_S\beta_{31}g_{11} + T_S\beta_{32}g_{21} + T_S^{1/2}\xi_1^{1/2}Y_{31}\right], \right. \\ &\qquad \left. \left[x_2(0) + T_S\beta_{31}g_{12} + T_S\beta_{32}g_{22} + T_S^{1/2}\xi_2^{1/2}Y_{32}\right]\right); \\ g_{41} &= h_1\left(\left[x_1(0) + T_S\beta_{41}g_{11} + T_S\beta_{42}g_{21} + T_S\beta_{43}g_{31} + T_S^{1/2}\xi_1^{1/2}Y_{41}\right], \right. \\ &\qquad \left. \left[x_2(0) + T_S\beta_{41}g_{12} + T_S\beta_{42}g_{22} + T_S\beta_{43}g_{32} + T_S^{1/2}\xi_2^{1/2}Y_{42}\right]\right); \end{aligned} \tag{6}$$

$$\begin{aligned} g_{42} \;=\; & h_2\Big(\Big[x_1(0)+T_S\beta_{41}g_{11}+T_S\beta_{42}g_{21}+T_S\beta_{43}g_{31}+T_S^{1/2}\xi_1^{1/2}Y_{41}\Big]\,, \\ & \Big[x_2(0)+T_S\beta_{41}g_{12}+T_S\beta_{42}g_{22}+T_S\beta_{43}g_{32}+T_S^{1/2}\xi_2^{1/2}Y_{42}\Big]\Big)\;; \end{aligned} \qquad (6(\text{cont.}))$$

where β_{ij} are constants given by

$$\begin{aligned} \beta_{21} &= 0.516719; & \beta_{31} &= 0.397300; & \beta_{32} &= 0.427690; \\ \beta_{41} &= 1.587731; & \beta_{42} &= 1.417263; & \beta_{43} &= 1.170469 \end{aligned} \qquad (7)$$

and Y_{ij} are random variables given by

$$\begin{aligned} Y_{01} &= \lambda_{01}Z_{11}+\lambda_{02}Z_{21} & Y_{02} &= \lambda_{01}Z_{12}+\lambda_{02}Z_{22} \\ Y_{11} &= \lambda_{11}Z_{11}+\lambda_{12}Z_{21} & Y_{12} &= \lambda_{11}Z_{12}+\lambda_{12}Z_{22} \\ Y_{21} &= \lambda_{21}Z_{11}+\lambda_{22}Z_{21} & Y_{22} &= \lambda_{21}Z_{12}+\lambda_{22}Z_{22} \\ Y_{31} &= \lambda_{31}Z_{11}+\lambda_{32}Z_{21} & Y_{32} &= \lambda_{31}Z_{12}+\lambda_{32}Z_{22} \\ Y_{41} &= \lambda_{41}Z_{11}+\lambda_{42}Z_{21} & Y_{42} &= \lambda_{41}Z_{12}+\lambda_{42}Z_{22} \end{aligned} \qquad (8)$$

where λ_{ij} are constants given by

$$\begin{aligned} &\lambda_{01}=1.0; \quad \lambda_{11}=0.0; \quad \lambda_{21}=0.516719; \quad \lambda_{31}=0.030390; \quad \lambda_{41}=1.0; \\ &\lambda_{02}=1.0; \quad \lambda_{12}=0.271608; \quad \lambda_{22}=0.499720; \quad \lambda_{32}=0.171658; \quad \lambda_{42}=0.0; \end{aligned} \qquad (9)$$

and Z_{ij} are independent Gaussian random variables with zero mean and variance equal to one. Expressions (6) show that the nonlinear functions $h_1(x_1,x_2,t)$ and $h_2(x_1,x_2,t)$ are evaluated at stochastically selected points. This choice has been made so that all moments of the extrapolated estimates after a time step are correct to third order in the step size [7]. The results achieved in Ref. [7] suggest that higher order algorithms do not exist. However this statement has not been demonstrated.

3.2 The modified Greenside–Helfand technique to solve SDEs with correlated noise sources

In order to use the GHT on practical communication systems, namely to simulate the dynamics of single-mode bulk lasers, some modifications have been introduced in this technique in order to take into account the possible noise sources correlation. So, the GHT has been modified to solve the set of SDEs

$$\begin{cases} \dfrac{dx_1(t)}{dt} = h_1(x_1,x_2,t)+r_3(t) \\[2ex] \dfrac{dx_2(t)}{dt} = h_2(x_1,x_2,t)+r_2(t) \end{cases} \qquad (10)$$

where $r_3(t)$ and $r_2(t)$ are Gaussian–distributed noises sources with zero mean and autocorrelation functions given by $\langle r_3(t)\cdot r_3(t')\rangle = \xi_3\cdot\delta(t-t')$ and $\langle r_2(t)\cdot r_2(t')\rangle = \xi_2\cdot\delta(t-t')$, respectively, where ξ_3 and ξ_2 are constants representing the power spectral densities of $r_3(t)$ and $r_2(t)$, respectively, and cross-correlation function given by $\langle r_3(t) \cdot r_2(t')\rangle = \xi_{32} \cdot \delta(t - t')$. It can be easily shown that $r_3(t)$ can be written as

$$r_3(t) = r_1(t) + K r_2(t) \tag{11}$$

where $r_1(t)$ is a Gaussian-distributed noise source independent of $r_2(t)$, with zero mean and autocorrelation function given by $\langle r_1(t) \cdot r_1(t')\rangle = \xi_1 \cdot \delta(t - t')$, with

$$K = \frac{\xi_{23}}{\xi_2}; \qquad \xi_1 = \xi_3 - K^2 \xi_2 \tag{12}$$

It results from (11)-(12) that if $K = 0$, $r_3(t)$ and $r_2(t)$ are independent noise sources and, so, the GHT is applicable directly to solve (10). If $K \neq 0$, the correlation between $r_3(t)$ and $r_2(t)$ increases with increasing $|K|$. In this situation, it can be heuristically thought that the GHT still remains correct if, in the numerical solution (4)-(9), the expressions related to r1(t) are substituted by the linear combination (11) of similar expressions related to the independent noise sources $r_1(t)$ and $r_2(t)$. In practical terms, the following substitutions should be accomplished:

$$\begin{aligned}
T_S^{1/2} Y_{01} &\longrightarrow T_S^{1/2} \xi_1^{1/2} Y_{01} + K T_S^{1/2} \xi_2^{1/2} Y_{02} \\
T_S^{1/2} Y_{11} &\longrightarrow T_S^{1/2} \xi_1^{1/2} Y_{11} + K T_S^{1/2} \xi_2^{1/2} Y_{12} \\
T_S^{1/2} Y_{21} &\longrightarrow T_S^{1/2} \xi_1^{1/2} Y_{21} + K T_S^{1/2} \xi_2^{1/2} Y_{22} \\
T_S^{1/2} Y_{31} &\longrightarrow T_S^{1/2} \xi_1^{1/2} Y_{31} + K T_S^{1/2} \xi_2^{1/2} Y_{32} \\
T_S^{1/2} Y_{41} &\longrightarrow T_S^{1/2} \xi_1^{1/2} Y_{41} + K T_S^{1/2} \xi_2^{1/2} Y_{42}
\end{aligned} \tag{13}$$

in expressions (4) and (6). In the following, this technique is called modified Greenside-Helfand technique (MGHT).

To validate the MGHT, the particular case of the set of equations (10), where $h_1(x_1, x_2, t)$ and $h_2(x_1, x_2, t)$ are linear functions, is considered. In this situation, the analytical solution of the set of equations (10) is easily obtained. The MGHT validation is achieved by comparison of the analytical results concerning the power spectral densities of $x_1(t)$ and $x_2(t)$, with the numerical ones achieved by using the MGHT to solve (10). The periodogram technique is used to estimate the power spectral densities [1].

Figures 1 (a) and (b) show the analytical power spectral densities of x1(t) and x2(t), respectively. Figures 1 (c) and (d) show the simulation results for power spectral densities of $x_1(t)$ and $x_2(t)$, respectively. We have assumed $h_1(x_1, x_2, t) = -x_2(t)$ and $h_2(x_1, x_2, t) = x_1(t) - x_2(t)$, $\xi_1 = \xi_2 = 0.01$ and $K = 100$, resulting in strong correlated noise sources.

All simulations results were obtained averaging over 128 periodograms, having each one 2^{13} points and a time width of 400 s. Excellent agreement between analytical and simulations results is achieved. It should be stressed that excellent agreement has been also achieved for other tested K values such as $K = \pm 1$, $K = \pm 10$ and $K = -100$. These results validate the MGHT.

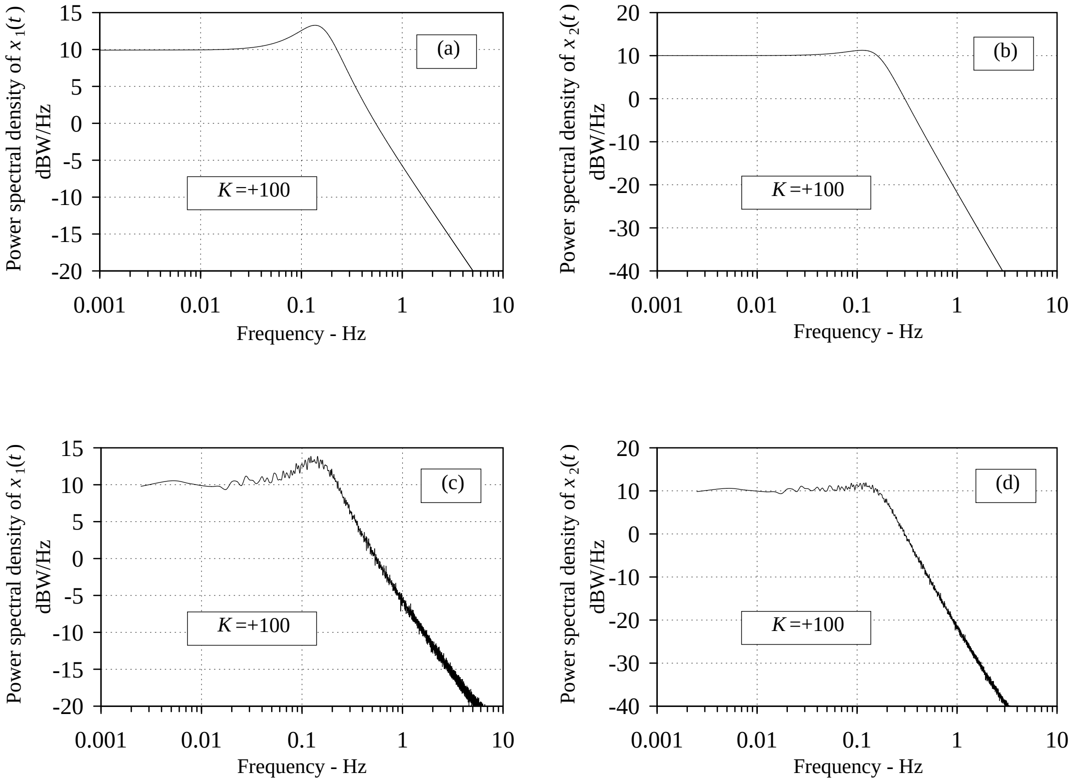


Figure 1: Theoretical results for power spectral density of (a) $x_1(t)$ and (b) $x_2(t)$. Simulation results for power spectral density of (c) $x_1(t)$ and (d) $x_2(t)$ using the MGHT.

3.3 Single-mode bulk laser rate equations simulation

In this section, the MGHT is used to solve the stochastic laser rate equations which govern the dynamics of a single-mode bulk laser and obtain the relative intensity noise (RIN) and frequency noise (FN) spectra at laser output and after fibre transmission. To validate the simulation results, simulation results are compared with theoretical predictions.

3.3.1 Theory

The single-mode bulk laser considered in this paper has been described in [8]. The stochastic laser rate equations which describe the relation between the carrier density $N(t)$ and photon density $S(t)$ at laser cavity, corresponding to the bias current $I(t)$ injected into the active region with volume V_{act}, can be written as [6]

$$\begin{cases} \dfrac{dN(t)}{dt} &= \dfrac{I(t)}{eV_{\mathrm{act}}} - R(N) - G(N,S)\cdot S(t) + F_N(t) \\[2ex] \dfrac{dS(t)}{dt} &= \Gamma G(N,S)\cdot S(t) - \dfrac{S(t)}{\tau_p} + R_S(N) + F_S(t) \end{cases} \tag{14}$$

where e is the electronic charge, Γ is the optical confinement factor, p is the photon lifetime, $G(N,S) = g_0(N - N_0)/(1 + \varepsilon S)$ is the cavity optical gain, where g_0 is the gain slope constant, N_0 is the electron density at which the net gain is zero and ε is the gain saturation parameter. $R(N)$ is the total recombination rate given by [6] $R(N) = A_{nr}N + B_rN^2 + C_{nr}N^3$, $R_s(N)$ is the spontaneous emission rate given by [6] $R_s(N) = \Gamma\beta B_rN^2$, where β is the spontaneous emission factor and A_{nr}, B_r and C_{nr} are respectively, the non-radiative, radiative and the Auger recombination coefficients [6]. $F_N(t)$ and $F_S(t)$ are the Langevin noise sources associated with the carrier and photon rate equations, respectively [6].

Within the semi-classical treatment, fluctuations arising from the spontaneous-emission process and the carrier-generation-recombination process, which are physically responsible for RIN, are incorporated by adding the Langevin noise sources $F_N(t)$ and $F_S(t)$ to the single-mode laser rate equations (14) [6]. Under the Markovian assumption, the Langevin noise sources are zero mean, correlated Gaussian ergodic stochastic processes [6], and their spectral densities are given by $\left\langle \left|\tilde{F}_N(f)\right|^2 \right\rangle = 8\pi\Delta f_0\overline{S}^2/\Gamma^2 + 2R(\overline{N})/V_{\mathrm{act}}$, $\left\langle \left|\tilde{F}_S(f)\right|^2 \right\rangle = 8\pi\Delta f_0\overline{S}^2$, and $\left\langle \tilde{F}_S(f)\cdot\tilde{F}_N^*(f) \right\rangle = -8\pi\Delta f_0\overline{S}^2/\Gamma^2$, where $\tilde{F}_N(f)$ and $\tilde{F}_S(f)$ are the Fourier transforms of $F_N(t)$ and $F_S(t)$, respectively, f is the modulating frequency, $\overline{S}$ and $\overline{N}$ represent the steady-state mean values of the photon and carrier densities, respectively, and $\Delta f_0 = R_S(\overline{N})/\left(4\pi\overline{S}\right)$ is the modified Schawlow-Townes linewidth [6].

The optical phase $\phi_T(t)$ of the electrical field emitted by the laser is given by [6]

$$\frac{d\phi_T(t)}{dt} = \frac{\alpha}{2}\Gamma g_0\left[N(t) - N_{th}\right] + F_{\dot{\phi}}(t) \tag{15}$$

where α is the linewidth enhancement factor, N_{th} is the threshold carrier density and $F_{\dot{\phi}}(t)$ is the Langevin noise source associated with the optical phase. Under the Markovian assumption, this Langevin noise source is a zero mean, Gaussian ergodic stochastic process, uncorrelated with $F_N(t)$ and $F_S(t)$, and with spectral density given by $\left\langle \left|\tilde{F}_{\dot{\phi}}(f)\right|^2 \right\rangle = 2\pi\Delta f_0$ [6].

The laser intrinsic parameters can be found in [8], except the carrier lifetime τ_n which is given by [6] $\tau_n = N/R(N) = \left(A_{nr} + B_{rN} + C_{nr}N^2\right)^{-1}$. $A_{nr} = 6.6 \times 10^7 \mathrm{s}^{-1}$, $B_r = 3 \times 10^{-16} \mathrm{m}^3\mathrm{s}^{-1}$ and $C_{nr} = 4 \times 10^{-41} \mathrm{m}^6\mathrm{s}^{-1}$, have been considered, which correspond to $\tau_n = 1$ ns at the threshold current [6].

To obtain analytically the laser noise characteristics, the steady-state values of the carrier and photon densities and optical phase are perturbed by small amounts whereas the injected current is kept in its steady state value [6], [9, 10, 11]. In this small-signal analysis, the stochastic nonlinear differential equations (14), (15) are first linearised and, then, solved in the frequency domain using the Fourier Transform. The amplitude of the power and phase fluctuations, in the frequency domain, at the laser output ($\tilde{\delta p}(z = 0, f)$ and $\tilde{\delta \phi}(z = 0, f)$) and fibre output ($\tilde{\delta p}(z, f)$ and $\tilde{\delta \phi}(z, f)$), assuming linear transmission, can be found in [11]. The two-sided RIN and FN spectra expressions are given, respectively, by [11]

$$RIN(z,f) = \frac{\left\langle \left| \tilde{\delta p}(z,f) \right| \right\rangle}{\overline{P}^2}; \qquad FN(z,f) = \left\langle \left| 2\pi f \cdot \tilde{\delta \phi}(z,f) \right|^2 \right\rangle \tag{16}$$

where $\overline{P}$ is the average power. Expressions for RIN and FN at laser and fibre output are presented in [11].

3.3.2 Simulation

To estimate the RIN and FN spectra at laser and fibre output, the periodogram technique has been used [1]. The estimator is obtained after averaging over M periodograms and is given by

$$\hat{E}(f) = \frac{\sum_{m=1}^{M} \hat{E}_m(f)}{M}$$

where $\hat{E}_m(f)$ is the m-th RIN or FN periodogram, respectively. This spectrum estimator is asymptotically unbiased and its variance decreases (approaches zero) with $1/M$ [1]. Each sample function $\delta p(z, t)$ and $\delta \phi(z, t)$ of the intensity and phase noise processes at laser ($z = 0$) or fibre output with duration T is sampled at N_P points (the sampling period is $T_S = N_P/T$) and each periodogram $\hat{E}_m(f)$ is obtained using the Fast Fourier Transform (FFT) as follows [1, 12]

$$\begin{aligned} R\hat{I}N_m(f) &= 10\log_{10}\left\{ \frac{\frac{FFT\left[\delta p(z,t)\right]^2}{(N_P T_S)}}{\overline{P}^2} \right\}; \qquad 0 \le i \le \frac{P}{2} \qquad (17) \\ \hat{F}N_m(f) &= \frac{\left|2\pi f \cdot FFT\left[\delta p(z,t)\right]\right|^2}{N_P T_S} \end{aligned}$$

Therefore, samples of $\hat{E}_m(f)$, calculated from (17), have a frequency resolution of $\Delta f = 1/T$ [1, 12]. So, larger T leads to better frequency resolution of the spectra estimators [12].

To obtain $\delta p(z,t)$ and $\delta\phi(z,t)$, the laser rate equations (14) taking into account the Langevin noise terms, are numerically integrated using the MGHT. As in [10] and [11], the transmission along the single-mode fibre is simulated in the frequency domain taking advantage of the speed of FFT computation.

3.3.3 Numerical results and discussion

Figure 2 shows the simulated results of RIN spectrum at laser output using the fifth order Runge-Kutta technique (deterministic technique) (Fig. 2-(a)), and the MGHT of solving a set of SDEs (stochastic technique) (Fig. 2-(b)). In Fig. 2-(a), the corresponding theoretical result is also shown for comparison purposes. Figs. 3-(a) and 3-(b) show, respectively, the theoretical results of FN spectrum at laser output, and the numerical estimate using the stochastic technique. A laser bias current of 80 mA has been considered. All simulations results were obtained averaging over M =128 periodograms, having each one $N_P = 2^{15}$ points and T =40 ns. These results show that the use of the MGHT to solve the laser rate SDEs provides statistically correct samples of the output stochastic processes, at least up to the second-order moment. Similar results have been obtained for other laser bias currents. Besides, Fig. 2-(a) shows that deterministic numerical methods utilized by many authors, see for instances [13, 14], to integrate the bulk laser rate equations (14) should not be used since they provide statistically incorrect description of the single-mode bulk laser noise.

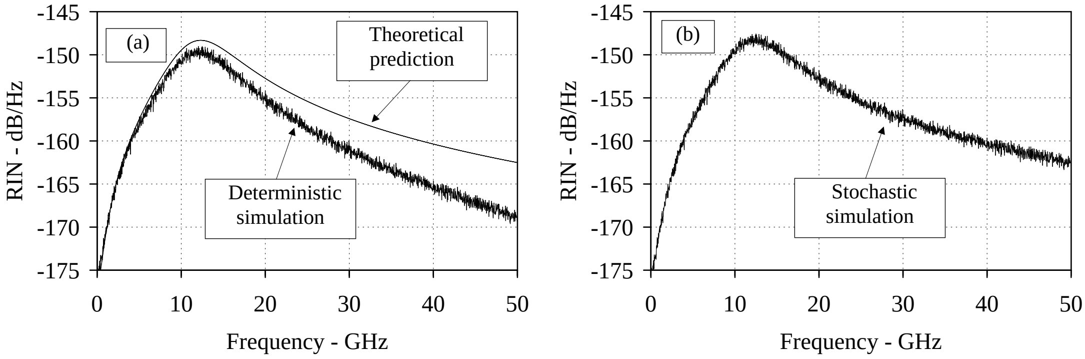


Figure 2: RIN spectrum at laser output for a bias current of 80 mA: (a) theoretical result and deterministic simulation result; (b) stochastic simulation result.

Figures 4 and 5 show, respectively, the RIN and FN spectra at fibre output at a laser bias current of 80 mA, after single-mode fibre transmission with dispersion of 1600 ps/nm. Figs. 4-(a) and 5-(a) show the theoretical results and Figs. 4-(b) and 5-(b) show the simulation results

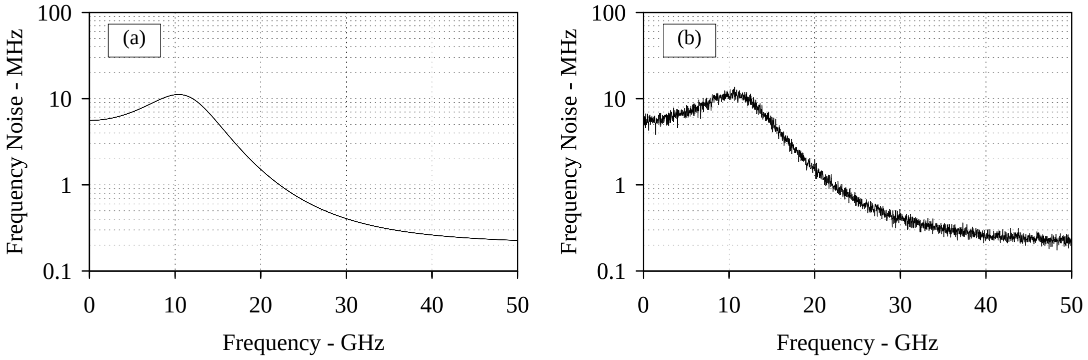


Figure 3: FN spectrum at laser output for a bias current of 80 mA: (a) theoretical result; (b) stochastic simulation result.

using the MGHT. All simulations results were obtained averaging over M =128 periodograms, having each one $N_P = 2^{15}$ points and T =40 ns. Figs. 1-5 show that very efficient and accurate power spectral density estimates of intensity and frequency noises of the field at the laser output and after linear transmission over a single-mode fibre are obtained using the MGHT. Further results showed that the MGHT leads also to good estimates of the probability density function of the intensity noise at the laser output in agreement with experimental data that indicates a quasi-Rice distributed optical power at the bulk laser output [15].

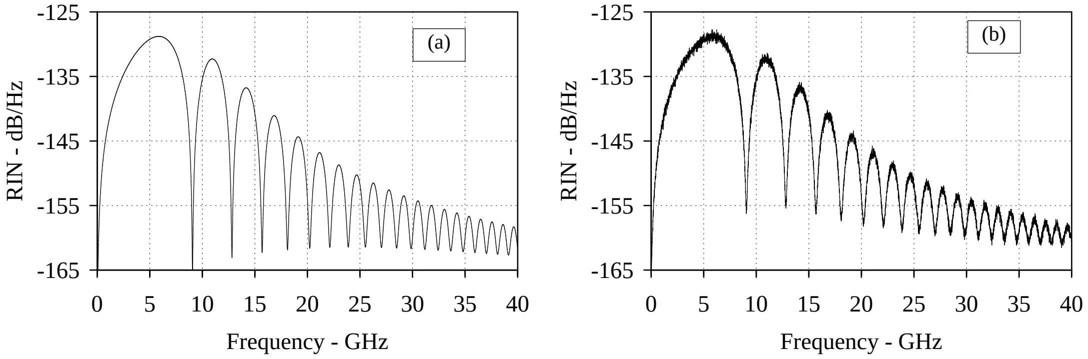


Figure 4: RIN spectrum at fibre output for a bias current of 80 mA and fibre dispersion of 1600 ps/nm: (a) theoretical result; (b) stochastic simulation result.

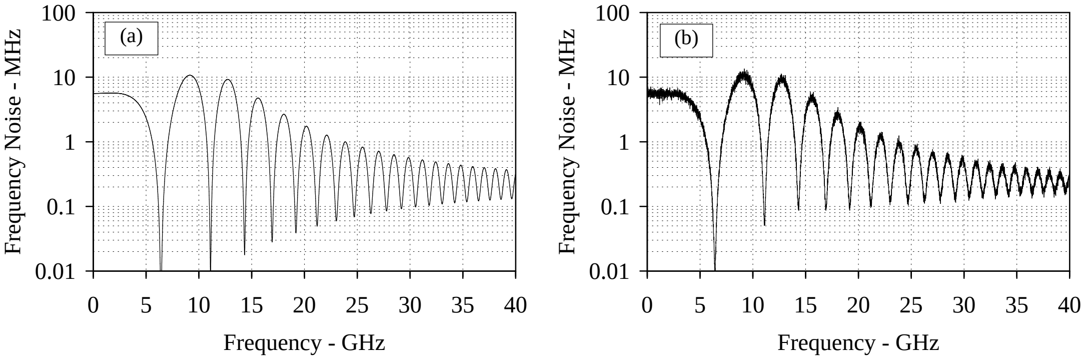


Figure 5: FN spectrum at fibre output for a laser bias current of 80 mA and fibre dispersion of 1600 ps/nm: (a) theoretical result; (b) stochastic simulation result.

4 Conclusions

To the authors knowledge, accurate numerical solution of a set of SPDEs is still an open problem. Tracks to find a solution have been provided. A technique used to solve a set of SDEs has been presented, discussed and validated with several examples. This technique, which is an extension of the Runge-Kutta techniques for numerical solution of deterministic differential equations, can be seen as a particular case from the one required for solving SPDEs. Its accuracy has been shown for the stochastic laser rate equations that govern the dynamics of a single-mode bulk laser. Therefore, a generalisation of the Runge-Kutta method to SPDEs or the development of a completely new technique with similar features, regarding accuracy, time efficiency and complexity, would be very desirable.

It has been also shown that the usual methods of solving deterministic nonlinear differential equations to estimate, by simulation, statistics of stochastic process should be avoided since they provide statistically incorrect estimates.

Acknowledgements

This work was supported, in part, by Fundação para a Ciência e Tecnologia from Portugal, and POSI within the project POSI/35576/CPS/1999-DWDM/ODC.

References

[1] M. Jeruchim, P. Balaban, and K. Shanmugan, *Simulation of Communication Systems.* New York: Plenum Press, 1992.

[2] G. Guekos, ed., *Photonic Devices for Telecommunications.* Springer-Verlag, 1999.

[3] E. Iannone, F. Matera, A. Mecozzi, and M. Settembre, *Nonlinear Optical Communication Networks.* J. Wiley & Sons, Inc., 1998.

[4] G. Morthier and P. Vankwikelberge, *Handbook of Distributed Feedback Laser Diodes.* Artech House, Inc., 1997.

[5] M. Werner and P. Drummond, "Robust algorithms for solving stochastic partial differential equations," *J. Computational Physics*, vol. 132, pp. 312–326, 1997.

[6] G. P. Agrawal and N. K. Dutta, *Semiconductor Lasers.* New York: Van Nostrand Reinhold, 1993.

[7] H. Greenside and E. Helfand, "Numerical integration of stochastic differential equations – II," *Bell Syst. Tech. Journal*, vol. 60, no. 8, pp. 1927–1940, 1981.

[8] R. Vodhanel, A. Elrefaie, R. Wagner, M. Iqbal, J. Gimlett, and S. Tsuji, "Ten-to-twenty gigabit-per-second modulation performance of 1.5μm distributed feedback lasers for frequency-shift-keying systems," *IEEE/OSA J. Lightwave Technol.*, vol. 7, no. 10, pp. 1454–1460, 1989.

[9] K. Peterman, *Laser Diode Modulation and Noise.* Kluwer Academic, 1991. (corrected edition).

[10] A. Cartaxo and J. Morgado, "Intensity and frequency noise transmission along single-mode fibre at zero-dispersion wavelength," *IEE Proc. Part J-Optoelect.*, vol. 145, pp. 211–216, Aug. 1998.

[11] A. Cartaxo and J. Morgado, "Rigorous assessment of small-signal analysis for linear and dispersive optical communication systems operating near the zero-dispersion wavelength," *IEEE/OSA J. Lightwave Technol.*, vol. 17, no. 1, pp. 1454–1460, 1989.

[12] E. O. Brigham, *The Fast Fourier Transform and its Applications.* Prentice-Hall International Editions, 1988.

[13] D. Marcuse, "Computer simulation of laser photon fluctuations: single-cavity laser results," *IEEE J. Quantum Electronics*, vol. 20, pp. 1148–1155, Oct. 1984.

[14] N. Schunk and K. Petermann, "Numerical analysis of the feedback regimes for a single-mode semiconductor laser with external feedback," *IEEE J. Quantum Electronics*, vol. 24, pp. 1242–1247, July 1988.

[15] P. Liu, L. Fencil, J. Ko, I. Kaminow, T. Lee, and C. Burrus, "Amplitude fluctuations and photon statistics of InGaAsP injection lasers," *IEEE J. Quantum Electronics*, vol. 19, pp. 1348–1351, July 1983.

