CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITECNICO NACIONAL

Departamento de Matemáticas

Análisis de Heurísticas de Optimización para Problemas Multiobjetivo

Tesis que presenta

Mario Alberto Villalobos Arias

Para obtener el grado de

Doctor en Ciencias

en la especialidad de

Matemáticas

Directores de Tesis: **Dr. Onésimo Hernández Lerma**
Dr. Carlos A. Coello Coello

México, D.F. agosto de 2005

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITECNICO NACIONAL

Departamento de Matemáticas

Análisis de Heurísticas de Optimización para Problemas Multiobjetivo

Tesis que presenta

Mario Alberto Villalobos Arias

Para obtener el grado de

Doctor en Ciencias

en la especialidad de

Matemáticas

Directores de Tesis: **Dr. Onésimo Hernández Lerma**
Dr. Carlos A. Coello Coello

México, D.F. agosto de 2005

A Marcela, Maricruz y Josué.

A mis Papás.

A mis Abuelos.

A mis Suegros.

Agradecimientos

Quiero agradecer al pueblo Mexicano que a través de su Secretaría de Relaciones Exteriores me otorgó una beca para financiar, en parte, mis estudios.

Agradezco:

A mis asesores:

Dr. Onésimo Hernández Lerma y Dr. Carlos A. Coello Coello

por su apoyo y el tiempo que me dedicaron para poder realizar esta tesis.

A los sinodales por el tiempo dedicado en revisar mi tesis y los consejos que me dieron.

A toda mi familia por el apoyo que me brindaron.

A todos los costarricenes (ticos) y mexicanos que han hecho que el tiempo que he estado en México sea más agradable.

A los compañeros de matemáticas por la ayuda y apoyo que me dieron.

A los compañeros de computación por la ayuda y apoyo que me dieron, especialmente a Gregorio y Nareli.

A Hilda, Laura, Roxana, Anabel, Norma, Sofi y Flor por la ayuda que me brindaron.

Agradezco el apoyo de las siguientes instituciones:

A la Universidad de Costa Rica por la beca complementaria que me otorgó para financiar mis estudios doctorales.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por la beca terminal que se me otorgó a través del Proyecto # 34201-A a cargo del Dr. Carlos A. Coello Coello.

Al Departamento de Matemáticas del CINVESTAV-IPN, y a este mismo, por todo el apoyo y las facilidades que me dieron.

Contents

List of Figures

List of Tables

Abbreviations

MOP Multiobjective optimization problem.

AIS Artificial immune systems.

EA Evolutionary algorithm.

SA Simulated annealing.

SAA Simulated annealing algorithm.

MhA Metaheuristic algorithms.

MISA Multi-objective immune system algorithm.

PSO Particle swarm optimizers

MOPSO Multi-objective particle swarm optimizers.

ST–MOPSO MOPSO with stripes.

TSC Two set coverage.

HV Hypervolume.

IGD Inverted generational distance.

SC Success counting.

POP Portfolio optimization problem .

BMV Mexican Stock Market (Bolsa Mexicana de Valores).

IPyC Index of Prices and Quotations (Indice de Precios y Cotizaciones).

Prefacio

En esta tesis se estudia la convergencia de varios tipos de heurísticas de optimización para problemas multiobjetivo (MOPs).

Las heurísticas de optimización son algoritmos computacionales basados en la simulación de ciertos procesos físicos o biológicos, como son la evolución de las especies, funcionamiento del sistema inmune del cuerpo humano, comportamientos sociales de ciertos animales, recocido de sólidos para formar cristales (annealing), etc. Estos procesos en sí mismos, son procesos que optimizan, por lo que algunos investigadores los han simulado por medio de computadoras y los algoritmos que resultan se aplican a problemas de optimización.

Algunos de estos algoritmos y referencias relacionadas son:

- recocido simulado [23, 39],
- algoritmos genéticos [17],
- estrategias evolutivas [45],
- programación evolutiva [15, 14],
- sistema inmune artificial [6, 37],
- optimización por enjambre de partículas [22, 12].

Algunos de estos problemas resultan muy complicados para los "métodos tradicionales" de optimización y en estos casos las heurísticas resultan de gran utilidad.

Por otro lado, existen situaciones en las que queremos obtener el máximo beneficio o rendimiento, pero al mismo tiempo deseamos minimizar los costos o el tiempo de realizar una cierta tarea. Usualmente, el mejorar el rendimiento conlleva que los costos o el tiempo sean mayores, por lo que los objetivos están en conflicto. A estos problemas en los que hay 2 o más objetivos es lo que llamamos un problema multiobjetivo (PMO).

En las aplicaciones, las heurísticas de optimización suelen dar buenos resultados, pero no había certeza en la convergencia de los algoritmos. Este es precisamente el propósito principal de este trabajo: demostrar la convergencia, en un sentido adecuado, de algunas de las heurísticas de optimización para PMO.

Una característica de los PMO es que, como regla general, tienen un conjunto de soluciones, que inclusive puede ser infinito, y entonces los algoritmos deben describir lo mejor posible este conjunto. En este trabajo, además de la convergencia de las heurísticas, proponemos algunas modificaciones que permiten obtener mejores representaciones de los conjuntos solución.

Para ilustrar nuestros resultados, consideramos el problema de selección de portafolio (de inversión) de Markowitz. En este problema se desea encontrar un portafolio que maximiza el rendimiento pero con riesgo mínimo.

Este trabajo está organizado como sigue: en el Capítulo 1 presentamos el problema de optimización multiobjetivo. En el Capítulo 2 se introduce el algoritmo de recocido simulado y la prueba de su convergencia, para el caso multiobjetivo. Después la convergencia de una heurística de optimización general, de nuevo para el caso multiobjetivo, se presenta en el Capítulo 3. En el Capítulo 4 presentamos el algoritmo del sistema inmune artificial y la correspondiente demostración de convergencia. Un nuevo esquema para mantener diversidad, basado en franjas, se presenta en el Capítulo 5. Finalmente, en el Capítulo 6 presentamos una aplicación de nuestros resultados al problema de selección de portafolio de Markowitz.

Preface

This thesis concerns the convergence of several heuristic algorithms for multiobjective optimization problems.

A heuristic optimization algorithm is a computational algorithm that tries to imitate some physical or biological processes such as the evolution of the species, the human immune system, the social behavior of some animal groups (for instance, bees and ants), and so on. These processes are themselves optimization processes and so they naturally suggest computational algorithms applicable to mathematical optimization problems.

Some of these algorithms and related references are

- simulated annealing [23, 39],
- genetic algorithms [17],
- evolution strategies [45],
- evolutionary programming [15, 14],
- artificial immune system algorithm [6, 37],
- particle swarm optimization [22, 12].

These heuristic techniques are extremely useful, in particular for optimization problems for which the traditional methods are difficult to apply.

As we already noted, our work deals with multiobjective optimization problems (MOPs). A typical example is when we try to maximize a certain utility or revenue function, but simultaneously we wish to minimize, say, an operation cost. Thus in a MOP we can have objective functions representing conflicting interests. In many of these situations, the heuristic algorithms usually perform quite well, but there were no mathematical results ensuring the convergence of the algorithms. Here is where the main contribution of our work comes in: we give conditions for the convergence, in a suitable sense, of some of the most common heuristic algorithms for MOPs.

A feature of MOP is that, as a rule, the solution set can be quite large, possibly infinite, and of course we would like our algorithms to describe as well as possible this set. Here we propose some modified algorithms that allow us to obtain very good representations of the solutions sets.

Finally, to illustrate our approach, we consider the Markowitz portfolio selection problem. This is an important problem in which one wishes to find an investment portfolio that maximizes the expected return, with minimum risk.

This work is organized as follows: in Chapter 1 we present the multiobjective optimization problem. The simulated annealing algorithm and the proof of its convergence, for the multiobjective case, are in Chapter 2. The convergence of a general heuristic optimization, again for the multiobjective case, appears in Chapter 3. Chapter 4 contains the algorithm of the artificial immune system and its convergence. A new scheme to maintain diversity, based on stripes, is presented in Chapter 5. Finally, in Chapter 6 we apply our results to the Markowitz portfolio selection problem.

Resumen

Esta tesis presenta un análisis de la convergencia de varios algoritmos heurísticos de optimización para problemas multiobjetivo. Los algoritmos que consideramos incluyen el recocido simulado, algunos algoritmos evolutivos y el sistema inmune artificial. En el caso de los algoritmos evolutivos, nos referimos a cualquier algoritmo en el que las probabilidades de transición utilizan una regla de mutación uniforme. Demostramos que estos algoritmos convergen si se utiliza elitismo.

Presentamos, además, un esquema para mantener la diversidad en este tipo de metaheurísticas. Este esquema se incorporó a un algoritmo de optimización de enjambre de partículas, dando lugar a un nuevo algoritmo evolutivo multiobjetivo. Finalmente, este nuevo algoritmo se aplicó al problema de selección de portafolio de Markowitz.

Abstract

This thesis presents the asymptotic convergence analysis of several heuristic algorithms for multiobjective optimization problems. The algorithms we consider include simulated annealing, some evolutionary algorithms and artificial immune system. In the case of evolutionary algorithms, we refer to any algorithm in which the transition probabilities use a uniform mutation rule. We prove that these algorithms converge if elitism is used.

In addition, we introduce a scheme to maintain diversity in this type of metaheuristics. This scheme is incorporated into a particle swarm optimization algorithm, giving rise to a new multiobjective evolutionary algorithm. Finally, this new algorithm is applied to the Markowitz problem of portfolio selection.

Chapter 1

Introduction

1.1 Motivation

In real-world, there are many problems with several objectives that we aim to optimize simultaneously. These problems are called "multiobjective" or "vector" optimization problems, and have been studied by many authors who have proposed a number of solution techniques [2, 7, 16, 32, 51].

The solution of a multiobjective optimization problem requires a suitable definition of "optimality" (usually called "Pareto optimality"). Such problems normally have not one, but an infinite set of solutions, which represent possible trade-offs among the objectives (such solutions constitute the so-called "Pareto optimal set", defined in Section 1.2).

In these multiobjective optimization problems (MOPs) one wishes to optimize a vector function, say $F(x) = (f_1(x), ..., f_n(x))$. A typical way to approach these problems is to transform the MOPs into single-objective (or "scalar") problems (e.g., by using a linear aggregating function). This approach indeed makes sense if the functions $f_1, ..., f_n$ are of the same type and expressed in the same units, but otherwise (for instance, if f_1 denotes distance, f_2 denotes time, and so on) the scalarized problem might be meaningless.

Diverse metaheuristics have been adopted to solve MOP [2]–[6], [9, 20]. In this thesis, we study three of them: simulated annealing (SA) [23, 39], artificial immune systems (AIS) [37] and general evolutionary algorithms (EA) [17, 13].

Metaheuristics such as those indicated above, have become a standard tool to solve both single-objective and multiobjective optimization problems. In the single-objective case, the convergence of a metaheuristic is reasonably well-understood [1, 40], under suitable simplifications.

For the multiobjective case there are also some convergence proofs [41, 42], but they are not quite rigorous. Most heuristics used for multiobjective optimization do not have a convergence proof reported in the literature. This thesis intends to bridge this gap for a class of algorithms.

For these metaheuristics that use a uniform mutation rule (see Section 3.3) we show that the associated Markov chain converges geometrically to its stationary distribution, but not necessarily to the optimal solution set of the multiobjective optimization problem. Convergence to the optimal solution set is ensured if elitism (whose definition is provided in section 3.2.2, page 17) is used.

However, when dealing with multiobjective optimization problems, there is not much work available in the literature, except for extremely particular cases (see for example [41, 42]).

Maintaining diversity in a population has been a problem that has attracted the attention from many researchers since the origins of evolutionary computation. Due to stochastic noise, evolutionary algorithms tend to converge to a single solution if run during a sufficiently long time. Thus, the problem of diversity in the context of multiobjective optimization basically focuses on blocking the selection mechanism of an evolutionary algorithm as to avoid this sort of convergence to a single solution. Instead, some sort of bias must be introduced in the selection mechanism as to allow the generation and maintenance of different nondominated solutions in the population of an evolutionary algorithm.

When dealing with MOPs, all metaheuristics are required not only to converge as closely to the true Pareto front as possible, but also to cover all the Pareto front (see Section 1.2) with well-distributed points. It is obviously possible that the solutions produced by a metaheuristic only cover a portion of the Pareto front, and this is an undesirable behavior. In this thesis, we present a proposal to efficiently solve this problem.

1.2 The Multiobjective Optimization Problem

Let X be a set and $F : X \longrightarrow I\!R^d$ a given vector function with components $f_i : X \longrightarrow I\!R$ for each $i \in \{1, \ldots, d\}$. The multiobjective optimization problem (MOP) we are concerned with is to find $x^* \in X$ such that

$$F(x^*) = \min_{x \in X} F(x) = \min_{x \in X}[f_1(x), \ldots, f_d(x)], \tag{1.1}$$

where the minimum is understood in the sense of the standard Pareto order in which two vectors in $I\!R^d$ are compared as follows.

If $\vec{u} = (u_1, \ldots, u_d)$ and $\vec{v} = (v_1, \ldots, v_d)$ are vectors in $I\!R^d$, then

$$\vec{u} \preceq \vec{v} \iff u_i \leq v_i \ \forall\, i \in \{1, \ldots, d\}.$$

This relation is a partial order. We also write $\vec{u} \prec \vec{v}$ if $\vec{u} \preceq \vec{v}$ and $\vec{u} \neq \vec{v}$. In this case we say that u *dominates* v. By example in Figure 1.1 point B dominates point E.

Definition 1.1 *A point $x^* \in X$ is called a* Pareto optimal solution *for the MOP (1.1) if there is no $x \in X$ such that $F(x) \prec F(x^*)$. The set*

$$\mathcal{P}^* = \{x \in X \ : \ x \textit{ is a Pareto optimal solution}\}$$

is called the Pareto optimal set *for the MOP (1.1), and its image under F, i.e.*

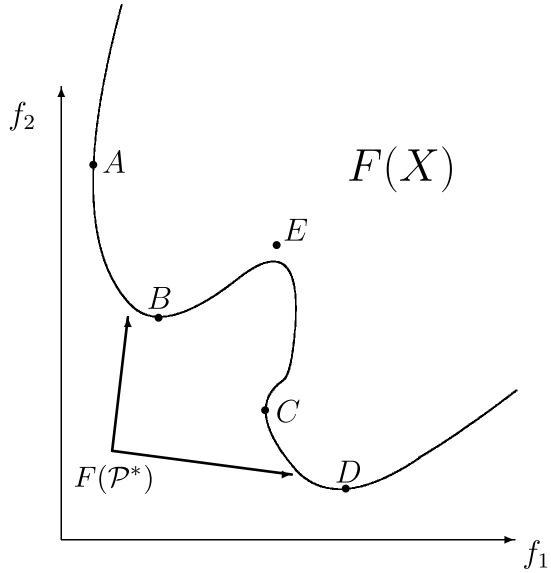


Figure 1.1: Example of a Pareto front for two objective case.

$$F(\mathcal{P}^*) \ := \ \{F(x) \ : \ x \in \mathcal{P}^*\},$$

is called the Pareto front.

In Figure 1.1 the Pareto front corresponds to the parts on the boundary of $F(X)$ joining the points A and B, and also the points C and D.

Here we say that x *dominates* y when $F(x) \prec F(y)$. Let $Y \subseteq X$ and $y \in Y$. If there is no $x \in Y$, that dominates y , we say that y is *nondominated* (with respect to Y). Observe that all the elements in the Pareto front are nondominated with respect to X.

As we are concerned with computational aspects, in the remainder of this thesis we will assume that the set X in (1.1) is *finite*. For an EA and the AIS, in which the elements are represented by strings of length l with 0 or 1 at each entry, we take $X = \mathbb{B}^l$, with $\mathbb{B} = \{0, 1\}$. For SA we only assume that X is finite.

Moreove, the algorithms we are concerned with will evolve as a Markov chains defined on an underlying probability space, say $(\Omega, \mathcal{F}, \mathbb{P})$.

Chapter 2

Simulated Annealing Algorithm

In this chapter we consider a simulated annealing algorithm for multiobjective optimization problems. With a suitable choice of the acceptance probabilities, the algorithm is shown to converge, that is, the Markov chain that describes the algorithm converges with probability one to the Pareto optimal set.

2.1 Introduction

Here, we consider a simulated annealing algorithm (SAA) for solving multiobjective optimization problems (MOPs). Under mild assumptions and a suitable choice of the acceptance probabilities, our SAA is shown to converge with probability one to the Pareto optimal set of the problem.

The remainder of this chapter is organized as follows. In Section 2.2 we introduce the SAA we are concerned with; we also briefly discuss the algorithm's acceptance probabilities, which are crucial for proving convergence. Our main result is stated in Section 2.3. Finally, our conclusions and future work are provided in Section 2.4 together with some general remarks.

As we are concerned with computational aspects, in the remainder of the chapter we will replace the set X in (1.1) with a *finite* set $S \subset I\!R^m$.

2.2 The Simulated Annealing Algorithm

Nicholas Metropolis et al. [31] originally proposed, in 1953, an algorithm to simulate the evolution of a solid in a heat bath until it reached its thermal equilibrium. The process started from a certain thermodynamic state of the system, defined by a certain energy and temperature. Then the state was slightly perturbed. If the change in energy produced by this perturbation was negative, the new configuration was accepted. If it was positive, it was accepted with a certain probability. This process was repeated until a frozen state was achieved [11, 44].