Optimal M-QAM/DAPSK allocation in Narrowband OFDM radio channels

Bárbara Coelho[*] António Navarro[†]

Abstract

This paper proposes and formulates a mathematical optimization problem in the context of multi-carrier communications. A particular multi-carrier modulation system is the orthogonal Frequency Division Multiplexing (ODFM) where modulation operations are implemented through a single Fast Fourier Transformer (FFT). A wireless communication system may use tens of thousands of orthogonal modulators. Given a finite set of possible digital modulators M-QAM or M-DAPSK and under certain constrains, the solution of the optimization problem should provide the optimum value of M.

Keywords: Digital Television, Integer Optimization, Optimal Multi-carrier, Adaptive Joint Source-modulation.

1 Introduction

Broadcasting is moving into a digital era allowing new and enriched services and applications. The broadcasting quality is improved significantly by using multicarrier systems. The first systems using MCM (Multi-Carrier Modulation) were military HF radio links in the late 1950s and early 1960s. OFDM, a special form of MCM was patented by R.W. Chang in the US in 1970. OFDM removed the bank of steep bandpass filters that completely separated the spectrum of individual subcarriers. Orthogonality of OFDM carriers allows subcarrier spectra overlapping without inter-carrier interference (ICI).

The most popular wireless broadcasting systems making use of OFDM are Digital Audio Broadcasting (DAB) and Digital Video Broadcasting (DVB). OFDM is nowadays efficiently implemented by applying the IDFT/IFFT at the emitter,

$$x_n = \frac{1}{N} \sum_{k=0+c}^{N-1+c} X_k e^{j\frac{2\pi}{N}kn}, \qquad 0 + d \leq n \leq N - 1 + d \tag{1}$$

[*] ESTGL Inst. Polit. de Leiria. E-mail:barbara@estg.ipleiria.pt

[†] Instituto de Telecomunicações. E-mail:navarro@det.ua.pt

and the DFT/FFT at the receiver,

$$X_k = \sum_{n=0+a}^{N-1+a} x_n e^{-j\frac{2\pi}{N}kn}, \qquad 0+b \leq k \leq N-1+b \tag{2}$$

where a, b, c and d can be any integer. For sake of simplicity, let us assume them equal to zero. X_k, $k = 0, 1, \ldots, N-1$ are integer complex numbers and represent the information to be transmitted to the receiver. As expressed in (1), X_k is modulated/multiplied by a complex exponential carrier and through the summation converted into a new discrete complex sequence x_n, $n = 1, 2, \ldots, N$ usually called a symbol. This sequence is delivered to the receiver suffering channel impairments. Thus x_n is changed by the channel, resulting in,

$$r_n = x_n \otimes h_n + w_n, \qquad n = 0, 1, \ldots, n-1 \tag{3}$$

where $\otimes$ denotes the convolution operation, h_n is a exponential decaying function and w_n is a zero mean complex Gaussian independent variable. All variables in (3) as well as X_n are random processes.

From (3), we have [1],

$$r_n = \text{IFFT}\left\{\text{FFT}(x_n) \cdot \text{FFT}(h_n)\right\} + w_n \tag{4}$$

resulting in,

$$r_n = \text{IFFT}\left\{X_n H_n\right\} + w_n \tag{5}$$

By applying the FFT to (5), we obtain,

$$R_n = X_n H_n + Z_n \tag{6}$$

with

$$Z_n = \text{FFT}(w_n), \tag{7}$$

representing a zero mean complex Gaussian independent random variable. The receiver performance is measured by its capability of removing H_n and Z_n effects in (6) and thus approaching R_n to X_n. The impairments caused by H_n and Z_n are greater and greater as M_n increases. However, the greater M_n is, the more information is delivered to the destination. Therefore a tradeoff is require to find out the best values of M_n, $n = 0, 1, \ldots, N-1$.

2 The Optimization Problem Formulation

We will confine our problem formulation to M-QAM with $M = 1, 2, 4, 8, 16, 32, 64$. Let b be given by $\log_2 M$. The impairments mentioned in the above section are modeled by the bit error probability. In consequence, we are interested in minimizing the following objective function defined implicitly,

$$\sum_{n=0}^{N-1} P_{b_n}(\overline{\gamma}_n), \qquad b \in \{0, 1, 2, \ldots, 6\} \tag{8}$$

where, the error probabilities, $P_b(\overline{\gamma})$ are derived from $P_b(\gamma)$, as described in Appendix A [2, 3, 4]. The variable follows a chi-square distribution with two degrees of freedom and average $\overline{\gamma}$ [5]. Expressions $P_b(\overline{\gamma})$ take into account the effects of H_n and Z_n described in Section 1 and therefore is a function of n.

The solution is the vector $\mathbf{B}$ with the following elements,

$$b_n, \qquad n = 0, 1, \ldots, N-1 \tag{9}$$

in which n is the vector index and represents the carrier frequency. The problem constraints are:

1. $$b \in \{0, 1, 2, \ldots, 6\}$$

2. $$\sum_{n=0}^{N-1} b_n = \frac{N}{7} \sum_{b=0}^{6} b = 3N \tag{10}$$

The latter expression constrains the total number of bits transmitted in the N carriers. For optimization algorithm evaluation purposes, we have assumed that the total number of bits is equal to N times the average number of bits in all possible constellations. Observing the latter constrain, the problem is non-convex and therefore several solutions may occur. However, we expect a solution for $\mathbf{B} \neq 3\mathbf{I}$ (the same constellation $M = 8$ in all carries) and

$$\sum_{n=0}^{N-1} P_{b_n}(\overline{\gamma}_n) < \sum_{n=0}^{N-1} P_{3_n}(\overline{\gamma}_n), \qquad b \in \{0, 1, 2, \ldots, 6\} \tag{11}$$

It would be interesting to find out a solution for some particular functions as for instance an exponential function type,

$$\overline{\gamma} = \frac{\alpha}{2}\left(1 - e^{-\alpha|n-k|}\right) \mathrm{dB}, \quad \alpha > 0, \; k \in \{0, 1, \ldots, N-1\} \tag{12}$$

We have been trying to propose to solve the problem through dynamic programming which provides a global optimum. However, the constrain (10) imposes some challenges. Figure 1 shows an hypothetical solution where higher modulations are assigned to carriers with lower signal-to-noise ratio.

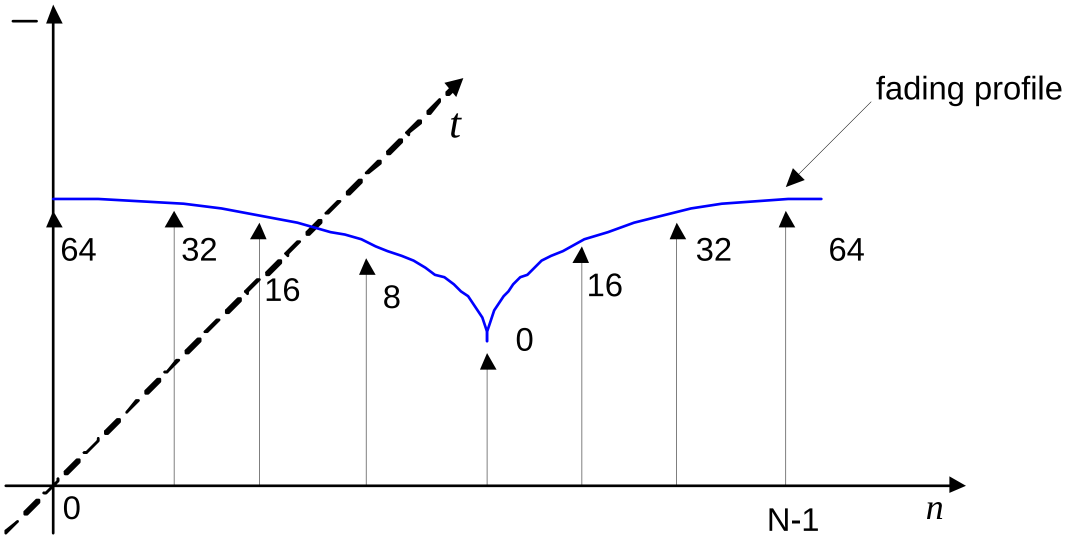


Figure 1: An hypothetical solution for a given profile at a particular instant of time.

3 Conclusions

Other problems could have been formulated. For instance, instead of minimizing the total error probability (8) and constraining on the bit error rate (10), we could formulate the problem by maximizing the total bit error rate and an inequality constrain given by the total error probability greater than a pre-defined value.

Appendix A

The bit error probabilities (BEP) are obtained by solving the integral:

$$P_E(\overline{\gamma}) = \int_0^\infty P_E p_\gamma(\gamma) d\gamma \tag{A.1}$$

where P_E is the conditional BEP in non-fading channel corrupted by AWGN and $p_\gamma(\gamma)$ is the probability density function (PDF) [4]. Rayleigh function is used to model the multipath fading with no direct line-of-sight (LOS). The PDF of Rayleigh model is:

$$p_\gamma(\gamma) = \frac{1}{\overline{\gamma}} \exp\left(-\frac{\gamma}{\overline{\gamma}}\right), \qquad \gamma \geq 0 \tag{A.2}$$

The expression for the BEP with AWGN involves the Gaussian Q-function and the square of this function,

$$Q(x) = \int_x^\infty \frac{\exp\left(-\frac{y^2}{2}\right)}{\sqrt{2}\,\pi} dy \tag{A.3}$$

Using an alternate representation for simplicity,

$$Q(x) = \int_0^{\frac{\pi}{2}} \exp\left(-\frac{x^2}{2\sin^2\theta}\right) d\theta \qquad \text{for } x \geq 0 \tag{A.4}$$

In this particular case,

$$P_{b_n}(\overline{\gamma}) = \int_0^{\infty} Q(a\sqrt{\gamma})p_\gamma(\gamma)d\gamma \tag{A.5}$$

resulting in the following expression,

$$P_{b_n}(\overline{\gamma}) = \frac{1}{\pi}\int_0^{\frac{\pi}{2}} \left(1 + \frac{a^2\overline{\gamma}}{2\sin^2\theta}\right)^{-1} d\theta = \frac{1}{2}\left(1 - \sqrt{\frac{\frac{a^2\overline{\gamma}}{2}}{1+\frac{a^2\overline{\gamma}}{2}}}\right) \tag{A.6}$$

Considering $b \in \{0, 1, 2, \ldots, 6\}$, the error probabilities are:

- For $b = 0$, no information is transmitted.
- For $b = 1$,

$$P_1(\overline{\gamma}) = \int_0^{\infty} Q\left(\sqrt{2\gamma}\right)\frac{1}{\overline{\gamma}}e^{-\frac{\gamma}{\overline{\gamma}}}d\gamma = \frac{1}{2}\left(1 - \sqrt{\frac{\overline{\gamma}}{\overline{\gamma}+1}}\right) \tag{A.7}$$

- For $b = 2$,

$$P_2(\overline{\gamma}) = \int_0^{\infty} 2Q\left(\sqrt{2\gamma}\right)\frac{1}{\overline{\gamma}}e^{-\frac{\gamma}{\overline{\gamma}}}d\gamma = \left(1 - \sqrt{\frac{\overline{\gamma}}{\overline{\gamma}+1}}\right) \tag{A.8}$$

- For $b = 3$,

$$P_3(\overline{\gamma}) = \int_0^{\infty} \frac{5}{6}Q\left(\sqrt{\frac{\gamma}{3}}\right)\frac{1}{\overline{\gamma}}e^{-\frac{\gamma}{\overline{\gamma}}}d\gamma = \frac{5}{12}\left(1 - \sqrt{\frac{\overline{\gamma}}{\overline{\gamma}+6}}\right) \tag{A.9}$$

- For $b = 4$,

$$\begin{aligned} P_4(\overline{\gamma}) &= \int_0^{\infty}\left[\frac{3}{4}Q\left(\sqrt{\frac{\gamma}{5}}\right) + \frac{1}{2}Q\left(\sqrt{\frac{9\gamma}{5}}\right) - \frac{1}{4}Q\left(\sqrt{5\gamma}\right)\right]\frac{1}{\overline{\gamma}}e^{-\frac{\gamma}{\overline{\gamma}}}d\gamma \\ &= \frac{3}{8}\left(1 - \sqrt{\frac{\overline{\gamma}}{\overline{\gamma}+10}}\right) + \frac{1}{4}\left(1 - \sqrt{\frac{9\overline{\gamma}}{9\overline{\gamma}+10}}\right) - \frac{1}{8}\left(1 - \sqrt{\frac{5\overline{\gamma}}{5\overline{\gamma}+2}}\right) \end{aligned} \tag{A.10}$$

- For $b = 5$,

$$P_5(\overline{\gamma}) = \int_0^{\infty} \frac{7}{10}Q\left(\sqrt{\frac{\gamma}{10}}\right)\frac{1}{\overline{\gamma}}e^{-\frac{\gamma}{\overline{\gamma}}}d\gamma = \frac{7}{20}\left(1 - \sqrt{\frac{\overline{\gamma}}{\overline{\gamma}+20}}\right) \tag{A.11}$$

- For $b = 6$,

$$\begin{aligned}
P_6(\overline{\gamma}) = & \int_0^\infty \left[\frac{7}{12}Q\left(\sqrt{\frac{\gamma}{21}}\right) + \frac{1}{2}Q\left(\sqrt{\frac{9\gamma}{21}}\right) - \frac{1}{12}Q\left(\sqrt{\frac{25\gamma}{21}}\right) + \frac{1}{12}Q\left(\sqrt{\frac{81\gamma}{21}}\right)\right. \\
& \left. -\frac{1}{12}Q\left(\sqrt{\frac{169\gamma}{21}}\right)\right] \frac{1}{\overline{\gamma}}e^{-\frac{\gamma}{\overline{\gamma}}}d\gamma \\
& = \frac{7}{24}\left(1 - \sqrt{\frac{\overline{\gamma}}{\overline{\gamma}+42}}\right) + \frac{1}{4}\left(1 - \sqrt{\frac{9\overline{\gamma}}{9\overline{\gamma}+42}}\right) - \frac{1}{24}\left(1 - \sqrt{\frac{25\overline{\gamma}}{25\overline{\gamma}+42}}\right) \\
& + \frac{1}{24}\left(1 - \sqrt{\frac{81\overline{\gamma}}{81\overline{\gamma}+42}}\right) - \frac{1}{24}\left(1 - \sqrt{\frac{169\overline{\gamma}}{169\overline{\gamma}+42}}\right)
\end{aligned} \tag{A.12}$$

References

[1] A. Navarro, “Half a century years later.” Lecture-notes, Video Signal Processing, Universidade de Aveiro, Feb. 2003.

[2] Y. Kim *et al.*, “Performance analysis of a coded OFDM system in time-varying multipath rayleigh fading channels,” *IEEE Trans. on Vehicular Technology*, vol. 48, pp. 1610–1615, Sep. 1990.

[3] L. Hanzo, W. Webb, and T. Keller, *Single- and Multi-carrier Quadrature Amplitude Modulation.* Chichester-England: Wiley, 2000.

[4] M. Simon and M. Alouini, *Digital Communications over Fading Channels.* Chichester-England: Wiley, 2000.

[5] J. Proakis, *Digital Communications.* Singapore: McGraw-Hill, 1995.

Computational Complexity of Discrete Fourier Transform

Vitor Silva[*] Fernando Perdigão[†]

Abstract

The development of a new mathematical theory on the computational complexity of the Discrete Fourier Transform is an important research topic. Lower bounds on the number of elementary operations (additions and multiplications) are needed in order to verify if current FFT algorithms are nearly optimal or if there is room for further improvements.

Keywords: DFT, FFT, computational complexity, lower bound.

1 Introduction

The discrete Fourier transform (DFT) is an important mathematical tool in modern digital signal processing and telecommunications fields. The direct transform is given by

$$X(k) = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}nk}, \tag{1}$$

for $0 \leq k \leq N-1$, and where N is the length of the sequence $x[n]$ (real or complex). Since the pioneer work by Cooley and Tukey [1], several families of fast DFT algorithms (FFT) have been developed [2, 3, 4, 5, 6]. The foundations of the modern work on efficient FFT algorithms were done by S. Winograd [7, 8, 9].

Several new algorithms have been published that require the least known amount of total arithmetic [10, 13, 12, 14, 16, 17]. Of these, the split-radix FFT [11, 21, 16] seems to have the best structure for programming. Other FFT approach is the prime factor algorithm (PFA) which access the data using an index map originally developed by Thomas and by Good [13]. The theory of the PFA was derived in [13] and some results on the PFA are given in [22, 23]. Another DFT method, valid for a general length N, based on a linear convolution operation, is the Chirp-z Transform (CZT) [24].

[*] Instituto de Telecomunicações. E-mail:vitor@co.it.pt

[†] Instituto de Telecomunicações. E-mail:fp@co.it.pt

A common issue shared by the mentioned algorithms is the arithmetic complexity minimization idea. Usually, it is measured as the number of complex (real) additions and multiplications necessary to compute (1). In the literature, several upper bounds on the number of arithmetic operations are available (for N prime, composite and power of primes). For $N = 2^n$ the best-known upper bound ($4N\log_2(N)$ real multiplications plus additions) is due to the split-radix algorithm [16]. For a generic N, a (good) upper bound is given in [25, 26]. Theoretical lower bounds on the number of multiplications required for the DFT, based on Winograd's theories, are given in [19, 20, 16] (multiplicative complexity). However, there is no similar result on the number of additions (additive complexity). The published work on linear complexity [27] and optimality of the FFT [28] are not enough generic. Furthermore, of most interest would be a lower bound on the number of both additions and multiplications.
Nowadays, arithmetic complexity and computer architecture capabilities (pipeline, cache, memory speed, etc) have a similar (comparable) effect on the FFT algorithm run time performance. A very interesting platform-adaptable FFT system, called FFTW, has been developed by Frigo and Johnson [18], which uses a library of efficient "codelets" and a decomposition method which searches the optimal DFT decomposition. Also, in another work [29], a generic algorithm derives fast versions for a broad class of discrete signal transforms symbolically, including the DFT.
In fact, the computational complexity of FFT algorithms is an intricate combination of arithmetic cost and computer implementation issues due to processor architectures, memory accesses and adequate code design.

2 The Problem

Independently of any kind of known efficient FFT algorithm, the proposed challenge is to develop a mathematical theory on the computational (arithmetic) complexity of the DFT, which leads to lower bounds on the number of elementary operations (additions and multiplications) necessary to compute (1) as functions of the sequence length N. This knowledge will allow us to verify the level of optimality of the available algorithms and if there is room for further research on new fast algorithms and related topics.

References

[1] J. Cooley, J. Tukey, "An algorithm for the machine calculation of complex Fourier series", Math. Computat., vol. 19, pp. 297-301, 1965.

[2] J. McClellan, C. Rader, *Number Theory in Digital Signal Processing*, Englewood Cliffs, Prentice-Hall, 1979.

[3] H. Nussbaumer, *Fast Fourier Transform and Convolution Algorithms*, Springer-Verlag, 2nd ed., 1982.

[4] Richard Blahut, *Fast Algorithms for Digital Signal Processing*, Addison-Wesley, 1985.

[5] C. S. Burrus, T. W. Parks, *DFT/FFT and Convolution Algorithms*, John Wiley & Sons, 1985.

[6] C. Van Loan, *Matrix Frameworks for the Fast Fourier Transform*, Philadelphia, SIAM, 1992.

[7] S. Winograd, "On computing the discrete Fourier transform", Math. Computation, vol. 32, pp. 175-199, Jan. 1978.

[8] S. Winograd, "On the multiplicative complexity of the discrete Fourier transform", Advances in Mathematics, vol. 32, pp. 83-117, May 1979.

[9] S. Winograd, *Arithmetic Complexity of Computation*, SIAM CBMS-NSF Series, No. 33, SIAM, 1980.

[10] M. Vetterli, H. Nussbaumer, "Simple FFT and DCT algorithms with reduced number of operations", Signal Proc., vol. 6, pp. 267-278, Aug. 1984.

[11] M. Vetterli, P. Duhamel, "Split-radix algorithms for length p^m DFT's", IEEE Trans. on ASSP, vol. 37, pp. 57-64, Jan. 1989.

[12] H. Guoy, G.Sittonz, C. Burrus, "The Quick Discrete Fourier Transform", ICASSP'94.

[13] D. Kolba, T. Parks, "A prime factor FFT algorithm using high speed convolution", IEEE Trans. ASSP, vol. 25, pp. 281-294, Aug. 1977.

[14] A. Saidi, "Decimation-in-time-frequency FFT algorithm", in Proc. ICASSP'94, pp. 453-456, Apr. 1994.

[15] P. Duhamel, H. Hollmann, "Split radix FFT algorithm", Elect. Letters, vol. 20, pp. 14-16, Jan. 1984.

[16] P. Duhamel, "Algorithms meeting the lower bounds on the multiplicative complexity of length 2^n DFT's and their connection with practical algorithms", IEEE Trans. Signal Processing, vol. 38, n°. 9, pp. 1504-1511, Sept. 1990.

[17] D. Tukahashi, "An Extended Split-Radix FFT Algorithm", IEEE Signal Processing Letters, vol. 8, no. 8, pp. 145-147, May 2001.

[18] M. Frigo, S. Johnson, "FFTW: An Adaptive Software Architecture for the FFT", Proc. ICASSP'98, vol.3, pp.1381-1384, 1998.

[19] M. Heideman, C. Burrus, ”On the number of multiplications necessary to compute a length-2n DFT”, IEEE Trans. ASSP, vol. 34, pp. 91-95, Feb. 1986.

[20] Michael Heideman, *Multiplicatice Complexity, Convolution, and the DFT*, Springer-Verlag, 1988.

[21] H. Sorensen, M. Heideman, C. Burrus, ”On computing the split-radix FFT”, IEEE Trans. ASSP, vol. 34, pp. 152-156, Feb. 1986.

[22] C. Burrus, P. Eschenbacher, ”An in-place, in-order prime factor FFT algorithm”, IEEE Trans. ASSP, vol. 29, pp. 806-817, Aug. 1981.

[23] C. Temperton, ”Nesting strategies for prime factor FFT algorithms”, J. Comp. Physics, vol. 82, pp. 247-268, June 1989.

[24] L. Rabiner, R. Schafer, C. Rader, ”The chirp z-transform algorithm”, IEEE Trans. Audio Electroacoustics, vol. AU-17, pp. 86-92, June 1969.

[25] U. Baum, M. Clausen, B. Tietz, ”Improved Upper Complexity Bounds for the Discrete Fourier Transform”, Communication and Computing 2, pp. 35-43, 1991.

[26] D. Maslen, D. Rockmore, ”Generalized FFTs-a Survey of Some Recent Results”, Discrete Math. Theoret. Comput. Sci., vol. 28, Amer. Math. Soc., pp. 183–237, 1997.

[27] J. Morgenstern, ”Note on a lower bound of the linear complexity of Fast Fourier Transform”, J. of ACM, vol. 20, n^o. 2, pp. 305-306, Apr. 1973.

[28] C. Papadimitriou, ”Optimality of the fast Fourier Transform”, Journal of ACM, vol. 26, n^o. 1, pp. 95-102, Jan. 1979.

[29] S. Egner, M. Püschel, ”Automatic Generation of Fast Discrete Signal Transforms”, IEEE trans. Signal Proc., vol. 49, no. 9, Sept. 2001.

Optimization of the dispersion profile in soliton links with dispersion–varying compensating fiber

Henrique J. A. da Silva[*] M. C. Gouveia[†]

Abstract

Recently, it has been shown that the soliton dynamics in fiber links employing compensating fiber with variable dispersion may lead to an improvement of the system performance, suggesting the possibility of an optimal dispersion compensating profile for soliton transmission in periodically amplified systems. The problem proposed here is to find an optimal dispersion profile, based on the variational approach for the solution of the nonlinear Schrödinger equation (NSE) and using a simple system criterion.

Keywords: soliton propagation, dispersion compensation, nonlinear Schrödinger equation.

1 Introduction

The equation that describes the propagation of soliton pulses in periodically amplified systems with variable dispersion is [1]:

$$i\frac{\partial u}{\partial \xi}+\frac{1}{2}d(\xi)\frac{\partial^2 u}{\partial \tau^2}+|u^2|u=-\frac{i}{2}\Gamma u+i\left(\sqrt{G}-1\right)\sum_{m=1}^{N_A}\delta(\xi-m\xi_A)u \tag{1}$$

where N_A is the number of cascaded amplifiers, $\xi_A = Z_A/L_D$ is the normalized amplifier separation, $\Gamma = \alpha L_D$ is the normalized loss coefficient (α is the loss coefficient and L_D is the dispersion length), $G = \exp(\Gamma\xi_A)$ is the gain provided to compensate for the losses, δ is the Dirac function, which indicates the periodic nature of the amplification, and $d(\xi)$ is the variable dispersion profile, given by:

$$d(\xi)=\begin{cases}\dfrac{\beta_2^{\text{DCF}}}{\beta_2^{\text{ave}}}, & \text{for the DCF, and}\\[2ex] \dfrac{\beta_2^{\text{DSF}}}{\beta_2^{\text{ave}}}, & \text{for the DSF.}\end{cases} \tag{2}$$

[*] Instituto de Telecomunicações. E-mail: henrique.silva@co.it.pt

[†] Instituto de Telecomunicações. E-mail: mcag@mat.uc.pt

where DCF and DSF stand for dispersion compensating fibre and dispersion shifted fibre, respectively, and

$$\beta_2^{\text{ave}} = \frac{\beta_2^{\text{DCF}} Z_{\text{DCF}} + \beta_2^{\text{DSF}} Z_{\text{DSF}}}{Z_A} \tag{3}$$

is the average dispersion coefficient.

In order to analyse the propagation of the pulse envelope, it is convenient to write the amplitude u as a function of a fast component, due to loss and periodic amplification, and a slow component, the envelope, through the transformation:

$$u(\xi,\tau) = a(\xi)v(\xi,\tau) \tag{4}$$

By applying this transformation to equation (1), the following equations are obtained:

$$i\frac{\partial v}{\partial \xi} + \frac{1}{2}d(\xi)\frac{\partial^2 v}{\partial \tau^2} + a^2(\xi)|v|^2 v = 0 \tag{5}$$

$$\frac{da}{d\xi} = -\frac{1}{2}\Gamma a + \left(\sqrt{G} - 1\right)\sum_{m=1}^{N_A} \delta(\xi - m\xi_A)a \tag{6}$$

Equation (6) has the following solution [1]:

$$a(\xi) = \begin{cases} a_0 \exp\left[-\frac{1}{2}\Gamma\left(\xi - m\xi_A\right)\right], & \text{for } m\xi_A < \xi < (m+1)\xi_A \\ a_0, & \text{for } \xi = m\xi_A \end{cases} \tag{7}$$

with

$$a_0 = \left[\frac{\Gamma\xi_A}{1 - \exp(-\Gamma\xi_A)}\right]^{1/2} \tag{8}$$

It should be noticed that, if the amplifier spacing Z_A is chosen much smaller than the dispersion length L_D, then $\xi_A = Z_A/L_D \ll 1$ and $a(\xi)$ is a function of the fast variation in each interval between amplifiers. With a suitable choice of the input power, the soliton shape will deviate very little from its shape in a lossless medium, and may be amplified hundreds of times with a behaviour very close to ideal propagation. The energy of the soliton in this propagation regime is the average energy in one amplification stage, and is therefore called average soliton regime.

2 Variational Method

2.1 Lagrangean of the system

The application of variational calculus to the solution of the nonlinear Schrödinger equation (NSE) was proposed for the first time by Anderson [2], in 1983. Since then, variational calculus has been a powerful tool in the study of soliton dynamics. It is worth noting that this method may be employed in systems with dispersion management, since the energy is conserved in average.

The differential equation that describes the optical field propagation is equation (5). This equation must be rewritten through variational calculus equations. The first step is to write the Euler-Lagrange equations for the system under analysis. Considering that the Lagrangean L of the system depends on the optical field and its derivatives relative to the propagation coordinate and to the temporal coordinate:

$$L = L\left(v, v^*, \frac{\partial v}{\partial \xi}, \frac{\partial v^*}{\partial \xi}, \frac{\partial v}{\partial \tau}, \frac{\partial v^*}{\partial \tau}\right) \tag{9}$$

the Euler-Lagrange equations are given by:

$$\frac{\partial L}{\partial v} - \left[\frac{\partial}{\partial \xi}\left(\frac{\partial L}{\partial v_\xi}\right) + \frac{\partial}{\partial \tau}\left(\frac{\partial L}{\partial v_\tau}\right)\right] = 0 \tag{10a}$$

$$\frac{\partial L}{\partial v^*} - \left[\frac{\partial}{\partial \xi}\left(\frac{\partial L}{\partial v^*_\xi}\right) + \frac{\partial}{\partial \tau}\left(\frac{\partial L}{\partial v^*_\tau}\right)\right] = 0 \tag{10b}$$

where

$$v_\xi = \frac{\partial v}{\partial \xi},\ v^*_\xi = \frac{\partial v^*}{\partial \xi},\ v_\tau = \frac{\partial v}{\partial \tau},\ v^*_\tau = \frac{\partial v^*}{\partial \tau}$$

The Lagrangean of the system must be such that, replaced in the Euler-Lagrange equations (10), produces the original NSE (5) or its complex conjugate. Using this principle, the Lagrangean of the system was found to be given by [3]:

$$L = \frac{i}{2}\left[v\frac{\partial v^*}{\partial \xi} - v^*\frac{\partial v}{\partial \xi}\right] + \frac{1}{2}d(\xi)\left|\frac{\partial v}{\partial \tau}\right|^2 - \frac{1}{2}c(\xi)|v|^4 \tag{11}$$

It may be easily verified that the NSE is obtained, if this Lagrangean is used in equation (10b). It should be noted, however, that here the variable is $c(\xi) = a^2(\xi)$.

2.2 *Ansatz* and average Lagrangean

Having defined the Lagrangean of the system and verified its validity through the Euler-Lagrange equations, an important step is to identify a trial function, or *ansatz*, which will be used as an approximation for the exact solution. The final solution, found through the variational method, is as more accurate as the *ansatz* is closer to the exact solution. In order to take into consideration the main system parameters that rule soliton propagation, the following *ansatz* was chosen [4]:

$$v(\xi,\tau) = \eta(\xi)\sec h\left[\eta(\xi)\left(\tau + \Omega(\xi)\xi - q(\xi)\right)\right] \cdot \exp\left[-i\Omega(\xi)\tau + i\left(\eta(\xi)^2 - \Omega(\xi)^2\right)\frac{\xi}{2} + i\phi(\xi)\right] \tag{12}$$

where $\eta(\xi)$, $\Omega(\xi)$, $q(\xi)$, $\phi(\xi)$ represent the amplitude, frequency, phase and position of the soliton, respectively.