About thirty years after the publication of Metropolis' approach, Kirkpatrick et al. [23] and Černy [39] independently pointed out the analogy between this "annealing" process and combinatorial optimization. Such analogy led to the development of an algorithm called "Simulated Annealing" which is a heuristic search technique that has been quite successful in combinatorial optimization problems (see [1] and [25] for details).

The SAA generates a succession of possible solutions of the optimization problem. These possible solutions are the states of a Markov chain and the "energy" of a state is the evaluation of the possible solution that it represents.

The temperature is simulated with a sequence of positive control parameters c_k.

A transition of the Markov chain occurs in two steps, given the value c_k of the control parameter. First, if the current state is i, a new state j is generated with a certain probability $G_{ij}(c_k)$, defined below. Then an "acceptance rule" $A_{ij}(c_k)$ is applied to j. Our main result hinges on a suitable selection of the acceptance rule, which we now discuss.

2.2.1 The Generation Probability

For each state i, let S_i be a subset of $S \setminus \{i\}$ called a neighborhood of i. We shall assume that the number of elements in S_i is the same, say Θ, for all $i \in S$, and also that the neighbor relation is symmetric, that is, $j \in S_i$ if and only if $i \in S_j$. Then, denoting by χ_{S_i} the indicator function of S_i (i.e. $\chi_{S_i}(j) := 1$ if $j \in S_i$ and 0 otherwise), we define the generation probability

$$G_{ij}(c_k) := \frac{\chi_{S_i}(j)}{\Theta} \quad \text{for all } i, j \in S. \tag{2.1}$$

2.2.2 The Acceptance Probability

The acceptance probability is crucial for the behavior of the SAA.

The idea of this probability or acceptance rule is that any new state that improves the actual state will be accepted with probability 1, whereas the others are accepted with certain probability that tends to zero as time goes to infinity.

When dealing with MOPs there are different options to define the acceptance rule. For instance, Serafini [47] proposes to use the L_∞–Tchebycheff norm given by

$$A'_{ij}(c) = \min\left\{1, \exp\left(\max_{s\in\{1,\ldots,d\}} \frac{\lambda_s(f_s(i) - f_s(j))}{c}\right)\right\},$$

where the λ_s are given positive parameters, and $c > 0$ is the control parameter, that simulates the temperature.

This acceptance probability has a possible drawback that if a single entry is improved (i.e. $f_s(i) > f_s(j)$ for some s) or has the same value, then the state j is accepted, which obviously is not very good. For example, in Figure 2.1, in which $f_1(j) = f_1(i)$, we have $A_{ij} = 1$ although $f_2(j)$ is too "bad" in comparison with $f_2(i)$.

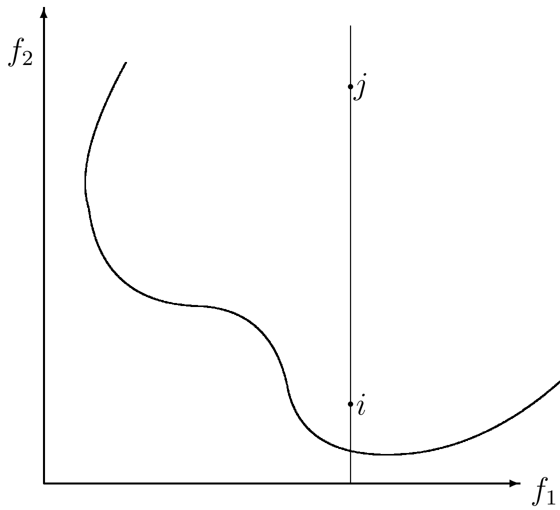


Figure 2.1: Graphical illustration of the "inconvenience" of the acceptance probability $A'_{ij}(c)$ proposed by Serafini [47].

On the other hand, Ulungu and coworkers [51, 52, 54, 53] use

$$\begin{aligned} A''_{ij}(c) &:= \min\left\{1, \exp\left(\sum_{s=1}^{d} \frac{\lambda_s(f_s(i) - f_s(j))}{c}\right)\right\} \\ &= \exp\left\{-\left(\sum_{s=1}^{d} \frac{\lambda_s(f_s(j) - f_s(i))}{c}\right)^+\right\}. \end{aligned} \tag{2.2}$$

where as usual, let a^+ be the positive part of a number $a \in I\!R$, namely

$$a^+ := \begin{cases} a & \text{if } a > 0, \\ 0 & \text{otherwise.} \end{cases}$$

But again, in this case it is possible to have the acceptance probability depending on the change of a *single* entry $f_s(i) - f_s(j),\ s = 1, \ldots, d$.

Here we shall use the acceptance probability [47]

$$A_{ij}(c) := \prod_{s=1}^{d} \min\left\{1, \exp\left(\frac{f_s(i) - f_s(j)}{c}\right)\right\},$$

which can be expressed in the simpler form

$$A_{ij}(c) = \exp\left(-\frac{\sum_{s=1}^{d}(f_s(j)-f_s(i))^+}{c}\right). \tag{2.3}$$

This acceptance probability is obviously "better" than the one in (2.2) because only the entries that do not improve are taken into account to calculate the probability; this probability could be improved changing c by an individual c_s for each entry $s = 1,\ldots,d$.

For the last two acceptance rules, we will prove that the SAA converges. (See Theorem 2.1)

2.2.3 The Transition Probability

Having the generation and the acceptance probabilities, we can now define the *transition probability* from i to j as

$$P_{ij}(c_k) := \begin{cases} G_{ij}(c_k)A_{ij}(c_k) & \text{if } i \neq j, \\ 1 - \sum_{l\in S, l\neq i} P_{il}(c_k) & \text{if } i = j \end{cases} \tag{2.4}$$

where A_{ij} is as in (2.3) (or as in (2.2)).

Note that for theoretical purposes we can use $f_s(i) - f_s(j)$ instead of $\lambda_s(f_s(i) - f_s(j))$ or $(f_s(i) - f_s(j))/c_s$, because the last two expressions can be transformed into the first one via the changes $g_s = \lambda_s f_s$ or $g_s = f_s/c_s$, respectively. Hence, at the remainder of this work we will use the first one.

2.3 Main Result

In the proof of the main result of this chapter, we will use the following well–known "scalarization" result.

Lemma 2.1 *If $\vec{x}^* \in X$ is a solution of the weighted problem:*

$$\min_{\vec{x}\in X} \sum_{s=1}^{d} w_s f_s(\vec{x}), \ \textit{where } w_s > 0 \ \forall s \in \{1,\ldots,d\} \ \textit{and} \ \sum_{s=1}^{d} w_s = 1,$$

then $\vec{x}^* \in \mathcal{P}^*$.

We omit the proof of this lemma because it is trivial.

Now we introduce some notation that will be used later on. Let

$$\Sigma_{opt} := \{x \in X \ : \ \textstyle\sum_{s=1}^{d} f_s(x) = \Sigma_m\},$$

where

$$\Sigma_m := \min_{x \in X} \sum_{s=1}^{d} f_s(x). \tag{2.5}$$

Then, by Lemma 2.1, the Pareto optimal set $\mathcal{P}^*$ contains Σ_{opt}, i.e.

$$\Sigma_{opt} \subset \mathcal{P}^*. \tag{2.6}$$

We next present our main result, which in particular states the convergence of the SAA for the MOP (1.1). The convergence, in this chapter, is understood in the following sense.

Definition 2.1 *Let $P(c) = (p_{ij}(c))$ be the transition matrix associated with the SAA defined by (2.1), (2.3), (2.4), and let $\{X_k(c), k = 0, 1, 2, ...\}$ be the corresponding Markov chain, at temperature c. The SAA is said to* converge with probability 1 *if*

$$\lim_{c \searrow 0} \lim_{k \to \infty} I\!P\{X_k(c) \in \mathcal{P}^*\} = 1.$$

The next theorem, which is the main result in this chapter, is an extension to MOPs of the results presented in [1]. Here we use ideas similar to those in that paper, with the appropriate changes.

In the proof of this theorem we show that the algorithm converges to the set $\Sigma_{opt} \subseteq \mathcal{P}^*$, because of the particular transition probability we use.

Theorem 2.1 *Let $P(c)$ be as in Definition 2.1 and, moreover, suppose that $G(c)$ is irreducible. Then:*

(a) The Markov chain has a stationary distribution $\vec{q}(c)$, whose components are given by

$$q_i(c) = \frac{1}{N_0(c)} \exp\left(-\frac{\sum_{s=1}^{d} f_s(i)}{c}\right) \quad \forall\, i \in S, \tag{2.7}$$

where

$$N_0(c) = \sum_{j \in S} \exp\left(-\frac{\sum_{s=1}^{d} f_s(j)}{c}\right) \tag{2.8}$$

(b) For each $i \in S$

$$q_i^* := \lim_{c \searrow 0} q_i(c) = \frac{1}{|\Sigma_{opt}|} \chi_{\Sigma_{opt}}(i),$$

where $|\Sigma_{opt}|$ denotes the number of elements in Σ_{opt}.

(c) The SAA converges with probability 1.

These results remain valid if (2.3) is replaced with (2.2).

Before presenting the proof of Theorem 2.1 we state some preliminary results. First, we note the following fact, which is due to $a^+ = a + (-a)^+$ $(= a + a^-)$.

Lemma 2.2 *For any real numbers* $a_1, a_2, \ldots, a_d,\ b_1, b_2, \ldots, b_d,$

$$\begin{aligned}\sum_{k=1}^{d}(a_k - b_k) + \left(\sum_{k=1}^{d}(b_k - a_k)\right)^+ &= \left(\sum_{k=1}^{d}(a_k - b_k)\right)^+, \\ \sum_{k=1}^{d}(a_k - b_k) + \sum_{k=1}^{d}(b_k - a_k)^+ &= \sum_{k=1}^{d}(a_k - b_k)^+.\end{aligned}$$

■

We will need some properties of the limiting distribution, which we present next. Recall that a probability distribution $\vec{q}$ is called the *limiting distribution* of a Markov chain with transition probability $P = (p_{ij})$ if

$$q_i = \lim_{k\to\infty} \mathbb{P}(X_k = i | X_0 = j) \text{ for all } i, j \in S.$$

If such a limiting distribution $\vec{q}$ exists and $a_i(k) = \mathbb{P}(X_k = i)$, for $i \in S$, denotes the distribution of X_k, then

$$\lim_{k \longrightarrow \infty} a_i(k) = q_i \text{ for all } i \in S.$$

Moreover, $\vec{q}$ is an *invariant* (or *stationary*) distribution of the Markov chain, which means that

$$\vec{q} = \vec{q}\,P; \tag{2.9}$$

that is, $\vec{q}$ is a left eigenvector of P with eigenvalue 1. A converse to this result (which is true for *finite* Markov chains) is given in Lemma 2.4 below.

Observe that (2.9) trivially holds if $\vec{q}$ is a probability distribution satisfying

$$q_i P_{ij} = q_j P_{ji} \quad \forall i, j \in S. \tag{2.10}$$

Equation (2.10) is called the *detailed balance equation*, and (2.9) is called the *global balance equation*.

It is well known that in an irreducible Markov chain all states have the same period. This observation yields the following.

Lemma 2.3 *An irreducible Markov chain with transition matrix* $P = (p_{ij})$ *is aperiodic if there exists* $j \in S$ *such that* $p_{jj} > 0$.

Lemma 2.4 ([27, pag.19]) *Let P be the transition matrix of a finite, irreducible and aperiodic Markov chain. Then the chain has a unique stationary distribution $\vec{q}$, that is $\vec{q}$ is the unique distribution that satisfies (2.9), and, in addition, $\vec{q}$ is the chain's limiting distribution.*

Now we present the proof of the main result of this chapter.

Proof of Theorem 2.1

(a) Since $G(c)$ is irreducible, using Lemma 2.3 it can be seen that the Markov chain associated to the SAA is irreducible and aperiodic (see [1, pag.39]). Hence, by Lemma 2.4 the chain has a unique stationary distribution. We now use (2.1) and (2.4) to see that (2.10) holds for all $i \neq j$. First note that

$$\begin{aligned} q_i(c)P_{ij}(c) &= q_i(c)G_{ij}(c)A_{ij}(c) \\ &= \begin{cases} \frac{1}{\Theta}q_i(c)A_{ij}(c) & \text{if } j \in S_i \\ 0 & \text{if } j \notin S_i. \end{cases} \end{aligned}$$

Similarly,

$$\begin{aligned} q_j(c)P_{ji}(c) &= q_j(c)G_{ji}(c)A_{ji}(c) \\ &= \begin{cases} \frac{1}{\Theta}q_j(c)A_{ji}(c) & \text{if } i \in S_j \\ 0 & \text{if } i \notin S_j. \end{cases} \end{aligned}$$

Thus, since $i \in S_j$ if and only if $j \in S_i$, to obtain (2.10) we only have to prove that

$$q_i(c)A_{ij}(c) = q_j A_{ji}(c).$$

But this follows from (2.3), (2.7) and Lemma 2.2, because

$$\begin{aligned} q_i(c)A_{ij}(c) &= \\ &= \frac{1}{N_0(c)} \exp\left(-\frac{\sum_{s=1}^d f_s(i)}{c}\right) \exp\left(-\frac{\sum_{s=1}^d (f_s(j) - f_s(i))^+}{c}\right) \\ &= \frac{1}{N_0(c)} \exp\left(-\frac{\sum_{s=1}^d f_s(j)}{c}\right) \exp\left(-\frac{\sum_{s=1}^d (f_s(i) - f_s(j)) + \sum_{s=1}^d (f_s(j) - f_s(i))^+}{c}\right) \\ &= \frac{1}{N_0(c)} \exp\left(-\frac{\sum_{s=1}^d f_s(j)}{c}\right) \exp\left(-\frac{\sum_{s=1}^d (f_s(i) - f_s(j))^+}{c}\right) \\ &= q_j(c)A_{ji}(c). \end{aligned}$$

This shows that (2.10) holds, which in turn yields part (a) in Theorem 2.1.

Note that this proof, with obvious changes, remains valid if the acceptance probability is given by (2.2) rather than (2.3).

(b) Note that for each $a \leq 0$

$$\lim_{x \searrow 0} e^{a/x} = \begin{cases} 1 & \text{if } a = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2.11}$$

Now, by (2.5), (2.7) and (2.8)

$$\begin{aligned}
q_i(c) &= \frac{\exp\left(-\frac{\sum_{s=1}^d f_s(i)}{c}\right)}{\sum_{j\in S}\exp\left(-\frac{\sum_{s=1}^d f_s(j)}{c}\right)} \\
&= \frac{\exp\left(\frac{\Sigma_m - \sum_{s=1}^d f_s(i)}{c}\right)}{\sum_{j\in S}\exp\left(\frac{\Sigma_m - \sum_{s=1}^d f_s(j)}{c}\right)} \\
&= \frac{\exp\left(\frac{\Sigma_m - \sum_{s=1}^d f_s(i)}{c}\right)}{\sum_{j\in S}\exp\left(\frac{\Sigma_m - \sum_{s=1}^d f_s(j)}{c}\right)}\left(\chi_{\Sigma_{opt}}(i) + \chi_{S-\Sigma_{opt}}(i)\right) \\
&= \frac{1}{\sum_{j\in S}\exp\left(\frac{\Sigma_m - \sum_{s=1}^d f_s(j)}{c}\right)}\chi_{\Sigma_{opt}}(i) \\
&\quad + \frac{\exp\left(\frac{\Sigma_m - \sum_{s=1}^d f_s(i)}{c}\right)}{\sum_{j\in S}\exp\left(\frac{\Sigma_m - \sum_{s=1}^d f_s(j)}{c}\right)}\chi_{S-\Sigma_{opt}}(i).
\end{aligned}$$

Now let $c \searrow 0$. Then, by (2.11), the second term of the latter sum tends to 0, whereas the denominator of the first term goes to $|\Sigma_{opt}|$. Hence

$$\lim_{c \searrow 0} q_i(c) = \frac{1}{|\Sigma_{opt}|}\chi_{\Sigma_{opt}}(i) \; + \; 0 \; = \; q_i^*,$$

which completes the proof of part (b).

(c) By (b) and Lemma 2.4

$$\lim_{c \searrow 0}\lim_{k\to\infty} I\!P\{X_k = i\} = \lim_{c \searrow 0} q_i(c) = q_i^*,$$

and so by (2.6)

$$\lim_{c\searrow 0}\lim_{k\to\infty} I\!P\{X_k \in \mathcal{P}^*\} \geq \lim_{c\searrow 0}\lim_{k\to\infty} I\!P\{X_k \in \Sigma_{opt}\} = 1. \tag{2.12}$$

Thus

$$\lim_{c\searrow 0}\lim_{k\to\infty} I\!P\{X_k \in \mathcal{P}^*\} = 1,$$

and (c) follows. ∎

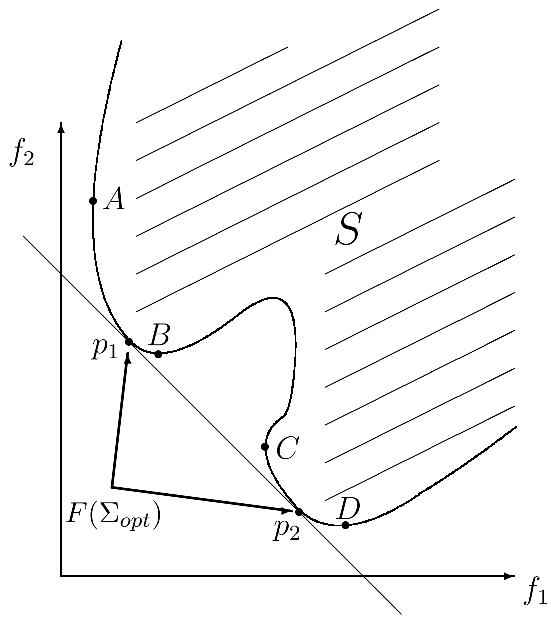


Figure 2.2: Comparison of Σ_{opt} and $\mathcal{P}^*$

2.4 Conclusions and Future Work

We have shown in Theorem 2.1 that a suitable choice of the acceptance probabilities yields the convergence of the SAA. This is reassuring, of course, because it means that the algorithm is indeed heading in the right direction. However, for computational purposes, our approach might not be very useful.

Indeed, what we actually prove is that, as in (2.12), the underlying Markov chain converges to the set Σ_{opt} which can be very "small" compared to the Pareto optimal set $\mathcal{P}^*$.

This is illustrated in Figure 2.2 in which the Pareto front corresponds to the parts on the boundary of S joining the points A and B, and also the points C and D, whereas $F(\Sigma_{opt})$ corresponds only to the points that give p_1 and p_2.

To improve our SAA one possibility would be to introduce an "elite set" (see Section 3.2.2), which is a standard procedure in multiobjective evolutionary algorithms [2, 7]. At each step of the algorithm, the elite set contains all the nondominated points generated so far. Thus, by introducing the elite set, the idea would be to make the contents of such elite set to converge to the Pareto optimal set.

Chapter 3

Metaheuristic Algorithms

In this chapter we analyze the convergence of metaheuristic algorithms for multiobjective optimization problems in which the transition probabilities use a uniform mutation rule. We prove that these algorithms converge only if elitism is used.

3.1 Introduction

This chapter concerns the use of metaheuristic algorithms (MhAs) for solving multiobjective optimization problem (MOPs) as defined in (1.1). For MhAs that use a uniform mutation rule we show that the associated Markov chain converges geometrically to its stationary distribution, but not necessarily to the MOP's optimal solution set. Convergence to the optimal solution set is ensured only if elitism is used.

MhAs are a standard tool to study both single-objective and MOPs. The convergence of a MhA in the single-objective case is reasonably well understood; see [40], for instance. For MOPs, however, the situation is quite different, and as far as we can tell the existing results deal with extremely particular cases; see, for example, [42]. This chapter is, therefore, the first one dealing with the convergence of a general class of MhAs.

The rest of the chapter is organized as follows. The class of MhAs we are interested in are described in Section 3.2, and the main results are presented in Section 3.3. These results are proved in Section 3.4. We conclude in Section 3.5 with some general remarks.

As we are concerned with a MhA in which the elements are represented by strings of length l with 0 or 1 in each entry, in the remainder of the chapter we will replace X in (1.1) with the *finite* set $\mathbb{B}^l$, where $\mathbb{B} = \{0, 1\}$.

3.2 Metaheuristic Algorithms

The MhAs are techniques in which there is a population that evolves applying some operations to the current population to obtain the next one. Some of these operations are

- mutation
- selection
- crossover
- reordering

Some examples of MhAs are

- genetic algorithms (see [17]),
- evolution strategies (see [45]),
- evolutionary programming (see [15, 14]).

The MhAs we are interested in are modeled as Markov chains with transition probabilities that use uniform mutation and possibly other operations. This mutation is made with a parameter or probability p_m, which is positive and less than 1/2, i.e.

$$p_m \in (0, 1/2) . \tag{3.1}$$

In some cases this mutation can be made with two or more parameters, namely the population is divided in subpopulations to each of which a different mutation rate is applied. For example in Chapter 4 we present the MISA algorithm that divides the population in two sets, and in which two different mutation rules are applied.

The algorithm we are concerned with is modeled as a Markov chain $\{X_k : k \geq 0\}$, whose state space S is the set of all possible populations of n individuals, each one represented by a bit string of length l. Hence $S = (\mathbb{B}^l)^n = \mathbb{B}^{nl}$, where $\mathbb{B} = \{0, 1\}$ and so S is the set of all possible vectors of n entries, each of which is a string of length l with 0 or 1 in each entry.

Let $i \in S$ be a state, so that i can be represented as

$$i = (i_1, i_2, \ldots, i_n),$$

where each i_s is a string of length l of 0's and 1's.

The chain's transition probabilities are given by

$$P_{ij} = \mathbb{P}(X_{k+1} = j \mid X_k = i).$$

Thus the transition matrix is of the form

$$P = (P_{ij}) = LM, \tag{3.2}$$

where M is the transition matrix corresponding to the mutation operation and L represents the other operations.

Note that these matrices are stochastic, i.e. $L_{ij} \geq 0$, $M_{ij} \geq 0$ for all i, j, and for each $i \in S$

$$\sum_{s \in S} L_{is} = 1 \quad \text{and} \quad \sum_{s \in S} M_{is} = 1. \tag{3.3}$$

3.2.1 The Mutation Probability

The mutation probability is very important in the convergence analysis of the MhA. To calculate it from state i to state j we use that the individual i_s is transformed into the individual j_s applying *uniform mutation*, i.e. with a fixed probability p_m, then each entry of i_s is transformed into the corresponding one of j_s with probability $1-p_m$ or p_m depending on whether the corresponding entries are equal or different. Schematically we have.

$$
\begin{array}{cc}
 & \begin{array}{cccc} 1 & 2 & \cdots & n \end{array} \\
i & \begin{array}{|c|c|c|c|} \hline i_1 & i_2 & \cdots & i_n \\ \hline \end{array} \\
 & \\
\text{mutation} & \begin{array}{cccc} \downarrow & \downarrow & \cdots & \downarrow \end{array} \\
 & \\
j & \begin{array}{|c|c|c|c|} \hline j_1 & j_2 & \cdots & j_n \\ \hline \end{array}
\end{array}
$$

Thus, for each individual in the population, the mutation probability can be calculated as

$$p_m^{H(i_s,j_s)}(1-p_m)^{l-H(i_s,j_s)} \quad \forall s \in \{1,\ldots,n\},$$

where $H(i_s,j_s)$ is the Hamming distance between i_s and j_s. Hence the mutation probability from i to j is:

$$M_{ij} = \prod_{s=1}^{n} p_m^{H(i_s,j_s)}(1-p_m)^{l-H(i_s,j_s)}. \tag{3.4}$$

3.2.2 Using Elitism

In our case, when dealing with MOPs, we say that we are using *elitism* in an algorithm if we use an extra set, called the *elite* set, in which we put the "best" elements (nondominated elements in our case) found. This elite set usually does not participate in the evolution (although, there are multi-objective evolutionary algorithms that use the elite set in the selection process, such as the Strength Pareto Evolutionary Algorithm [59]), since it is used only to store the nondominated elements.

After each transition we apply an *elitism operation* that accepts a new state if there is an element in the population that improves some element in the elite set (i.e., if there is an element in the population that dominates, in the Pareto sense, some element in the elite set). Additionally, all the elements in the elite set that are dominated by the new element are taken off of the elite set.

If we are using elitism the representation of the states changes to the following form:

$$\hat{i} = (i^e; i) = (i_1^e, \cdots, i_r^e; i_1, \cdots, i_n),$$

where $i_1^e, \cdots, i_r^e$ are the members of the elite set of the state, r is the number of elements in the elite set. Of course, we assume that the cardinality of $\mathcal{P}^*$ is greater than or equal to r, and also that $r \leq n$.

Note that in general $i_1^e, \cdots, i_r^e$ are not necessarily the "best" elements of the state $\hat{i}$, but after applying the elitism operation in i^e they are indeed the "best" elements.

Let $\hat{P}$ be the transition matrix associated with the new states. If all the elements in the elite set of a state are Pareto optimal, then any state that contains an element in the elite set that is not a Pareto optimal will not be accepted, i.e.

$$\text{if } \{i_1^e, \cdots, i_r^e\} \subset \mathcal{P}^* \text{ and } \{j_1^e, \cdots, j_r^e\} \not\subset \mathcal{P}^* \text{ then } \hat{P}_{ij} = 0. \tag{3.5}$$

3.3 Main Results

Before stating our main results we introduce the definition of convergence of an algorithm, which uses the following notation: if $V = (v_1, v_2, \ldots, v_n)$ is a vector, then $\{V\}$ denotes the set of entries of V, i.e.

$$\{V\} = \{v_1, v_2, \ldots, v_n\}.$$

Definition 3.1 *Let $\{X_k : k \geq 0\}$ be the Markov chain associated to an algorithm. We say that the algorithm converges to $\mathcal{P}^*$ with probability 1 if*

$$I\!P(\{X_k\} \subset \mathcal{P}^*) \to 1 \quad as \quad k \to \infty.$$

In the case that we are using elitism we replace X_k by X_k^e, the elite set of the state (i.e. if $X_k = i$ then $X_k^e = i^e$).

Our first result is related to the existence of a stationary distribution for the Markov chain of the MhA.

Theorem 3.1 *Let P be the transition matrix of a MhA. Then P has a stationary distribution π such that*

$$|P_{ij}^k - \pi_j| \leq \left(1 - 2^{nl} p_m^{nl}\right)^{k-1} \quad \forall i, j \in S \quad \forall k = 1, 2, \ldots. \tag{3.6}$$

Moreover, π has all entries positive.

Theorem 3.1 states that P^k converges geometrically to π. Nevertheless, in spite of this result, the convergence of the MhA to the Pareto optimal set cannot be guaranteed. In fact, from Theorem 3.1 and using the fact that π has all entries positive we will immediately deduce the following.

Corollary 3.1 *The MhA does not converge.*

To ensure convergence of the MhA we need to use elitism.

Theorem 3.2 *The MhA using elitism converges.*

3.4 Proofs

We first recall some standard definitions and results.

The next result gives an upper bound on the rate of convergence of P^k as $k \to \infty$. We will use it to show the existence of the stationary distribution in Theorem 3.1.

Lemma 3.1 *Let N be the cardinality of S, and let P_{ij}^k be the entry ij of P^k. Suppose that there exists an integer $\nu > 0$ and a set $J \subset S$ with $N_1 \geq 1$ elements and such that*

$$\min_{\substack{1 \leq i \leq N \\ j \in J}} P_{ij}^{\nu} = \delta > 0.$$

Then there are numbers $\pi_1, \pi_2, \ldots, \pi_{N_1}$ such that

$$\lim_{k \to \infty} P_{ij}^k = \pi_j \ \forall i = 1, \ldots, N, \ \textit{with} \ \ \pi_j \geq \delta > 0, \ \forall j \in J,$$

and $\pi_1, \pi_2, \ldots, \pi_{N_1}$ form a set of stationary probabilities. Moreover

$$|P_{ij}^k - \pi_j| \leq (1 - N_1 \delta)^{\frac{k}{\nu} - 1} \quad \forall k = 1, 2, \ldots.$$

Proof See, for example, [10, p. 173].

The next lemma will allow us to use Lemma 3.1 to prove Theorem 3.1.

Lemma 3.2 *Let P be the transition matrix of the MhA. Then*

$$\min_{i,j \in S} P_{ij} = p_m^{nl} \ > \ 0. \tag{3.7}$$

Proof

By (3.1) we have

$$p_m < \frac{1}{2} < 1 - p_m.$$

Thus, from (3.4), for all i, j:

$$\begin{aligned} M_{ij} &= \prod_{s=1}^{n} p_m^{H(i_s,j_s)} (1 - p_m)^{l - H(i_s,j_s)} \\ &> \prod_{s=1}^{n} p_m^{H(i_s,j_s)} p_m^{l - H(i_s,j_s)} \quad = \quad \prod_{s=1}^{n} p_m^l \\ &= p_m^{nl}. \end{aligned}$$

On the other hand, from (3.2) and (3.3), for all i, j:

$$\begin{aligned} P_{ij} &= \sum_{s \in S} L_{is} M_{sj} \\ &\geq p_m^{nl} \sum_{s \in S} L_{is} \\ &= p_m^{nl} \; > \; 0, \end{aligned}$$

To verify (3.7), observe that P_{ij} attains the minimum in (3.7) if i has 0 in all entries and j has 1 in all entries. Thus the desired conclusion follows. ■

Proof of Theorem 3.1

Since inequality (3.7) holds for all $j \in S$ we have that $N_1 = N = 2^{nl}$ and $\nu = 1$ in Lemma 3.1. Thus, by Lemma 3.1, P has a stationary distribution π with all entries positive and we get (3.6). ■

Theorem 3.2 is an extension of a result originally presented by Rudolph [40]. However our proof is more general, because it is valid for any algorithm that uses uniform mutation and it is valid for MOPs. Additionally, we do not make any assumptions regarding the existence of a single optimal point, due to the use of essential and inessential states as defined next.

Definition 3.2 *Let X be as in section 1.2. We say that X is* complete *if for each $x \in X \setminus \mathcal{P}^*$ there exists $x^* \in \mathcal{P}^*$ such that $F(x^*) \preceq F(x)$.*

For instance, if X is finite then X is complete.

Definition 3.3 *Let $i, j \in S$ be two arbitrary states, we say that i* leads *to j, and write $i \to j$, if there exists an integer $k \geq 1$ such that $P_{ij}^k > 0$. If i does not lead to j we write $i \not\to j$.*

We call a state i inessential *if there exists a state j such that $i \to j$ but $j \not\to i$. Otherwise the state i is called* essential.

We denote the set of essential states by E and the set of inessential states by I. Clearly,

$$S = E \cup I.$$

We say that the transition matrix P is in *canonical form* if it can be written as

$$P = \begin{pmatrix} P_1 & 0 \\ R & Q \end{pmatrix}.$$

Observe that P can be put in this form by reordering the states, that is, the essential states at the beginning and the inessential states at the end. In this case, P_1 is the matrix associated with the transitions between essential states, R with transitions from inessential to essential states, and Q with transitions between inessential states.

Note also that P^k has Q^k in the position of Q in P, i.e.

$$P^k = \begin{pmatrix} P_1^k & 0 \\ R_k & Q^k \end{pmatrix},$$

where R_k is a matrix that depends on P_1, Q and R.

Now we present some results that will be necessary in the proof of Theorem 3.2.

Lemma 3.3 *Let P be a stochastic matrix, and let Q be the submatrix of P associated with transitions between inessential states. Then, as $k \to \infty$,*

$$Q^k \to 0 \textit{ elementwise geometrically fast.}$$

Proof See, for instance, [46, p.120]. ∎

As a consequence of Lemma 3.3 we have the following.

Corollary 3.2 *For any initial distribution,*

$$I\!P(X_k \in I) \to 0 \quad as \quad k \to \infty.$$

Proof For any initial distribution vector p_0, let $p_0(I)$ be the subvector that corresponds to the inessential states. Then, by Lemma 3.3,

$$I\!P(X_k \in I) = p_0(I)' Q^k \mathbf{1} \to 0 \text{ as } k \to \infty. \qquad \blacksquare$$

Proof of Theorem 3.2

By Corollary 3.2, it suffices to show that the states that contain elements in the elite set that are not Pareto optimal are inessential states. To this end, first note that $X = I\!B^l$ is complete, because it is finite.

Now suppose that there is a state $\hat{i} = (i^e; i)$ in which the elite set contains elements $i^e_{s_1}, \ldots, i^e_{s_k}$ that are not Pareto optimal. Then, as X is complete, there are elements, say $j^e_{s_1}, \ldots, j^e_{s_k} \in \mathcal{P}^*$, that dominate $i^e_{s_1}, \ldots, i^e_{s_k}$, respectively.

Take $\hat{j} = (j^e; j)$ such that all Pareto optimal points of i^e are in j^e and replace the other elements of i^e with the corresponding $j^e_{s_1}, \ldots, j^e_{s_k}$. Thus all the elements in j^e are Pareto optimal.

Now let

$$j = (j_1^e, \ldots, j_r^e, \underbrace{i_{s_1}^e, \ldots, i_{s_1}^e}_{n-r \text{ copies}}).$$

By Lemma 3.2 we have $i \to j$. Hence with positive probability we can pass from (i^e, i) to (i^e, j), and then we apply the elitism operation to pass from (i^e, j) to (j^e, j). This implies that $\hat{i} \to \hat{j}$. On the other hand, using (3.5), $\hat{j} \not\to \hat{i}$ and, therefore, $\hat{i}$ is an inessential state.

Finally, from Corollary 3.2 we have

$$I\!P(\{X_k^e\} \subset \mathcal{P}^*) = I\!P(X_k \in E) = 1 - I\!P(X_k \in I) \to 1 - 0 = 1$$

as $k \to \infty$. This completes the proof of Theorem 3.2. ■

3.5 Conclusions and Future Work

We have presented a general convergence analysis of a MhA for MOPs in which uniform mutation is used. It was proven in Theorem 3.2 that it is necessary to use elitism to ensure that our algorithm converges. This result is of course reassuring, but it is not quite complete in the sense that we have been unable to provide a result such as in (3.6), on the speed of convergence. The latter fact as well as a convergence analysis of a MhA with nonuniform mutation rule, require further research.

Chapter 4

Multiobjective Artificial Immune System Algorithm

This chapter presents the convergence of a multiobjective artificial immune system algorithm (based on clonal selection theory). An specific algorithm, previously reported in the specialized literature, is adopted as a basis for the mathematical model presented herein.