Once defined the trial function, a new calculation of the system Lagrangean is required, starting with the chosen *ansatz*. In this way, the Lagrangean becomes a function of the *ansatz* parameters.

Replacing equation (12) in equation (11), it is found that:

$$L = \eta^2 \mathrm{sech}(x) \left\{ -\tau \frac{\partial \Omega}{\partial \xi} + \frac{1}{2} \left[(\eta^2 - \Omega^2) + 2\xi\eta \frac{\partial \eta}{\partial \xi} - 2\xi\eta \frac{\partial \Omega}{\partial \xi} \right] \right.$$
$$\left. + \frac{\partial \psi}{\partial \xi} + \frac{i}{4} d(\xi) \left[\Omega^2 + \eta^2 \tanh(x)^2 \right] - \frac{i}{4} c(\xi) \eta^2 \mathrm{sech}(x)^2 \right\} \quad (13)$$

where $x = \eta(\tau + \Omega\xi - q)$.

The variational principle establishes that:

$$\delta \iint L d\xi d\tau = 0 \quad (14)$$

It is possible to reduce the variational principle to only one dimension, by integrating the Lagrangean over all time τ. For this it is useful to define the reduced or average Lagrangean, given by:

$$\langle L \rangle = \int_{-\infty}^{+\infty} L d\tau \quad (15)$$

Using equation (13), the following average Lagrangean is obtained:

$$\langle L \rangle = -2q \frac{\partial \Omega}{\partial \xi} + 2\Omega\xi \frac{\partial \Omega}{\partial \xi} + \eta^3 - \eta\Omega^2 + 2\xi\eta^2 \frac{\partial \eta}{\partial \xi} - 2\xi\eta^2 \frac{\partial \Omega}{\partial \xi} + \eta \frac{\partial \phi}{\partial \xi} + \frac{i}{2} \eta d(\xi) \Omega^2 + \frac{2}{3} \eta^3 - \frac{i}{3} \eta^3 c(\xi) \quad (16)$$

The variational principle is then replaced by the reduced variational principle, expressed by:

$$\delta \int \langle L \rangle d\xi = 0 \quad (17)$$

2.3 *Ansatz* parameters

The reduced Lagrangean defines a Hamiltonian system with finite dimension. Therefore, the equations that describe the ansatz parameters can be obtained from the canonical Hamilton equations, given by:

$$\frac{dx_j}{dz} = \frac{\partial H}{\partial p_j} \quad (18)$$

$$\frac{dp_j}{dz} = \frac{\partial H}{\partial x_j} \quad (19)$$

where the generalized system momentum, p, and the system Hamiltonian, H, for the reduced variational problem, are given respectively by:

$$p_j = \frac{\partial \langle L \rangle}{\partial \left[\frac{dx_j}{dz} \right]} \tag{20}$$

$$H = \sum_{j=1}^{N} p_j \frac{dx_j}{dz} - \partial \langle L \rangle \tag{21}$$

The generalized coordinates x_j are those that, in the reduced Lagrangean, have derivative relative to z. In the case under analysis, $z = \xi$ and the generalized coordinates of the problem are $x_j = \Omega, \eta, \phi$, for $j = 1, 2, 3$. Therefore:

$$p_1 = \frac{\partial \langle L \rangle}{\partial \left[\frac{d\Omega}{d\xi} \right]} = -2q + 2\Omega\xi - 2\xi\eta^2 \tag{22}$$

$$p_2 = \frac{\partial \langle L \rangle}{\partial \left[\frac{d\eta}{d\xi} \right]} = 2\xi\eta^2 \tag{23}$$

$$p_3 = \frac{\partial \langle L \rangle}{\partial \left[\frac{d\phi}{d\xi} \right]} = \eta \tag{24}$$

Using equation (21), the following Hamiltonian is found:

$$H = \eta\Omega^2 - \eta^3 - \frac{2}{3}\eta^3 - \frac{i}{2}\eta\Omega^2 d(\xi) + \frac{i}{3}\eta^3 c(\xi) \tag{25}$$

Using this in the Hamilton equations (18, 19), we obtain:

$$\frac{d\Omega}{d\xi} = \frac{\partial H}{\partial \left(-2q + 2\Omega\eta - 2\xi\Omega^2\right)} \Rightarrow \frac{d\Omega}{d\xi} = 0 \Rightarrow \Omega = \Omega_0 = \text{const.} \tag{26}$$

$$-\frac{dq}{d\xi} + \xi\frac{d\Omega}{d\xi} + \Omega - \eta^2 - 2\xi\eta\frac{d\eta}{d\xi} + \eta\Omega - \frac{i}{2}\eta\Omega d(\xi) = 0 \tag{27}$$

$$\frac{d\phi}{d\xi} = \frac{\partial H}{\partial \phi} \Rightarrow \frac{d\phi}{d\xi} = -3\eta^2 + \Omega^2 - \frac{i}{2}\Omega^2 d(\xi) - 2\eta^2 + i\eta^2 c(\xi) \tag{28}$$

$$\frac{d\eta}{d\xi} = -\frac{\partial H}{\partial \phi} \Rightarrow \frac{d\eta}{d\xi} = 0 \Rightarrow \eta = \eta_0 = \text{const.} \tag{29}$$

$$\frac{d\eta}{d\xi} = \frac{\partial H}{\partial (2\xi\eta^2)} \Rightarrow \frac{d\eta}{d\xi} = 0 \Rightarrow \eta = \eta_0 = \text{const.} \tag{30}$$

$$\frac{d(2\xi\eta^2)}{d\xi} = -\frac{\partial H}{\partial \eta} \Rightarrow 3\eta^2 - \Omega^2 + \frac{i}{2}d(\xi)\Omega^2 - i\eta^2 c(\xi) = 0 \tag{31}$$

Equation (27) may now be rewritten using equation (26):

$$\frac{dq}{d\xi} = \Omega - \eta^2 + \eta\Omega - \frac{i}{2}\eta\Omega d(\xi) \tag{32}$$

In summary, the equations for the soliton parameters are:

$$\frac{d\eta}{d\xi} = 0 \Rightarrow \eta = \eta_0 = \text{const.} \tag{33}$$

$$\frac{d\Omega}{d\xi} = 0 \Rightarrow \Omega = \Omega_0 = \text{const.} \tag{34}$$

$$\frac{d\phi}{d\xi} = -2\eta_0^2 \tag{35}$$

$$\frac{dq}{d\xi} = \Omega_0 - \eta_0^2 + \eta_0\Omega_0 - \frac{i}{2}\eta_0\Omega_0 d(\xi) \tag{36}$$

From these results it may be concluded that, for this type of propagation:

- The amplitude of the envelope of the slow variation function is constant;
- The soliton frequency is constant;
- The phase varies linearly with propagation distance;
- The position of the soliton during propagation depends on the dispersion map and on the initial amplitude and frequency values.

3 Optimization problem

In [5], a new dispersion compensation scheme is reported where the uniform DCF is replaced by fiber with decreasing and increasing dispersion profiles. In this study, three cases were considered for the dispersion coefficient of the compensating fiber: uniform (conventional DCF), exponential decreasing, and exponential increasing. In order to do a fair comparison as general as possible, the average dispersion of the dispersion-varying compensating fiber (DVCF) was set to be equal to the dispersion of the uniform fiber.

The NSE has been solved with the variational approach with a more general *ansatz* of the type [3]:

$$Q(z,\tau) = a(z) f\left[\tau/b(z)\right] \exp\left[i\lambda(z) + i\mu(z)\tau^2\right] \tag{37}$$

where $f(x)$ is an arbitrary pulse waveform, $a(z)$, $b(z)$, $\lambda(z)$ and $\mu(z)$ account for the complex amplitude, pulse width, phase and pulse chirp, respectively, and $z = Z/L_{NL} = \xi L_D/L_{NL}$ is the distance normalized by the fiber nonlinear length L_{NL}. This normalization is more convenient to analyse the interaction between the fiber nonlinearity and the residual dispersion given by equation (3), which is significant on the scale of the complete transmission line, which may have a length with the same order of magnitude of $L_{NL} \gg Z_A$.

By applying the variational principle, the NSE reduces to a set of ordinary differential equations:

$$a(z)^2 b(z) = N^2 = \text{const.} \tag{38}$$

$$\frac{db(z)}{dz} = \frac{2L_{NL}d(z)}{L_D} b(z)\mu(z) \tag{39}$$

$$\frac{d\mu(z)}{dz} = \frac{L_{NL}d(z)C_1}{2L_D b(z)^4} - \frac{c(z)a(z)^2 C_2}{b(z)^2} - \frac{2L_{NL}d(z)}{L_D}\mu(z)^2 \tag{40}$$

where C_1 and C_2 are constants that depend on the shape of the input pulse:

$$C_1 = \frac{\displaystyle\int_{-\infty}^{+\infty} \left| \frac{df(x)}{dx} \right| dx}{\displaystyle\int_{-\infty}^{+\infty} x^2 \left| f(x) \right|^2 dx} \tag{41}$$

$$C_2 = \frac{\displaystyle\int_{-\infty}^{+\infty} \left| f(x) \right|^4 dx}{\displaystyle\int_{-\infty}^{+\infty} x^2 \left| f(x) \right|^2 dx} \tag{42}$$

For $f(x) = \text{sech}(x)$ pulses, $C_1 = 2C_2 = 4/\pi^2$, and for Gaussian pulses $f(x) = \exp(-x^2)$, $C_1 = 4$ and $C_2 = 1\sqrt{2}$. Function $c(z)$ is related to the fiber losses and amplifier gain:

$$c(z) = \exp\left[2\int_0^z g(z')dz'\right] \tag{43}$$

with

$$g(z') = L_{NL}\left\{-\gamma + [\exp(\gamma Z_A) - 1]\sum_{k=1}^{N} \delta(z' - z_k)\right\} \tag{44}$$

where $\gamma = 0.115\alpha$ describes the fiber losses, Z_A is the amplification period, and $z_k = kz_A$ are the amplifier locations.

In the framework of project TRANSPARENT (POSI/34559/CPS/2000) [5], other dispersion profiles were considered besides the exponential one, namely the Gaussian and the hyperbolic profiles, in order to compare the soliton dynamics for different kinds of dispersion-varying compensating fiber (DVCF). For these profiles, the increasing and decreasing cases were also considered, and a good agreement was observed between the solution obtained with the variational equations (38) to (40) and the direct numerical solution of the NSE with the split-step Fourier method (SSFM) [6].

The results reported in [5] show that the pulse dynamic behavior depends on the dispersion profile of the compensating fiber, and that the chirp acquired by the pulse as it propagates is responsible for the different evolution of the pulse with the three profiles considered. Therefore, the preliminary study presented indicates that it is possible to improve the system performance with a proper control of the chirp parameter evolution, and that this may be achieved through the use of DVCF instead of conventional DCF.

The assumption concerning system performance made here is that the pulse dynamics whose amplitude and width values deviate less from the input amplitude and width values would be less detrimental to the system performance. With this simple criterion, the optimization problem may be reduced to a simpler minimization problem.

Other system parameters, namely the amplifier span, the DVCF length and the pulse width, might also be considered in this optimization problem, with the objective of finding the best operating regions for the propagation of solitons in long-haul links. This would however make the problem much harder, due to the higher number of variables required.

As a further motivation for this study, it should be noted that the use of a DVCF with the same average dispersion as its DCF counterpart might be applied to any dispersion map. This would mean that the performance of a system employing a conventional DCF with an arbitrary dispersion value might always be improved through the use of its DVCF counterpart (i.e., a DVCF with average value of the dispersion coefficient equal to the uniform dispersion coefficient of the DCF), and even more so if an optimal dispersion profile could be found.

4 Soliton dynamics

The variational equations (38) to (40) include both the fast dynamics, due to loss and amplification, and the slow dynamics, due to the fiber nonlinearity and residual dispersion. By introducing the variable $v(z) = \mu(z)b(z)$, we have, from (38) and (39):

$$\frac{dv}{dz} = \frac{L_{NL}d(z)C_1}{2L_D b^3} - \frac{c(z)N^2C_2}{b^2} \tag{45}$$

To obtain the solution for the linear case we must consider only the dispersive effects. Therefore, for *sech* pulses (45) reduces to:

$$\frac{dv}{dz} = \frac{L_{NL}d(z)C_1}{2L_D b^3} = \frac{2L_{NL}d(z)C_1}{L_D b^3 \pi^2} \tag{46}$$

By dividing (39) by (46) and integrating (through separation of variables), we obtain:

$$v^2 = \frac{1}{\pi^2}\left(1 - \frac{1}{b^2}\right) \tag{47}$$

By substitution of this result in (39) and integration, the linear solution for $b(z)$ is then obtained:

$$b_l^2 = 1 + \left[\frac{4}{\pi}R(z)\right]^2 \tag{48}$$

where

$$\frac{dR(z)}{dz} = \frac{L_{NL}}{2L_D}d(z) \tag{49}$$

The slow dynamics due to the fiber nonlinearity and the residual dispersion can be considered as perturbations of the linear solution. Equations (39) and (45) may be linearized about the linear solution by assuming $b = b_l + \tilde{b}_{\pm}$, with $\tilde{b}_{\pm} \ll b_l$ (+ and − corresponding to SMF and DCF, respectively; for simplicity, these subscripts are not used in the following):

$$\frac{d\tilde{b}}{dz} = \frac{2L_{NL}}{L_D} d(z)\tilde{b}\mu = \frac{2L_{NL}}{L_D} d(z)\tilde{v} \tag{50}$$

$$\frac{d\tilde{v}}{dz} = -\frac{6L_{NL}d(z)}{\pi^2 L_D b_l^4}\tilde{b} - \frac{2c(z)N^2}{\pi^2 b_l^2} \tag{51}$$

The initial conditions at $z = 0$ are $\tilde{b} = 0$ and $\tilde{v} = 0$. Therefore, the system formed by these equations can be written in the form:

$$\begin{bmatrix} \dfrac{d\tilde{b}}{dz} \\ \\ \dfrac{d\tilde{v}}{dz} \end{bmatrix} = \begin{bmatrix} 0 & k \\ \\ -\dfrac{3k}{\pi^2 b_l^4} & 0 \end{bmatrix} \begin{bmatrix} \tilde{b} \\ \\ \tilde{v} \end{bmatrix} + \begin{bmatrix} 0 \\ \\ -\dfrac{2N^2 c(z)}{\pi^2 b_l^2} \end{bmatrix} \tag{52}$$

with the initial conditions

$$\begin{bmatrix} \tilde{b}(0) \\ \tilde{v}(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

and where $b_l = b_l(z)$ and

$$k = \frac{2L_{NL}d(z)}{L_D}$$

is constant for DCF and SMF, but varies with z in DVCF. This system is of the type $y'(z) = A(z)y(z) + e(z)$.

According to theorem 8.3 in [7], if $\phi(z)$ is a fundamental matrix for the system $y'(z) = A(z)y(z)$, the unique solution for this system will be given by:

$$y(z) = \phi(z)\phi^{-1}(z_0)\eta + \int_{z_0}^{z} \phi(z)\phi^{-1}(s)e(s)ds \tag{53}$$

where $\eta = y(z_0)$.

If $\phi(z)$ is nonsingular for all z and satisfies equation $\phi'(z) = A(z)\phi(z)$, then it is a fundamental matrix for the system $y' - Ay = 0$. This matrix exists if $A(z)$ is continuous. Moreover, if $x_i(z)$, with $i = 1, \ldots, n$, are solutions of $y' - Ay = 0$, then $\phi(z) = \begin{bmatrix} x_1 & x_2 & \ldots x_n \end{bmatrix}$ satisfies $\phi'(z) = A(z)\phi(z)$.

In order to determine $\phi(z)$ for system (52), we consider:

$$x_1 = \begin{bmatrix} \cosh(Mz) \\ \frac{M}{k}\sinh(Mz) \end{bmatrix} \text{ and } x_2 = \begin{bmatrix} \frac{k}{M}\sinh(Mz) \\ \cosh(Mz) \end{bmatrix} \tag{54}$$

where k is defined as in (52) and $M = i\dfrac{\sqrt{3}k}{\pi b_l^2}$.

Since x_1 and x_2 are solutions of $y' - Ay = 0$, then:

$$\phi(z) = \begin{bmatrix} \cosh(Mz) & \frac{k}{M}\sinh(Mz) \\ \frac{M}{k}\sinh(Mz) & \cosh(Mz) \end{bmatrix} \tag{55}$$

such that $|\phi(z)| = 1$ and:

$$\phi^{-1}(z) = \begin{bmatrix} \cosh(Mz) & \frac{k}{M}\sinh(Mz) \\ -\frac{M}{k}\sinh(Mz) & \cosh(Mz) \end{bmatrix} \tag{56}$$

with $\phi^{-1}(0) = I$.

Then, with the initial conditions

$$y(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

solution (53) takes the form:

$$y(z) = \int_0^z \phi(z)\phi^{-1}(s)e(s)ds \tag{57}$$

Therefore:

$$\begin{bmatrix} \tilde{b}_- \\ \\ \tilde{v}_- \end{bmatrix} = \begin{bmatrix} -\dfrac{2kN^2}{\pi^2}\displaystyle\int_0^z \frac{1}{M}\sinh\left[Mz - Ms\right]\frac{c(s)}{b_l^2(s)}ds \\ \\ -\dfrac{2N^2}{\pi^2}\displaystyle\int_0^z \frac{1}{M}\sinh\left[Mz + Ms\right]\frac{c(s)}{b_l^2(s)}ds \end{bmatrix} \tag{58}$$

if k is constant, as happens with DCF and SMF.

5 Conclusion

Result (58) is valid for uniform DCF and for SMF, which have constant dispersion. If the DCF is replaced by a DVCF with arbitrary dispersion profile, $A(z)$ becomes a matrix with variable coefficients. In this case, $\phi(z)$ in solution (53) may be obtained by applying the Wei–Normann theorem, taking into consideration that $A(z)$ belongs to the Lie algebra of 2×2 matrices with null trace.

At the time of writing we are still working on the solution for this case, which we expect will enable the identification of the DVCF dispersion profile that minimizes the perturbation of the soliton parameters.

References

[1] A. Hasegawa and Y. Kodama, *Solitons in Optical Telecommunications.* Oxford: Clarendon Press, 1995.

[2] D. Anderson, “Variational approach to nonlinear pulse propagation in optical fibers,” *Phys. Rev. A*, vol. 27, no. 6, p. 3135, 1983.

[3] I. Gabitov, E. G. Shapiro, and S. K. Turitsyn, “Optical pulse dynamics in fiber links with dispersion compensation,” *Opt. Comm.*, no. 134, p. 317, 1997.

[4] G. P. Agrawal, *Fiber-Optic Communication Systems.* New York: John Willey and Sons, 1997.

[5] A. M. Melo, M. C. Gouveia, and H. J. A. da Silva, “Soliton dynamics in dispersion varying compensating fibers (DVCF),” in *Advances in Communications and Software Technologies*, (Skiathos, Greece), pp. 127–130, WSEAS Int. Conf. on Global Optical and Wireless Networks 2002 (GOWN ’02), WSEAS Press, 25-28 September 2002.

[6] G. P. Agrawal, *Nonlinear Fiber Optics.* London: Academic Press, 1995.

[7] P. Waltman, *A Second Course in Elementary Differential Equations.* Academic Press, 1986.

Effective Permittivity of 1D- 2D- and 3D-Wire Structures: An Analytical Approach

Mário Silveirinha[*] Carlos A. Fernandes[†]

Abstract

In this paper we discuss the propagation of electromagnetic waves in an artificial medium that consists of a regular array of thin metallic wires. The problem involves the calculation of the eigenvalues of a differential operator. The challenge is to obtain an approximate closed-form solution for the first few eigenvalues (the long wavelength limit). The proposed solution is based on a variational formulation and physical considerations. We homogenize the periodic structure and we prove that it can be described in terms of an effective permittivity. Our results show that the effective permittivity of the wire medium depends explicitly on the wave vector, and thus that the wire medium is not isotropic in the long wavelength limit. We compare our model with the standard plasma model commonly accepted in the literature, and we discuss the physical implications of the results.

Keywords: metamaterials, left-handed media, wire media, homogenization theory.

1 Introduction

In recent years the propagation of electromagnetic waves in periodic dielectric/metallic structures has received great attention [1, 2]. These structures consist of a three-dimensional regular array of metallic or dielectric particles (inclusions) embedded in a host homogeneous material. It has been shown that these artificial materials possess remarkable properties that find many applications in several branches of physics and engineering [1]. The main features of the interaction of composite materials with electromagnetic waves depend on the wavelength of operation.

For wavelengths smaller or comparable to the lattice constant (i.e. the spacing between the inclusions) the propagation of electromagnetic waves in the periodic material may be forbidden in certain frequency bands [1]. In general, the frequency bands depend on the polarization and direction of propagation. However, it has been shown [3] that it is possible to design a periodic

[*] Instituto de Telecomunicações. E-mail: mario.silveirinha@co.it.pt

[†] Instituto de Telecomunicações. E-mail: carlos.fernandes@lx.it.pt

dielectric material in which propagation of electromagnetic waves is completely forbidden, irrespective of the polarization and direction of propagation. These structures were initially investigated in the framework of solid-state physics and electronics. It was suggested that spontaneous emission in semiconductor lasers can be rigorously eliminated if the band-gap of the periodic dielectric structure overlaps the electronic band edge [4]. Spontaneous emission limits the performance of semiconductor lasers and other devices. Other applications have been suggested over the years. These include high-Q electromagnetic cavities and waveguides for short wavelengths at which metals are useless due to strong losses [5], monolithic waveguide filters [6], and the improvement of the radiation characteristic of antennas [7, 8].

For wavelengths much larger than the lattice constant, the properties and characterization of periodic materials are substantially different from those of the band gap regime described above. In the long wavelength limit, the propagation of electromagnetic waves in the artificial material can be described from an average perspective, in analogy with propagation in matter (due to this reason composite structures in the long wavelength limit are also known as “metamaterials”). In this way, it is possible to homogenize the metamaterial and to define effective parameters that completely characterize the average electromagnetic fields (in the simpler formulation the effective parameters are the effective permittivity and permeability [9]; in the most general case the medium is bianisotropic [10]). Fifty years ago, these facts motivated an intense research on artificial dielectrics [11]. Recently, the investigation of these structures regained interest after the extraordinary breakthrough that it is possible to synthesize a material having simultaneously negative permittivity and permeability over a certain frequency band [12]. These materials are known as left-handed media or double negative materials, and their unconventional electrodynamics was investigated long time before they were actually found [13]. Among other exotic and unexpected properties, double negative materials have negative index of refraction. An interesting implication is that rays refracted at an interface with air bend with a negative transmission angle. A remarkable consequence is that negative refraction makes a perfect lens [13, 14]. Indeed, as discussed in [14], with a conventional lens the resolution of the image is always limited by the wavelength of light. Quite differently, a slab of material with negative refractive index has the power to focus all Fourier components of an image, even those that do not propagate in a radiative manner (i.e. the Fourier components that decay exponentially in free-space).

The composite structure that was first shown to have a negative index of refraction [12], consists of a periodic array of metallic wires and split ring resonators, as illustrated in Figure 1. This structure is not isotropic and strictly speaking the negative index of refraction is *seen* only by electromagnetic waves that propagate in the direction indicated in Figure 1, with magnetic field normal to the split ring resonators. As discussed in [12], the negative permeability effect emerges due to the split ring resonators, whereas the negative permittivity effect emerges due to the metallic wires.

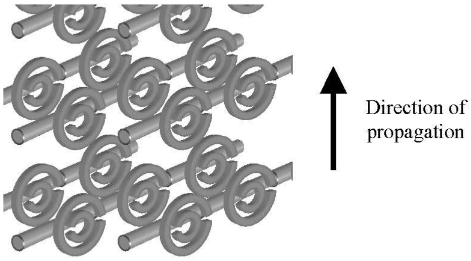


Figure 1: Geometry of the medium proposed by Smith et al. [12], which interacts with electromagnetic waves as a double negative medium

To a first approximation the properties of the composite structure can be described by characterizing the effect of each of its basic components individually. This approach assumes small coupling between the two basic inclusions.

In this paper, we are interested in modeling uniquely the interaction of electromagnetic waves with the array of metallic wires (and other similar structures with increased symmetry described ahead), and to assess the validity of the standard plasma model [12]. In fact, it was recently proved that the standard plasma model is insufficient to describe the electrodynamics of the array of wires because strong spatial dispersion emerges at very long wavelengths [15]. The analysis of [15] is however restricted to the case in which metallic wires are all oriented in the same direction (1D-wire medium). The objective of this paper is to investigate whether this effect emerges or not in other wire structures with increased symmetry. The geometry of these structures is depicted in Figure 2. The relevance and motivation of the study is that the natural solution to synthesize an isotropic metamaterial with negative permittivity is based on the 3D-wire medium configuration depicted in Figure 2. Is such a structure really isotropic at long wavelengths? Later in the paper we will give the answer to this fundamental question.

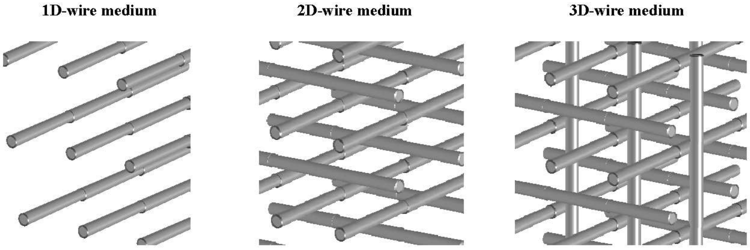


Figure 2: Geometry of the 1D- 2D- and 3D- wire medium

2 Propagation of electromagnetic waves in periodic media

In this section we discuss the propagation of electromagnetic waves in periodic media, and the mathematical formulation of the problem. For simplicity, we restrict the discussion to media with metallic inclusions (perfect conductors). A three-dimensional periodic medium is invariant to translations along three independent vectors $\mathbf{a}_1$, $\mathbf{a}_2$ and $\mathbf{a}_3$, which are known as the lattice primitive vectors [1]. The unit cell $\Omega = \{\alpha_1\mathbf{a}_1 + \alpha_2\mathbf{a}_2 + \alpha_3\mathbf{a}_3 : \ |\alpha_i| \leq \frac{1}{2}\}$ completely defines the geometry of the structure. In a metallic crystal the unit cell consists of a homogeneous dielectric region, and a metallic domain D. The boundary of domain D is surface ∂D, and the outward unit vector normal to D is $\hat{\mathbf{v}}$. The translation of D into the lattice point $\mathbf{r}_1 = i_1\mathbf{a}_1 + i_2\mathbf{a}_2 + i_3\mathbf{a}_3$ is $\partial D_{\mathbf{I}}$, where $\mathbf{I} = (i_1, i_2, i_3)$ is a multi-index of integers.

The unit cell is depicted in Figure 3, assuming the particular case in which the periodic medium is the 3D-wire medium depicted in Figure 2. In this situation, the primitive vectors are parallel to the coordinate axes and such that $\alpha = |\mathbf{a}_1| + |\mathbf{a}_2| + |\mathbf{a}_3|$ (i.e. the lattice is simple cubic [1]). The α parameter is the spacing between adjacent parallel wires, and is referred to as the lattice constant. The spacing between adjacent orthogonal wires is assumed to be half-lattice constant.

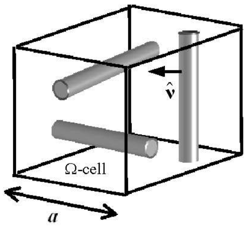


Figure 3: Unit cell for the 3D-wire medium. The lattice constant is α.

The propagation of electromagnetic waves in a periodic structure is described in terms of a theory of bands. An arbitrary solution of Maxwell-Equations [16] in the periodic structure can be decomposed into electromagnetic Floquet modes. The Floquet modes are the analogue of plane waves in free-space, and are characterized by a wave vector $\mathbf{k}$. The wave vector is restricted to the so-called Brillouin zone [1] (e.g. in case of a simple cubic lattice the Brillouin zone is the cube $\left[-\frac{\pi}{\alpha}, \frac{\pi}{\alpha}\right]^3$).

The objective is to compute the electromagnetic Floquet modes $(\mathbf{E}, \mathbf{H})$ of the periodic structure. A Floquet mode associated with wave vector $\mathbf{k}$ is a solution of the (frequency-dependent) Maxwell-Equations, such that $(\mathbf{E}, \mathbf{H}) \exp(j\mathbf{k} \cdot \mathbf{r})$ is periodic (i.e. invariant to translations along the primitive vectors). Thus, an electromagnetic mode must satisfy the following equations:

$$\nabla \times \mathbf{E} = -j\beta Z_0 \mathbf{H}, \text{ dielectric region} \tag{1a}$$

$$\nabla \times \mathbf{H} = j\frac{\beta}{Z_0}\mathbf{E}, \text{ dielectric region} \tag{1b}$$

$$\hat{\mathbf{v}} \times \mathbf{E} = \mathbf{0}, \text{ on } \partial D_1 \tag{1c}$$

$$(\mathbf{E}, \mathbf{H}) \exp(j\mathbf{k} \cdot \mathbf{r}), \text{ is periodic} \tag{1d}$$

where $\mathbf{k} = (k_1, k_2, k_3)$ is the wave vector, $j = \sqrt{-1}$, Z_0 is the impedance of free-space, $\beta = \frac{\omega}{c}$ is the free-space wave number, ω is the angular frequency, and c is the velocity of light in vacuum. Equations (a) and (b) are the frequency-dependent Maxwell-Equations, (c) is the boundary condition at the metallic interfaces, and (d) is the Floquet wave condition.

For a given wave vector $\mathbf{k}$, system (1) has non-trivial solutions only for a countable set of resonant wave numbers (i.e. it is an eigenvalue problem). The resonant wave numbers $\beta_n = \beta_n(\mathbf{k})$ $n = 1, 2, \ldots$, form the so-called band structure of the metallic crystal. The calculation of the band structure of a periodic medium is a difficult problem. No analytical solutions are in general available. Thus we have to resort to numerical methods [17, 3], which are computationally demanding and give no insight of the physical problem.