4.1 Introduction

It was until recent years that researchers on optimization problems became aware of the potential of population-based heuristics such as artificial immune systems [18, 2]. The main motivation for using population-based heuristics in solving multiobjective optimization problems (MOPs) is because such a population makes possible to deal simultaneously with a set of possible solutions (the so-called population), which allows us to find several members of the Pareto optimal set in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of traditional mathematical programming techniques [32]. Additionally, population-based heuristics are less susceptible to the shape or continuity of the Pareto front (e.g., they can easily deal with discontinuous and concave Pareto fronts), whereas these two issues are a real concern for mathematical programming techniques [7, 2].

Despite the considerable amount of research related to artificial immune systems in the last few years [6, 37], there is still little work related to issues as important as mathematical modelling (see for example [49, 43]). Other aspects, such as convergence, have been practically disregarded in the current specialized literature.

This chapter studies the convergence of an artificial immune system algorithm used for multiobjective optimization problems.

First, we present a simplified form of the algorithm for which the convergence proof is somewhat similar to the one in the previous chapter. Afterwards we deal with the general model.

The remainder of this chapter is organized as follows. In Section 4.2 we briefly describe the specific algorithm adopted for developing our mathematical model of convergence. Then, in Subection 4.3.1, we present the results for the simplified model. The proofs are presented in Subection 4.3.2. Section 4.4 deals with the general algorithm. Finally, our conclusions and some possible paths for future research are presented in Section 4.5.

As we are concerned with the artificial immune system algorithm in which the elements are represented by a string of length l with 0 or 1 in each entry, in the remainder of the chapter we will replace the set X in (1.1) with the *finite* set $\mathbb{B}^l$, where $\mathbb{B} = \{0, 1\}$.

4.2 The Artificial Immune System Algorithm

The Artificial Immune System (AIS) algorithm is a technique that, as its name indicates, simulates in a computer certain aspects of the human immune system. When an antigen enters our immune system, it is immediately detected and generates a response from the immune system. As a consequence, antibodies are generated by the immune system. Antibodies are molecules that play the main role in the immune response. They are capable of adhering to the antigens in order to neutralize and mark them for elimination by other cells of the immune system. Successful antibodies are cloned and hypermutated. This is called the *clonal selection principle* [36] and has been the basis for developing the algorithm on which we base the work reported in this chapter.

For our mathematical model, we will consider the AIS (based on clonal selection theory [36]) for multiobjective optimization proposed in [5], called a "Multi-objective Immune System Algorithm" (MISA for short). Next, we will focus our discussion only on the aspects that are most relevant for its mathematical modelling. For a detailed discussion on this algorithm, readers should refer to [5].

In MISA the antigens are simulated with a population of strings of 0's and 1's. The population is divided in two parts, a primary set and a secondary set; the primary set contains the "best" individuals (or elements) of the population. The transition of one population to another is made by means of two mutation rules and a reordering operation. First, the elements of the primary set are copied several times, then in each of these copies or **clones** a **fixed** number of bits are mutated, at random. Regarding the secondary set, a uniform mutation with parameter p_m is applied. This parameter is positive and less than 1/2, i.e. $p_m \in (0, 1/2)$. After that, the elements are reordered, moving the "best" individuals to the primary set.

MISA can be modeled by a Markov chain $\{X_k : k \geq 0\}$, with state space $S = \mathbb{B}^{nl}$, where $\mathbb{B} = \{0, 1\}$. Suppose that the primary set has n_1 individuals, so that the secondary set has $n - n_1$ individuals. Let $i \in S$ be a state (population). Then we can express i as

$$i = (i^1, i^2) = (i_1, i_2, \ldots, i_{n_1}, i_{n_1+1}, \ldots, i_n), \tag{4.1}$$

i^1, i^2 represent the primary and the secondary set, respectively, whereas each i_s is a string of length l of 0's and 1's.

Use of Elitism

In [5], the elite set is called *the secondary population*, and again it is used only to store the nondominated elements found along the process.

In our case, if we are using elitism, the representation of the states changes to the following form:

$$\hat{i} = (i^e; i) = (i^e; i^1, i^2) = (i_1^e, \cdots, i_r^e; i_1, \cdots, i_{n_1}, i_{n_1+1}, \cdots, i_n)$$

where $i_1^e, \cdots, i_r^e$ are the members of the elite set of the state; r is the number of elements in the elite set, and of course we assume that the cardinality of $\mathcal{P}^*$ is greater than r. In addition we assume that $r \leq n$.

4.3 MISA: Simplified Model

In this model we do not "mutate" a fixed number of entries of the primary set; instead a uniform mutation with probability p_m is applied to the copies of the primary set, whereas a mutation with parameter ρ_m is applied to the secondary set. These parameters are positive and less than 1/2, i.e.

$$p_m, \rho_m \in (0, 1/2)\,. \tag{4.2}$$

Again, we model this algorithm with a Markov chain $\{X_k : k \geq 0\}$, with state space $S = \mathbb{B}^{nl}$, where $\mathbb{B} = \{0, 1\}$. Hence S is the set of all possible vectors of n individuals each one represented by a string of length l with 0 or 1 in each entry.

In our model we omitted to make clones and the mutation is made directly to the elements of the primary set. We do not use clones because this operation is not important for our current purposes.

As in previous chapters the chain's transition probability is denoted by

$$P_{ij} = \mathbb{P}(X_{k+1} = j \mid X_k = i).$$

We also write

$$P(i, A) = \mathbb{P}(X_{k+1} \in A \mid X_k = i).$$

The transition matrix is of the form

$$P = (P_{ij}) = RM,$$

where R and M are the transition matrices of reordering and mutation, respectively. These matrices are stochastic, of course.

The Mutation Probability

In order to calculate the mutation probability from state i to state j we use that the individual i_s is transformed into the individual j_s by applying uniform mutation with probability p_m or ρ_m depending on whether i_s is part of i^1 or i^2, as in the following scheme.

$$
\begin{array}{cccccccc}
 & 1 & \cdots & n_1 & n_1+1 & \cdots & n \\
i & \multicolumn{1}{|c|}{i_1} & \multicolumn{1}{c|}{\cdots} & \multicolumn{1}{c||}{i_{n_1}} & \multicolumn{1}{c|}{i_{n_1+1}} & \multicolumn{1}{c|}{\cdots} & \multicolumn{1}{c|}{i_n} \\
\text{mutation} & \downarrow & \cdots & \downarrow & \downarrow & \cdots & \downarrow \\
j & \multicolumn{1}{|c|}{j_1} & \multicolumn{1}{c|}{\cdots} & \multicolumn{1}{c||}{j_{n_1}} & \multicolumn{1}{c|}{j_{n_1+1}} & \multicolumn{1}{c|}{\cdots} & \multicolumn{1}{c|}{j_n}
\end{array}
$$

Thus, for each individual in the primary set of the population, the mutation probability is given by

$$p_m^{H(i_s,j_s)}(1-p_m)^{l-H(i_s,j_s)} \quad \forall s \in \{1,\ldots,n_1\},$$

where $H(i_s,j_s)$ is the Hamming distance between i_s and j_s. Similarly, for the secondary set we have

$$\rho_m^{H(i_s,j_s)}(1-\rho_m)^{l-H(i_s,j_s)} \quad \forall s \in \{n_1+1,\ldots,n\}.$$

Hence the mutation probability from i to j is:

$$M_{ij} = \prod_{s=1}^{n_1} p_m^{H(i_s,j_s)}(1-p_m)^{l-H(i_s,j_s)} \prod_{s=n_1+1}^{n} \rho_m^{H(i_s,j_s)}(1-\rho_m)^{l-H(i_s,j_s)}. \qquad (4.3)$$

4.3.1 Main Results

For the simplified model of MISA our results and the corresponding proofs are similar to those in the previous chapters. However, now the population is divided in two subpopulations, and we introduce suitable changes. For instance, as in (3.6), we now obtain the following.

Theorem 4.1 *Let P be the transition matrix of MISA. Then P has a stationary distribution π such that*

$$|P_{ij}^k - \pi_j| \leq \left(1 - 2^{nl} p_m^{n_1 l} \rho_m^{(n-n_1)l}\right)^k \quad \forall\ i,j \in S \ \ \forall\ k = 1,2,\ldots. \qquad (4.4)$$

Moreover, π has all its entries positive.

Hence, as in Corollary 3.1 of Theorem 3.1, we immediately deduce the following fact.

Corollary 4.1 *The MISA does not converge.*

Again, as in Chapter 3 to ensure convergence of the MISA we need to use elitism.

Theorem 4.2 *The elitist version of MISA* does *converge.*

4.3.2 Proofs

We first recall some standard definitions and results.

Definition 4.1 *A nonnegative matrix P is said to be* primitive *if there exists $k > 0$ such that the entries of P^k are all positive.*

Definition 4.2 *A Markov chain $\{X_k : k \geq 0\}$ with transition matrix P is said to satisfy a* minorization condition *if there is a pair (β, μ) consisting of a positive real number β and a probability distribution μ on S, and such that*

$$P(i, A) \geq \beta\mu(A) \quad \forall i \in S, \forall A \subseteq S.$$

The following result gives an upper bound on the convergence rate of a Markov chain that satisfies a minorization condition.

Lemma 4.1 *Consider a Markov chain $\{X_k : k \geq 0\}$ with transition matrix P and suppose that it satisfies a minorization condition (β, μ). Then P has a unique stationary distribution π. Moreover for any initial distribution we have*

$$||P^k - \pi|| \leq (1 - \beta)^k \quad \forall k = 1, 2, \ldots.$$

Proof see for example [19, pp. 56-57]

We will use the next result to show the existence of the stationary distribution in Theorem 4.1.

Lemma 4.2 *Let P be a stochastic primitive matrix. Then, as $k \to \infty$, P^k converges to a stochastic matrix $P^\infty = \mathbf{1}'p^\infty$, where $\mathbf{1}'$ is a column vector of 1's and $p^\infty = p^0 \lim_{k\to\infty} P^k = p^0 P^\infty$ has positive entries and it is unique, independently of the initial distribution p^0.*

Proof [21, p. 123]

The next lemma will allow us to use either Lemma 4.1 or Lemma 4.2.

Lemma 4.3 *Let P be the transition matrix of the MISA. Then*

$$\min_{i,j\in S} P_{ij} = p_m^{n_1 l} \rho_m^{(n-n_1)l} \;>\; 0 \tag{4.5}$$

and therefore P is primitive. Moreover, P satisfies a minorization condition (β, μ) with

$$\beta = 2^{nl} p_m^{n_1 l} \rho_m^{(n-n_1)l}, \quad \mu(A) = \frac{|A|}{2^{nl}} \;\; \forall A \subset S, \tag{4.6}$$

where $|A|$ is the cardinality of A.

Proof

The proof of the first is similar to the proof of Lemma 3.2.

By (4.2) we have

$$p_m < \frac{1}{2} < 1 - p_m, \quad \rho_m < \frac{1}{2} < 1 - \rho_m.$$

Thus, from (4.3),

$$\begin{aligned}
M_{ij} &= \prod_{s=1}^{n_1} p_m^{H(i_s,j_s)}(1-p_m)^{l-H(i_s,j_s)} \prod_{s=n_1+1}^{n} \rho_m^{H(i_s,j_s)}(1-\rho_m)^{l-H(i_s,j_s)} \\
&> \prod_{s=1}^{n_1} p_m^l \prod_{s=n_1+1}^{n} \rho_m^l \\
&= p_m^{n_1 l}\rho_m^{(n-n_1)l}.
\end{aligned}$$

On the other hand, since $P = RM$, where R and M are stochastic matrices,

$$\begin{aligned}
P_{ij} &= \sum_{s\in S} R_{is}M_{sj} \\
&\geq p_m^{n_1 l}\rho_m^{(n-n_1)l}\sum_{s\in S} R_{is} \\
&= p_m^{n_1 l}\rho_m^{(n-n_1)l} \;>\; 0,
\end{aligned}$$

To verify (4.5), it suffices to note that P_{ij} attains the minimum in (4.5) if i has 0 in all entries and j has 1 in all entries.

Now we will show that the pair (β, μ) given by (4.6) is a minorization condition for P. Indeed, from (4.5) we have

$$\begin{aligned}
P(i,A) = \sum_{j\in A} P_{ij} &\geq \sum_{j\in A} p_m^{n_1 l}\rho_m^{(n-n_1)l} \\
&= |A|\; p_m^{n_1 l}\rho_m^{(n-n_1)l} \\
&= \frac{|A|}{2^{nl}}\; 2^{nl}\; p_m^{n_1 l}\rho_m^{(n-n_1)l} \\
&= \beta\; \mu(A)
\end{aligned}$$

and the desired conclusion follows. ■

Now we are ready to present the proof of Theorems 4.1 and 4.2.

Proof of Theorem 4.1

By Lemma 4.3, P is primitive. Thus, by Lemma 4.2, P has a unique stationary distribution π with all entries positive. Finally, using Lemma 4.1 and the minorization in (4.6), we get (4.4). ■

Proof of Theorem 4.2

The proof of Theorem 4.2 is same as the proof of Therorem 3.2 (see Chapter 3). ■

4.4 MISA: General Model

In section 4.3 we presented the convergence of a simplified version of MISA.

Here, we present a proof of a more general version of MISA. Thus, to the clones of the primary set a **fixed** number of bits are mutated, at random, and a uniform mutation is applied to the elements of the secondary set.

The idea is similar to Theorem 3.2, and is presented in the next lemma and in Theorem 4.3.

Lemma 4.4 *If any state in MISA has in its elite set an element that is not Pareto optimal, then this state is inessential.*

Proof. Note that $X = \mathbb{B}^l$ is complete, because it is finite.

Let $\hat{i} = (i^e; i^1, i^2)$ be a state in which the elite set contains elements that are not Pareto optimal.

1. From i^1, a set of clones is generated. Next, a fixed number of (randomly chosen) string positions of these clones are mutated. Then we change the initial positions in all the strings of the clones (there exists a positive probability of doing this). The set obtained from this process is called $ClonesM(i^1)$.

2. Since a uniform mutation is applied to i^2, we change whatever is necessary in all the elements within this set, so that we can obtain the worst element of $ClonesM(i^1)$. As before, there exists a positive probability of doing this, so that none of these elements enters the primary set.

3. Then, all the elements are rearranged and we select the nondominated elements and they are placed in j^1. Now, let j^2 contain a number of individuals of the remainder of the elements available, until completing n (n is the population size).

4. Applying elitism we obtain the set j^e.

5. To the clones of j^1, we mutate the same initial string positions. Then

$$ClonesM(j^1) \subseteq ClonesM(i^1).$$

Therefore, the best elements of $ClonesM(j^1)$ will be in j^1 again. When we apply elitism to the elements of j^1, we do not modify the set j^e.

6. Let $j^e_{s_1}, \ldots, j^e_{s_k}$ be the elements of j^e that are not Pareto optimal. As X is complete, there exist elements $i^\star_{s_1}, \ldots, j^\star_{s_k} \in \mathcal{P}^*$ that dominate $j^e_{s_1}, \ldots, j^e_{s_k}$, respectively.

7. Now, since we apply uniform mutation to j^2, we can obtain $i^\star_{s_1}, \ldots, j^\star_{s_k}$ from $j^2_1, \ldots, j^2_k$, respectively, and the other elements of j^2 are left as they were before.

8. Like $ClonesM(j^1)$ and $\{j_{k+1}, \ldots, j_{n_2}\}$ had already been modified j^e, when applying elitism we will not modify again j^e. Thus, the only part of j^e that is modified will be $i^\star_{s_1}, \ldots, j^\star_{s_k}$ and they will replace the nondominated elements of j^e.

9. Finally, let $i^\dagger$ be the resulting state of this process. Using the previous process (1-8), we can go from $\hat{\imath}$ to $i^\dagger$ $(\hat{\imath} \rightarrow i^\dagger)$, but as in $i^{\dagger e}$ there are only Pareto optimal solutions, from (3.5) $P_{i^\dagger \hat{\imath}} = 0$ $(i.e.\ \ i^\dagger \not\rightarrow \hat{\imath})$. This proves that $\hat{\imath}$ is an inessential state. ■

From Lemma 4.4, the convergence of MISA is easily obtained as follows.

Theorem 4.3 *The MISA using elitism converges.*

Proof.
From Lemma 4.4 and Corollary 3.2 in chapter 3 we have

$$\mathbb{P}(\{X^e_k\} \subset \mathcal{P}^*) = \mathbb{P}(X_k \in E) = 1 - \mathbb{P}(X_k \in I) \rightarrow 1 - 0 = 1$$

as $k \rightarrow \infty$. This completes the proof. ■

4.5 Conclusions and Future Work

We have presented a proof of convergence of the multiobjective artificial immune system algorithm (MISA) presented in [5]. The convergence analysis indicates that the use of elitism (which is represented in the form of an *elite set* in the case of multiobjective optimization) is necessary to guarantee convergence. To the author's best knowledge, this is the first proof of convergence presented for a multiobjective artificial immune system.

As part of our future work, we plan to extend our theoretical analysis to other types of artificial immune systems [6]. We are also interested in defining a more general framework for proving convergence of heuristics based on a mutation operator. Such a framework would allow us to prove convergence of a family of heuristics that comply with a certain (minimum) set of requirements, rather than having to devise a specific proof for each of them.