The objective of the first part of this paper sections 3 and 4 is to obtain an approximate analytical formula for the first few bands (i.e. eigenvalues) of eigensystem (1) assuming that the wire radius, r_w , is very small. Thus, the metallic regions (i.e. the high-permittivity regions) are extremely localized in space. These results will allow us to characterize the average electromagnetic fields in terms of an effective permittivity dyadic [9] in section 5. In section 6, we discuss the physical implications of the results, and in section 7 we draw the conclusions.

3 Variational Formulation

In this section, we derive a variational formulation for problem (1). To this end, we obtain first an integral representation for the electric field $\mathbf{E}$.

To begin with, we introduce the lattice Green function $\Phi_p = \Phi_p(\mathbf{r}|\mathbf{r}')$, which is the Floquet solution of the following equation [18, 19]:

$$\nabla^2 \Phi_p + \beta^2 \Phi_p = -\sum_{\mathbf{I}} \delta(\mathbf{r} - \mathbf{r}' - \mathbf{r}_{\mathbf{I}}) e^{-j\mathbf{k}\cdot(\mathbf{r}-\mathbf{r}')} \tag{2}$$

where $\mathbf{I} = (i_1, i_2, i_3)$ is a multi-index of integers, $\mathbf{r} = (x_1, x_2, x_3)$ is the observation point, $\mathbf{r}' = (x_1', x_2', x_3')$ is a source point, $\mathbf{r}_1 = i_1\mathbf{a}_1 + i_2\mathbf{a}_2 + i_3\mathbf{a}_3$ is a lattice point, and δ is Dirac's distribution. The lattice Green function can be efficiently evaluated as explained in [18]. In this paper we shall consider instead the so-called spectral representation of the Green function, which is obtained by expanding Φ_p in a Fourier series. The result is the following slow converging series:

$$\Phi_p(\mathbf{u}) = \frac{1}{V_{\text{cell}}} \sum_{\mathbf{J}} \frac{e^{-j\mathbf{k_J}\cdot\mathbf{u}}}{|\mathbf{k_J}|^2 - \beta^2}, \quad \mathbf{k_J} = \mathbf{k} + \mathbf{k_J^0} \tag{3}$$

where $\mathbf{u} = \mathbf{r} - \mathbf{r}'$, $V_{\text{cell}} = |\mathbf{a}_1 \cdot \mathbf{a}_2 \times \mathbf{a}_3|$ is the volume of the unit cell, $\mathbf{J} = (j_1, j_2, j_3)$ is a multi-index of integers, $\mathbf{k_J^0} = j_1\mathbf{b}_1 + j_2\mathbf{b}_2 + j_3\mathbf{b}_3$, and $\mathbf{b}_1$, $\mathbf{b}_2$ and $\mathbf{b}_3$ are the reciprocal lattice primitive vectors defined by the relations $\mathbf{a}_n \cdot \mathbf{b}_m = 2\pi\delta_{n,m}$, $n, m = 1, 2, 3$ ($\delta_{n,m}$ is Kronecker's delta symbol, which equals 1 if $n = m$ and 0 otherwise). Using (1c), (2) , and arguments similar to those employed in [20, pp.151], for the case in which the Green function is the free-space kernel, we can readily prove that the electric field has the following integral representation:

$$\mathbf{E}(\mathbf{r}) = \frac{Z_0}{j\beta}\nabla \times \nabla \times \int\limits_{\partial D} \mathbf{J}_c(\mathbf{r}')\Phi_p(\mathbf{r}|\mathbf{r}')ds', \quad \mathbf{r} \text{ in the dielectric region} \tag{4}$$

In the above, the surface integral is over the primed coordinates (the integration is over the boundary of the metallic region in the unit cell), and $\mathbf{J}_c = \hat{\mathbf{v}} \times \mathbf{H}$. Physically, $\mathbf{J}_c$ is the surface current that flows over the metallic surface. Using the vector identity $\nabla \times \nabla\times = \nabla\nabla \cdot -\nabla^2$ and (2), we can rewrite (4) as:

$$\mathbf{E}(\mathbf{r}) = \frac{Z_0}{j\beta}\nabla \int\limits_{\partial D} \nabla_s \cdot \mathbf{J}_c(\mathbf{r}')\Phi_p(\mathbf{r}|\mathbf{r}')ds' - j\beta Z_0 \int\limits_{\partial D} \mathbf{J}_c(\mathbf{r}')\Phi_p(\mathbf{r}|\mathbf{r}')ds' \tag{5}$$

where $\nabla_s\cdot$ stands for the surface divergence of a tangential vector density.

Let $\mathbf{w}$ be an arbitrary tangential vector density defined over ∂D. Letting the observation point $\mathbf{r}$ approach ∂D in (5), and using the boundary condition (1c), we obtain that:

$$0 = \mathbf{w}(\mathbf{r}) \cdot \nabla \int\limits_{\partial D} \nabla_s \cdot \mathbf{J}_c(\mathbf{r}')\Phi_p(\mathbf{r}|\mathbf{r}')ds' + \beta^2\mathbf{w}(\mathbf{r}) \cdot \int\limits_{\partial D} \mathbf{J}_c(\mathbf{r}')\Phi_p(\mathbf{r}|\mathbf{r}')ds', \quad \mathbf{r} \in \partial D \tag{6}$$

Next, we integrate the above equation over ∂D (in the unprimed coordinates). After simple manipulations we find that:

$$\int\limits_{\partial D}\int\limits_{\partial D} \nabla_s \cdot \mathbf{w}(\mathbf{r})\nabla_s \cdot \mathbf{J}_c(\mathbf{r}')\Phi_p(\mathbf{r}|\mathbf{r}')ds'ds - \beta^2 \int\limits_{\partial D}\int\limits_{\partial D} \mathbf{w}(\mathbf{r}) \cdot \mathbf{J}_c(\mathbf{r}')\Phi_p(\mathbf{r}|\mathbf{r}')ds'ds = 0 \tag{7}$$

Thus, we conclude that if there is an electromagnetic mode associated with wave vector $\mathbf{k}$ and wave number β, then the bilinear form defined by the left-hand side of the previous formula is degenerate. Note that in the above formula the Green function depends on both $\mathbf{k}$ and β.

For a given wave vector $\mathbf{k}$, we can compute the resonant wave numbers $\beta_n = \beta_n(\mathbf{k})$ $n = 1, 2, \ldots$ using the standard approach described next. First, we expand $\mathbf{J}_c$ in a set of basis functions (with unknown coefficients), and replace it in (7). Then, we test the resulting equation with a basis of test functions $\mathbf{w}$. In this way, we obtain an homogeneous linear system for the unknown coefficients. A non-trivial solution exists only if the determinant of the linear system vanishes. This occurs only if β is coincident with a resonant wave number.

The approach described before is "numerically" rigorous, but still complicated. In next section, we describe a simplified procedure that will allow us to obtain an approximate analytical solution for the first few eigenvalues.

4 Approximate analytical solution for wire structures

The formalism of the previous section is completely general. We apply now the results to the wire structures depicted in Figure 2. The unit cell of the periodic medium contains N wire (cylindrical) sections with length α and radius r_w, where $N = 1$, 2, or 3 depending on the geometry being that of the 1D- 2D- or 3D-wire medium, respectively. The geometry for the unit cell of the 3D-wire medium is depicted in Figure 3.

We put $\partial D = \partial D_1 \cup \ldots \cup \partial D_n$ where ∂D_n represents the surface of the wire directed along the x_n-axis. The cylindrical section ∂D_n is completely defined by the wire radius r_w, and by a point $\mathbf{r}_{n,0}$ in the wire axis. As referred in section 2, we admit that the distance between adjacent orthogonal wires is half-lattice constant, $\frac{\alpha}{2}$, i.e. orthogonal wires are as far as possible.

Since we admit that the wire radius is very small, we expect, based on physical grounds, that for relatively low frequencies the current will flow along the wire axes as a propagating wave. Thus, it seems reasonable to assume that the surface current over ∂D_n is to a first approximation:

$$\mathbf{J}_c\,|_{\partial D_n} \approx \frac{I_n}{2\pi r_w} e^{-j\mathbf{k}\cdot\mathbf{r}} \hat{\mathbf{u}}_n \tag{8}$$

where $\hat{\mathbf{u}}_n$ is the unit vector along the x_n-axis, and I_n is the (unknown) current over the n-th wire (a complex constant). Within this hypothesis, we know the functional dependence of the current associated with the eigenmodes. In what follows, we explore this fact to calculate the first few resonant wave numbers of the metallic crystal.

To this end, we apply the procedure outlined in the end of the previous section. Hence, we replace $\mathbf{J}_c$ given by (8) in (7), and then we test the resulting equation with several test functions $\mathbf{w}$. The m-th test function is taken equal to $e^{+j\mathbf{k}\cdot\mathbf{r}}\hat{\mathbf{u}}_n$ over ∂D_m , and 0 elsewhere. Since,

$$\nabla_s \cdot \left(\frac{1}{2\pi r_w} e^{-j\mathbf{k}\cdot\mathbf{r}} \hat{\mathbf{u}}_n \right) = \frac{jk_n}{2\pi r_w} e^{-j\mathbf{k}\cdot\mathbf{r}} \tag{9}$$

we easily obtain that:

$$\sum_n g_{m,n}\left(k_m k_n - \beta^2\delta_{m,n}\right) I_n = 0, \text{ where } g_{m,n} = \frac{1}{(2\pi r_w)^2}\int\limits_{\partial D_n}\int\limits_{\partial D_n} e^{+j\mathbf{k}\cdot(\mathbf{r}-\mathbf{r}')}\Phi_p(\mathbf{r}|\mathbf{r}')ds'ds \tag{10}$$

In the previous equation, $m, n = 1, \ldots, N$, with $N = 1$, 2, or 3, depending on the geometry of the wire medium. Next, we calculate the coefficients $g_{m,n}$ defined as above. To this end, we use the spectral representation (3) of the Green function. Since the lattice is simple cubic, it is straightforward to obtain by direct integration that:

$$g_{m,n} = \frac{1}{a}\sum_{\mathbf{J}}\delta_{j_m,0}\delta_{j_n,0}\frac{\left[J_0(|\mathbf{k}_\mathbf{J}^0|r_w)\right]^2}{|\mathbf{k}_\mathbf{J}|^2-\beta^2}e^{-j\mathbf{k}_\mathbf{J}^0(\mathbf{r}_{m,0}-\mathbf{r}_{n,0})} \tag{11}$$

where J_0 is the Bessel function of the first kind and order 0, $\mathbf{J} = (j_1, j_2, j_3)$ is a multi-index of integers, $\mathbf{k}_\mathbf{J}^0 = \frac{2\pi}{\alpha}(j_1, j_2, j_3)$, and the rest of the symbols are defined as in the previous section. Note that if $m = n$ the range of summation can be reduced to a pair of integers, whereas if $m \neq n$ it can be reduced to the set of integers.

Formula (11) is exact. Note that $g_{m,n}$ depends on both $\mathbf{k}$ and β. We recall that the objective is to calculate the first few resonant wave numbers of (1). More specifically, our approximation is supposed to hold in the long-wavelength limit where $|\mathbf{k}|a \ll 1$ and $\beta a \ll 1$. Within this approximation, we can put $\mathbf{k} = 0$ and $\beta = 0$ in each term of the series in (11) , with the exception of the $\mathbf{J} = 0$ term. Thus, we obtain that:

$$g_{m,n} \approx \frac{1}{a}\left(\frac{1}{|\mathbf{k}|^2-\beta^2}+\frac{1}{\beta_{m,n}^2}\right), \quad \frac{1}{\beta_{m,n}^2} = \sum_{\mathbf{J}\neq 0}\delta_{j_m,0}\delta_{j_n,0}\frac{\left[J_0(|\mathbf{k}_\mathbf{J}^0|r_w)\right]^2}{|\mathbf{k}_\mathbf{J}^0|^2}e^{-j\mathbf{k}_\mathbf{J}^0(\mathbf{r}_{m,0}-\mathbf{r}_{n,0})} \tag{12}$$

Notice that $\beta_{m,n}^2$, defined as above is a constant independent of $\mathbf{k}$ and β.

Since we admit that the distance between adjacent orthogonal wires is half-lattice constant, $\frac{\alpha}{2}$, it can be easily verified that for $m \neq n$ we have that,

$$\frac{1}{\beta_{m,n}^2} = \left(\frac{a}{2\pi}\right)^2\sum_{l\neq 0}\frac{\left[J_0(2\pi l r_w/a)\right]^2}{l^2}(-1)^l, \ m \neq n \tag{13}$$

where l is an integer different from zero. The above series is nearly alternate (it is exactly alternate if $\frac{r_w}{\alpha} \longrightarrow 0$). Hence, it seems clear that for $m \neq n$, $\frac{1}{\beta_{m,n}^2}$ can be neglected as compared with the first term in right-hand side of the approximate formula for $g_{m,n}$ in (12) (notice that in the long wavelength limit – $|\mathbf{k}|a \ll 1$ and $\beta a \ll 1$ – the first term is certainly very large). Quite differently, the amplitude of $\frac{1}{\beta_{m,n}^2}$ can be very large if $m = n$ (it is a double and non-alternate series), and thus it cannot be neglected. Therefore, we have that:

$$g_{m,n} \approx \frac{1}{a}\left(\frac{1}{|\mathbf{k}|^2-\beta^2}+\frac{1}{\beta_0^2}\delta_{m,n}\right), \quad \frac{1}{\beta_0^2} = \sum_{\mathbf{J}\neq 0}\delta_{j_3,0}\frac{\left[J_0(|\mathbf{k}_\mathbf{J}^0|r_w)\right]^2}{|\mathbf{k}_\mathbf{J}^0|^2} \tag{14}$$

Replacing (14) into (10) we obtain that:

$$\sum_n \left(k_m k_n - \beta^2 \delta_{m,n}\right) \left(\frac{1}{|\mathbf{k}|^2 - \beta^2} + \frac{1}{\beta_0^2}\delta_{m,n}\right) I_n = 0 \tag{15}$$

Hence, we obtain a homogeneous linear system for the unknown currents I_n. The dimension of the linear system is 1, 2 or 3 depending on the geometry of the considered wire structure. We can now easily obtain the dispersion characteristic $\beta = \beta(\mathbf{k})$ of the first few eigenvalues, by equaling the determinant of the associated matrix to zero. Before that, it is appropriate to homogenize the artificial medium, i.e. relate the average electromagnetic fields with the wave vector. This topic is discussed in next section.

5 Homogenization of the structure

In this section, we prove that the average electromagnetic fields can be related with the wave vector $\mathbf{k}$ using an effective permittivity dyadic (tensor). To begin with, we present some introductory results and definitions.

Let $(\mathbf{E}, \mathbf{H})$ be an electromagnetic Floquet mode in a generic metallic crystal, i.e. a solution of (1). We define the average fields $\mathbf{E}_{\text{av}}$ and $\mathbf{H}_{\text{av}}$ as follows,

$$\mathbf{E}_{\text{av}} = \frac{1}{V_{\text{cell}}} \int_{\Omega} \mathbf{E}(\mathbf{r}) e^{+j\mathbf{k}\cdot\mathbf{r}} d^3\mathbf{r}, \qquad \mathbf{H}_{\text{av}} = \frac{1}{V_{\text{cell}}} \int_{\Omega} \mathbf{H}(\mathbf{r}) e^{+j\mathbf{k}\cdot\mathbf{r}} d^3\mathbf{r} \tag{16}$$

Using (1), it can be verified that the following equations hold:

$$-\mathbf{k} \times \mathbf{E}_{\text{av}} + \beta Z_0 \mathbf{H}_{\text{av}} = 0 \tag{17a}$$

$$\beta \mathbf{E}_{\text{av}} + \mathbf{k} Z_0 \mathbf{H}_{\text{av}} = \frac{j Z_0}{V_{\text{cell}}} \int_{\partial D} \mathbf{J}_c e^{+j\mathbf{k}\cdot\mathbf{r}} ds \tag{17b}$$

where ∂D denotes the surface of the metallic region in the unit cell, and $\mathbf{J}_c = \hat{\mathbf{v}} \times \mathbf{H}$ is the surface current over the metallic boundaries. From (17), we obtain after straightforward manipulations that:

$$\left[\left(\beta^2 - |\mathbf{k}|^2\right) \overline{\overline{I}} + \mathbf{k}\mathbf{k}\right] \cdot \mathbf{E}_{\text{av}} = \frac{j Z_0}{V_{\text{cell}}} \int_{\partial D} \mathbf{J}_c e^{+j\mathbf{k}\cdot\mathbf{r}} ds \tag{18}$$

where $\overline{\overline{I}}$ is the identity dyadic.

In the rest of this section, we assume for simplicity that the metallic crystal is the 3D-wire medium depicted in Figure 2. In the end, we explain how the derived results can be generalized to the other wire geometries.

Since we admit that the surface current over the wires is given by (8), we readily obtain from (18) that:

$$\left[\left(\beta^2 - |\mathbf{k}|^2\right) \overline{\overline{I}} + \mathbf{k}\mathbf{k}\right] \cdot \mathbf{E}_{\text{av}} = \frac{j Z_0}{a^2} \left(I_1 \hat{\mathbf{u}}_1 + I_2 \hat{\mathbf{u}}_2 + I_3 \hat{\mathbf{u}}_3\right) \tag{19}$$

We also note that (15) is equivalent to:

$$\frac{1}{|\mathbf{k}|^2-\beta^2}\left(\mathbf{kk}-|\mathbf{k}|^2\overline{\overline{I}}+\beta^2\overline{\overline{\tilde{\varepsilon}}}\right)(I_1\hat{\mathbf{u}}_1+I_2\hat{\mathbf{u}}_2+I_3\hat{\mathbf{u}}_3)=0 \tag{20}$$

where,

$$\overline{\overline{\tilde{\varepsilon}}}=\frac{|\mathbf{k}|^2-\beta^2}{\beta^2}\left(\overline{\overline{\varepsilon}}-\overline{\overline{I}}\right)^{-1}\cdot\overline{\overline{\varepsilon}} \tag{21}$$

and $\overline{\overline{\varepsilon}}$ is the diagonal dyadic (in the canonical basis associated with the coordinates axes) such that:

$$\hat{\mathbf{u}}_m\cdot\overline{\overline{\varepsilon}}\cdot\hat{\mathbf{u}}_m=\varepsilon_{m,m}=1-\frac{\beta_0^2}{\beta^2-k_m^2},\qquad m=1,2,3 \tag{22}$$

Therefore, using (19), we find that the average electric field satisfies the following homogeneous system:

$$\frac{1}{|\mathbf{k}|^2-\beta^2}\left(\mathbf{kk}-|\mathbf{k}|^2\overline{\overline{I}}+\beta^2\overline{\overline{\tilde{\varepsilon}}}\right)\cdot\left(\left(\beta^2-|\mathbf{k}|^2\right)\overline{\overline{I}}+\mathbf{kk}\right)\cdot\mathbf{E}_{\mathrm{av}}=0 \tag{23}$$

Next, we multiply the left-hand side of the above equation by the dyadic $\overline{\overline{\varepsilon}}-\overline{\overline{I}}$. After straightforward manipulations we conclude that:

$$\left(\mathbf{kk}-|\mathbf{k}|^2\overline{\overline{I}}+\beta^2\overline{\overline{\varepsilon}}\right)\cdot\mathbf{E}_{\mathrm{av}}=0 \tag{24}$$

Comparing the above equation with the characteristic equation for the average electric field in an anisotropic medium [21, pp.202], we recognize that $\overline{\overline{\varepsilon}}$ is necessarily the (relative) effective permittivity dyadic. The dispersion characteristic $\beta=\beta(\mathbf{k})$ for the first few bands is obtained by setting the determinant of the dyadic to zero, and solving for β:

$$\det\left(\mathbf{kk}-|\mathbf{k}|^2\overline{\overline{I}}+\beta^2\overline{\overline{\varepsilon}}\right)=0 \tag{25}$$

As is well-known [21, pp.202], the average electric field is then given by:

$$\mathbf{E}_{\mathrm{av}}\propto\left(\frac{k_1}{|\mathbf{k}|^2-\beta^2\varepsilon_{1,1}},\frac{k_2}{|\mathbf{k}|^2-\beta^2\varepsilon_{2,2}},\frac{k_3}{|\mathbf{k}|^2-\beta^2\varepsilon_{3,3}}\right) \tag{26}$$

where $\varepsilon_{m,m}$ is given by (22).

As referred before, the derived results assume the 3D-wire medium geometry. In general, for N wire sections in the unit cell, we could verify that $\varepsilon_{m,m}$ is given by (22) for $m\leq N$ and that $\varepsilon_{m,m}=1$ for $m>N$. In particular, in the 1D-wire medium case our results are consistent with those obtained in [15] using a completely different approach. Furthermore, the constant β_0, defined by (14), is necessarily the so-called plasma wave number. In order that our definition is consistent with that of [15], we must have that [22]:

$$(\beta_0 a)^2=\frac{2\pi}{\ln\left(\frac{a}{2\pi r_w}\right)+0.5275} \tag{27}$$

Indeed, numerical simulations and a more detailed analysis show that the above formula is a very good approximation of (14).

6 Physical implications of the results

In this section, we obtain the dispersion characteristic $\beta = \beta(\mathbf{k})$ for the first few modes of the different wire structures, and briefly discuss the physical implications of the results.

As explained in the previous section, the dispersion characteristic is obtained by solving (25) for β . In general the solution cannot be obtained in a closed-analytical form. To circumvent this problem we proceed as delineated next. First, we write the wave vector in polar coordinates, i.e. we put $\mathbf{k} = |\mathbf{k}|(\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta)$. Then, we admit that $\beta^2 = a_0 + a_2|\mathbf{k}|^2 + \ldots$, where a_0, a_2, etc, are unknown coefficients that in general depend on angles θ and φ. We insert the indicated formulas for β and $\mathbf{k}$ in (25). After simplifications, we obtain an identity of the form $F\left(|\mathbf{k}|^2, a_0, a_2, \ldots\right) = 0$, where F is a polynomial function of its arguments. In order to calculate recursively the unknown coefficients a_0, a_2, etc, we impose that the successive derivatives of function F (in $|\mathbf{k}|^2$) at the origin vanish. In this way, we obtain an approximation for the desired dispersion characteristic (the formulas are expected to hold in the long wavelength limit). In what follows, we describe the results for the different wire geometries.

6.1 1D-Wire medium

The 1D-wire medium depicted in Figure 2 (the wires are oriented in the x_1-direction) is characterized by three bands in the long wavelength limit. The dispersion characteristic of these bands is:

$$\beta^2 = |\mathbf{k}|^2 \tag{28a}$$

$$\beta^2 = k_1^2 \tag{28b}$$

$$\beta^2 = \beta_0^2 + |\mathbf{k}|^2 \tag{28c}$$

The band (28a) is associated with (transverse electric) modes with average electric field normal to the wires. The band whose dispersion characteristic is (28b) is associated with transverse electromagnetic (TEM) modes. Finally, the band (28c) is the plasma mode (which sees a negative permittivity in the static limit). More details can be found in [15].

6.2 2D-Wire medium

In the 2D-wire medium case the wires are directed along the x_1- and x_2-directions. There are four relevant bands, which are defined by:

$$\beta^2 = k_1^2 + k_2^2 + o\left(|\mathbf{k}|^4\right) \tag{29a}$$

$$\beta^2 = \frac{2k_1^2k_2^2}{\beta_0^2}\frac{|\mathbf{k}|^2}{k_1^2 + k_2^2} + o\left(|\mathbf{k}|^6\right) \tag{29b}$$

$$\beta^2 = \beta_0^2 + |\mathbf{k}|^2 \pm k_1k_2 + o\left(|\mathbf{k}|^4\right) \tag{29c}$$

where $o(t)$ represents a quantity that vanishes as t. The band (29a) is associated with a mode whose power flow is restricted to directions in the x_1ox_2 plane. The polarization (i.e. the direction of the average electric field) is practically normal to the x_1ox_2 plane. The band (29b) is associated with a mode that propagates only at extremely low frequencies. Quite interestingly, the intersection of the contours $\beta =$ const. (known as the wave normal surfaces) with the $k_3 = 0$ plane are hyperbolic curves. This situation is rather peculiar and is not observable in standard (non-artificial) dielectric materials, which invariably have elliptic wave normal surfaces. For $k_3 = 0$, the polarization is $\mathbf{E}_{\text{av}} \propto (k_1, -k_2, 0)$. In particular, if the wave vector is along the wire axes the mode is longitudinal (i.e. has polarization parallel to the wave vector). Finally, the two bands defined by (29c) are associated with plasma modes (which see a negative permittivity in the static limit). To a first approximation the polarization of these modes is $\mathbf{E}_{\text{av}} \propto (1, -(\pm 1), 0)$, i.e. independent of the wave vector. More details are omitted for conciseness.

6.3 3D-Wire medium

The 3D-wire medium is characterized by five different bands in the long wavelength limit. The general formulas for the different bands are a bit cumbersome, and in some cases cannot be obtained in closed-analytical form. Due to that reason, we restrict our discussion to the case in which $k_3 = 0$. The dispersion characteristic of the different bands is given by:

$$\beta^2 = 0 + o\left(|\mathbf{k}|^6\right), \qquad k_3 = 0 \tag{30a}$$

$$\beta^2 = \frac{2k_1k_2^2}{\beta_0^2} + o\left(|\mathbf{k}|^6\right), \qquad k_3 = 0 \tag{30b}$$

$$\beta^2 = \beta_0^2 + |\mathbf{k}|^2 \pm k_1k_2 + o\left(|\mathbf{k}|^4\right), \qquad k_3 = 0 \tag{30c}$$

$$\beta^2 = \beta_0^2 + |\mathbf{k}|^2 + o\left(|\mathbf{k}|^4\right), \qquad k_3 = 0 \tag{30d}$$

The first two bands defined by (30a) and (30b) are associated with modes that only propagate at extremely low frequencies. Indeed, the band defined by (30a) only exists in the static limit (zero frequency). The corresponding polarization is longitudinal. On the other hand, the band defined by (30b) has an hyperbolic wave normal contour. The polarization properties are identical to those of band (30b) discussed in the previous section.

The three bands defined by (30c) and (30d) correspond to the plasma modes, which see a negative permittivity in the static limit. The two bands defined by (30c) have elliptical wave normal contours and polarization as in the previous section (i.e. practically independent of the wave vector). On the other hand, the band defined by (30d) has a circular wave normal contour, and polarization normal to the x_1ox_2 plane.

It is now appropriate to compare our results with the standard plasma model for the 3D-wire medium. Within the standard plasma model, the effective permittivity of the wire medium is given by $\bar{\bar{\varepsilon}} = 1 - \frac{\beta_0^2}{\beta}$, i.e. the medium is isotropic and does not suffer spatial dispersion (the effective permittivity is independent of $\mathbf{k}$). Within the standard model, only two electromagnetic modes should exist in the long wavelength limit. The two modes are degenerate and have dispersion characteristic $\beta^2 = \beta_0^2 + |\mathbf{k}|^2$ (plasma modes). The electromagnetic modes are transverse electromagnetic (i.e. the electric and magnetic field are normal to the wave vector).

Thus, we conclude that the standard model fails to predict the existence of bands (30a) and (30b), which correspond to modes that propagate at very low frequencies. Furthermore, the standard model fails to predict the existence of three plasma modes. Indeed, only the mode (30d) is correctly predicted. It is thus apparent that the propagation of electromagnetic waves in the 3D-wire medium is much more intricate than it was thought. There is no isotropy even for very long wavelengths (this is a consequence of the strong spatial dispersion caused by the infinitely long wires). Quite interestingly the existence of three propagating plasma modes in a similar wire structure was speculated in [23] based on numerical and experimental data. To conclude, we refer that we have checked our analytical formulas with numerical results obtained using the hybrid method proposed in [17]. The agreement was good.

7 Conclusions

In this paper, we discussed the propagation of electromagnetic waves in several wire structures in the long wavelength limit. Based on simple physical considerations and using a variational formulation, we were able to reduce the calculation of the spectrum of a differential operator to the calculation of the zeros of a simple characteristic equation that is known in closed-analytical form. Then, we proved that the propagation of electromagnetic waves in the considered periodic structures can be described in terms of a spatially dispersive permittivity dyadic. The permittivity dyadic is a generalization of that derived in [15] for the 1D-wire medium case. Finally, we discussed the physical implications of the results. We showed that there is no isotropy in the 3D-wire medium for long wavelengths, and that the standard model may be insufficient to describe the electrodynamics of the structure. Our results predict the existence of three distinct plasma modes. In general, a plasma mode may be longitudinal. The results also show that two modes propagate at extremely low frequencies. The wave normal surfaces of these modes are intrinsically hyperbolic. The implication of the results in the realization of double negative media merits further investigation.

Acknowledgement

This work was funded by Fundação para Ciência e a Tecnologia under project POSI 34860/99.

References

[1] J. Joannopoulos, R. Meade, and J. Winn, *Photonic Crystals.* Princeton University Press, 1995.

[2] K. Sakoda, "Optical properties of photonic crystals." Springer Series in Optical Sciences 80, 2001.

[3] K. Ho, C. Chan, and C. Soukoulis, "Existence of a photonic gap in periodic dielectric structures," *Phys. Rev. Letts.*, vol. 65, p. 3152, December 1990.

[4] E. Yablonovitch, "Inhibited spontaneous emission in solid-state physics and electronics," *Phys. Rev. Letts.*, vol. 58, p. 2059, May 1987.

[5] E. Yablonovitch, T. Gmitter, R. Meade, A. Rappe, K. Brommer, and J. Joannopoulos, "Donor and acceptor modes in photonic band structure," *Phys. Rev. Letts.*, vol. 67, p. 3380, December 1991.

[6] C. Kyriazidou, H. Contopanagos, and N. Alexopoulos, "Monolithic waveguide filters using printed photonic-bandgap materials," *IEEE Trans. on MTT*, vol. 49, p. 297, December 2001.

[7] H. Yang, N. Alexopoulos, and E. Yablanovitch, "Photonic band-gap materials for high-gain printed circuit antennas," *IEEE Trans. on AP*, vol. 45, p. 185, January 1997.

[8] M. Thevenot, C. Cheype, A. Reinex, and B. Jecko, "Directive photonic-bandgap antennas," *IEEE Trans. on MTT*, vol. 47, p. 2115, November 1999.

[9] A. Sihvola, "Electromagnetic mixing formulas and applications," *IEE Electromagnetic Waves Series*, vol. 47, 1999.

[10] I. Lindell, A. Sihvola, S. Tretyakov, and A. Viitanen, *Electromagnetic Waves in Chiral and Bi-isotropic Media.* Artech House, 1994.

[11] J. Brown, "Artificial dielectrics," *Progr. Dielectrics*, vol. 2, pp. 195–225, 1960.

[12] D. Smith, W. Padilla, D. Vier, S. Nemat-Nasser, and S. Schultz, "Composite medium with simultaneously negative permeability and permittivity," *Phys. Rev. Letts.*, vol. 84, p. 4184, May 2000.

[13] V. Veselago, "Electrodynamics of substances with simultaneously negative electrical and magnetic permeabilities," *Sov. Phys. USPEKHI*, vol. 10, p. 509, 1968.

[14] J. B. Pendry, *Negative Refraction Makes a Perfect Lens.* Physical Review Letters, 2000.

[15] P. Belov, R. Marques, I. Nefedov, M. Silveirinha, and S. T. C.R Simovsky, “Strong spatial dispersion in wire media in the very large wavelength limit,” *Physical Review B*, vol. 67, p. 113, 2003.

[16] H. D. Jackson, *Classical Electrodynamics.* John Wiley & Sons, 3rd ed., 1999.

[17] M. Silveirinha and C. A. Fernandes, “Efficient calculation of the band structure of artificial materials with cylindrical metallic inclusions,” *IEEE Trans. MTT*, vol. 51, pp. 1460–1466, May 2003.

[18] M. Silveirinha and C. A. Fernandes, “A new method with exponential convergence to evaluate the periodic green function,” in *IEEE APS/URSI Symposium*, vol. 2, pp. 805–808, Ohio , USA, June 2003.

[19] P. Ewald, “Die berechnung optischer und elektrostatischer gitterpotentiale,” *Ann. Der Physik*, vol. 64, pp. 253–287, 1921.

[20] D. Colton and R. Kress, *Integral Equation Methods in Scattering Theory.* John Wiley & Sons, 1983.

[21] R. Collin, *Field Theory of Guided Waves.* IEEE Press, 2nd ed., 1991.

[22] P. A. Belov *et al.*, “Dispersion and reflection properties of artificial media by regular lattices of ideally conducting wires,” *Journal of Elect. Waves and Appls.*, vol. 16, p. 1153, 2002.

[23] D. Sievenpiper, M. Sickmiller, and E. Yablanovitch, “3D wire mesh photonic crystals,” *Phys. Rev. Letts.*, vol. 76, p. 2480, 1996.

Analysis of chordal rings

Slawomir Bujnowski* Bozydar Dubalski* Antoni Zabludowski*

Abstract

In the paper the analysis of third and fourth degree chordal rings has been given. In the first part of the paper some definition of chordal rings, existing in the literature, has been discussed. The main goal of the second part is to search for the general formulas that describe chord lengths in chordal rings with degree 3. This chord gives the graph, describing the communication network, with minimum diameter and average paths length. The results obtained were set against the theoretically obtained values which determine the lower limits of these parameters together with results obtained while examining the real structures. The main obtained results related to the graphs fourth degree have been presented in the third part of the paper. Two classes of chordal rings, namely ideal and optimal graphs have been defined. It has been shown that the optimal chordal rings posses strictly defined number of nodes, there exist only one optimal chord with the length $2d(G)+1$ for those structures, and the optimal chordal rings are built with the use of two Hamiltonian cycles ($d(G)$ means the diameter of graph). The ideal chordal rings (real or virtual), can be treated as the lower bound of any analysed chordal ring.

Keywords: Chordal ring, diameter of graph, average path length.

1 Introduction

As the part of research project, conducted in the Institute of Telecommunications the University of Technology and Agriculture in Bydgoszcz, that concerns to the switching systems, an analysis distributed structures of telecommunication server has been done. A distributed telecommunication server consists of a number of identical, sophisticated, switching modules that communicate each other with the use of interconnection network. Recently, some of the leading producers of telecommunication equipment has implemented such a solutions in their products [1, 2]. It is obvious, that the telecommunication servers provide for their subscribers the same functions and services as the large switches do. The main problem which has appeared during analysis of such the systems was the problem of choosing the interconnection network structure that links the telecommunication modules. At the beginning, the solutions from distributed computer systems, especially the interconnection network structure for

*Inst. Telekom. ATR Poland. E-mail: antoni.zabludowski@atr.bydgoszcz.pl

telecommunication servers has been studied. Distributed computer systems were developed to increase the computation power for applications in engineering and science and were based on a concept of parallel processes, implemented by a set of processors connected with the use of given interconnection network. In fact, the distributed computer systems are very similar in concept to the studied distributed telecommunication server. It is obvious that the topology of interconnection network determines the efficiency of the entire system [3], both in distributed computer system and distributed telecommunication server. The interconnection network structure should provide very high level of reliability as well as the level of service quality. Among the analyzed interconnection structures (i.e. hypercubes, meshes, Cayley's graphs, rings etc.) the rings are the cheapest and the easiest to implement ones, but they have the lowest connectivity and the highest diameter [4]. So, the transmissions properties of such the networks (we take the probability of call rejection as the of quality parameter) are very poor in comparison with the other graphs. The simplest possible method of improvement of transmission properties of interconnection structures is the use of additional edges called chords, that would connect the nodes of the ring in a given way. The rings with additional edges linking nodes are denoted in the literature the *chordal rings*.

Definition 1 *Chordal ring is a circulant graph with chord of length 1. It is defined by the pair* (w, S)*, where* w *means the number of nodes of the ring and* S *is the set of chords* $S \subseteq \{2, ..., \lfloor w/2 \rfloor\}$*. Each chord* $s \in S$ *connects every pair of nodes of ring that are at distance* s *in the ring. This structure is denote by symbol* $G(w; s_1, ..., s_i)$*,* $s_1 < \ldots < s_i$*, the chordal ring defined by* $(w, \{s_1, \ldots, s_i\})$*. The dimension of* $G(w; s_1, \ldots, s_i)$ *is* $i + 1$*. Degree of chordal rings is* $2i$ *in general whenever there is a chord of length* $w/2$*, in this case* w *is even and ring?s degree is* $2i - 1$ *[5].*

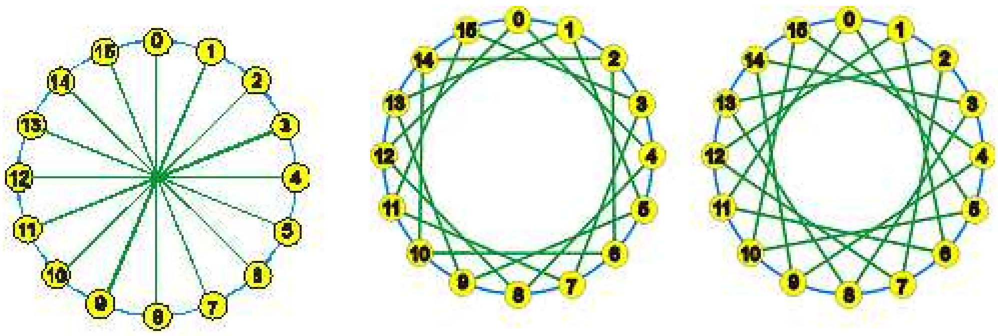


Figure 1: Examples of chordal rings of third and fourth degrees $G(16; 1, 8)$, $G(16; 1, 4)$ and $G(16; 1, 5)$

The term *chordal ring* denotes also in many papers the regular graphs with nodes of degree three.

Definition 2 *Chordal ring is a ring structure network in which each node has an additional link, called a chord, that connect some other node in the network. In the ring each odd-numbered node $i = \{1, 3, 5, \ldots, w-1\}$ is connected to node numbered $(i+s) \mod w$, where s is odd number, so node numbered $(i+s)$ is even-numbered. Thus, in the chordal ring each even-numbered node $j = \{0, 2, 4, \ldots, w-2\}$ is connected to odd-numbered node $(j-s) \mod w$, where $s \geq w/2$ means length of chord and w means number of nodes. Length of chord is positive and odd.[6]*

To describe chordal rings third degree defined by definition 2 we will introduce additional index $\mathbf{G}_3$.

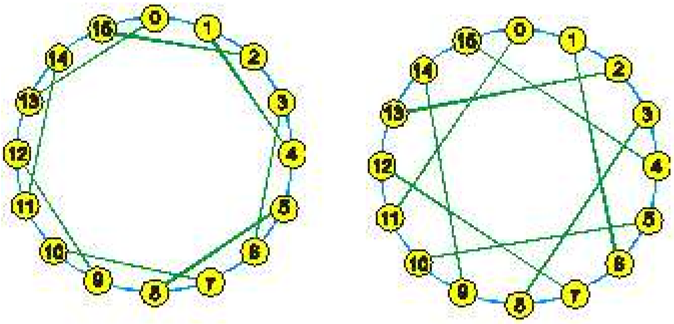


Figure 2: Examples of graphs of third degree $\mathbf{G}_3(16; 1, 3)$, $\mathbf{G}_3(16; 1, 5)$

Before we start with the analysis of the chordal ring we recall basic graph parameters, we will use for graph comparison.

The network diameter:

$$d(G) = \max_{v_i v_j}\{d_{\min}(v_i, v_j)\} \tag{1}$$

is the largest value among all of the shortest path lengths between all pair of nodes.

The average length of the paths between all the pair of nodes is defined by the following formula:

$$d_{\text{av}} = \frac{1}{w(w-1)} \sum_{i=0}^{w-1} \sum_{j=0}^{w-1} d_{\min}(v_i, v_j) \tag{2}$$

The parameters defined above will be used as the main measure of communication ability of analysed networks. In the next parts of this paper we will derive the formulas which give near the best value of chord. However, before we will be able to define those formulas we should describe chordal rings that can be treated as the ideal ones, not necessary existing in real.

2 Third degree cordal rings

As we already have explained, chordal rings are very useful as communication structures (communication network) connecting distributed modules. The only problem in analysis of such a networks is to determine the parameters describing the properties of the ring that should be taken into account. In the most papers the authors use the diameter of chordal rings as the main parameter that describe the properties of the ring [7]. They assert that this parameter decides of the usefulness of the structure to design multi-computers, multiprocessors or telecommunications networks. On the basis of results obtained, they suggest moreover, that the chord lengths equal to $\sqrt{w}+3$ or $\sqrt{w}+5$ gives the shortest diameter of chordal rings, and the networks with topology based on such rings posses better transmission abilities than the other ones [6, 8, 9, 10].

We have observed however, that the diameter of chordal rings is not sufficient parameter for analysing of communication network properties, as this parameter does not determine the transmissions abilities of the networks. We have studied several hundred of networks determined by the chordal rings of degree 3 (as well as with the higher node degrees), and observed, that very often the networks with the same diameter posses different abilities for passing telecommunication traffic. In order to examine the transmission abilities of the networks we have used the computer simulation tests using Monte-Carlo method. The following assumptions have been taken into account:

- the comparison of different chordal rings (interconnection networks) with the same number of nodes and transmission capacity of the links, were done;
- the multicommodity flow (Poissson, exponential) among the nodes have been studied;
- the value of the traffic intensity generated in each node have been increased. For fixed value of the traffic there were done several thousands of attempts;
- in each attempt with the fixed traffic value the source and destination nodes were taken with the uniform distribution;
- the shortest paths between source and destination have been chosen to carry the traffic. In the event of existence of many possible paths between the nodes, the least loaded has been chosen;
- probability of call rejection described by the formula

$$p_{cr} = \frac{\sum PN}{\sum GP},$$

where

$$\sum PN \quad - \quad \text{rejected call number,}$$
$$\sum GP \quad - \quad \text{total call number,}$$

has been calculated.

As the example, we will discuss the communication properties of two networks ? first one with 30 nodes and the second one with 56 nodes. The simulation results obtained for those networks are depicted in figure 3. For chordal rings with 30 nodes, the diameter of all the rings was the same, equal to 5, but for rings with 56 nodes, the diameter of the graph $\mathbf{G}_3(56, 1, 13)$ was equal to 7, and $\mathbf{G}_3(56, 1, 11)$ was equal to 8 (the best graph $\mathbf{G}_3(56, 1, 9)$ has diameter equal to 7).

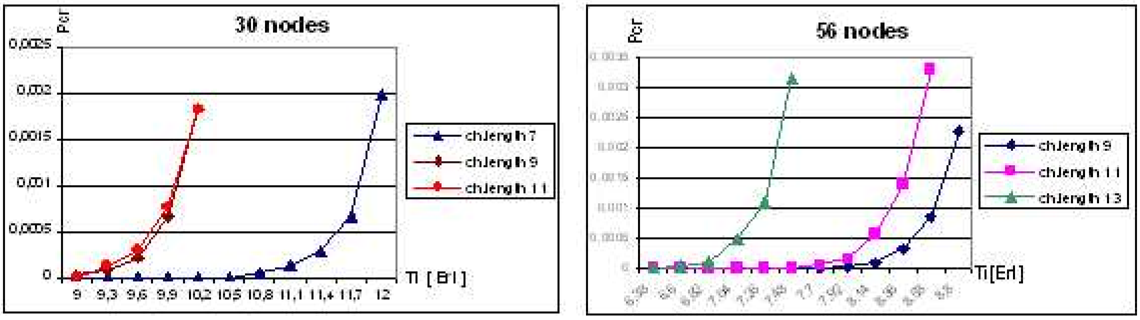


Figure 3: Results of simulation. T_i – total value of traffic generated in each node, P_{cr} – probability of call rejection

The probability of call rejection was different for analysed chordal rings. For the graphs with 30 nodes (we remind that all the graphs have the same diameter) the probability of call rejection (for fixed traffic value) reaches the minimal value for chord length equal to 7, but for chordal ring with 56 nodes the probability of call rejection for the ring with a bigger diameter is smaller than for the ring with smaller one. From the above discussion, one can conclude that in order to find the chordal rings having better ability of carrying communication traffic with the minimal probability of call rejection, we should choose such a chord that minimise a completely different parameter as the graph diameter is. A more detailed inspection of chordal rings shows however, that the biggest influence on communication ability of chordal rings possesses the average value of the length of shortest paths d_{av} between all the pairs of nodes. Thus, the following postulate was formulated:

Postulate 1 *In chordal rings of degree 3 the probability of call rejection depends on the average length of the paths between all pairs of the nodes.*

Thus, the analysis of chordal rings should concern two basic parameters of rings, namely the network diameter and the average length of the paths between all pairs of nodes.

Turning back to the analysed examples, it was easy to calculate that for the graph $\mathbf{G}_3(30,1,7)$ the average value of paths is $d_{\mathrm{av}} = 3.07$, while for $\mathbf{G}_3(30,1,9)$ or $\mathbf{G}_3(30,1,11)$, $d_{\mathrm{av}} = 3.14$. For rings with 56 nodes in turns, for $\mathbf{G}_3(56,1,11)$ (the diameter is equal to 8), $d_{\mathrm{av}} = 4.25$ and for $\mathbf{G}_3(56,1,13)$ (the diameter is equal to 7) - $d_{\mathrm{av}} = 4.29$. The best graph $\mathbf{G}_3(56,1,9)$ has the diameter equal to 7 and $d_{\mathrm{av}} = 4.18$.

2.1 Optimal and ideal chordal rings with degree 3

In this part of the paper, two type of chordal rings will be defined, the first one called the optimal chordal ring and the second one called the ideal graph. Those rings will be used for comparison of "ideal" and real obtained structures.

Before the ideal graph will be defined (i.e. the graph which will be used for comparison of parameters of examined graphs), the optimal graph should be defined.

The optimal graph is the chordal ring characterising by following features:

1. The number of nodes w_d in a d-th layer (the layer means the subset of nodes that are reached from any source node with the use of d edges), is given by the formula:

$$w_{do} = 3d \tag{3}$$

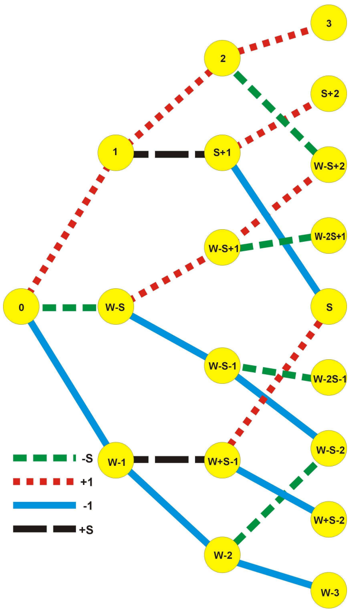


Figure 4: Node distribution at first three layers

2. The total number of nodes w_o in the graph with a diameter $d(G)$ is equal to:

$$w_0 = 1 + 3\sum_{d=1}^{d(G)} d = 1 + 3\frac{d(G)\,[d(G)+1]}{2} \tag{4}$$

3. The diameter of the graph with the w_o nodes is equal to:

$$d(G)_o = \frac{\sqrt{24w_o - 15} - 3}{6} \tag{5}$$

4. The average length of the path is equal to:

$$d_{\text{avo}} = 3\frac{\sum_{d=1}^{d(G)} d^2}{w_o - 1} = d(G)\frac{[d(G)+1]\;[2d(G)+1]}{2(w_o-1)} = \frac{2d(G)+1}{3} \tag{6}$$

At the table below the number of nodes and the average length of the paths versus the diameter are given.

$d(G)$	1	2	3	4	5	6	7	8
w_o	4	10	19	31	46	64	85	109
d_{avo}	1	1.67	2.33	3	3.67	4.33	5	5.67

Theorem 1 *There exist no optimal chordal rings of degree 3 with the diameters equal to $4j$ or $4j-1$ $(j = 1, 2, \ldots)$.*

Proof: The necessary condition for existence of graph in the assumed premises is an even number of nodes. Total number of nodes existing in ideal graph of $d(G)$ diameter is equal to

$$w_o = 1 + 3\frac{d(G)\,[d(G)+1]}{2},$$

so

$$3\frac{d(G)\,[d(G)+1]}{2}$$

must bring an odd number for every $d(G)$ value as a result. By using values $4j$ and $4j-1$ in the mentioned formula we obtain

$$3\frac{4j(4j+1)}{2}$$

and

$$3\frac{(4j-1)4j}{2}$$

respectively. As the any number multiplied by an even number gives an even number, therefore total number of nodes in the described graph will be odd. □

The only one real optimal graph is a graph with 4 nodes and diameter $d(G) = 1$, simply it is a complete graph. It was not stated that there should exist any possibilities of practical construction of the ideal graph with a diameter $d(G) > 1$. Optimal graphs are the special group of the ideal graphs. The ideal graph is a theoretical chordal ring which fulfils, in turns, the following conditions:

1. The diameter of the ideal graph with the w_i nodes is equal to:

$$d(G)_i = \left\lceil \frac{\sqrt{24w - 15} - 3}{6} \right\rceil \tag{7}$$

2. The numbers of the nodes w_{di} in a d-th – layer are:

$$\begin{array}{lll} \text{if } d \neq d(G) & \Rightarrow & w_{di} = 3d \\ \text{if } d = d(G) & \Rightarrow & w_{di} = w - w_{o\{d(G)_r - 1\}} \end{array} \tag{8}$$

   where: w – number of nodes in the ideal graph, $w_{o\{d(G)_i - 1\}}$ – number of nodes in the optimal graph with diameter $d(G)_i - 1$.

3. The average length of the paths is equal to:

$$d_{\text{avi}} = d(G)_i \frac{[d(G)_i - 1]\,[2d(G)_i - 1] + 2(w - w_{o\{d(G)_i - 1\}}}{2(w-1)} \tag{9}$$

Parameters of the ideal graphs determine the lower limits of the diameter and average length paths and they are a basis to evaluate results obtained while examining the real structures.

2.2 Approximation method of chords calculation

Examining the diameter distribution in the chordal rings degree 3 versus the chord length, a typical diagram shown in figure below was obtained.

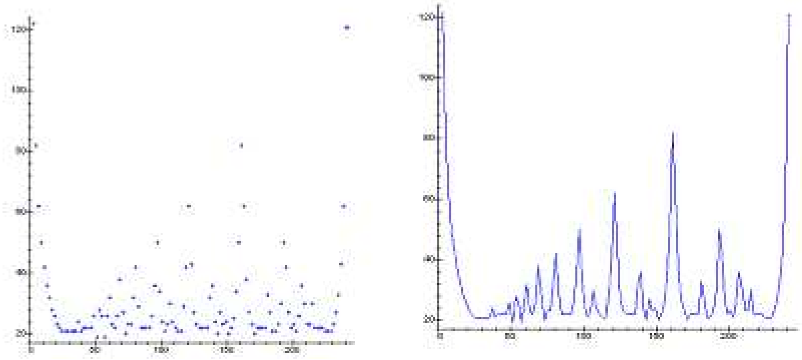


Figure 5: Diameter of chordal graphs degree 3 versus chord length – 484 nodes

Postulate 2 *In chordal rings degree 3 the choice of a chord length s_2 equal to*

$$s_2 = \left|\sqrt{2w}\right| \qquad or \qquad s_2 = \frac{w}{2} - \left|\sqrt{\frac{w}{2}}\right| \tag{10}$$

guarantees that this graph has the close diameter to the one of a reference graph.

Proof: On the obtained diagrams we can observe two ranges in which the distribution reaches the local minimal values.

Functions approximating the distribution of maximum diameter values versus the chord length (we the use of the Least Square Method [11]) are given by the formulas:

$$\text{(i) } d(G)_{s_2} = \frac{s_2}{2} + \frac{w}{s_2} + C \tag{11}$$

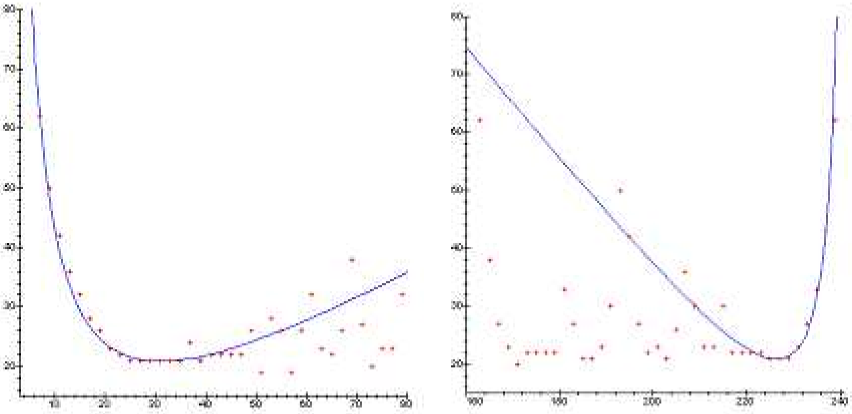
$$\text{(ii) } d(G)_{s_2} = \frac{w}{2} - s_2 + \frac{w}{2\left(\frac{w}{2} - s_2\right)} + C \tag{12}$$

C means a constant without any influence on function minimum.

First derivatives are:

$$\text{(i) } d(G)'_{s_2} = \frac{1}{2} - \frac{w}{s_2^2} \tag{13}$$

$$\text{(ii) } d(G)'_{s_2} = \frac{w}{2\left(\frac{w}{2} - s_2\right)^2} - 1 \tag{14}$$



(a) (i)

(b) (ii)

Figure 6: Diagrams of the functions approximating the maximum diameter distribution in chordal rings with 484 nodes

So these functions reach the minima for:

$$\text{(i) } s_2 = \left|\sqrt{2w}\right| \tag{1}$$

$$\text{(ii) } s_2 = \frac{w}{2} - \left|\sqrt{\frac{w}{2}}\right| \tag{2}$$

□

To illustrate postulate 2, the chordal ring with 484 nodes was analysed. In this case the constant C is equal to -10.25, but the results are as follows:

	(i)	(ii)
Computed chord length	31.11	226.44
Computed diameter value	20.86	20.86
Real cord length	31	227
Real diameter value	21	21

The minimal diameter $d(G)$ equal to 19 is obtained for $\mathbf{G}_3(484, 1, 51)$ and $\mathbf{G}_3(484, 1, 57)$ respectively. Figure 7 shows in turns the distribution of average path length versus chord diameter. This diagram has a similar character (comb function) to the diameter distribution shown in the figure 5.

Postulate 3 *In chordal rings of degree 3 the choice of a chord length s_2 equal to*

$$\textit{(i) } s_2 = \left|\sqrt{1.8182w}\right| \tag{15}$$

$$\textit{or (ii) } s_2 = \frac{w}{2} - \sqrt{0.4545w} \tag{16}$$

guarantees that the average length of the paths in this graph will be close to the value of this parameter existing in a reference graph.

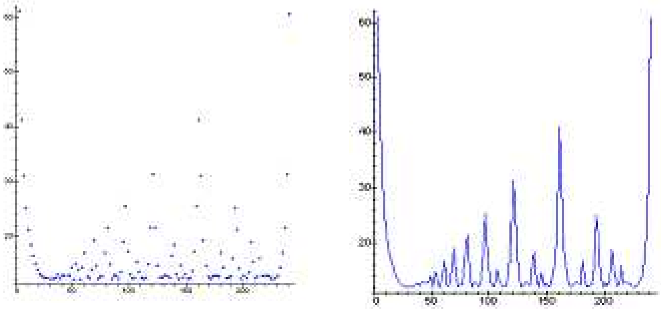


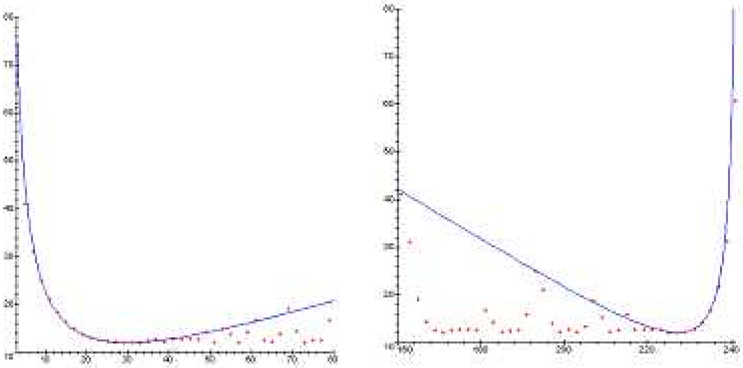
Figure 7: Average length of the path in chordal rings with 484 nodes versus chord lengths

Proof: Approximating functions are given as:

$$\text{(i) } d_{\max}(s_2) = 0.275 s_2 + \frac{w}{2s_2} + C \tag{17}$$

$$\text{(ii) } d_{\max}(s_2) = 0.55\left(\frac{w}{2} - s_2\right) + \frac{w}{4\left(\frac{w}{2} - s_2\right)} + C \tag{18}$$

and they reach the minima for (i) $s_2 = \left|\sqrt{1.8182w}\right|$ or (ii) $s_2 = \frac{w}{2} - \sqrt{0.4545w}$ □



(a) (i)

(b) (ii)

Figure 8: Diagrams of approximating functions in chordal rings with 484 nodes

To illustrate postulate 3, the chordal ring with 484 nodes was analysed. In this case the constant C is equal to -4.25, but the results are as follows:

	(i)	(ii)
Computed chord length	29.66	227.17
Computed average length path value	12.06	12.06
Real cord length	29	227
Real average length path value	12.21	12.10
Diameter	21	21

The minimal average length paths is $d_{\text{av}} = 12.02$ for $\mathbf{G}_3(484, 1, 51)$ and $\mathbf{G}_3(484, 1, 57)$.

In the tables below are the results of choice of chord length using the presented and other methods with comparison the parameters of those graphs and the parameters of reference graph have been given.

Nodes number	Chord Length	Diameter	Average length	Chord length	Diameter	Average length
50	9	7	3.94	19	7	3.98
100	15	9	5.52	43	10	5.95
500	31	22	12.44	235	21	12.30
1000	45	30	17.52	477	30	17.46
1600	57	41	22.39	771	38	22.13
2000	63	42	24.64	969	42	24.57
3000	77	56	30.59	1461	51	30.16
4000	89	64	35.30	1955	59	34.82
5000	99	67	38.89	2451	74	39.48
	$s_2 = \left\lvert \sqrt{2w} \right\rvert$			$s_2 = \frac{w}{2} - \left\lvert \sqrt{\frac{w}{2}} \right\rvert$		

Table 4: The results obtained with the use of postulate 2.

Nodes number	Chord Length	Diameter	Average length	Chord length	Diameter	Average length
50	9	7	3.94	21	7	3.97
100	13	9	5.52	43	10	5.95
500	31	22	12.44	235	21	12.30
1000	43	30	17.46	479	30	17.44
1600	53	38	22.10	773	38	22.06
2000	61	42	24.65	969	42	24.57
3000	73	54	30.38	1463	51	30.15
4000	85	64	35.15	1957	59	34.81
5000	95	67	38.91	2453	70	39.22
	$s_2 = \left\lvert \sqrt{1.8182w} \right\rvert$			$s_2 = \frac{w}{2} - \sqrt{0.4545w}$		

Table 5: The results obtained with the use of postulate 3.

Nodes number	Chord Length	Diameter	Average length	Chord length	Diameter	Average length
50	11	7	3.94	13	7	4.14
100	13	9	5.51	15	9	5.51
500	25	22	12.58	27	21	12.41
1000	35	31	17.83	37	32	17.65
1600	43	39	22.74	45	39	22.47
2000	47	44	25.60	49	43	25.28
3000	55	55	31.89	57	54	31.46
4000	65	63	36.49	67	62	36.11
5000	73	70	40.73	75	69	40.35
	$s_2 = \lfloor\sqrt{w}\rfloor + 3$			$s_2 = \lfloor\sqrt{w}\rfloor + 5$		

Table 6: The results obtained with the use of the theorem given in [8].