Chapter 5

Using Stripes to Maintain Diversity in a MOPSO

In this chapter, we propose a new mechanism to maintain diversity in multi-objective optimization problems. The proposed mechanism is based on the use of stripes that are applied on objective function space and that are independent of the search engine adopted to solve the multi-objective optimization problem. In order to validate the proposed approach, we included it in a multi-objective particle swarm optimizer. Our approach was compared with respect to two multi-objective evolutionary algorithms which are representative of the state-of-the-art in the area. The results obtained indicate that our proposed mechanism is a viable alternative to maintain diversity in the context of multi-objective optimization.

5.1 Introduction

In the last few years, several multi-objective particle swarm optimizers (MOPSOs) have been proposed in the specialized literature (see for example [4, 35, 48, 20, 28, 29, 50, 38]).

Most of this work, however, focuses mainly on the design of novel selection or archiving mechanisms. Nevertheless, the design of effective mechanisms to maintain diversity remains as a key issue when extending particle swarm optimizers so that they can deal with multi-objective optimization problems.

In some recent work, a few authors have proposed or adopted novel mechanisms to maintain diversity in their MOPSOs (e.g., [34, 33, 50]). Such approaches have led to the development of very successful multi-objective particle swarm optimizers.

In this chapter, we propose a new mechanism to maintain diversity, which we show to overcome the main drawbacks of other popular mechanisms such as ε-dominance [26] and the sigma method proposed in [34].

The remainder of the chapter is organized as follows. Section 5.2 presents the most relevant previous related work. Our proposed approach is described in Section 5.3. Section 5.4 presents a comparison of the results produced by our approach (coupled to a MOPSO) and two multi-objective evolutionary algorithms that are representative of the state-of-the-art. Finally, in Section 5.5, we present our conclusions and some possible paths for future research.

5.2 Previous Related Work

There are two main approaches to maintain diversity of MOPSOs that have been reported in the specialized literature: the sigma method proposed by Mostaghim et al. [34] and the ε-dominance method proposed by Laumanns et al. [26].

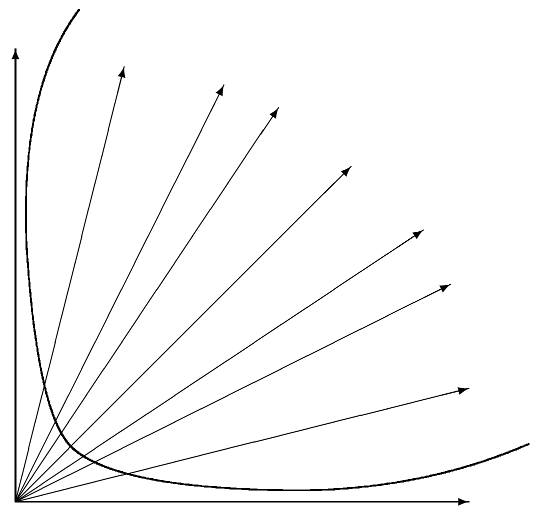


Figure 5.1: This figure illustrates a situation that causes problems to the sigma method proposed by Mostaghim et al. [34].

The sigma method uses the vector represented by the evaluation of $F(\vec{x})$ of the particle $\vec{x}$, and the leader of this particle is the individual in the elite set whose sigma is closest to the sigma of $\vec{x}$ (sigma is a direction and is computed using an expression provided by the authors of this method [34]). The core idea in the sigma method is to form clusters using the particles in the elite set as the centers of such clusters. Note however that the elite set could be very large. Since the number of elements in each cluster is not bounded, there could be leaders with many "followers" and some leaders with no "followers". In consequence, the approach may fail to cover all the Pareto front. Also, the approach requires that all the objective function values are positive (some sort of scaling is required when this is not the case). Figure 5.1 shows a case in which the sigma method could fail. In this figure, all the directions go to the portion of the Pareto front which is closer to the “ideal vector”. Thus, it is possible that the solutions generated do not cover all the Pareto front.

Let us consider again the MOP (1.1).

The concept of ε-dominance [26] refers to a relaxed form of dominance. A decision vector x_1 is said to ε-dominate a decision vector x_2 for some $\varepsilon > 0$ iff: $f_i(x_1)/(1+\varepsilon) \leq f_i(x_2), \forall i = 1, ..., d$, and $f_i(x_1)/(1+\varepsilon) < f_i(x_2)$; for at least one $i = 1, ..., d$ (d is the total number of objective functions of the problem). It is worth noting that ε is a user-defined parameter.

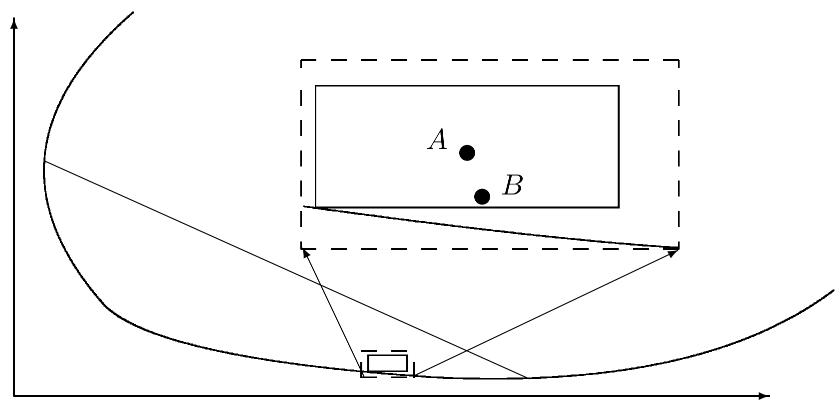


Figure 5.2: An example in which the ε-dominance approach retains the wrong point.

This concept is normally used to fix the size of the external archive (or secondary population) in which a multi-objective evolutionary algorithm retains the nondominated vectors found during the search.

The main drawback of the $\varepsilon-$dominance method is the number of comparisons and distances that have to be computed. Another possible problem with the ε-dominance approach is shown in Figure 5.2 because . In this case, the point A is closer to the lower lefthand corner than point B, but point B is closer to the Pareto front than point A. So, in this case, the ε-dominance approach retains point A. In contrast, our approach will retain point B.

5.3 Our Proposal

Throughout the remainder of this chapter the functions $f_1, \ldots, f_d$ are supposed to be bounded below.

First, we present the next result and definition that is part of our idea.

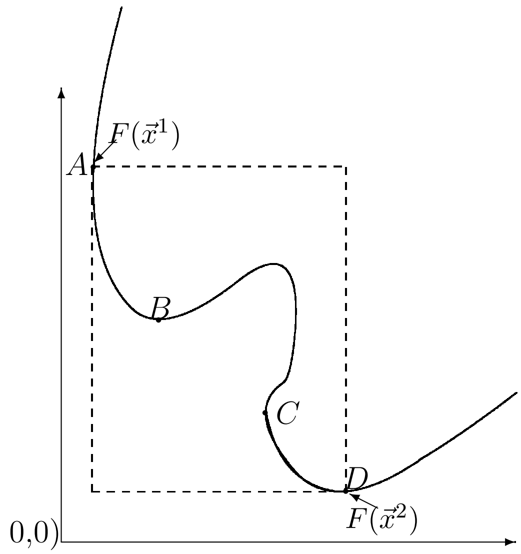


Figure 5.3: $F(\mathcal{P}^*)$ is contained in the "hyper-box" defined by $F(\vec{x}^1)$, $F(\vec{x}^2)$

Lemma 5.1 *Let $\vec{x}^1, \vec{x}^2, \ldots, \vec{x}^d \in X$ be the minimizers of the functions $f_1, f_2, \ldots, f_d$ respectively. Then the Pareto front is contained in the "hyper-box" defined by the points $F(\vec{x}^1)$, $F(\vec{x}^2), \ldots, F(\vec{x}^d)$.*

The proof of Lemma 5.1 is trivial and is, therefore, omitted here. The lemma is illustrated in Figure 5.3, for the case in which $d = 2$ and the Pareto front corresponds to the parts on the boundary of S joining the points A and B, and also the points C and D,

Definition 5.1 *Let $X_1, \ldots, X_d \in I\!R^d$ then the* **convex hull** *of these vectors is*

$$CH(X_1, \ldots, X_d) = \left\{ \sum_{i=1}^{d} \alpha_i X_i / \sum_{i=1}^{d} \alpha_i = 1, \alpha_i \geq 0, \alpha_i \in I\!R \right\}.$$

The core idea of the approach proposed in this chapter (which we call "stripes") is that the convex hull generated by the points $F(\vec{x}^1), F(\vec{x}^2), \ldots, F(\vec{x}^d)$ (defined in Lemma 5.1) is "similar" to the Pareto front (see Figures 5.4 and 5.5). Thus, we can use several points (which we call stripe centers) uniformly distributed along this convex hull, and we assign the individuals of the population to the nearest stripe center. This way, we are distributing the individuals in several stripes determined by the stripe centers (see Figure 5.4). Now, if we set an upper bound on the number of individuals in each stripe and on the number of elements of the Pareto front, the approach will provide a distribution of points, avoiding an excessive clustering in any particular region from those defined by the stripes.

In this chapter, we use the notion of clustering, but the center of each cluster is fixed and uniformly distributed along the convex hull, as shown in Figures 5.4 and 5.5 (the small circles are the centers of the clusters).

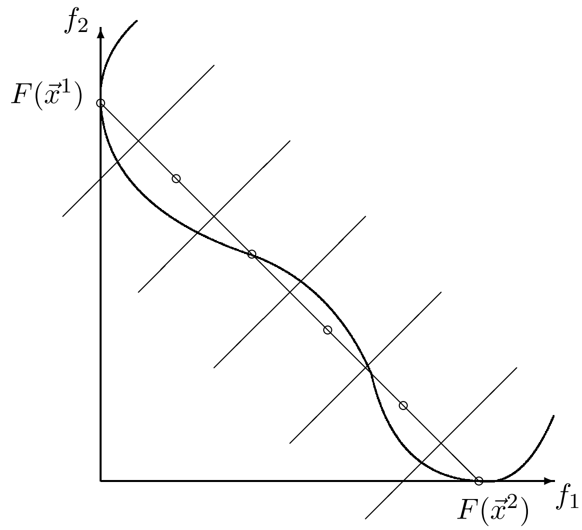


Figure 5.4: Graphical representation of the stripes proposed in this chapter.

The stripe center set can be computed using:

$$\left\{ X_{i_1,\ldots,i_d} = \sum_{j=1}^{d} \frac{i_j F(\vec{x}^i)}{nl} \;/\; i_1 + \cdots + i_d = nl,\; i_1, \ldots, i_d \in I\!N \cup \{0\} \right\},$$

where nl, a parameter provided by the user, is the number of points in each edge of the convex hull.

Special case $d = 2$

In the case $d = 2$, we can do some simplifications that are presented next.

In this case the stripe center set can be computed using

$$\left\{ X_i = \frac{iF(\vec{x}^1) + (ns - 1 - i)F(\vec{x}^2)}{ns - 1}, i \in \{0, 1, \ldots, ns - 1\} \right\}, \tag{5.1}$$

where ns is the number of stripes, which is equal to nl.

In the case in which there are only two objective functions, $d = 2$, we can apply a rotation to all elements in the population and to all elements in the elite set, such that the vector $F(\vec{x}^1) - F(\vec{x}^2)$ is parallel to the $x-$axis. Then the stripe of every element in the population is calculated using the coordinate x of the rotated element, as follows. Let θ be the angle between the $x-$axis and the vector $F(\vec{x}^1) - F(\vec{x}^2)$. Thus, this angle is what we need to rotate all the elements. Further, if $F^r(\vec{x}) = (f_1^r(\vec{x}), f_2^r(\vec{x}))$ are the rotated

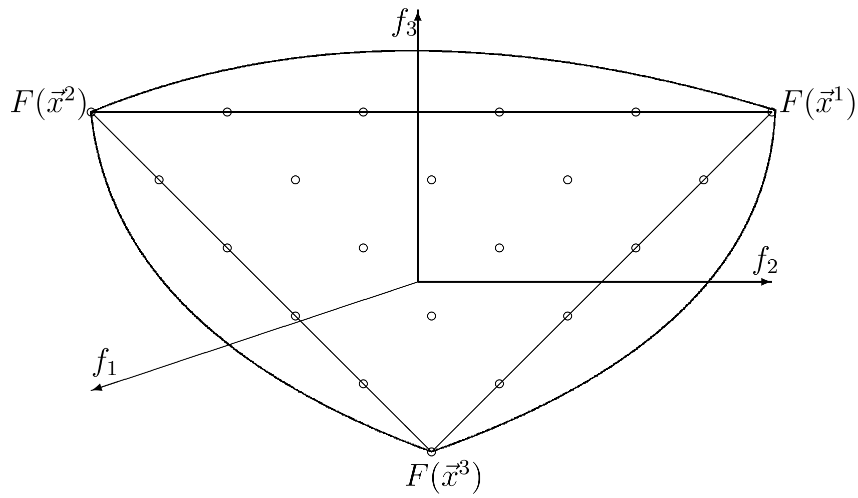


Figure 5.5: Distribution of the stripes center for $d = 3$, $nl = 6$

coordinates of $F(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}))$, we have

$$\begin{aligned} f_1^r(\vec{x}) &= \cos(\theta) f_1(\vec{x}) - \sin(\theta) f_2(\vec{x}) \\ f_2^r(\vec{x}) &= \sin(\theta) f_1(\vec{x}) + \cos(\theta) f_2(\vec{x}) \end{aligned} \tag{5.2}$$

Now, to determine the stripe of the individual whose evaluation is $F(\vec{x})$ we use the following expressions. Let

$$h = \frac{f_1^r(\vec{x}^2) - f_1^r(\vec{x}^1)}{ns - 1}, \quad \text{and} \quad h_{\vec{x}} = \frac{f_1^r(\vec{x}) - f_1^r(\vec{x}^1)}{h}.$$

Then

$$\text{stripe}(\vec{x}) = \begin{cases} 1 & \text{if} \quad h_{\vec{x}} < 0.5 \\ [\![\, h_{\vec{x}} + 1.5]\!] & \text{if} \quad 0.5 \leq h_{\vec{x}} < ns - 0.5 \\ ns & \text{if} \quad h_{\vec{x}} \geq ns - 0.5, \end{cases}$$

where $[\![r]\!]$ denotes the integer part of $r \in I\!R$.

This procedure to assign the stripe is simpler than the method for calculating distances to the stripe center and for computing the minimum of these distances.

To illustrate the way in which our proposed approach works, we show in Figure 5.6 an example for a problem with two objectives. In the figure, it can be seen that the approach (which was coupled to the MOPSO proposed in [4]) gives a good representation of the Pareto front.

5.3.1 PSO with stripes

In order to validate the effectiveness of our proposed approach to maintain diversity, we used as our search engine the MOPSO previously proposed in [4]. However, in this case, the diversity maintenance scheme are the stripes proposed in this chapter instead of the adaptive grid originally adopted [4]. We call our MOPSO with stripes ST–MOPSO.

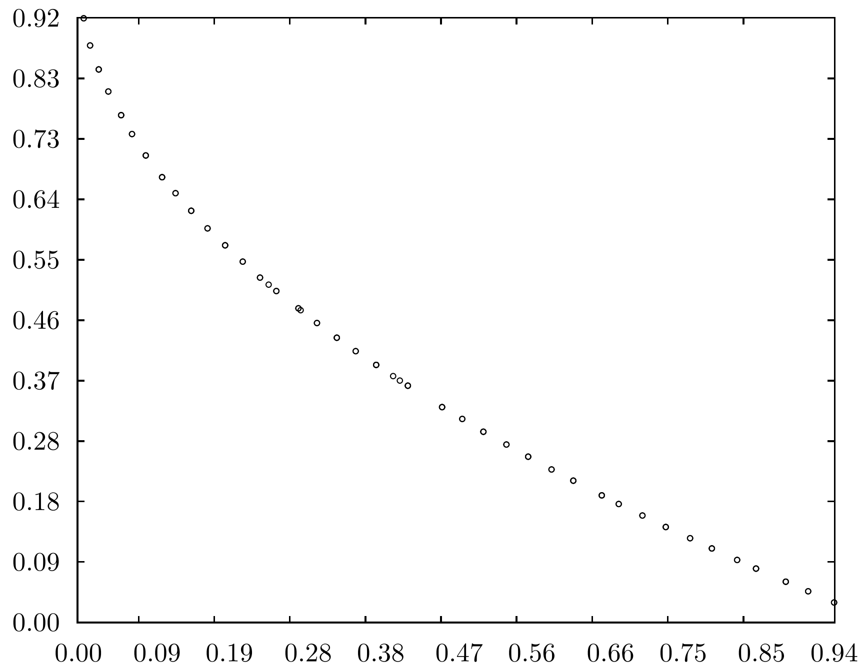


Figure 5.6: An example of the type of distribution of non-dominated solutions produced by our approach.

Our proposal consists of using one leader in each stripe and to compute a weighted sum determined by the points $F(\vec{x}^1), F(\vec{x}^2), \ldots, F(\vec{x}^d)$ defined in Lemma 5.1, to select the leaders, where the leader of a stripe is the point that minimizes this weighted sum.

To compute the parameter of the scalarization we use the coefficients of the normal vector $\vec{n}$ of the "affine subspace" (hyper–plane) that contains the points $F(\vec{x}^1), F(\vec{x}^2), \ldots,$ $F(\vec{x}^d)$.