Nodes number	Diameter	Average length	Chord length	Diameter	Average length
50	6	3.87	9	7	3.94
100	8	5.45	13	9	5.51
500	18	12.17	59	20	12.24
1000	26	17.22	269	27	17.23
1600	33	21.77	97	33	21.78
2000	37	24.34	611	39	24.38
3000	45	29.81	133 **933**	48 **49**	29.99 **29.87**
4000	52	34.43	151	53	34.46
5000	58	38.49	965 1293	59	38.50
	Reference graphs (theory)		Reference graphs (practice)		

Table 7: Reference graphs and real best graphs parameters.

3 Fourth degree chordal rings

Fourth degree chordal rings consist of one ring and chords with length s that link two different nodes in the network. Fourth degree chordal rings possess some properties, that make them very suitable for construction of interconnection networks. The basic feature of fourth degree chordal rings is their ability to physical realization some of them as the optimal graphs and the majority of them as the ideal ones (definitions of these graphs will be given further in the paper). This type of interconnection structures were widely discussed in [12, 13, 14], as well as in [15] where this structure was used for the construction of transputers networks for image processing application.

Chordal rings of fourth degree ($d(V) = 4$) denoted as $G(w; 1, s)$, where w means number of nodes and s – chord length, form two disjoint classes of graphs (shown for example in figure 1):

- class in which the chord linking nodes i and j, where $j = i \oplus s \mod w$, generates the Hamiltonian cycle – the Hamiltonian cycle is obviously formed by the edges of the ring;
- class in which the chord linking nodes i and j where $j = i \oplus 1 \mod w$ generates some separate cycles of the same length $< w$. In this graphs there is only one Hamilton cycle formed by edges of the ring.

Theorem 2 *In the chordal ring $G(w; 1, s)$ of degree four, the shortest path length (defined as the number of edges) between any two nodes v_i and v_j, can be calculated as follows:*

$$\begin{aligned} d(v_i, v_j) = \min\{&(|k-j|div(s) + \{(k-j) - [|k-j|div(s) \times s]\}), \\ &(|w+j-k|div(s) + \{[(w+k) - (|w+j-k|div(s) \times s] - k\}), \\ &(|k-j|div(s) + 1) + \{[(j + (|k-j|div(s) + 1) \times s]) - k\}), \\ &((|w+j-k|div(s) + 1) + [(k - ((w+k) - (|w+j-k|div(s) + 1) \times s)])\} \end{aligned} \tag{19}$$

Proof: To find the shortest path between two nodes in chordal ring all the possible routes from node v_i to node v_j has to be calculated. It is obvious that wherever it is possible the route use the long jumps (long jump = chord length s). This route can end either on left or right side of node v_j. From the node we have reached by the long jumps, we have to go by the edges of the ring. Four routes to the node of chordal ring does exist:

- moving right by long jumps (the last node has a number smaller than j),
- moving right by long jumps (the last node has a number larger than j),
- moving left by long jumps (the last node has number smaller than j),
- moving left by long jumps (the last node has a number larger than j).

□

To search for the graph diameter there is no need to check the maximum for all pairs (v_j, v_k), as the chordal ring is the symmetrical structure. It is only need to check all paths length from one node to all remaining ones. One of the very important graph parameter for chordal rings is the average length of paths. This parameter shows the number of edges, which have to be crossed on average route for communication between any two nodes. In the chordal ring the layer ω_d contains all the nodes reachable from source node by the shortest path of length d. The number of nodes in layer ω_d (cardinality of layer) will be denoted by w_d.

The average length of the path (average distance) d_{av} is defined as:

$$d_{\text{av}} = \frac{\sum\limits_{d=1}^{d(G)} d w_d}{w - 1} \tag{20}$$

where w_d denotes the cardinality of layer ω_d and d the distance between nodes of layer ω_d and the source node.

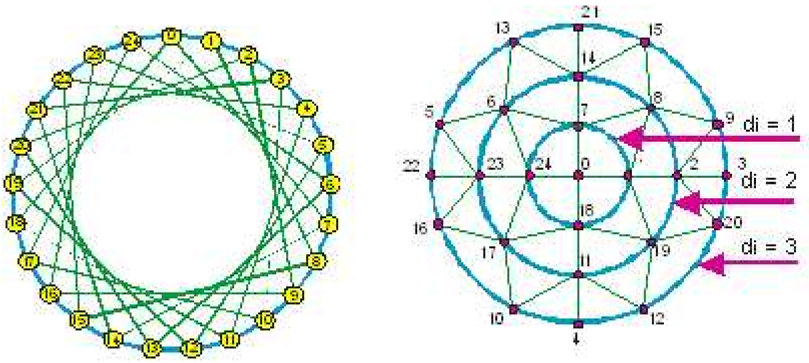
Example: In chordal ring $G(16; 1, 5)$, the nodes that are connected from the source node (zero node) by the use of d - length paths consist three following overlapping layers ω_d (in ω_3 the nodes already appeared in the lower layer are written in bold):

- the layer ω_1 contains the nodes connected by the paths with the length $d = 1$, i.e. the nodes with indexes from subset $\{1, 5, 11, 15\}$,
- the layer ω_2 contains the nodes connected by the paths with the length $d = 2$, i.e. the nodes with indexes from subset $\{2, 4, 6, 6, 10, 10, 12, 14\}$,
- the layer ω_3 contains the nodes connected by the paths with the length $d = 3$, i.e. the nodes with indexes from subset $\{3, 3, 7, 7, 9, 9, 13, 13, \mathbf{1}, \mathbf{5}, \mathbf{11}, \mathbf{15}\}$.

			15			
		9	10	**11**		
	3	4	5	6	7	
13	14	15	0	1	2	3
	9	10	11	12	13	
		5	6	7		
			1			

Figure 9: Node distribution table for graph $G(16; 1, 6)$, by bold are depicted the nodes that appear in the lower layers

For construction of interconnection network we should take such a topology of the network structure, for which the diameter as well as the average path length reach minimal value. It is easy to conclude that the minimal value of those parameters possess the chordal rings in which all layers ω_d are disjoint. This type of chordal rings is called the *optimal graph*. In chordal ring the generated layers can be depicted by the use of, so called, "atomic" model. An example of "atomic" model of chordal ring $G(25;1,7)$ is shown in figure 10.



(a) (i)

(b) (ii)

Figure 10: Example of optimal chordal ring $G(25;1,7)$ and its "atomic" model

In the next part of this paper it will be shown that the optimality of fourth degree chordal ring depends on nodes number, diameter of the ring and chord length $s > 1$. As an example of the optimal graph with node degree four and diameter 3 a graph with 25 nodes and chord equal to 7 can be taken. The successive layers of this graphs are as follows:

$$\begin{aligned}\omega_1 &= \{1,7,18,24\}\\ \omega_2 &= \{2,6,8,11,14,17,19,23\}\\ \omega_3 &= \{3,4,5,9,10,12,13,15,16,20,21,22\}\end{aligned}$$

The number of nodes appearing in the respective layers makes up the arithmetic sequence. Both the first expression in the sequence a_0 – corresponding to the node number in the first layer, as well as the difference – corresponding to increase of the node numbers in the respective layers – are equal 4. So it is possible to estimate the diameter of this ring by means on solving the inequality being a summation formula of arithmetical sequence.

$$w - 1 \leq \frac{d(G)}{2}\left[2a_0 + r(d(G)-1)\right] \tag{21}$$

By substituting $a_0 = 4$ and $r = 4$ we get the main formula:

$$w - 1 \leq d(G)\left[4 + 2(d(G)-1)\right] \quad \text{or} \quad 2d(G)^2 + 2d(G) + 1 \geq w \tag{22}$$

3.1 Ideal and optimal graph

Maximum numbers of nodes that appear in each layer form the number sequence. The total number of nodes is determined by:

$$w_i = 1 + w_1 + w_2 + \ldots + w_{d(G)-1} + w_{d(G)} = 1 + \sum_{d=1}^{d(G)-1} w_d + w_{d(G)} \tag{23}$$

where $w_{d(G)}$ is equal to node number of last layer.

Definition 3 *Chordal ring fulfilling the formula (23) under assumption that two layers numbered from* 1 *to* $d(G) - 1$ *are disjoint but the last layer contains the remain nodes, is referred as an ideal graph.*

Definition 4 *The ideal chordal ring with the last layer that reaches the maximal value is called the optimal graph.*

In optimal graph the total number of nodes is expressed in the form:

$$w_o = 1 + \sum_{d=1}^{d(G)} w_d \tag{24}$$

but the average paths length is equal to:

$$d_{\text{avo}} = \frac{\sum\limits_{d=1}^{d(G)} dw_d}{w-1} \tag{25}$$

The formulas (24) and (25) allow to define ideal graph parameters in the form:

$$\begin{aligned} w_i &= w_{o(d(G)-1)} + w_{d(G)} \\ d_{\text{avi}} &= d_{\text{avo}(d(G)-1)} + \frac{d(G)w_{d(G)}}{w-1} \end{aligned} \tag{26}$$

The ideal graphs described above constitute a reference point for estimation the studied chordal ring structures. The expression (22) that describes a dependence between the node number and chordal ring diameter, allows to determine fourth degree chordal ring in which the nodes number reaches the maximal value.

Definition 5 *In the optimal chordal ring of fourth degree the node number* w_o *can be expressed as the function of diameter* $d(G)$ *in the following form:*

$$w_o = 2d(G)^2 + 2d(G) + 1 \tag{27}$$

For optimal chordal ring with wo nodes the diameter is equal to:

$$d(G) = \frac{\sqrt{2w_o - 1} - 1}{2} \tag{28}$$

The study of optimal chordal rings has shown that the optimal structures can be obtained only for one value of chord, called in this paper the optimal chord – s_o. In the table 8, the node numbers of optimal chordal rings, as well as the optimal chord for different diameters $d(G)$ were presented :

$d(G)$	1	2	3	4	5	6	7	8
w_o	5	13	25	41	61	85	113	145
s_o	3	5	7	9	11	13	15	17

Table 8

Before the optimal chord can be determined, the upper and the lower bounds will be defined.

Theorem 3 *For the optimal chordal ring the optimal chord is bounded as follows:*

$$2d(G) + 2 \geq s_o \geq 2d(G) \tag{29}$$

Proof: In the optimal chordal ring each node can be reached from any other one by the use of at most $d(G)$ chords. In the ring there are nodes that are reachable from the starting ones by $d(G)$ chords of length s_o. So, in the optimal ring the maximal distance between any two nodes cannot be greater than the half of the node number, i.e.:

$$s \times d(G) < w \tag{30}$$

On the other hand, using $d(G) + 1$ chords of length s_o, it is possible to pass the entire ring, so the expression:

$$s_o \times [d(G) + 1] > w \tag{31}$$

is true. By putting the number of nodes of optimal chordal ring into expression $s_o \times d(G) < w$, one can obtain:

$$s_o \times d(G) < w = 2d(G)^2 + 2d(G) + 1 \tag{32}$$

As all the calculations are done in the set of integer numbers, so the following is true:

$$s_o \times d(G) \leq d(G)^2 + 2d(G) \tag{33}$$

Therefore the left side of the expression (29) is fulfilled. Transforming in turns the expression $s_o \times [d(G) + 1] > w$ we obtain:

$$s_o \times [d(G) + 1] > 2d(G)^2 + 2d(G) + 1 \tag{34}$$

and next:

$$s_o \times [d(G) + 1] > 2d(G)^2 + 2d(G) \tag{35}$$

so $s_o > 2d(G)$. Obtained expression proves the theorem 3. □

Having found the lower and the upper bounds of the optimal chord it is possible to find its real value. This value is defined by the theorem 4.

Theorem 4 *In the optimal chordal rings of fourth degree the length of optimal chord is equal to:*

$$s_o = 2d(G) + 1 \tag{36}$$

Proof: In order to prove theorem 4, one has to show that all the nodes existing between the source node (to simplify let?s assume its number equal to 0) and the final node of the chord (number s_o), can be reached from starting node with the use at most $d(G) chords.$ There exist three routes to reach these nodes:

- by moving clockwise with the use $d(G)$ chords with length 1, those nodes have the numbers $1, \ldots, d(G)$;
- by the use of one so length chord clockwise and $(d(G) - 1)$ counter clockwise chords of length 1 – numbers of the nodes $s_o - [d(G) - 1], \ldots, s_o$;
- precisely by the use of $d(G)$ counter clockwise chords of length s_o – number of the reached node is equal to $d(G) + 1$.

Let us consider the last route. Due to that we move counter clockwise from node 0, the following expression is fulfilled:

$$w - s_o \times d(G) = d(G) + 1 \tag{37}$$

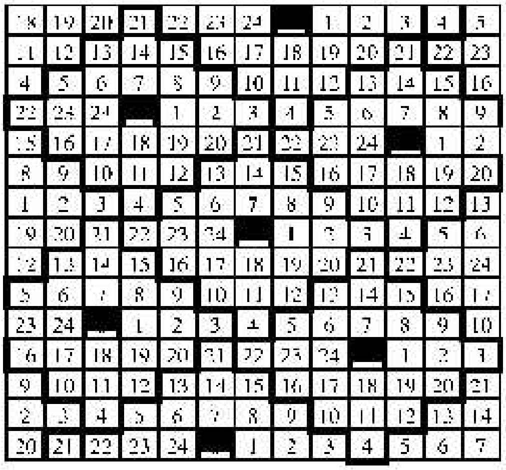
Substituting w by $2d(G)^2 + 2d(G) + 1$ we get:

$$2d(G)^2 + 2d(G) + 1 - s_o \times d(G) = d(G) + 1 \tag{38}$$

and finally $s_o = 2d(G) + 1$ what proves theorem 4. □

The proof of theorem 4 has also been shown in [12]. For the proof the authors has constructed the plane (figure 11) and have analysed all the vectors that connect the starting node with the other. From this table they have concluded that the only value of the optimal chord length is equal to $s_o = 2d(G) + 1$. This chord enables to arrange the cells which gives graph $G(2d(G)2 + 2d(G) + 1; 1, s_o)$.

Theorem 5 *The optimal chordal ring of fourth degree consists of two Hamiltonian cycles.*



18	19	20	21	22	23	24	■	1	2	3	4	5
11	12	13	14	15	16	17	18	19	20	21	22	23
4	5	6	7	8	9	10	11	12	13	14	15	16
22	23	24	■	1	2	3	4	5	6	7	8	9
15	16	17	18	19	20	21	22	23	24	■	1	2
8	9	10	11	12	13	14	15	16	17	18	19	20
1	2	3	4	5	6	7	8	9	10	11	12	13
19	20	21	22	23	24	■	1	2	3	4	5	6
12	13	14	15	16	17	18	19	20	21	22	23	24
5	6	7	8	9	10	11	12	13	14	15	16	17
23	24	■	1	2	3	4	5	6	7	8	9	10
16	17	18	19	20	21	22	23	24	■	1	2	3
9	10	11	12	13	14	15	16	17	18	19	20	21
2	3	4	5	6	7	8	9	10	11	12	13	14
20	21	22	23	24	■	1	2	3	4	5	6	7

Figure 11: Supplementary table used in paper [12]

Proof: Due to the fact that the ring with chords of the length 1 is Hamiltonian cycle, it is enough to prove that optimal chord of the length so also generates Hamilton cycle. To prove that, it is enough to show that the number of nodes w_o and optimal chord length s_o are relatively prime. By converting the expression (27) defining the nodes number in the optimal chordal ring we get:

$$w_o = 2d(G)^2 + 2d(G) + 1 = \frac{s_o^2 + 1}{2} \tag{39}$$

We need to show that there does not exist the number (different to 1) that divides the numbers $(s_o^2 + 1)$ and s_o. If s_o is a prime then theorem is always true. So let us assume, that s_o is not a prime number and there is a number p, which divide s_o. We will prove than that p does not divide $(s_o^2 + 1)$. As s_o must be odd, so p must be also odd. Let us write s_o as the product of the form $p \times q$. Therefore $(s_o^2 + 1) = p^2 \times q^2 + 1$. The first element of this expression is divided by p but the rest equal to 1 is not divided by p. So $(s_o^2 + 1)$ cannot be divided by p, what proves the theorem 5. □

The average path length of optimal chordal ring of fourth degree is expressed in the form:

$$d_{\text{avo}} = \frac{1}{w_o - 1} \sum_{j=0}^{w_o - 1} d_{\min}(v_0, v_j) = 4 \frac{\sum_{j=1}^{d(G)} j^2}{w_o - 1} \tag{40}$$

By substituting value of expression $\sum_{j=1}^{d(G)} j^2$ and the value of w_o we obtain:

$$d_{\text{avo}} = 4 \frac{d(G)[d(G) + 1][2d(G) + 1]}{6[2d(G)^2 + 2d(G)]} = \frac{2d(G) + 1}{3} = \frac{s_o}{3} \tag{41}$$

Optimal chordal rings can be built for a specific number of nodes only. Those rings, as we explained before, form a specific group of ideal graphs that are extended for any number of nodes. The ideal graph can be treated as the reference graph – virtual or real ones for analysed structure, as the parameters of this graph, i.e. its diameter and average path length are the lower bounds of parameters of analysed structures.

Definition 6 *Chordal ring with w_i nodes is called an ideal graph if it fulfils the following conditions:*

- *graph diameter $d(G)_i$ is expressed by:*

$$d(G)_i = \left\lceil \frac{\sqrt{2w_i - 1} - 1}{2} \right\rceil \tag{42}$$

- *in layer d number of nodes w_{di} is defined in the form:*

$$\begin{aligned} &\textit{if } d \neq d(G) \quad \Rightarrow \quad w_{di} = 4d \\ &\textit{if } d = d(G) \quad \Rightarrow \quad w_{di} = w_i - w_o \end{aligned} \tag{43}$$

  where: w_i – number of nodes in ideal graph, w_o – number of nodes in optimal graph with diameter $d(G)_i$?1,

- *the average paths length is expressed by the form:*

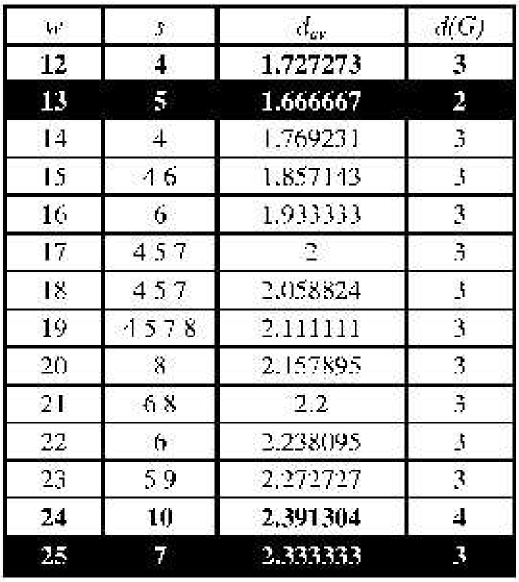
$$\begin{aligned} d_{avi} &= \frac{\frac{2[d(G)_i - 1] + 1}{3}(w_o - 1) + d(G)_i(w_i - w_o)}{w_i - 1} \\ &= d_{avo}\frac{w_o - 1}{w_i - 1} + d(G)_i \frac{w_i - w_o}{w_i - 1} \end{aligned} \tag{44}$$

  where $d(G)_i$ – diameter of ideal graph, w_i – number of the nodes of ideal graph, w_o – number of nodes of optimal graph with diameter $d(G)_i - 1$.

In the table 9 results for some ideal graphs has been depicted.

4 Conclusions

In this paper the properties of third and fourth degree chordal rings have been discussed. On the basis the results shown in first part of the paper the conclusions can be formulated as follows: The average length of the path in chordal rings has a bigger influence on the value of call rejection probability than diameter. The proposed formulas give similar or better results of the diameter values and average length of the paths to those met in the accessible literature. In [16] another way of the choice of structures possessing the best transmissions abilities is presented. In second part of the paper two classes of rings forth degree, namely ideal and optimal graphs, have been defined. In fact, the class of optimal chordal rings belongs to the



w	s	d_{av}	$d(G)$
12	**4**	**1.727273**	**3**
13	**5**	**1.666667**	**2**
14	4	1.769231	3
15	4 6	1.857143	3
16	6	1.933333	3
17	4 5 7	2	3
18	4 5 7	2.058824	3
19	4 5 7 8	2.111111	3
20	8	2.157895	3
21	6 8	2.2	3
22	6	2.238095	3
23	5 9	2.272727	3
24	**10**	**2.391304**	**4**
25	**7**	**2.333333**	**3**

Table 9

class of ideal ones. The optimal chordal rings posses strictly defined number of nodes (that depends on the diameter $d(G)$ of ring) and there exist only one optimal chord with the length $2d(G) + 1$ that allows to build the optimal chordal ring. It has been also proved that the optimal chordal rings of fourth degree can be built with the use of two Hamiltonian cycles. The ideal chordal rings in turns, have been introduced as the reference graphs (sometimes virtual only eg. when graph posses $2d(G)^2 + 2d(G)$ nodes – table 9) for comparison of properties of any analysed chordal ring.

References

[1] "www.alcatel.com/products."

[2] "www.alcatel.pl/technology/ngn.html."

[3] L. N. Bhuyan, "Interconnection networks for parallel and distributed processing," *IEEE Computer*, vol. 20, no. 6, pp. 9–12, 1987.

[4] G. Kotsis, "Interconnection topologies and routing for parallel processing systems," Tech. Rep. ACPC/TR92-19, ACPC, 1992.

[5] C. Gavoille, "A survey on internal routing."

[6] W. Arden and H. Lee, "Analysis of chordal ring network," *IEEE Transactions on Computers*, vol. 30, no. 4, pp. 291–295, 1981.

[7] P. Morillo, F. Comellas, and M. Fiol, "The optimisation of chordal ring networks," in *Communications Technology* (Q. Yasheng and W. Xiuying, eds.), pp. 295–299, World Scientific, 1987.

[8] M. Freire and H. da Silva, "Wavelength-routed chordal ring networks," in *Business Briefing: Global Optical Communications*, pp. 151–154, 2001.

[9] M. Freire and H. da Silva, "Performance comparison of wavelength routing optical networks with chordal ring and mesh-torus topologies," *ICN*, vol. 1, pp. 358–367, 2001.

[10] M. Freire and H. da Silva, "Influence of chord length on the blocking performance of wavelength routed chordal ring networks," in *5th Working Conference on Optical Network Design and Modeling (ONDM?2001)*, (Vienna).

[11] S. Brandt, "Statistical and computational methods in data analysis," tech. rep., Wydawnictwo Naukowe PWN, Warszawa, 2002.

[12] L. Narayanan and J. Opatrny, "Compact routing on chordal rings of degree four," *Algorithmica*, vol. 23, pp. 72–96, 1999.

[13] L. Narayanan, J. Opatrny, and D. Sotteau, "All-to-all optical routing in chordal rings of degree 4," *Algorithmica*, vol. 31, pp. 155–178, 2001.

[14] A. L. Liestman, J. Opatrny, and M. Zaragoza, "Network properties of double and triple fixed step graphs," *International Journal of Foundations of Computer Science*, vol. 9, pp. 57–76, 1998.

[15] R. Browne and R. Hodgson, "Symmetric degree-four chordal ring networks," *IEEE Proceedings*, vol. 137, no. 4, 1990.

[16] S. Bujnowski, *Analysis & Synthesis Homogeneous Structure Networks Connecting Communications Modules*. PhD thesis, ATR Bydgoszcz, 2003.

Solving puzzles and games by Evolutionary Algorithms

Agostinho Rosa*

Abstract

This presents the application of Evolutionary Algorithms to several puzzles and games. The solutions of the puzzles Pentominoes and Mastermind and two player games gomoku and chess are presented. As pedagogical and illustrative examples the presentation stress on the search space coding and the fitness function construction. Performance results are presented.

Keywords: Evolutionary Algorithms, Pentominoes, Mastermind, GoMoku, Chess, Genetic algorithms, meta-heuristics, co-evolutionary algorithms.

1 Introduction

Supporting the pedagogic framework of learning by example is the underline motivation of the set of works to be presented in this paper. Puzzles and games are and has been appropriate paradigms as learning examples for several reasons. They are well known for large population, easy understanding of the underlining rules and operations, real life problems, could be simple or very challenging, association to an intelligent behavior and could also be entertaining. The Artificial Intelligence community had used games and puzzles extensively for building intelligent systems, so it is a known and effective path. Evolutionary Algorithms gained popularity very recently and it is still a relative unknown subject and sometimes hard to understand not it principles but the reasons of it success. The examples studied covers a wide range of different problems from simple to very complex and the ultimate objective is to shed some light on possible ways to apply evolutionary algorithms to combinatorial search problems.

Introduction to evolutionary algorithms and detailed description of the puzzles and games is not given in this paper, one may find them elsewhere [1, 2, 3, 4, 5, 6]. What will be described with some details are the coding and fitness function described in the next few chapters.

*Instituto de Sistemas e Robótica. E-mail: acrosa@isr.ist.utl.pt

2 Brief Introduction to Evolutionary (Genetic) Algorithms

The operational approach or pseudo-code of the algorithms can be summarized as follows:

1. Candidates/possible solution to the problem (individual), is the phenotype. The coding of the candidate solution in terms of computer representation is called the genotype, in most of the cases, the chromosome. It is usually a set of variables, each variable is called a gene and the possible values of the gene variable are the alleles. All the possible combinations of the alleles in the different genes of a chromosome is the search space.

2. The algorithms is composed by a set of individuals (or chromosomes) making the population; each individual in the population corresponds to a candidate solution to the problem.

3. The evolutionary paradigm bias is introduced in the process of transforming one population to another along time steps. More fit (adapted to the environment) individuals have higher probability to reproduce or in other words the number of off springs is proportional to the individual fitness or performance. This part implements the selective process and is the selection operator. A pragmatic approach to avoid loosing already good individuals in the population is to artificially force their presence in the next population, this forcing procedure is known as elitism.

4. The reproduction process itself is the generation of new individuals from current individuals. The process known as variation may have many forms and implementations. In the genetic paradigm, the crossover operator, where (two) parents generate children through the exchange of their genes. The crossover operator is view as a combinatorial search on the alleles present in the population (exploitation). On the other hand the mutation operators change the allele to any possible allele, enabling the scan of any possible position in the search space (exploration).

5. The "intelligence" of the algorithms resides in the fitness function. All individuals are evaluated through this fitness function designed to achieve the target objectives of the candidate solutions.

6. This process repeats until runs out of time, or a specific target is reached or a number of repetition was performed or from an external termination request.

2.1 Basic Genetic Algorithm

Genetic algorithms (GA) are a stochastic computational technique based on the theory of evolution. [7] They operate on a population of solutions represented by chromosomes. Each chromosome consists of a number of genes and any chromosome is one possible solution for the problem. The evolution of the new populations are obtained repeatedly by reproduction and genetic operators (Figure 1) in order to search the optimum solution or to satisfy our goals like to win the opponent.

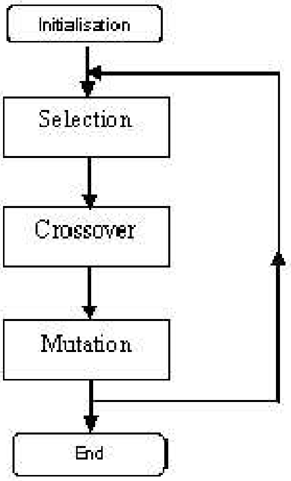
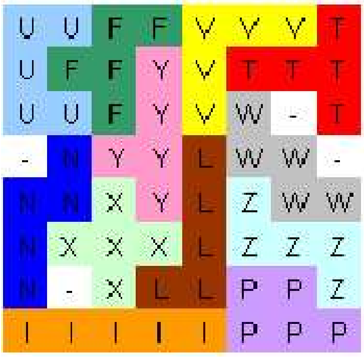


Figure 1: GA-Cycle

3 Pentominoes [3]

The Pentominoes puzzle [8] used consists in trying to place twelve pieces, of five squares each, on an $8 \times 8 = 64$ board, without overlapping (there are four places left vacant). These twelve pieces are the possible configurations one can obtain joining together five squares at least by one side. Figure 2, shows, in a solution, the twelve pieces and their usually associated names.



U	U	F	F	V	V	V	T
U	F	F	Y	V	T	T	T
U	U	F	Y	V	W	-	T
-	N	Y	Y	L	W	W	-
N	N	X	Y	L	Z	W	W
N	X	X	X	L	Z	Z	Z
N	-	X	L	L	P	P	Z
I	I	I	I	I	P	P	P

Figure 2: a solution to the 8×8 Pentomino Puzzle

3.1 Coding

The genotype is a single chromosome built as a sequence of twelve *pieces* (genes), each one having x and y coordinates, as well as orientation and mirroring (Figure 3).

X	Y	R	M

Figure 3: A gene. Each gene represents the location of a piece in the board. (R – rotation; M – mirroring)

Considering all varieties of orientation and mirroring (eight varieties for each board location), the size of the search space is 3.25×10^{32} (approximately $2^{9\times12}$). The fitness evaluation is obtained by counting the number of occupied board locations. The objective of the search is to find a board with 60 occupied locations. The observation of a piece classifies it as good or bad. Good, in this context means that the piece does not overlap with any other in the same chromosome, and it is within the board boundaries. The mutation operator acts upon genes (pieces), flipping its bits randomly. Each gene have a counter (initialized to zero) that are incremented by one unit when the corresponding piece overlaps or is not completely inside the board boundaries and it is decremented otherwise. The counter saturates at a constant pre defined value MAXINF.

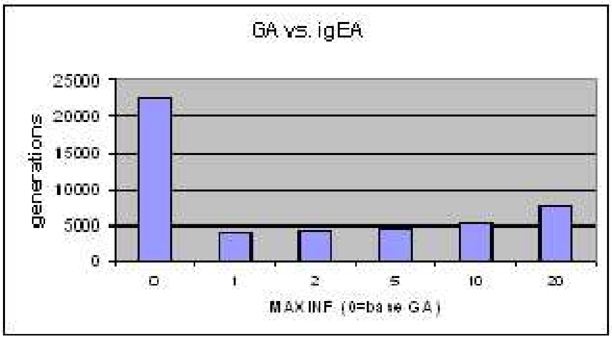


Figure 4: **(GA comparison):** mean number of generations to find a solution; when MAXINF=0, it is the base GA, otherwise it is igEA with different MAXINF values.

3.2 Results

Figure 4 presents the comparison between a basic Evolutionary Genetic Algorithms (bGA) [7] for this problem and an igEA [3] version with several MAXINF values. The population size was 100 and the results are expressed in terms of mean number of generations to reach a solution. For each combination of parameters, 100 runs were made. Results clearly show that igEA outperforms the bGA.

4 Mastermind [4]

The MASTERMIND game [9] is known as a logical game for false two players, an encoder and a decoder. The encoder builds a Secret Code using any combination (with or without repetition) of the existing N colors, taking in consideration that exists P positions to be filled. The decoder tries to duplicate the exact colors and positions of the Secret Code. Each time the decoder establishes a code as a possible solution, the encoder should provide information to decoder by presenting a Code Key. The Code Key is a set of black, white and null tokens (absence of tokens) built with the following rules:

1. **Black tokens -** one black token for each color in the code, given by the decoder, which matches at the same time the cooler and its position in the Secret Code.

2. **White Pieces -** one white token for each color in the code, given by the decoder, which matches one color but not the position in the Secret Code.

3. The game ends when the decoder finds the Secret Code. The encoder should present P black tokens as the Code Key and reveal the Secret Code.

The number of possible combinations for the Secret Code is given by N^P. The two most widely commercial configurations for the game is the 6×4 with 6 colors and 4 Pegs and the 8×5 with 8 colors and 5 pegs. For the 6×4 game, the optimal strategy based on exhaustive search needs 4.3 moves in average to solve the problem [10]. A dynamic constraint optimization using GA. without crossover to the Master Mind problem is described in [11].

In our approach, the fitness function is formulated as a dynamic self constraint optimization problem, but the crossover and mutation operators are dependent of the fitness of the individuals of the population and this characteristic provides them effective probabilistic self-adaptive behavior.

4.1 Coding

For the implementation of the algorithm the following elements were used: Each peg is coded by a gene with N alleles. Each chromosome with P pegs (genes) is a candidate solution to the problem. Integer coding is used in order to allow genes boundary crossover.

Since this particular problem has the final goal completely unknown we needed to make some adjustment to the standard GA. The fitness of the individuals can only be estimated from the incomplete knowledge gathered during the sequence of guesses. The initial population of fixed size is randomly generated, having however the particularity that all the individuals are

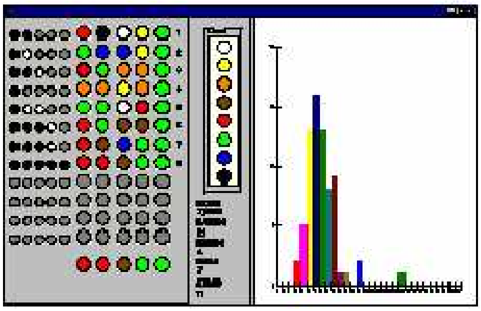


Figure 5: shows a picture of a version 5 × 8 of the game with the histogram of the number of trials.

made out of chromosomes without repeated colors. This particularity is the result of a series of tests, which indicated that in this case the information provided is richer. Exception on the initial population where the trial element is randomly chosen, on every generation one element (trial element), with the best fitness is chosen to be played. A new population is built based on the preceding one by applying the genetic operators described in the following sections.

4.2 Selection

The selection is proportional to the fitness (roulette wheel). The next population replaces completely the original population. When elitism is applied a percentage of best elements of the parent population will substitute the corresponding number of worst elements of the child population. The crossover operator is dependent on the (conditional) fitness of the elements calculated in relation to the current trial.

4.2.1 Conditional Fitness

Lets define XB_i and XW_i the number of black and white pieces, respectively, attributed to the individual i of the population after selection, by comparing to the trial element T. For each individual i the number of pegs that may be correct in both position and color (HB_i), and in color only (XW_i) are calculated as:

$$\begin{aligned}\mathrm{HB}_i &= \min(\mathrm{XB}_i, \mathrm{NB}) + \min(\mathrm{XW}_i, \mathrm{NW}) \\ \mathrm{HW}_i &= \mathrm{abs}\,(\mathrm{NB} - \mathrm{XB}_i)\end{aligned}$$

Where min is the minimum operator, NB and NW are the number of black and white tokens obtained by the last trial, respectively and abs is the absolute value operator.

4.2.2 Conditional Crossover

Two individuals from the population after selection i and j are chosen randomly, thenpositions genes are picked randomly for the new individual, in these positions the pegs will be replicas of the pegs of the individual i , additional HW_i positions are also selected Randomly, for these positions the colors of individual i will be attributed to the new individual but at different and non yet occupied positions. The rest of the $(\mathrm{P} - \mathrm{HB}_i - \mathrm{HW}_i)$ pegs of the new individuals are obtained from the individual j. The second individual is generated by the same method, by exchanging the functions of the individuals i and j and selecting the positions using HB_j and HW_j. For a better understanding of the operator, an example of creating the first new individual with $\mathrm{HB}_i = 2$ and $\mathrm{HW}_i = 1$ is shown in figure 6:

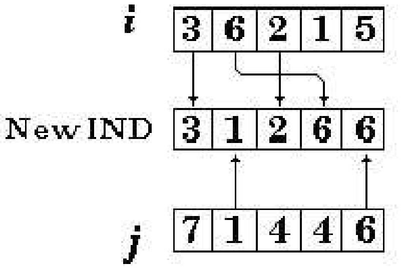


Figure 6: Application of Conditional Crossover.

Two genes ($\mathrm{HB}_i = 2$), the first with value 3 and the third with value 2 are selected randomly from the **i** element to be passed directly without change to the new individual/element. Only one ($\mathrm{HW}_i = 1$) gene, the second with value 6 is passed to the new individual but to a different non occupied location, in this example to the fourth position. The remaining locations are filled by the corresponding genes value of the **j** element.

This step is applied successively, without repeating the already selected individuals, until the new population is obtained. The Crossover process is different when all the colors of the Secret Code were found but not their positions. In this case only the last trial will be used (the population is substituted by clones of the last trial), only conditional fitness crossover operator is actually applied, since, for this situation, it is obvious that HB = NB and HW = NW. The crossover operator is always applied (probability 1), it may be considered as a self adaptive operator in the sense that the actual number and extent of crossovers are dependent on the fitness of each individual it reduces as the fitness increases.

4.2.3 Mutation

An individual of the population after crossover is randomly selected for mutation. A peg of this individual is then randomly selected for application of the mutation operator. The mutation on a peg is made by replacing the color of the peg by the color in the Color Pool (CP, a varying set of tagged colors available for mutation) with the smallest Number of Occurrence (NoC), but different from the color to be mutated.

The NoC of each CP is calculated in the following manner: When a color appears in a peg of an individual then the NoC of the color in the CP, is incremented by a parametric value, named Play Color (PC). If a color is used in a mutation, than the NoC of the color in CP is decremented by a parametric value, named Mutation Color (MC).

The number of mutations (NM) performed on an element is proportional to the number of null pieces (NN) attributed to the last trial, $\mathrm{NN} = \mathrm{P} - \mathrm{NB} - \mathrm{NW}$, to the Size of the population (SP), to the present Generation Number (GN) and to the Mutation Weight (MW) value:

$$\mathrm{NM} = \mathrm{NN} \times \mathrm{SP} \times \mathrm{GN} \times \frac{\mathrm{MW}}{100}$$

The mutation operator is also always applied. It is also a self-adaptive operator in the sense that the actual mutation rate of the gene also depends on the changing fitness.

4.2.4 Elitism

Two types of Elitisms were implemented. The first type is applied when there is an element in the trial set where Code Key has NB = 0 and NW = 0, which means that all colors in the trial element are not in the Secret Code, two actions are taken: first, all colors present are eliminated from the color pool, therefore are not used by the mutation operator; second, these colors are also taken away from all the individuals in the new population, and replaced by the ones in the CP. The second type of elitism is the replacement, of a percentage, of the new generation of individuals with low fitness by the same percentage of individuals of previous generation with highest fitness (the replacement is made only for the individuals of previous generation that are really better than the individuals of the new generation). The second type of elitism is not applied if all the pegs in the Secret Code had already been found.

4.3 Fitness function

The Fitness Function measure how a given chromosome satisfies all trials constraints already obtained. For a chromosome fitness calculation it is required to have in account the following variables: NB(GN), NW(GN) and NN(GN) - number of black, white and null pieces, respectively, attributed to the trial element associated with the generation number, GN, evaluated by the Secret Code.

$$\mathrm{NN(GN)P} - \mathrm{NB(GN)} - \mathrm{NW(GN)}$$

Where $\mathrm{XB}_I(\mathrm{GN})$, $\mathrm{XW}_I(\mathrm{GN})$ and $\mathrm{XN}_I(\mathrm{GN})$ are the number of black, white and null pieces respectively, attributed to the individual I having as reference the trial element GN.

For a given GN:

$$\begin{aligned}\mathbf{A}_i(\mathrm{GN}) &= 10 - |\mathrm{NB}(\mathrm{GN}) - \mathrm{XB}_i(\mathrm{GN})|\mathbf{B}_i(\mathrm{GN}) = \mathrm{P} - |\mathrm{NN}(\mathrm{GN}) - \mathrm{XN}_i(\mathrm{GN})| \\ \mathbf{C}_i(\mathrm{GN}) &= \mathrm{P} - |\mathrm{NB}(\mathrm{GN}) - \mathrm{XB}_i(\mathrm{GN})|\end{aligned}$$

The values $\mathbf{A}_i(\mathrm{GN})$ and $\mathbf{C}_i(\mathrm{GN})$ are proportional to the numbers of positional agreements between individual i, and generation GN of the Code Key. However, for the calculation of $\mathbf{C}_i(\mathrm{GN})$ it is already known that the correct colors have already been found, so the value $\mathbf{B}_i(\mathrm{GN})$ is, when the individual i, agrees, in terms of colors, with the GN generation Code Key.

4.3.1 Linear scaling

In the Linear case, the Fitness Function for the individual i of the GN generation is given by:

$$\mathrm{f}_i = \sum_{n=0}^{\mathrm{GN}} \mathrm{PositionWeight} \times \mathbf{A}_i(n) + \mathrm{ColourWeight} \times \mathbf{B}_i(n)$$

or when all colors of the Secret Code is found:

$$f_i = \sum_{n=0}^{\mathrm{GN}} \mathrm{PositionWeight} \times \mathbf{C}_i(n) + \mathrm{ColourWeight} \times \mathbf{B}_i(n)$$

4.3.2 Power scaling

In the Power case, the Fitness Function for the individual i of the NG generation is given by:

$$\mathrm{f}_i = \sum_{n=0}^{\mathrm{GN}} \mathrm{PositionWeight} \times \mathbf{A}_i(n) \times \mathrm{ColourWeight} \times \mathbf{B}_i(n)$$

or when all colors of the Secret Code is found:

$$f_i = \sum_{n=0}^{\mathrm{GN}} \mathrm{PositionWeight} \times \mathbf{C}_i(n) \times \mathrm{ColourWeight} \times \mathbf{B}_i(n)$$

4.4 Results

Due to the large number of parameters a default set is defined as follows:

The tests were done with different population size and fitness function scaling. In all tests the runs were repeated 500 times and the histogram of the number of the moves needed and corresponding statistical data are shown. In all tests performed the algorithm reached always the solution within 24 trials.

The test with default parameters set is shown in figure 3, where the average number of moves necessary for solution is 7.538.

The histogram of figure 7 is obtained by running the algorithm with linear scaling in the fitness function instead of power law scaling. For this problem the power scaling performs better than linear scaling as shown by the number of average moves needed for solution, 7.538 and 8.894 respectively, see figure 8.

POPULATION SIZE = 150
ELITISM_WEIGHT = 0.02
PLAY_COLOR = 0.5
MUTATION_COLOR = 1.0
MUTATIONWEIGHT = 1.0
GENERATION* = 30
POWER FITNESS FUNCTION
COLOR WEIGHT = 2.0 (8.0)
POSITION WEIGHT = 8.0 (2.7)

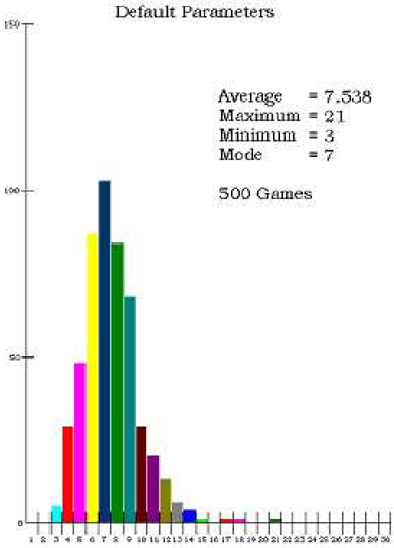


Figure 7: Histogram of the trials with default parameters.

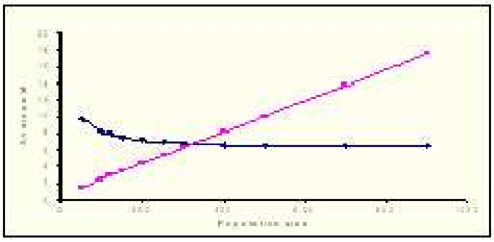


Figure 8: Histogram of the trials with default parameters with Linear Scaling

The effect of population size on the number of average moves needed for solution using the default parameter set is shown in figure 9.

The initial increase of the population size has a noticeable effect on the number of average moves (#AM) for solution. But after 250 elements in the population, the curve is still decreasing monotonically but very slowly (for 250 the #AM=6.954 to 900 with #AM=6.401), become almost flat, i.e. the size does not influence much the number of average moves. On the other hand the total number of combinations searched is increasing linearly with the population size (PS=250 5.3% search space and PS=900 17.6%). These results suggest the use of niching or speciation in order to take advantage of a larger population.

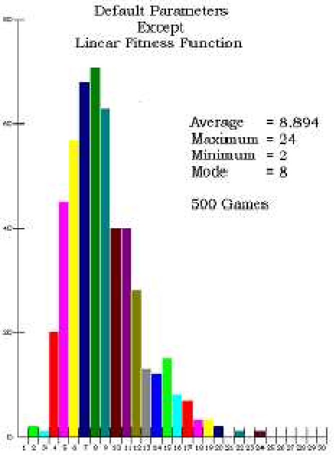


Figure 9: Effect of population size on the average moves

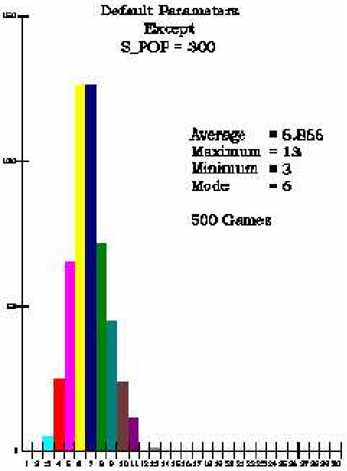


Figure 10: Reduction of Population size to 300

The histogram of 500 runs with population size equals to 300 is shown in figure 10. By increasing the size of the population, the performance of the algorithm is also increased

The number of average moves is 6.866.

4.5 Conclusions

The solution presented includes a fitness function that represents simultaneously the incomplete information of each trial and the cumulative changing indications of the sequence of trials and a problem specific crossover and mutation operators conditioned by the outcome of previous trials. This combination yielded a more efficient solution for the problem. The algorithm was designed to play a trial for every generation, although it is simple to adapt it in order to give a trial only when a meaningful improvement in the fitness function is found. This improvement can be asserted using a pre-defined criteria (such as: until the fitness achieves a certain increment or a desired value or satisfaction of additional constraints; etc). It is not implemented due to trade off between the average number of moves and the number of fitness function evaluated (computation time).

It is important to evidence that this algorithm evaluates SP × NG (size of population number of generations) combinations repeated or not until it achieves the correct solution. That is, for SP=150, the number of evaluations made are, in average, $150 \times 7.538 = 1131$. This means that the algorithm searches in the worst situation the average of 3.5% of all possible combinations (32768). For SP=300, (a good compromise of a low #AM and percentage of combinations searched) the algorithm examined 6.3% of the search space, with a gain of 9% in the average number of trials (7.538 to 6.866). It seems that for population size beyond a certain point it did not provide much improvement in performance. Results presented in reference [3], for a problem of similar size, (7×5 16807 combinations) needed 5.45 trials and 7755 combinations in average (1422 combinations per trial), leading to a percentage of searches of around 46%.

5 Gomoku [5]

Board games such as Gomoku, Chess, Checkers and Go etc, are interesting in our study because they offer pure and abstract competition without the confusions on the 2-gun's game or on the war's game. Five-in-Line (Gomoku) is a ancient Japanese strategic board-game. This game is a two players game. The players have unlimited number of pieces (stones). Each move consists of putting one piece in the crossing points of a 19×19 square board (Go-board) by the players in sequence. The moves of players are permitted in any direction and any free-position (non-occupied position by stones). The game is "over" when one of the player made five pieces in a row (horizontal, vertical or diagonal) or when the board is filled up. We explain how to apply the Genetic algorithm to this game and show how to avoid the use of search-tree. The next section shows some of the aspects of Genetic algorithms. Section 3 describes how to apply GA to this game and section 4 presents some experimental results. Finally we discuss future aspects and extensions to this work. The game is represented as a search problem through a space of possible game positions. This section describes both encoding and a states of game-board in order to represent our genetic information (gene, chromosome and population).

5.1 Coding

The board is a two dimensional 20×20 elements array. Each element may take the following values:

$$\begin{aligned} -1 &: \text{Free position in neighbour zone} \\ 0 &: \text{Free position} \\ 1 &: \text{Piece of computer-player (GA)} \end{aligned}$$

Piece of human-player (external opponent)

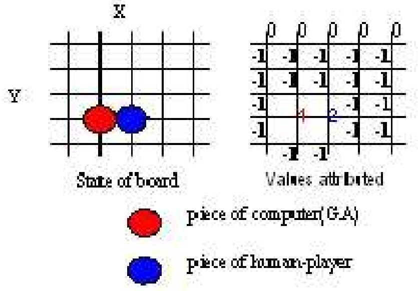


Figure 11: Representing the board's state and the associated values with the 2 dimensional array after two plays

The *neighbor zone* is the set of positions adjacent to all played positions (elements with values 1 and 2). Figure 11 shows the actual state in the game-board and the values of the 2 dimensional array after two moves. The neighbor zone provides us some information about the actual state of the board and it is updated after each player move. In order to use a genetic algorithm to solve this game, it is necessary to devise a representation for a move and a strategy in which maps onto gene and chromosome respectively. Each player move is a gene defined by 3 integers (x, y and fitness) where x and y represent the horizontals and verticals coordinates of the board respectively. The fitness is the resulting value from the calculation of fitness using the fitness function, i.e. the fitness value is the resulting of calculation after the playing on position (x,y).

Players / number of pieces	1	2	3	4
GA (red pieces)	4	16	100	5000
Human-player (blue pieces)	2	10	80	3000

Table 1: Weights table used in the algorithm

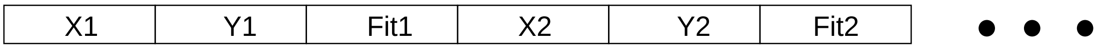


Figure 12: Structure of chromosome

Each chromosome represents a sequence of alternative plays by the algorithm and opponent. There is a predefined number Ng of genes in the chromosome. The population is composed by a predefined number Np of chromosomes, so they represent Np different strategies.

5.2 Initialization

```
The pseudo–code A is used to generate the first conflict–free population.
  After the first human–player move
    Generate chromosomes randomly
    GA-Cycle with the repair method
```

This pseudo-code produces the first population of valid moves or strategies. Clearly, it will produce any feasible population but it is used in order to obtain the first GA population.

5.3 Fitness function

The fitness value of a gene is the simple sums of the specific weights (see Table 1) of all sequences of pieces surrounding this gene. The algorithm search for all four directions (horizontal, vertical, two diagonals) to localize sequences of pieces from the both players.

Figure 13 showed how the weights can be used to calculate the fitness of gene X. In this case, this gene has a value of 18.([*16+2*] *16* for the couple[red pieces] and 2 for the single one[blue piece]).

5.4 Special operators

Special genetic operators are introduced because normal domain independent operators [6] are hardly to produce feasible strategies or moves. To obtain the feasible strategy or move, the special genetic operators must satisfy the following restrictions:

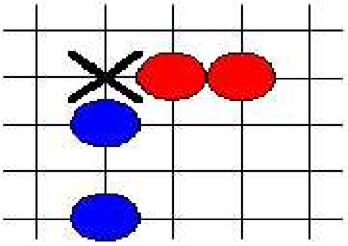


Figure 13: An example of calculation of fitness of gene using the weights table

R1: Chromosomes cannot contain equal genes.

R2: The genes must be in free-positions of the board.

5.4.1 Crossover

In the case of the absence of genes that violate the restrictions R1 and/or R2, the crossover is an insertion operator. Firstly, the GA selects randomly two chromosomes from the population and secondly it inserts a percentage Tx (rate) of randomly selected genes from one chromosome onto the end of the another. (Figure 14) The crossover rate in our case is 50%. The length of the resulting chromosome after crossover varies between 1 to 1.5 Ng. In the selection process only the first Ng sorted genes (see section 3.4.3) are retained to keep a fixed length chromosome.

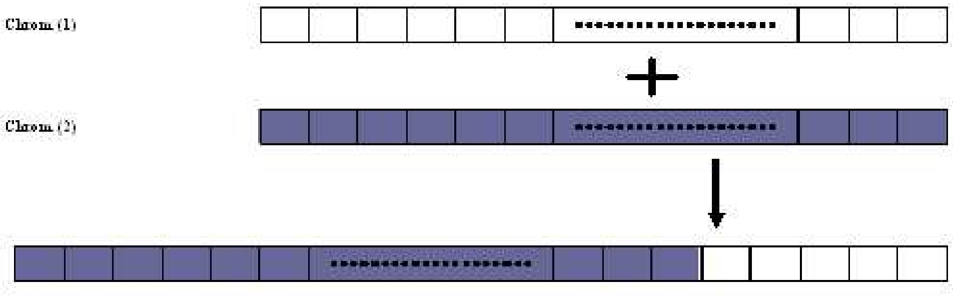


Figure 14: Offspring–chromosome resulting by crossover operator in absence of bad genes

In other cases, the special crossover uses the heuristic repair method (next section) to produce the feasible chromosomes.

5.4.2 Heuristic Repair

In order to guarantee the feasibility of strategies (chromosomes), the heuristic repair method is necessary because there are some bad genes (i.e. genes that violate any of the restrictions) during the crossover. In practice, the algorithm detects any of the two possible situations: the existences of two identical genes. genes in non-free positions. In the first situation, one of the infected gene is eliminated and his "place" is filled by another different gene that is generated randomly. In second situation, the genes will be moved into other free-positions.

5.4.3 Sorting with prediction

After crossover the chromosomes contains non-ordered new moves. In order to achieve the best possible strategies in the next $\mathrm{N}_{\mathrm{cop}}$ (parameter of prediction) moves with the available moves, the re-sorting of genes is necessary. The genes are sorted in descending order so the first gene has the highest fitness value. The second gene represents the answer of the move of the first gene, its choice must take into account the last move of the algorithm. Therefore all the fitness values must be recalculated in a virtual board where these moves are simulated. The other fitness values of genes are ordered by the same way, i.e. the fitness value of the next n-Moves is obtained in the conditions in which all fitness valued are recalculated taking into account the last (n-1)-Moves of human-player and of GA. With this coding from the ordered genes of the chromosomes the prediction of the next moves can be done.

The Elitism is applied and the best Ng genes of each chromosome are kept.

5.4.4 Decision-Making

The next computer's move is obtained from the actual population. It is necessary to decide which chromosome has the best strategy. The decision of the algorithm consists of two steps: For each chromosome of the actual population, the algorithm calculates the sum of the values (fitness) of the first Ncop genes. Choose the maximum of the step 1 and the first gene of the chosen chromosome is used by GA in the next move. There are many possible criterions in order to make the decision but it's not easy to choose the right one for each stage of the game. Let us to suppose one state of game shown in Figure 4 with one simplified board in which the GA is doing one attack's strategy and the last 2 moves are:

1. GA (red piece) to (7,6)
2. human-player (blue piece) to (3,5)

After these two moves, it's again the move of red piece (GA). There are 3 interesting sites –(3,6), (7,5) and (7,9) for GA. The Table 2 shows the values of genes in these three positions. It is obvious that the gene in the point (3,6) is the unique position to survive. But this point(3,6) has the smallest fitness value in Table 2. Let's suppose again the population of GA

Gene-marked position	Fitness value
X-(3,6)	**92**
X-(7,5)	106
X-(7,9)	102

Table 2: Comparison of fitness value of the 3 genes

has only these 3 genes. If the GA just use simply the best gene (the gene has the biggest fitness value) to play and therefore the GA will not be the winner. But in practice, the GA played a red piece in the point (3,6) using our criterion of decision-making. In fact sometimes the algorithm should to make some *sacrifices* (i.e. defense instead of attack) to survive. We verified this *sacrifices* in our experiments (under the condition without the action of the A.H.M.Defense, see next section).

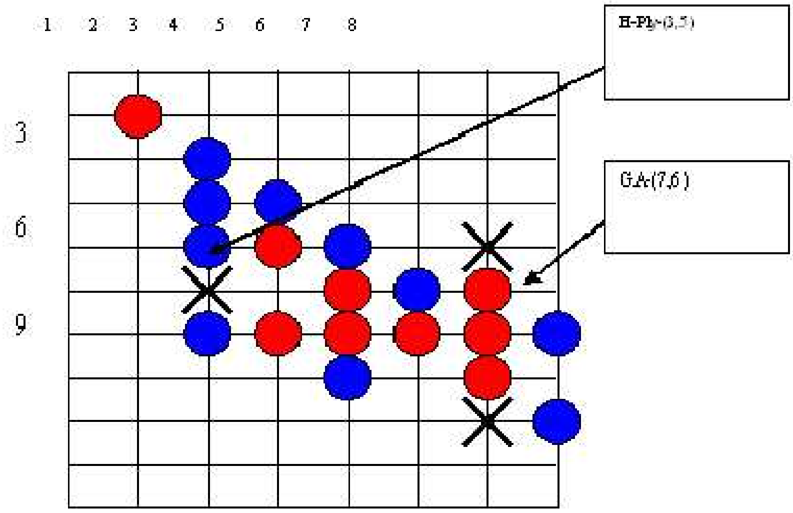


Figure 15: An example of decision making

5.4.5 Additional Heuristic Mechanism of Defense

In our first test of the algorithm, we considered only one generation during the lifetime of GA. Sometimes the GA hasn't "sufficient intelligence" to organize strategies of defense because of the lack in genetic information. We implemented one additional heuristic defense's mechanism and it is explained by the pseudo-code 1.

There are two parameters *HumanMoveFitness* and *LimiarDefense* in the pseudo-code 1. The first parameter is the fitness value of gene or chromosome resulting in the calculation by fitness function for the human player move. The second one is, a predefined value, the key of control in this mechanism. Clearly, if one very big number is used, the mechanism is eliminated because in practice, during all states of the game, there's not any gene with fitness value more than 20006. In fact, this number is the maximum fitness value for any gene.(Figure 16).

```
:
:
If (HumanMoveFitness > LimiarDefense)
{
   // mechanism of defense
   Make one new gene;
   Choice one chromosome randomly;
   Insert the gene into the chromosome chosen;
}
Else { do nothing}
:
:
```

Pseudo-code 1: Heuristic mechanism of defense.

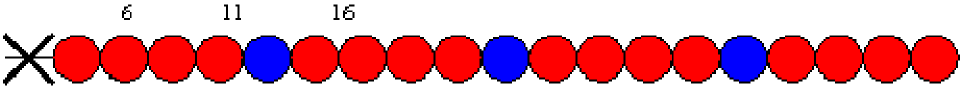


Figure 16: An example in A.H.M. Defense in which gene X has the fitness value of 20006.

The A.H.M.D. consists of the simple test of game's state and an insertion of one new gene onto one chosen chromosome randomly. There is no guaranty that the new gene will been chosen by GA but we used this mechanism to increase the genetic information in our first test of the algorithm. But when we used more than one generation on the lifetime of GA, the algorithm conducted alone some adaptable defense's and attack's strategies without the help of the heuristic defense.

5.5 Experiments and results

The default values set is shown in the following Table 3.

In this section we describe some important results. The parameter Ncop is the most sensitive one in computation time resulting in the increments of the time to simulate moves

Ng=20	Number of genes per chromosome
Np=20	Number of chromosome per population
Ncop=3	Prediction step
Tx = 0.5	Crossover rate
Ge = 15	Number of generations
LimiarDefense=1E8	Control of A.H.M.D.

Table 3: Default values list

in virtual board. The dimension of the population (Ng × Nc) is not determinant at least from tenth to hundredth of genes. The Crossover rate Tx should be close to 0.5 for an efficient search of the genetic information interchange and fast adaptation to the evolution of the game. We verified that the used weights (section 3.2) influence strongly the quality of the moves of GA. To test the performance of the algorithm, one set of 15 consecutive human-computer (the first author-computer) games were done and the results are showed in Figure 6,7,8 and 9. The average values in the defensive cases is 3.27 and 3.33 for the sequences of 3 and 4 pieces respectively. This two numbers mean that on average the GA had 3.27 and 3.33 times to blockade the sequences of 3 and 4 pieces of opponent respectively. The results showed that on average the GA attacked more than defended in the game. Our preference of strategy (attack is almost always more advantage than defend) is verified by this results. But if we change the used weights, for example, the 2-nd row change with the 3-rd row in the Table 1. In this situation, the GA would use the strategy of defense. But it isn't a good strategy in this game. Although the GA only won 8 times in all 15 tests with the first author but the revelatory outcome by the algorithm is still satisfactory. Perhaps the results aren't fair so 3 occasional players were invited to another set of 15 (5 games for each one) human-computer games. In these games, the GA won totally 12 times in all 15 games.(each player won one time in 5 games) So far the algorithm played against human opponents only. The human opponent needs a great deal of concentration in order to beat the algorithm otherwise will lose. The execution time for each move is below hundredth of a second for the default parameters set using a Pentium 133Mhz CPU,16Mb RAM, under Windows 95. The paper has demonstrated the practical utility of GA in the area of game-playing. At the same time, it has shown how we could implement a board-game without using the search tree or game-tree. In many situations, to solve the board's problem using the search tree isn't practical because we have to keep at least some parts of the search-tree in the physical memories. [8, 9, 10] Another fact is the simplicity of the fitness function allows large possibilities of improvement but it's not easy to arrange some appropriate weights to produce the replies of brilliant quality both in attack and in defense. It is difficult for us to classify the intelligence of the algorithm because of the lack in International Tournament of the game Five-in-Line. In our opinion, this plain

GA implemented could simulate a medium level human-player. For further information visit the small WWW page at: (see `http://laseeb.org/Portas_abertas/gomoku/`)

6 Chess [6]

Very few attempts have been made to address two players games by evolutionary algorithms. For the Go-moku game a genetic algorithm (GA) based program is described in [4]. For the checkers game a GA based search program is presented during in Gecco01 [12]. For the Chess game there is only a few experiments have been described [13]. Since chess is a well known game and there are many references describing the rules of the game and also different type of machine intelligence solutions, e.g. [14]. In this paper we restrict to the presentation of the implementation aspects of the co-evolutionary algorithm based chess machine player.