Thus for $d = 2$ the normal vector is:

$$\vec{n} = (|f_2(\vec{x}^1) - f_2(\vec{x}^2)|, |f_1(\vec{x}^1) - f_1(\vec{x}^2)|).$$

For the case $d = 3$ we will use the vector product as follows.

Let $(a, b, c) = [F(\vec{x}^1) - F(\vec{x}^2)] \times [F(\vec{x}^1) - F(\vec{x}^3)]$ then

$$\vec{n} = (|a|, |b|, |c|).$$

In the other cases we will use Gram-Schmidt orthogonalization process to obtain the normal vector or the orthogonal projection of some special vector on the "Affine Subspace".

Thus, the leader of a stripe is the element that minimizes.

$$\vec{n} \cdot F = \sum_{i=1}^{d} n_i f_i(\vec{x}), \tag{5.3}$$

with $\vec{n} = (n_1, \ldots, n_d)$.

From Lemma 2.1 of Chapter 2 if the leader is a minimal point of (5.3) it is in $\mathcal{P}^*$. Thus, this form to select the leaders makes sense.

However, in the case in which $d = 2$, we have the rotated coordinates $F^r(\vec{x})$ and the parameters of the scalarization can be taken as:

$$n_1 = \sin(\theta), \text{ and } n_2 = \cos(\theta),$$

the coefficients of the rotation. In this case the leader of a stripe is the particle in the elite set that minimizes f_2^r (the y-coordinate of F^r)

5.4 Comparison of Results

Several test functions were taken from the specialized literature to validate our approach, and here we will include results obtained by ST–MOPSO for three of these test functions for two objetive problem. Our results are compared with respect to those produced by two multi-objective evolutionary algorithms representative of the state-of-the-art in the area: the NSGA-II [9] and ε-MOEA [8].

After that, we present an example in which we consider a MOP with 3 objective funcions (see subsection 5.4.4. Our results are, again, compared with NSGA II and ε-MOEA.

For the first three test funtions, in the results shown next, each approach performed 3000 fitness function evaluations. The results shown correspond to 30 independent runs.

For the example with 3 objectives function, each approach perform 4000 fitness function evaluations and in this case the results was obtained for 20 independent runs.

In order to allow a quantitative comparison of results we adopted the following *performance measures*: two set coverage [59, 58], hypervolume [58], inverted generational distance [55], and success counting (which is a variation of the performance measure called "error ratio" [55]).

The definition of each of these performance measures is presented next.

Two Set Coverage (TSC)

This performance measure can be termed relative coverage comparison of two sets.

Definition 5.2 *Let $\wp(X)$ be the power set of X. Then TSC is defined as follows:*

$$TSC : \wp(X) \times \wp(X) \longrightarrow [0,1],$$

$$TSC(X', X'') := \frac{|\{x'' \in X''; \exists\ x' \in X' : F(x') \preceq F(x'')\}|}{|X''|},$$

$\forall\ X', X'' \subseteq X.$, where $|A|$ denotes the cardinality of A

If all points in X' dominate or are equal to all points in X'', then by definition $TSC = 1$. $TSC = 0$ implies that none of the points in X'' is dominated by a point in X'. In general, $TSC(X', X'')$ and $TSC(X'', X')$ both have to be considered due to set intersections not being empty.

Hypervolume (HV)

This performance measure, denoted $HV(A, B)$ for the algorithms A and B, was proposed by Zitzler and Thiele [58].

Let $\mathcal{P}_A^*$ be the solution (nondominated elements) of an algorithm A . First, we have to define the hypervolume[1] of the region determined by the image $(F(\mathcal{P}_A^*))$ of the solution of the algorithm A and the origin.

For example, a vector $\vec{x} \in \mathcal{P}_A^*$ for a two-objective problem defines a rectangle bounded by the origin and $(f_1(\vec{x}), f_2(\vec{x}))$. The area of the union of all such rectangles defined by each vector in $\mathcal{P}_a^*$ is the hypervolume. Then the hypervolume $V(\mathcal{P}_A^*)$ determined by the algorithm A is defined as:

$$V(\mathcal{P}_A^*) := \text{Hypervolume}\left\{\bigcup_{\vec{x} \in \mathcal{P}_A^*} a_{\vec{x}}\right\},$$

where $a_{\vec{x}}$ is the hyperbox determined by the components of $F(\vec{x})$ and the origin.

Thus the comparative performance measure is defined as follows.

Definition 5.3 *Let $\mathcal{P}_A^*$ and $\mathcal{P}_B^*$ be the solution of two algorithms A and B, respectively. Then the performance measure HV is defined as*

$$HV(A, B) \;=\; \frac{V(\mathcal{P}_A^* \cup \mathcal{P}_B^*) - V(\mathcal{P}_A^*)}{V(\mathcal{P}_A^* \cup \mathcal{P}_B^*)}$$

Observe that this measure is not symmetric, hence we need to consider both $HV(A, B)$ and $HV(B, A)$.

In the case that all elements in B are dominated by elements in A then $HV(A, B) = 0$ and $HV(B, A) > 0$.

[1] The hypervolume is the area under the curve for the 2-dimensional case or the volume under the surface for the 3-dimensional case.

Inverted Generational Distance (IGD)

The concept of generational distance was introduced by Van Veldhuizen & Lamont [55, 56] as a way of estimating how far are the elements in the Pareto front produced by our algorithm from those in the true Pareto front of the problem.

Definition 5.4 *This measure is defined as:*

$$GD = \frac{\left(\sum_{i=1}^{N} d_i^2\right)^{1/2}}{N} \tag{5.4}$$

where N is the number of nondominated vectors found by the algorithm being analyzed and d_i is the Euclidean distance (measured in objective space) between each of these vectors and the nearest member of the true Pareto front.

It should be clear that a value of $GD = 0$ indicates that all the elements generated are in the true Pareto front of the problem. Therefore, any other value will indicate how "far" we are from the global Pareto front of our problem. In our case, we implemented an "inverted" generational distance measure (IGD) in which we use as a reference the true Pareto front, and we compare each of its elements with respect to the front produced by an algorithm. In this way, we are calculating how far are the elements of the true Pareto front, from those in the Pareto front produced by our algorithm. Computing this "inverted" generational distance value reduces the bias that can arise when an algorithm does not fully cover the true Pareto front.

Success Counting (SC)

We define the *success counting* measure based on the idea of the measure called *Error Ratio* proposed by Van Veldhuizen [55] which indicates the percentage of solutions (from the nondominated vectors found so far) that are not members of the true Pareto optimal set. In this case, we count the number of vectors (in the current set of nondominated vectors available) that are members of the Pareto optimal set:

Definition 5.5 *The* success counting *measure SC is defined by*

$$SC = \sum_{i=1}^{N} s_i,$$

where N is the number of vectors in the current set of nondominated vectors available; s_i = 1 if vector i is a member of the Pareto optimal set, and $s_i = 0$ otherwise.

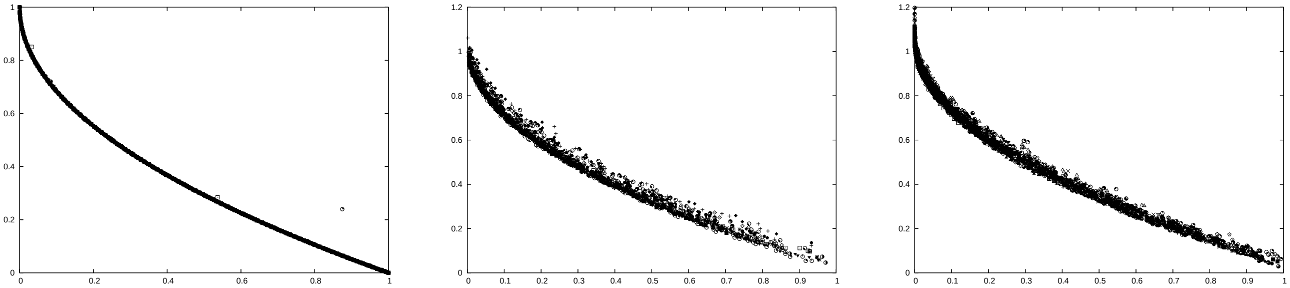


Figure 5.7: Pareto fronts produced by ST-MOPSO (left), ε-MOEA (center) and NSGA-II (right) for the ZDT1's test function.

It should then be clear that $SC = N$ indicates an ideal behavior, since it would mean that all the vectors generated by our algorithm belong to the true Pareto optimal set of the problem. For a fair comparison, when using this measure, all the algorithms should limit their final number of non-dominated solutions to the same value. Note that SC avoids the bias introduced by the Error Ratio measure, which normalizes the number of solutions found (which belong to the true Pareto front) and, therefore, provides only a percentage of solutions that reached the true Pareto front. This percentage does not provide any idea regarding the actual number of non-dominated solutions that each algorithm produced.

Now we present the results obtained for these three test function.

5.4.1 ZDT1's test function

$$
\begin{aligned}
\text{Minimize} \quad & (f_1(\vec{x}), f_2(\vec{x})) \\
f_1(\vec{x}) &= x_1 \\
f_2(\vec{x}) &= g(\vec{x})\ h(f_1(\vec{x}), g(\vec{x})) \\
g(\vec{x}) &= 1 + 9 \sum_{i=2}^{m} \frac{x_i}{(m-1)}, \\
h(x, y) &= 1 - \sqrt{\frac{x}{y}}
\end{aligned}
$$

where $m = 30$, and $x_i \in [0, 1]$.

Figure 5.7 shows the graphical results produced by ST-MOPSO, ε-MOEA and the NSGA-II in the first test function chosen. (The true Pareto front of the problem is shown as a continuous line in the left–hand side image of Figure 5.7). Tables 5.1 and 5.2 show the comparison of results among the three algorithms considering the performance measures

TSC	**ST-MOPSO**	**ε-MOEA**	**NSGA-II**
ST-MOPSO	–	0.999333	0.999666
ε-MOEA	0	–	0.411102
NSGA-II	0	0.0983892	–

HV	**ST-MOPSO**	**ε-MOEA**	**NSGA-II**
ST-MOPSO	–	0	0
ε-MOEA	0.0203594	–	0.000509322
NSGA-II	0.0303835	0.0107699	–

Table 5.1: Results of the Two Set Coverage and Hyper Volume performance measures for the ZDT1's test function.

IGD	**ST-MOPSO**	**ε-MOEA**	**NSGA-II**
Best	**0.000343805**	0.00167212	0.0020484
Worst	**0.000670735**	0.0190439	0.0276651
Mean	**0.000430186**	0.00795849	0.00641032
Stdev	**7.39891e-05**	0.0050593	0.0052243
Median	**0.000419506**	0.00655261	0.00495141

SC	**ST-MOPSO**	**ε-MOEA**	**NSGA-II**
Best	100	2	8
Worst	95	0	0
Mean	99.3	0.3	1.1
Stdev	1.20773	0.534983	1.66816
Median	100	0	1

Table 5.2: Results of the Inverted Generational Distance and Success Counting performance measures for the ZDT1's test function.

previously described. It can be seen that the performance of ST-MOPSO is the best with respect to all the performance measures tested. By looking at the Pareto fronts produced by each algorithm in this test function, it should be clear that ST-MOPSO was the only algorithm that could reach the true Pareto front in most of the runs performed (the output of the 30 independent runs was combined in a single file in order to generate the plots from Figure 5.7).

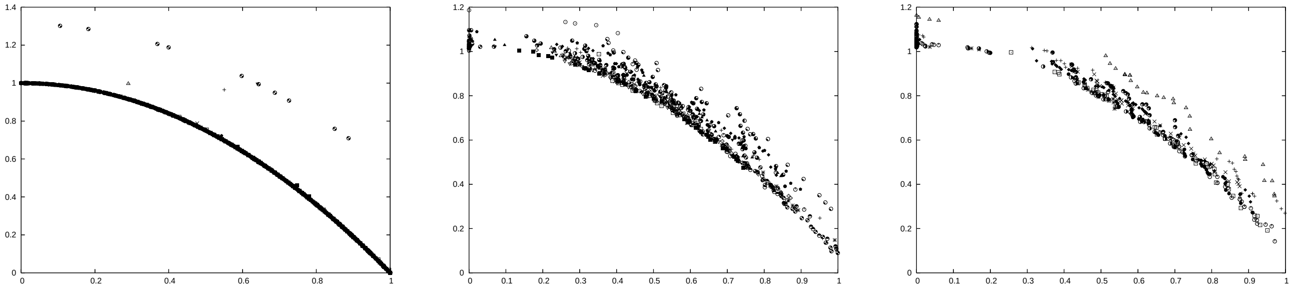


Figure 5.8: Pareto fronts produced by our ST-MOPSO (left), ε-MOEA (center) and NSGA II (right) for the ZDT2's test function.

5.4.2 ZDT2's test function

$$
\begin{aligned}
\text{Minimize} \quad & (f_1(\vec{x}), f_2(\vec{x})) \\
f_1(\vec{x}) &= x_1 \\
f_2(\vec{x}) &= g(\vec{x})\ h(f_1(\vec{x}), g(\vec{x})) \\
g(\vec{x}) &= 1 + 9 \sum_{i=2}^{m} \frac{x_i}{(m-1)} \\
h(x, y) &= 1 - \left(\frac{x}{y}\right)^2
\end{aligned}
$$

where $m = 30$, and $x_i \in$[0,1].

Figure 5.8 shows the graphical results produced by ST-MOPSO, the NSGA-II [9], and ε-MOEA [8] in the second test function adopted.

TSC	**ST-MOPSO**	**ε-MOEA**	**NSGA-II**
ST-MOPSO	–	0.993918	0.993918
ε-MOEA	0	–	0.485246
NSGA-II	0	0.012	–

HV	**ST-MOPSO**	**ε-MOEA**	**NSGA-II**
ST-MOPSO	–	0	0
ε-MOEA	0.0377096	–	0.000175564
NSGA-II	0.0610739	0.0235162	–

Table 5.3: Results of the Two Set Coverage and Hypervolume performance measures for the ZDT2's test function.

Tables 5.3 and 5.4 show the comparison of results among the three algorithms considering the performance measures previously indicated. As in the previous example, the performance of ST-MOPSO was the best with respect to all the performance measures adopted.

Graphically (see Figure 5.8), it can be seen that in this case, our ST-MOPSO generated a few points outside the true Pareto front in one of the runs. However, when looking at the graphical output generated by the other algorithms, it is clear that our ST-MOPSO had the most robust behavior in this problem, since the others produced a considerably large number of solutions outside the true Pareto front of the problem.

IGD	**ST-MOPSO**	**ε-MOEA**	**NSGA-II**
Best	**0.000346726**	0.0038222	0.00461677
Worst	**0.051142**	0.0516044	0.0550343
Mean	**0.0104091**	0.016797	0.0373733
Stdev	**0.0186685**	0.0128839	0.0202878
Median	**0.000439715**	0.0112029	0.0518229

SC	**ST-MOPSO**	**ε-MOEA**	**NSGA-II**
Best	**100**	0	0
Worst	**1**	0	0
Mean	**75.6**	0	0
Stdev	41.8384	**0**	**0**
Median	**100**	0	0

Table 5.4: Results of the Inverted Generational Distance and Success Counting performance measures for the ZDT2's test function.

5.4.3 ZDT3's test function

$$\begin{aligned} \text{Minimize} \quad & (f_1(\vec{x}), f_2(\vec{x})) \\ f_1(\vec{x}) &= x_1 \\ f_2(\vec{x}) &= g(\vec{x})\ h(f_1, g) \\ g(\vec{x}) &= 1 + 9\sum_{i=2}^{m} \frac{x_i}{(m-1)} \\ h(x,y) &= 1 - \sqrt{\frac{x}{y}} - \frac{x}{y}\sin(10\pi x) \end{aligned}$$

where $m = 30$, and $x_i \in[0,1]$.

Figure 5.9 shows the graphical results produced by the ST-MOPSO, the NSGA-II, and ε-MOEA in the third test function chosen.

Tables 5.5 and 5.6 show the comparison of results among the three algorithms considering the performance measures previously described.

Once more, our ST-MOPSO had the best performance with respect to all the performance measures considered. Graphically (see Figure 5.9), it can be seen that in this

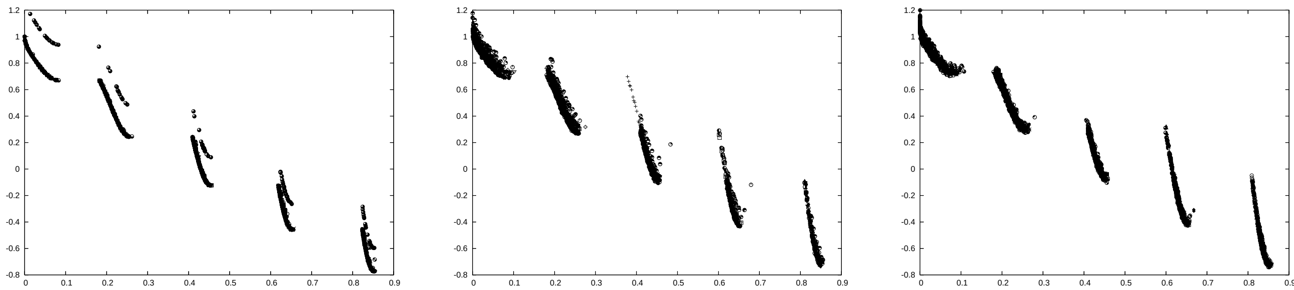


Figure 5.9: Pareto fronts produced by ST-MOPSO (left), ε-MOEA (center) and NSGA II (right) for the ZDT3's test function.

case, our ST-MOPSO generated some points outside the true Pareto front in some of the runs. However, when looking at the graphical output generated by the other algorithms, it is clear that our ST-MOPSO had the most robust behavior in this problem, since the others produced a considerably large number of solutions outside the true Pareto front of the problem (this is corroborated by the values of the performance measures).