6.1 Game

Chess game is a 2 player strategic game played in an 8 × 8 "chess" board (alternating black and white squares). Each player has the same set of pieces (8 Pawns, 2 Knights, 2 Bishops, 2 Rooks, 1 Queen and 1 King); the different pieces have different movement patterns. The objective is to take the opponent king (check mate). Each player makes their moves alternatively [21] The search space in a game of chess problem is N × M, where N is number of possible choices and M the depth level (number of look ahead moves) is in average (N=35 and M=4) will ends up to 1500625 choices. The chess player could be implemented as a usual in evolutionary algorithms (EA), where the population represents candidate sequences of alternated (white and black) moves. Another possibility is to use two different populations, where each element of the population is a list of moves of only one of the opponents, black or white. The situation is usually known as co-evolution, where more than one population evolves together without competing directly with each other, only influences each other. The first option is simpler to implement but needs a very large population in order to cover a representative number of play sequences. The advantage of the second is a more compact representation of the moves and also provides a finer control on the number of play sequences to analyze. If K is the size of each of the two equal size populations, then we can obtain KxK possible combinations of alternated play sequences. Different strategies can be used to reduce the number of evaluations, like for example the most promising ones.

6.2 The co-evolutionary algorithm

The CA is an EA with two distinct populations, one for the black and one for the white pieces. The EA used for each population is the standard binary coded GA with fitness proportional selection with elitism, crossover and variation operators. The variation operators, crossover and mutation, are applied to these two populations independently, obtaining two offspring populations. The fitness of each individual is calculated in each generation, takes into account not only the quality of individual moves in his own population but also the quality of the possible moves of the other population. For example it does not payoff a move that takes a knight but loose in the next move the queen.

6.3 Population and coding

The two populations have the same number K of individuals; there is no specific reason to make them different. The size of the population depends on the depth level of the moves analysed in order to maintain a suitable percentage of coverage of the search space. Each individual is coded by a binary chromosome of variable number of genes, as shown in Figure 17. The number of genes is the depth level or the number of the play-ahead moves.

Chromosome with depth level 3:

$$\underbrace{011010}_{\text{gene 1}}\ \underbrace{100010011}_{\text{gene 2}}\ \underbrace{1011101}_{\text{gene 3}}$$

Chromosome with depth level 5:

$$\underbrace{011010}_{\text{gene 1}}\ \underbrace{100010011}_{\text{gene 2}}\ \underbrace{1011101}_{\text{gene 3}}\ \underbrace{11000010}_{\text{gene 4}}\ \underbrace{11110110}_{\text{gene 5}}$$

Figure 17: Chromosomes of varying length dependent on the depth level

The genes are binary codes, length and coding depends on the specific piece. Each gene represents a possible move and contains the information of the piece type, the move and the distance of the move. The generic gene structure is shown in Figure 18.

Piece (4 bits)	Direction (3 bits)	Displacement (3 bits)

Figure 18: Gene coding: Type of Piece, Move Direction and Displacement.

The Direction coding depends on the type of piece. The Pawns, Rooks, and Bishops have only 4 different directions; Knights and Queen have 8 and the King has 10 (8 for direction and 2 for Left and Right castle). Therefore, a maximum of 4 bits is needed for the Direction coding. The displacement is the number of squares a piece can move in any direction. Pawns, Knights and King have a fixed displacement of 1, only Rooks, Bishops and Queen need the displacement code. In order to have a more compact code and avoiding redundancy, a variable size coding is used. For the Pawns a total of 6 bits is needed and a total of 10 bits for the

Queen.

6.4 Variation Operators

The Variation operators used are: bit level uniform crossover operator with probability 0.7 and bit mutation and or simple inversion with probability 0.02 per bit.

6.5 Evaluation Strategy

As mentioned before, the fitness function evaluation is the heart of the player intelligence. A new set of chromosomes are formed through the combination of a pair of black and white chromosomes. Each chromosome is formed by alternated white and black genes, as shown in figure 19.

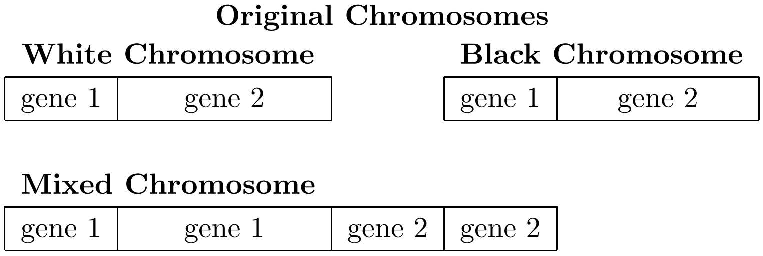


Figure 19: Mixed chromosomes, a combination of a pair of white and black chromosomes.

The new population is evaluated by a static fitness function and the fitness of best elements of the two populations is elaborated further through the mixed chromosomes. The P best white chromosomes are combined with the best Q black chromosomes, resulting in PxQ mixed chromosomes. The first move of white chromosome with the best mixed chromosome fitness is played. The corresponding first gene of both, black and white chromosomes is discarded and a new randomly generated gene is appended at the end.

6.6 Fitness function

Each piece in the game has a relative weight factor, absolute and relative positional (AP and RP) and menace-protection (MP) scorings. The relative weight is dependent on the relative value given to the different pieces in the game. There are several proposals for the relative weight and some are even optimized by GA through simulated game plays [15]. Here an empirical weight system was adopted, and it is similar to most often adopted ones, as shown in table 4.

Piece	Weight
Pawn	100
Knight	300
Bishop	320
Rook	500
Queen	900
King	3000

Table 4: Relative weight factor for the pieces.

The absolute positional scoring is the corresponding value an 8 × 8 weight matrix, it depends only on the position of the piece in the game board. It reflects the strategic positional value of the piece and is dynamic along the game. The relative positional scoring takes into account of the synergetic value of the interaction of pieces when they are close together. The menace-protection scoring depends on the balance value between the number of pieces protecting a specific piece and the number of menacing pieces from the opponent. When a piece is under menace the MP scoring is calculated by, subtracting the value of the menaced piece, adding the value of the attacked piece and subtracting the value of the protected piece. An example is provided in figure 4; the Black Knight is under the menace/attack of 3 white pieces and is protected by only 2 black pieces. If the last move was the Black Knight then the MP scoring of the Black knight only will be: subtract Black Knight (-300), add White Knight (300), subtract Black Bishop (-320), add White Bishop (320) and subtract Black Queen (-900). The total MP scoring will be -900.

6.6.1 Rooks

The two Rooks have a distance action and can protect each other, so the absolute position is not important; the AP scoring is substituted by The Proximity and Mobility Scorings. The Mobility scoring is the total number of squares each Rook can move to. The Proximity Score of the Rook is the sum of column and row distances of the Rook to the opponent King's position. The reason of this scoring is due to the movement restriction inflicted to the opponent King. For example a Rook at the distance of 1 Row and 2 Columns will add 24 points (14 +10).

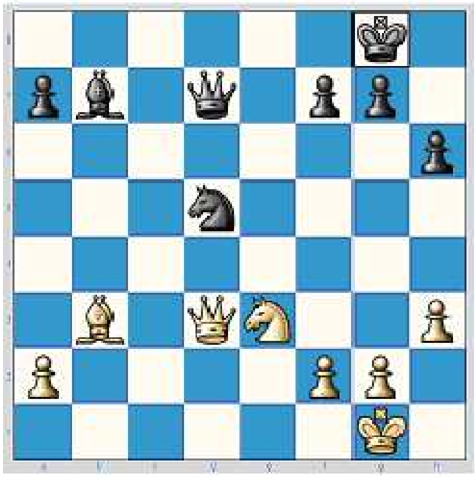


Figure 20: Menace-protection of a piece.

A Passed and Blocked white Pawn at E5 by a Black Knight at E6 is shown in figure 23.

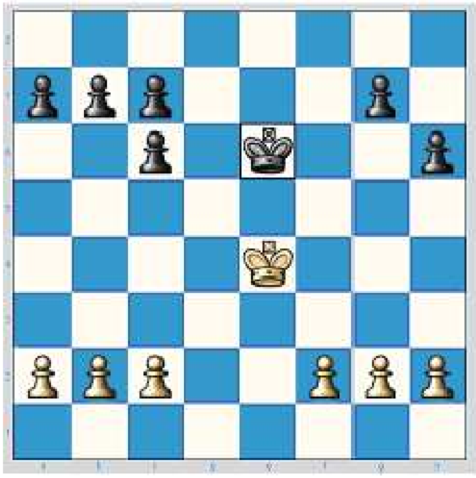


Figure 21: Doubled Back Pawn at C7.

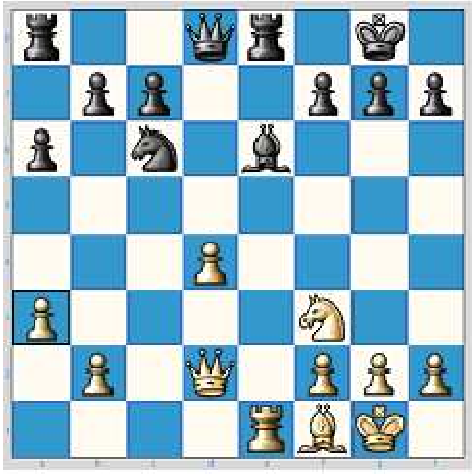


Figure 22: An Isolated White Pawn at D4.

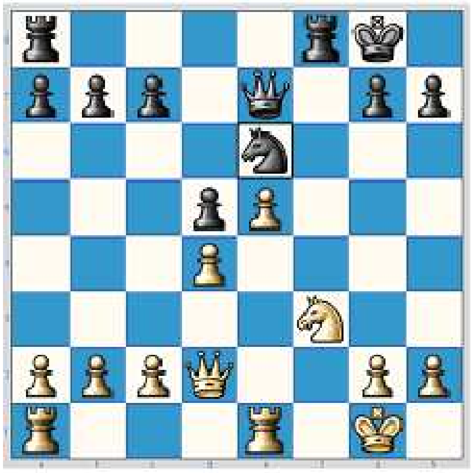


Figure 23: Passed and Blocked white Pawn at D5 by a Black Knight at D6.

The RP scoring is the following:

- Add 20 points for each Rook of the same color present in line 7 (or 2), as shown in figure 24.
- Add 15 points for the presence of 2 or more Rooks in the same column.
- Add 3 points if opponent Pawns are under menace.
- Add 4 points for absence of opponent Pawns in the same column.

- Subtract 12 points, if the King's Rook is moved before the King. (Will disable the Left Rook)
- Subtract 8 points, if the Queen's Rook is moved before the King. (Will disable the Right Rook).

The RP Scoring of Knight is:

- Add 3 points for each protecting Pawn to Knights at a proximity lower than 7.

6.6.2 Bishops

The AP scoring of the Bishop is very similar to the Knight AP Scoring. As reflected in the score differences, the movement restriction of the Bishop is less severe at the board edges than for the Knight. Figure 24 shows an example of free (D3) and blocked (D7) Bishops.

The RP scoring of the Bishop is:

- Add 20 points, if both Bishops of the same color are present. (Bishops are complementary, each acting on black or white squares exclusively).
- Subtract 3 points for each Pawn (independent of color) present in the adjacent diagonal. (The Bishops loose its effectiveness when obstructed).

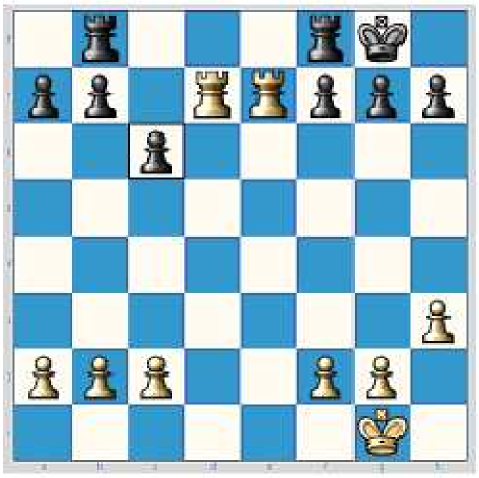


Figure 24: Two white Rooks, present at row 7 (add 40 points).

6.6.3 Queen

The Queen as the Rooks do not have AP scoring matrix. The Mobility can be very large, the upper limit is 28. Two different matrixes are used for the beginning and end stages of

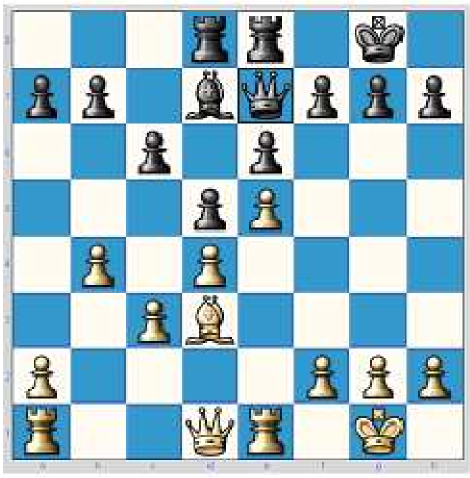


Figure 25: Free white Bishop (D4) and blocked Black Bishop (D7).

the game, reflecting the increased importance of the queen, when there are few pieces in play. The Proximity scoring is very high, especially for small values of proximity. When the Queen is very close to the opponent King it restricts drastically its movements, The RP scoring of the Queen is:

- Add 9 points for the presence of a Bishop in the same diagonal occupied by the Queen. (A protected Queen is a serious menace for the opponent King).
- Subtract 9 points, if the Queen is moved before two minor pieces (Knight or Bishops). (The Queen is a very powerful and valuable piece, should not be too exposed prematurely).
- Add 6 points if the Queen is on the row 7 (or 2)
- Add 6 points if the column of the queen is free from any Pawn.

6.6.4 King

The King is the most valuable piece in the game, there is no widely accepted weighting and scoring values, but it is of general consensus that it should at least be more than the some of all other pieces.

The RP Scoring of the King is:

- Subtract 10000 points if suffer check-mate
- Add 30 points if Castle
- Subtract 30 points if the first move of the King is not a Castle

- Add 10 points for each piece difference of friendly and foe pieces surrounding the King (the Queen counts here as 3 pieces).
- Subtract 10 points for each movement of protecting pawns after Castle.

There is also two AP scoring for the beginning and end stages of the game for the King. During the beginning of the game a well protected and covered positions are rewarded but advanced positions are highly penalized, as shown In table 12. At the end stages the centre of the board has more strategic value. Figure 26 shows an example of protected white King and unprotected black King.

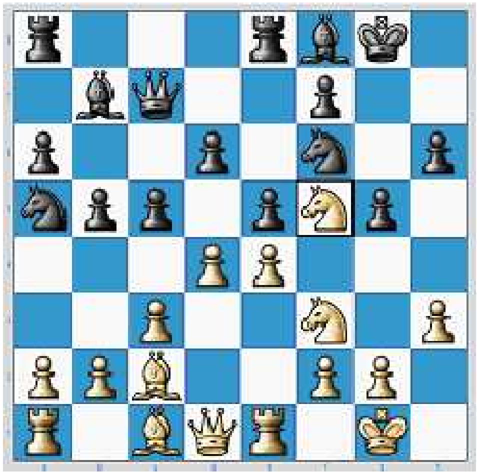


Figure 26: Protected white King and unprotected black King.

6.6.5 End stages

The end stage threshold condition is the presence of less than 6 minor pieces (Knights and Bishops) in the game.

6.6.6 Technical Tie and Checks

A Technical Tie condition is declared when one of the following conditions is met:

- King against King
- King and Knight against King
- King and Bishop against King
- King and Bishop against King and Bishop
- King and Bishop against King and Knight

- King and 2 Knights against King
- Repetition of the same last 3 moves by both players.

Technical Tie is not possible at presence of any Queen, Rooks or Pawn still in play. The same also applies when more than 2 Knights or Bishops are present in the game. The technical tie check is performed whenever a piece is taken. For a more detailed description of the calculation and scoring procedure of fitness function, see [16].

Before each move is executed, all the rules are checked first. Forbidden moves like exposing the King to check or signaling a check situation to the opponent will be .the defeat is awarded to the blocked player. The Stalemate is also detected (when the only valid moves will expose the King to check, situation in which a defeat is awarded.

6.7 Implementation aspects

The CA Chess player satisfies all the internal rules of Chess, namely the Pawn impuissant move (when a Pawn steps 2 squares in the first move and cross adjacent columns opponent pawns, the Pawn can be taken by the opponent Pawn as if only one square has been moved) and the Pawn promotion (when a Pawn reaches the last row in the opponent side it is promoted to any piece of choice, except the King and Pawn. The CA Chess Player automatically chooses the Queen, which is the piece of choice, except very rare situation, where a different piece could be chosen. King Left (or Right) Castle is a complex move where the King (Queen) Rook and the King exchange positions simultaneously. The program is implemented using Java 2; it is available by request through the authors. In a Pentium IV 1 GHz the average time for a move using the default configuration is 10 seconds.

6.8 Results

Two sets of test is done and presented here. The first is algorithm vs. algorithm. These tests aim to observe the behaviour of different CA settings. In the different configurations a fair comparison in terms of computation time is tried, but due to the characteristics of the algorithm it is difficult to ensure for tests. Besides the differences in the algorithms, its stochastic nature and the uncertainty of the repair function will make the computation different on every run. The second test is algorithm vs. human players; it aims to classify the performance of the CA against different level of human players.

6.8.1 Algorithm vs. Humans

Two 3 players groups, beginner and experienced human players were tested. A total of 90 games are played for each group against the default CA.

Default CA

The default CA has the following parameters:

Population:	100
Generations:	20
Crossover probability:	0.7
Uniform crossover bits %:	20
Mutation probability per bit:	0.04
Depth Level:	4

Beginner

The beginners lost all games to the default CA. A typical result is shown in figure 27.

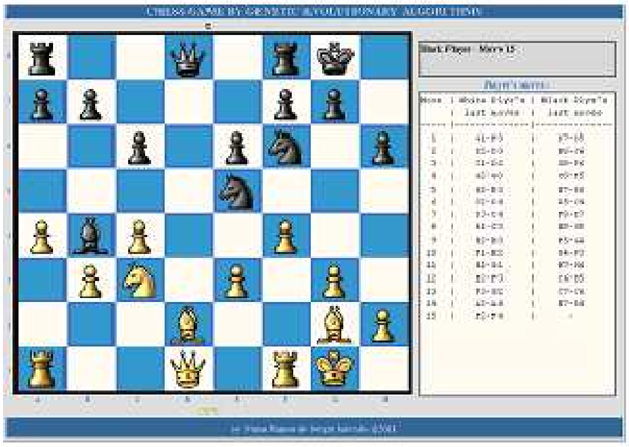


Figure 27: Typical game between beginner and default CA.

Experienced

The result between experienced players vs. default CA is 46 vs. 54%, favourable to default CA. A typical game is shown in figure 28. It can be noted long diagonal chains of Pawns (situation of sequences of protection). The Bishops usually occupies empty diagonals, a situation that increases its influence. Castles are always performed since it is a highly rewarded move. The position of a Bishop at G2 is a very common situation, because it puts strong pressure to opponent positions.

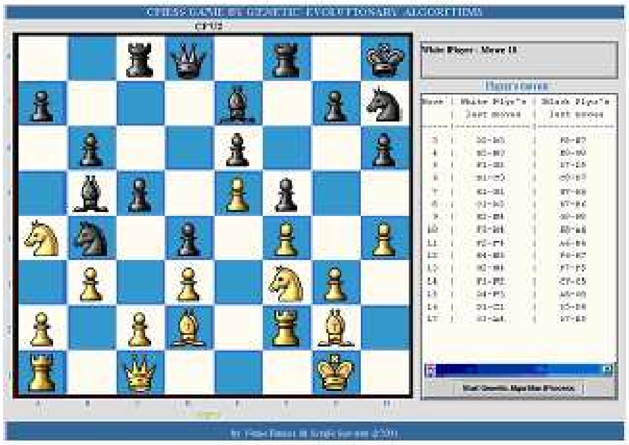

CHESS GAME BY GENETIC EVOLUTIONARY ALGORITHMS
CPU2
White Player : Move 18
White Ply's last moves
Black Ply's last moves
Start Genetic Algorithm Process

Figure 28: Typical game between Experienced Player and default CA.

6.9 Conclusions and discussions

A Co-evolutionary based chess player is implemented and the performance of the default CA player (that depends on the depth level) is comparable to an experienced human player. Since finals situations are well known, they could be incorporated in order to reduce the search space. Although the scoring system used seems to work well, it has room for further improvements. The performance of the CA player worsens in the more advanced stages of the game when the search space is much larger than in the beginning. A dynamic population and generation schedule could improve further the performance. Currently the fitness function of the mixed chromosomes is the sum of all moves; a possibly better approach could be the fitness due to the last move in the chromosome. The final move at the specified depth is the one that matters not the intermediate moves. The danger of this strategy is the assumption that the opponent will always play the response moves coded by the simulated opponent best chromosome that is not always true. A meta level EA could be used to learn the weights and scorings to be used during the games and can be adapted to the opponent plat styles. Adaptation to the international computer chess rules and platforms is under way in order to have a more precise and quantitative characterization of the CA chess Player.

References

[1] M. Mitchell, *Introduction to Genetic Algorithms.* MIT Press, 2000.

[2] T. Beck, *Handbook of Evolutionary Computation.* Addison-Wesley, 2001.

[3] R. Tavares, A. Teófilo, and A. C. Rosa, “Infected gene EA.” ACM SAC 99.

[4] A. C. Rosa, L. Pereira, and L. Bento, “Mastermind by evolutionary algorithms.” ACM SAC 99.

[5] W. H. Tang and A. Moura, “Using genetic algorithms in the game five-in-line,” in *Proc. of GAAL 98*, pp. 26–28, 1998. http://laseeb.ist.utl.pt/workshops/agva98/agva98.html.

[6] “Co-evolutionary solution of chess game.”

[7] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, 1989.

[8] S. W. Golomb, *Polyominoes.* New York: Scribner, 1965.

[9] D. Viaud, “Une formulisation du jeu de mastermind,” *RAIRO-Recherche Operationelle*, vol. 13, no. 3, pp. 307–321, 1979.

[10] E. Newirth, “Some strategies for mastermind,” *Zeitschrift fur Operations Research B*, vol. 26, no. 8, pp. 257–278, 1982.

[11] J. Merelo, “Genetic mastermind, a case of dynamic constrain optimisation,” Tech. Rep. GeNeura Technical Report G-96-1, Univ. Granada, 1996.

[12] D. Fogel, *Blondie24, Playing at the Edge of AI.* Morgan Kaufmann Publishers, 2002.

[13] J. Furznkrauz, “http://www.spkrane.org/spk/abalone/mligb.html,” 2003.

[14] C. Moreton, “The rival chess engine.” http://www.chrismo.com/, 1999.

[15] J. Stanback, “Chess GA experiment.” http://www.geocities.com/SilliconWalley/Bay/253/chess-ga.html, 1996.

[16] N. Ramos and S. Salvado, “Jogos de xadrez por algoritmos genético-evolutivos.” Graduation Thesis Report, DEEC-IST-UTL, 2001. http://www.laseeb.org/ChessGA.

List of Participants

- Adolfo Cartaxo (Instituto de Telecomunicações) adolfo.cartaxo@lx.it.pt
- Agostinho Rosa (Instituto de Sistemas e Robótica) acrosa@isr.ist.utl.pt
- Ana Lemos (ESTGL Inst. Polit. de Leiria) aclemos@estg.ipleiria.pt
- Ana Maria Faustino (FE Univ. Porto) afausti@fe.up.pt
- Ana Vieira (ESTT Inst. Polit. de Tomar) avieira@ipt.pt
- Anibal Ferreira (FE Univ. Porto) ajf@fe.up.pt
- Antoni Zabludowski (Inst. Telekom. ATR Poland) antoni.zabludowski@atr.bydgoszcz.pl
- António Almeida (Instituto de Telecomunicações) antonio.almeida@lx.it.pt
- António Caetano (DM Universidade de Aveiro) acaetano@mat.ua.pt
- António Navarro (Instituto de Telecomunicações) navarro@det.ua.pt
- Bárbara Coelho (ESTGL Inst. Polit. de Leiria) barbara@estg.ipleiria.pt
- Carlos Alves (Instituto Superior Técnico) calves@math.ist.utl.pt
- Carlos Belo (Instituto de Telecomunicações) belo@lx.it.pt
- Carlos Fernandes (Instituto de Telecomunicações) carlos.fernandes@lx.it.pt
- Carlos Baptista (ESTT Inst. Polit. de Tomar) carlos.perquilhas@aim.estt.ipt.pt
- Carlos Salema (Instituto de Telecomunicações) carlos.salema@lx.it.pt
- Celeste Gouveia (Instituto de Telecomunicações) mcag@mat.uc.pt
- Cristina Pimentel (Inst. Polit. de Beja)
- Eckart Zitzler (TIK Swiss Federal Instutite of Technology, Zurich) zitzler@tik.ee.ethz.ch
- Enrique Zuazua (DM Universidade Autonoma, Madrid) enrique.zuazua@uam.es
- Fernando Perdigão (Instituto de Telecomunicações) fp@co.it.pt
- Francisco Nunes (ESTT Inst. Polit. de Tomar) fnunes@ipt.pt
- Henrique Silva (Instituto de Telecomunicações) henrique.silva@co.it.pt
- Isabel Cristina Ribeiro (FE Univ. Porto) iribeiro@fe.up.pt
- Isabel Narra de Figueiredo (DM Universidade de Coimbra) isabel.figueiredo@mat.uc.pt
- Ivette Gomes (DEIO Universidade de Lisboa) ivette.gomes@fc.ul.pt
- Joana Rita Silva Fialho jotafialho@hotmail.com
- Joana Soares (DM Universidade do Minho) jsoares@math.uminho.pt
- João Patrício (ESTT Inst. Polit. de Tomar) Joao.Patricio@aim.estt.ipt.pt
- Joaquim João Júdice (Instituto de Telecomunicações) joaquim.judice@co.it.pt
- José Carvalho (DM Universidade de Aveiro) abel@mat.ua.pt

- José Craveirinha (DEEC Universidade de Coimbra) jcrav@deec.uc.pt
- José Luis Santos (DM Universidade de Coimbra) zeluis@mat.uc.pt
- José Morgado (Instituto de Telecomunicações) j.morgado@lx.it.pt
- Lígia Rodrigues (ESTT Inst. Polit. de Tomar) Ligia.Rodrigues@aim.estt.ipt.pt
- Luis Grilo (ESTT Inst. Polit. de Tomar) lgrilo@ipt.pt
- Luis Merca Fernandes (ESTT Inst. Polit. de Tomar) lmerca@co.it.pt
- Luis Vieira Sá (Instituto de Telecomunicações) luis.sa@co.it.pt
- Manuel Barros (ESTT Inst. Polit. de Tomar) fmbarros@ipt.pt
- Manuela Fernandes (ESTT Inst. Polit. de Tomar) manuela.fernandes@aim.estt.ipt.pt
- Manuela Simões (ESTG Inst. Polit. da Guarda) msimoes@ipg.pt
- Maria Cristina Costa (ESTT Instituto Politécnico de Tomar) ccosta@ipt.pt
- Maria do Carmo Brás (DM Universidade Nova de Lisboa) mb@fct.unl.pt
- Maria do Carmo Miranda Guedes (DMA Universidade do Porto) mmguedes@fc.up.pt
- Mário Figueiredo (Instituto de Telecomunicações) mtf@lx.it.pt
- Mário Silveirinha (Instituto de Telecomunicações) mario.silveirinha@co.it.pt
- Marta Pereira (Inst. Polit. de Beja)
- Marta Sofia Ferreira Umbelino Marta-Umbelino@iol.pt
- Maurício Resende (ATT, USA) mgcr@research.att.com
- Nuno Bastos (ESTV Inst. Polit. de Viseu) nbasto@mat.estv.ipv.pt
- Nuno Rodrigues (DEEC Universidade de Coimbra) nuno.rodrigues@co.it.pt
- Odete Ribeiro (ESTV Inst. Polit. de Viseu) odetecr@mat.estv.ipv.pt
- Paola Festa (DM Applicazioni, Itália)
- Paula Cristina Sarabando dos Santos paula-sarabando@hotmail.com
- Paulo Paiva Monteiro (DEEC Universidade de Coimbra) pm@co.it.pt
- Pedro Correia (ESTT Inst. Polit. de Tomar) pcorreia@ipt.pt
- Pedro Martins (ISCAC Inst. Polit. de Coimbra) pmartins@iscac.pt
- Pedro Simões Patrício (DM Universidade da Beira Interior)
- Pedro Carrasqueira (ESTT Inst. Polit. de Tomar) pedro.carrasqueira@aim.estt.ipt.pt
- Pedro Oliveira (DPS Universidade do Minho) pno@dps.uminho.pt
- Rui Valadas (Instituto de Telecomunicações) rv@av.it.pt
- Silvério Rosa (Universidade da Beira Interior) rosa@noe.ubi.pt
- Teresa Godinho (Inst. Polit. de Beja)
- Victor Anunciada (Instituto de Telecomunicações) avaa@lx.it.pt
- Vitor Bastos Fernandes (Universidade da Minho) vbastos@uminho.pt
- Vitor Silva (Instituto de Telecomunicações) vitor@co.it.pt

Contents