TSC	**ST-MOPSO**	**ε-MOEA**	**NSGA-II**
ST-MOPSO	–	0.934675	0.93291
ε-MOEA	0	–	0.24594
NSGA-II	0	0.0680684	–

HV	**ST-MOPSO**	**ε-MOEA**	**NSGA-II**
ST-MOPSO	-	0	0
ε-MOEA	0.019099	-	0.000320688
NSGA-II	0.026448	0.0076528	-

Table 5.5: Results of the Two Set Coverage and Hyper Volume performance measures for the ZDT3's test function.

Summarizing our results, it can be seen that the performance of our ST-MOPSO is the best with respect to all the performance measures tested. By looking at the Pareto fronts of the three test functions adopted, it can be easily seen that most of the executions of the ST-MOPSO algorithm reached the true Pareto front, which is an indicative of the robustness of the approach. This contrasts with the other approaches, which not only showed a higher variation of results, but were also unable to reach the true Pareto front in most of the runs, which is due to the relatively low number of fitness function evaluations considered. When using a larger number of evaluations the two other approaches are able to reach consistently the true Pareto front of the test functions adopted. Note that our ST-MOPSO was the only algorithm able to cover the entire Pareto front of the test problems adopted in our comparative study.

IGD	ST-MOPSO	ε-MOEA	NSGA-II
Best	**0.000705166**	0.00236056	00163111
Worst	**0.0397353**	0.0225697	0.0231257
Mean	**0.0036884**	0.00842278	0.0065121
Stdev	**0.00707846**	0.00397113	0.00445668
Median	**0.00203803**	0.00804916	0.0059502

SC	ST-MOPSO	ε-MOEA	NSGA-II
Best	**100**	4	1
Worst	0	0	0
Mean	**82.7333**	0.566667	0.0666667
Stdev	25.8082	1.16511	**0.253708**
Median	**88.5**	0	0

Table 5.6: Results of the Inverted Generational Distance and Success Counting performance measures for the ZDT3's test function.

5.4.4 An example with 3 objective function

We presented in the previous subsections optimization problems with 2 objective functions Now we present an example with 3 objectives, the so–called Viennet test funcion (see [2, p. 458]):

$$\begin{aligned} \text{Minimize} \quad & (f_1(\vec{x}), f_2(\vec{x}), f_3(\vec{x})) \\ f_1(\vec{x}) &= x_1^2 + (x_2 - 1)^2 \\ f_2(\vec{x}) &= x_1^2 + (x_2 + 1)^2 + 1 \\ f_3(\vec{x}) &= (x_1 - 1)^2 + x_2^2 + 2 \end{aligned}$$

Figure 5.10 on page 47 shows the graphical representation of the solution produced by the ST-MOPSO (only the nondominated elements are shown), the ε-MOEA and the NSGA-II for this optimization problem. (in each graphic the true Pareto front of the problem is shown using dots)

We can see that the the solution obtained by ε-MOEA is well distributed, but it does reach the extrema of the Pareto front, whereas the solution of ST-MOPSO and NSGA-II do it. Thus, this is a drawback of the ϵ-dominance mechanism. Also we can see NSGA-II and ST-MOPSO obtain well distributed solutions with points covering all the Pareto front.

Tables 5.7 and 5.8 show the comparison of results among the three algorithms considering the performance measures previously described.

We can observe that for TSC and IGD performance measures, the results from the three algorithms are similar. However, for the SC metric, the best results were obtained

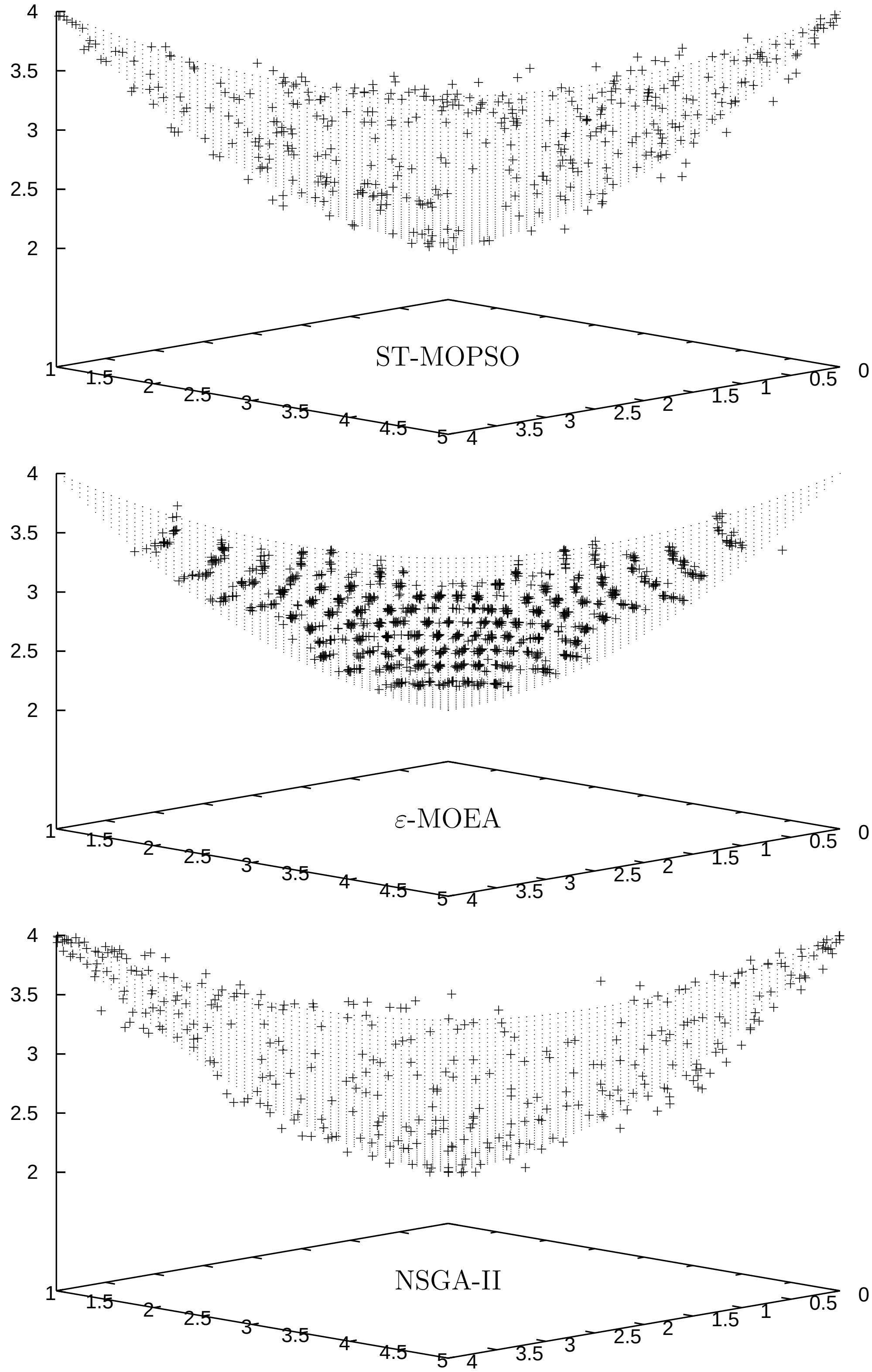


Figure 5.10: Pareto fronts produced by ST-MOPSO (top), ε-MOEA (center) and NSGA II (bottom) for the Viennet's test function (3 objectives functions).

TSC	**ST-MOPSO**	**ε-MOEA**	**NSGA-II**
ST-MOPSO	–	0.9920	0.9765
ε-MOEA	0.9995	–	0.9990
NSGA-II	0.9175	0.9665	–

HV	**ST-MOPSO**	**ε-MOEA**	**NSGA-II**
ST-MOPSO	–	0.002250206	0.02687405
ε-MOEA	0.235307772	–	0.002490852
NSGA-II	-0.000842701	0.256664038	–

Table 5.7: Results of the Two Set Coverage and Hyper Volume performance measures for Viennet's test function.

IGD	**ST-MOPSO**	**ε-MOEA**	**NSGA-II**
Best	0.0033421	0.00377578	0.0031531
Worst	0.0038932	0.00430738	0.0037456
Mean	0.0034570	0.00412982	0.0034246
Stdev	0.0001919	0.00015831	0.0002381
Median	0.0033956	0.00418837	0.0034085

SC	**ST-MOPSO**	**ε-MOEA**	**NSGA-II**
Best	100	100	88
Worst	90	93	77
Mean	95.7	96.81	83.05
Stdev	4.54336	2.2682	3.76235
Median	90	93.5	82

Table 5.8: Results of the Inverted Generational Distance and Success Counting performance measures for the Viennet's test function.

from $\varepsilon-$MOEA. ST-MOPSO obtained the second place with very similar results. The worst performance was obtained by the NSGA-II algorithm showing a significant difference with the others.

On the other hand, if we consider the HV performance measure, we can observe that the NSGA-II algorithm obtained the best results and ST-MPSO obtained similar results to it. In this case $\varepsilon-$MOEA shows poor results. A similar analysis could be obtained if we observe the graphics in Figure 5.10.

Summarizing, if we consider the all four performance measures and the graphical representation, we can conclude that ST-MOPSO outperforms the other algorithms, for this function. But it is necessary to conduct more experiments in order to be able to provide a more reliable assessment of the effectiveness of our approach

5.5 Conclusions and Future Work

In this chapter, we have proposed a new mechanism to maintain diversity which is based on the use of stripes. The mechanism was incorporated into a multi-objective particle swarm optimizer (MOPSO) in order to validate its effectiveness. The results indicate that the approach is a viable alternative to maintain diversity in a multi-objective evolutionary algorithm (not necessarily a particle swarm optimizer).

As part of our future work, we would like to extend the approach to handle any number of dimensions (i.e., objectives), since our current version only deals with optimization problems with 2 or 3 objectives. We also intend to test this approach with other types of multi-objective optimization heuristics, such as the artificial immune system [3]. Finally, we are also developing a new performance measure based on the stripes introduced in this chapter. The idea is that this new performance measure can be used to assess the performance of multi-objective evolutionary algorithms regarding spread and distribution of nondominated solutions.

Chapter 6

Portfolio Optimization using PSO with Stripes

As an aplication of ST-MOPSO, we consider the Portfolio Optimization Problem developed by Markowitz [30]. The basic assumption is that the investor tries to maximize his/her profit and at the same time wants to minimize the risk. This problem is usually solved using a scalarization approach. Here we solve it as a bi-objective optimization problem.

6.1 Description of the Model

We considere a market where s different securities (i.e. stocks) are traded. These securities have prices $p_1, p_2, \ldots, p_s$ at the initial time $t = 0$. We restrict ourselves to a one-period model. This means that the investor makes his decisions at the beginning of the period and is not allowed to revise his decisions until the end of the period. Let $P_1(T), P_2(T), \ldots, P_s(T)$ be the prices of the securities at the final time $t = T$, we assume that these final prices are not foreseeable. Therefore, they are modeled as non-negative random variables on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$.

The return of the stocks is given by the variables $r_1, r_2, \ldots, r_s$ given by

$$r_i = \frac{P_i(T) - p_i}{p_i}, \quad i = 1, \ldots, s. \tag{6.1}$$

Observe that r_i is also a random variable.

We assume that we know (or have estimated) their means, variances and covariances.

$$\begin{aligned} E(r_i) &= \mu_i \quad \text{for all } i = 1, \ldots, s, \\ Cov(r_i, r_j) &= \sigma_{ij} \quad \text{for all } i, j = 1, \ldots, s. \end{aligned} \tag{6.2}$$

Using the variables x_i for the share of the $i-$th security on the portfolio, we can calculate the return of the portfolio $R_p = R_p(x_1, \dots, x_s)$ by

$$R_p = \sum_{i=1}^{s} x_i r_i, \tag{6.3}$$

with the restrictions on the shares

$$\sum_{i=1}^{s} x_i = 1 \quad \text{and } x_i \geq 0 \quad i = 1, \dots, s.$$

We have observed that the r_i are random variables with means μ_i and covariances $\sigma_{ij} = E(r_i - E(r_i))(r_j - E(r_j))$. Thus the return of the portfolio R_p is a random variable as well, and its mean μ_p is given by

$$\mu_p \;=\; E(R_p) \;=\; \sum_{i=1}^{s} x_i E(r_i) \;=\; \sum_{i=1}^{s} x_i \mu_i.$$

We measure the risk contained in the portfolio by the variance of its return

$$\sigma_p^2 \;= Var(R_p) = \; E\left[\{R_p - E(R_p)\}^2\right] \;=\; \sum_{j=1}^{s}\sum_{i=1}^{s} x_i \sigma_{ij} x_j \;=\; \sum_{i,j=1}^{s} x_i x_j \sigma_{ij}.$$

We will also impose the constraints

$$x_i \leq c_i, \quad \text{for all} \quad i = 1, \dots, s,$$

where the c_i are constants.

Therefore, the investor wants to find a vector $\vec{x} = (x_1, x_2, \dots, x_s)$ that maximizes the mean return

$$\mu_p \;=\; \sum_{i=1}^{s} x_i \mu_i =: -f_1(\vec{x})$$

and at the same time minimizes the risk

$$\sigma_p^2 \;=\; \sum_{i,j=1}^{s} x_i x_j \sigma_{ij} =: f_2(\vec{x}),$$

subject to the constraints

$$\sum_{i=1}^{s} x_i = 1 \quad \text{and } 0 \leq x_i \leq c_i \quad \forall\, i = 1, \dots, s.$$

Thus, we have the next definition.

Definition 6.1 *The classical* portfolio optimization problem *(POP) with two objective functions is to find the vector* $\vec{x}^* = (x_1^*, x_2^*, \ldots, x_s^*)$ *such that*

$$\begin{aligned} (f_1(\vec{x}^*), f_2(\vec{x}^*)) &= \min_{\vec{x}} \left(-\sum_{i=1}^{s} x_i \mu_i \ , \ \sum_{i,j=1}^{s} x_i x_j \sigma_{ij} \right) \\ \textit{subject to} \quad & \sum_{i=1}^{s} x_i = 1, \\ & 0 \leq x_i \leq c_i \quad \forall \ i = 1, \ldots, s. \end{aligned} \tag{6.4}$$

Classical Solution

The classical way to solve this problem is by solving a single–objective (or scalar) problem, (see for example, [24]). One can also consider several variants of (6.4).

For instance we may require a lower bound (R_c) on the mean return, and then choose the portfolio with minimal variance, that is

$$\begin{aligned} \min_{\vec{x}} \sigma_p^2 &= \min_{\vec{x}} \sum_{i,j=1}^{s} x_i x_j \sigma_{ij} \\ \text{subject to} \quad & \mu_p \geq R_c \\ & \sum_{i=1}^{s} x_i = 1, \\ & 0 \leq x_i \leq c_i \quad \forall \ i = 1, \ldots, s \end{aligned} \tag{6.5}$$

Alternatively, one can consider the dual problem of setting up an upper bound (σ_c) on the portfolio variance, and then maximize the mean return.

$$\begin{aligned} \max_{\vec{x}} \mu_p &= \max_{\vec{x}} \sum_{i=1}^{s} x_i \mu_i \\ \text{subject to} \quad & \sigma_p^2 \leq \sigma_c \\ & \sum_{i=1}^{s} x_i = 1, \\ & 0 \leq x_i \leq c_i \quad \forall \ i = 1, \ldots, s. \end{aligned} \tag{6.6}$$

In any of these two forms of the POP, we usually find a single point of the Pareto front. (see Figure 6.1).

Still another variant of the POP is

$$\begin{aligned}
\min_{\vec{x}}(\sigma_p^2 - \mu_p) \quad &= \quad \min_{\vec{x}} \left(\sum_{i,j=1}^{s} x_i x_j \sigma_{ij} - \sum_{i=1}^{s} x_i \mu_i \right) \\
\text{subject to} \quad & \quad \sum_{i=1}^{s} x_i = 1, \\
& \quad 0 \le x_i \le c_i \quad \forall\ i = 1, \dots, s.
\end{aligned} \tag{6.7}$$

Again, the solution of this single–objective problem gives only one point of the Pareto front, and the investor does not have the option to select another portfolio with a similar risk and/or a better return.

This situation is illustrated in Figure 6.1, which shows a classical Pareto front for the POP. If the value of σ_c is close to 0, we can see that a small increase in the risk can give a much higher return. In contrast, if σ_p^2 is large, then to obtain a small increase in the return requires a large increase in the risk.

In the single–objective formulation of the POP, the investor cannot appreciate these subtleties.

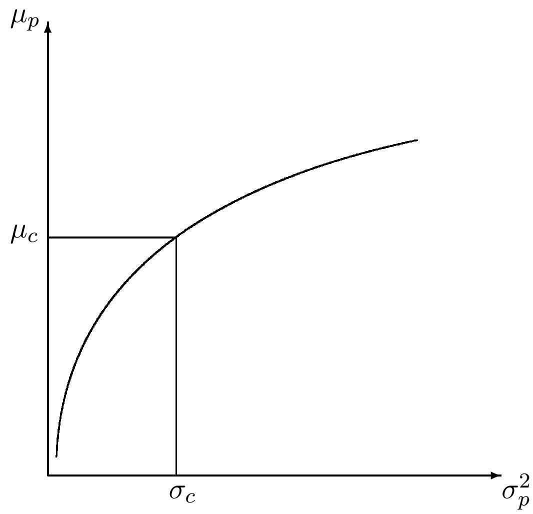


Figure 6.1: Graphical illustration of the Pareto front for the POP

6.2 The Data

To test our algorithm we took 20 securities (i.e. $s = 20$) from the "Mexican Stock Market" (BMV = Bolsa Mexicana de Valores). These securities appear in the "Index of Prices and Quotations" (IPyC = Indice de Precios y Cotizaciones).

We took the prices of the 20 stocks for 100 days (see Table 6.1; the whole data is in Tables A.1, A.2, A.3, A.4, in Appendix A). Then we calculated the return of each

date	AlfaA	AmTelA1	Amxl	BImboA	Cemex CPO	Elektra	Femsaubd	gcarsoa1	$\cdots$
28/09/2004	42.090	23.800	22.027	24.752	63.800	76.200	50.200	51.843	$\cdots$
29/09/2004	42.880	24.420	22.226	24.655	64.720	76.790	50.620	52.787	$\cdots$
30/09/2004	43.060	24.600	22.206	24.439	64.090	76.480	50.300	51.992	$\cdots$
01/10/2004	43.480	24.890	22.756	25.203	64.800	76.750	50.830	52.250	$\cdots$
04/10/2004	43.280	25.250	23.185	25.350	65.760	76.400	50.870	52.558	$\cdots$
05/10/2004	43.100	24.600	22.956	25.340	65.470	76.610	51.040	52.648	$\cdots$
06/10/2004	42.860	24.300	22.526	25.144	67.140	76.690	51.140	52.518	$\cdots$
07/10/2004	42.990	24.310	22.506	25.291	66.580	78.000	51.010	52.379	$\cdots$
08/10/2004	42.150	23.810	22.007	24.214	65.120	79.500	50.920	52.131	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Table 6.1: Example of table of prices.

Fecha	AlfaA	AmTelA1	Amxl	BImboA	Cemex CPO	Elektra	Femsaubd	gcarsoa1	$\cdots$
29/09/2004	1.877	2.605	0.907	-0.395	1.442	0.774	0.837	1.821	$\cdots$
30/09/2004	0.420	0.737	-0.090	-0.873	-0.973	-0.404	-0.632	-1.506	$\cdots$
01/10/2004	0.975	1.179	2.474	3.124	1.108	0.353	1.054	0.497	$\cdots$
04/10/2004	-0.460	1.446	1.888	0.583	1.481	-0.456	0.079	0.590	$\cdots$
05/10/2004	-0.416	-2.574	-0.991	-0.039	-0.441	0.275	0.334	0.170	$\cdots$
06/10/2004	-0.557	-1.220	-1.871	-0.772	2.551	0.104	0.196	-0.245	$\cdots$
07/10/2004	0.303	0.041	-0.089	0.584	-0.834	1.708	-0.254	-0.265	$\cdots$
08/10/2004	-1.954	-2.057	-2.219	-4.257	-2.193	1.923	-0.176	-0.474	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Table 6.2: Example of table of returns.

security, for each day, using equation (6.1). See Table 6.2, in tables A.5, A.6, A.7, A.8 in Appendix A are the whole data.

To compute estimates of the mean returns and the covariances in (6.2) we used $5-$day moving averages, that is, using the data from day $n-4$ to day n with $n = 5, 6, \ldots, 100$. This procedure gave us 95 matrices of order 21×20 whose first row are the mean returns. An example of these matrices appears in Table 6.3.

Then to these data we applied the ST-MOPSO algorithm to obtain a Pareto front for each of the 95 matrices. In each case we used the constraints $c_i = 0.2$ for all $i = 1, \ldots, s$.

6.3 The Results

To apply the results obtained in the previous section, the idea was to use the 5-day data to decide the portfolio for the sixth day. Hence for each of the 95 matrices we tried to obtain a Pareto front with 100 points. The Table 6.4 is an example of the resulting solution, and Figure 6.2 shows the corresponding graph. (Appendix B contains all the graphs). The solutions tell us, according to the POP, the fraction (or the share (in percent)) of our wealth that we should invest in each of the 20 securities.

Since each of the 100 points in the Pareto front is a possible portfolio, handling this information turns out to be quite complicated. Therefore, we sorted the solutions according to their risks and took only three solutions per day: the solution with the minimal

0.48	0.68	0.84	0.48	0.52	0.11	0.33	0.31	0.78	1.08	0.52	1.02	0.26	1.36	0.81	0.27	0.16	0.60	-0.12	0.19
3.89	5.13	1.61	0.54	1.63	1.46	1.36	2.18	4.89	9.07	0.44	-0.05	-0.58	4.25	-0.14	2.89	1.30	1.30	10.88	1.23
5.13	15.14	7.65	1.32	5.85	0.40	1.08	3.57	12.53	11.03	5.41	0.43	1.90	11.16	0.25	8.03	2.89	1.42	9.91	4.39
1.61	7.65	7.99	6.58	5.18	0.02	1.84	2.64	6.23	1.75	6.97	3.74	2.84	-0.23	4.65	3.19	1.81	2.99	-2.37	2.19
0.54	1.32	6.58	9.87	3.36	0.61	2.74	1.73	1.61	-1.75	6.48	5.70	2.07	-8.08	7.23	-1.10	0.79	4.17	-6.42	-0.15
1.63	5.85	5.18	3.36	5.27	0.82	2.08	4.62	4.13	5.17	3.44	1.70	2.41	0.77	2.36	4.66	2.39	2.92	4.95	2.79
1.46	0.40	0.02	0.61	0.82	1.11	1.15	1.80	0.50	4.35	-0.62	0.13	-0.29	-0.18	0.30	0.95	0.71	1.22	6.10	0.35
1.36	1.08	1.84	2.74	2.08	1.15	1.77	2.58	0.94	3.80	1.03	1.37	0.55	-1.82	1.90	1.23	1.09	2.22	4.46	0.66
2.18	3.57	2.64	1.73	4.62	1.80	2.58	5.72	2.26	8.35	0.57	0.58	1.45	0.43	1.13	4.71	2.57	3.15	11.17	2.56
4.89	12.53	6.23	1.61	4.13	0.50	0.94	2.26	10.86	9.65	4.60	0.64	0.98	9.20	0.46	5.86	2.10	1.15	8.47	3.11
9.07	11.03	1.75	-1.75	5.17	4.35	3.80	8.35	9.65	25.47	-2.17	-2.10	-1.11	11.01	-2.40	9.46	4.23	3.52	33.95	4.24
0.44	5.41	6.97	6.48	3.44	-0.62	1.03	0.57	4.60	-2.17	6.81	3.86	2.40	-1.71	4.68	0.98	0.78	2.06	-7.50	1.03
-0.05	0.43	3.74	5.70	1.70	0.13	1.37	0.58	0.64	-2.10	3.86	3.34	1.23	-4.95	4.22	-1.01	0.26	2.20	-5.20	-0.25
-0.58	1.90	2.84	2.07	2.41	-0.29	0.55	1.45	0.98	-1.11	2.40	1.23	1.71	-1.02	1.65	1.45	0.79	1.08	-2.60	1.11
4.25	11.16	-0.23	-8.08	0.77	-0.18	-1.82	0.43	9.20	11.01	-1.71	-4.95	-1.02	17.07	-6.62	6.85	1.27	-3.02	14.28	3.22
-0.14	0.25	4.65	7.23	2.36	0.30	1.90	1.13	0.46	-2.40	4.68	4.22	1.65	-6.62	5.37	-1.12	0.46	2.98	-6.00	-0.23
2.89	8.03	3.19	-1.10	4.66	0.95	1.23	4.71	5.86	9.46	0.98	-1.01	1.45	6.85	-1.12	6.72	2.64	1.40	11.85	3.62
1.30	2.89	1.81	0.79	2.39	0.71	1.09	2.57	2.10	4.23	0.78	0.26	0.79	1.27	0.46	2.64	1.28	1.35	5.18	1.45
1.30	1.42	2.99	4.17	2.92	1.22	2.22	3.15	1.15	3.52	2.06	2.20	1.08	-3.02	2.98	1.40	1.35	2.93	3.55	0.86
10.88	9.91	-2.37	-6.42	4.95	6.10	4.46	11.17	8.47	33.95	-7.50	-5.20	-2.60	14.28	-6.00	11.85	5.18	3.55	48.78	5.02
1.23	4.39	2.19	-0.15	2.79	0.35	0.66	2.56	3.11	4.24	1.03	-0.25	1.11	3.22	-0.23	3.62	1.45	0.86	5.02	2.03

Table 6.3: Example of a table of mean return μ_i and covariances σ_{ij}

day\\x_i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	var.	return
5	0	0	0	0	0	0	0	0	0	1.3	0	20	15.1	20	20	0	0	20	0	0	0.962696	0.788671
5	0	0	0	0	0	0	0	0	0	3.1	0	20	12.0	20	20	0	0	20	0	0	0.994871	0.799627
5	0	0	0	0	0	0	0	0	0	3.6	0	20	11.8	20	20	0	0	20	0	0	1.017500	0.804341
5	0	0	1.7	0	0	0	0	0	0	3.0	0	20	12.0	20	20	0	0	20	0	0	1.078990	0.812383
5	0	0	0	0	0	0	0	0	0	4.8	0	20	13.7	20	20	0	0	20	0	0	1.111420	0.821429
5	0	0	0	0	0	0	0	0	0	5.4	0	20	12.3	20	20	0	0	20	0	0	1.123690	0.823776
5	0	0	0	0	0	0	0	0	0	6.1	0	20	11.5	20	20	0	0	20	0	0	1.151900	0.828414
5	0	0	0	0	0	0	0	0	0	5.6	0	20	13.6	20	20	0	0	20	0	0	1.155620	0.829170
5	0	0	0	0	0	0	0	0	0	7.1	0	20	9.0	20	20	0	0	20	0	0	1.185270	0.832727
5	0	0	0	0	0	0	0	0	0	6.7	0	20	12.1	20	20	0	0	20	0	0	1.200290	0.836255
5	0	0	0	0	0	0	0	0	0	6.6	0	20	13.4	20	20	0	0	20	0	0	1.215770	0.838955
5	0	0	3.4	0	0	0	0	0	0	5.6	0	20	6.3	20	20	0	0	20	0	0	1.230710	0.838862
5	0	0	0	0	0	0	0	0	0	7.8	0	20	8.6	20	20	0	0	20	0	0	1.232530	0.839415
5	0	0	0	0	0	0	0	0	0	7.5	0	20	12.5	20	20	0	0	20	0	0	1.260490	0.845516
5	0	0	1.6	0	0	0	0	0	0	6.6	0	20	11.6	20	20	0	0	20	0	0	1.271040	0.846919
5	0	0	0	0	0	0	0	0	0	8.4	0	20	10.4	20	20	0	0	20	0	0	1.296070	0.849498
5	0	0	0.5	0	0	0	0	0	0	8.0	0	20	11.4	20	20	0	0	20	0	0	1.308080	0.852146
5	0	0	5.9	0	0	0	0	0	0	4.6	0	20	8.7	20	20	0	0	20	0	0	1.342850	0.854090
5	0	0	2.8	0	0	0	0	0	0	7.2	0	20	9.8	20	20	0	0	20	0	0	1.348560	0.858057
5	0	0	2.9	0	0	0	0	0	0	8.1	0	20	6.8	20	20	0	0	20	0	0	1.374120	0.860402
5	0	0	5.6	0	0	0	0	0	0	5.9	0	20	8.0	20	20	0	0	20	0	0	1.393930	0.863270
5	0	0	4.5	0	0	0	0	0	0	6.9	0	20	8.6	20	20	0	0	20	0	0	1.407580	0.866164
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 6.4: Example of a solution (the values are percent)

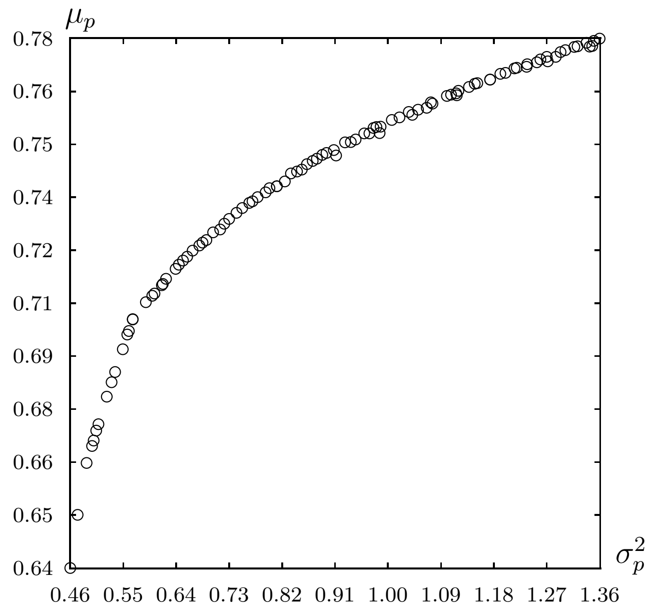


Figure 6.2: Example of the graph of a solution of POP

risk, the solution with the maximal risk, and a solution with a medium risk. Then we computed the return of for day 6 of these 3 solutions using equation 6.3, and we compared the return of these three solutions with the return of the IPyC of the BMV. The results are shown in Table 6.5.

To see how this would work in a real situation, we did an experiment beginning with "one unit" of investment (say, one peso) and following the corresponding wealth day-per-day; that is, every day we multiply the current value by $1 + r$ to obtain the value of our investment the day after. The results are shown in the Table 6.6 and their graphical representation appears in Figure 6.3. It can be seen that each of our three solutions gives a better return than the IPyC–in some cases the return is up to 8% above the IPyC return (day 72 of Table 6.6). Only in the last few days our solutions were similar to the IPyC–perhaps because the IPyC was behaving "optimally". For instance, from the Table 6.4 we can see that the IPyC return of 21.1% is very close to our solutions with minimal and maximum risks, 20.7% and 21.0%, respectively, but below the 24.6% given by our medium risk solution.

day	IPyC	min. risk	med. risk	max. risk	day	IPyC	min. risk	med. risk	max. risk
1	-0.24%	0.48%	0.23%	-0.21%	48	1.11%	1.24%	2.01%	2.45%
2	0.06%	0.70%	0.37%	0.25%	49	0.24%	-0.29%	-0.20%	-0.53%
3	-1.62%	-0.81%	-1.04%	-1.39%	50	1.13%	0.81%	0.52%	0.25%
4	0.49%	0.52%	0.42%	0.36%	51	0.50%	0.38%	0.80%	0.98%
5	0.43%	0.09%	0.09%	-0.05%	52	0.09%	-0.02%	-0.22%	-0.51%
6	-0.74%	-1.45%	-1.47%	-1.56%	53	0.11%	-0.23%	-0.22%	-0.13%
7	-0.61%	-0.58%	-0.75%	-0.41%	54	-0.05%	0.19%	-0.17%	-0.41%
8	1.05%	0.73%	0.83%	0.56%	55	1.09%	0.79%	1.09%	0.62%
9	0.58%	0.30%	0.39%	0.77%	56	0.48%	0.55%	1.21%	1.26%
10	-0.29%	-0.13%	0.04%	0.04%	57	0.28%	0.21%	0.49%	0.67%
11	0.48%	1.12%	1.16%	1.34%	58	0.42%	0.26%	0.22%	0.07%
12	0.81%	1.00%	1.12%	1.15%	59	0.13%	0.04%	0.43%	0.82%
13	0.53%	0.32%	0.59%	0.33%	60	0.72%	0.80%	0.53%	0.51%
14	-0.43%	0.28%	-0.05%	0.19%	61	0.91%	0.65%	0.88%	0.85%
15	1.44%	1.29%	2.00%	1.94%	62	-0.48%	-0.60%	-0.58%	-0.68%
16	1.53%	0.53%	0.17%	0.37%	63	-0.39%	-0.22%	-0.22%	-0.18%
17	-0.66%	-0.11%	-0.69%	-1.38%	64	0.81%	-0.41%	0.35%	0.21%
18	1.05%	0.77%	0.90%	1.38%	65	-1.92%	-1.66%	-1.65%	-2.47%
19	0.50%	0.41%	0.43%	-0.09%	66	-1.48%	-1.01%	-1.47%	-1.57%
20	-0.10%	-0.01%	-0.10%	-0.05%	67	0.88%	0.05%	0.28%	1.28%
21	1.32%	1.06%	1.61%	1.43%	68	-2.01%	-1.16%	-1.10%	-2.67%
22	0.83%	0.72%	0.85%	0.96%	69	-0.06%	-0.45%	-0.39%	-0.49%
23	-0.60%	-0.50%	-0.29%	-0.61%	70	-1.88%	-0.58%	-0.44%	-0.56%
24	-0.29%	0.00%	0.05%	-0.80%	71	0.91%	0.57%	0.76%	0.83%
25	0.23%	1.05%	0.11%	0.47%	72	1.07%	1.38%	1.22%	1.19%
26	0.06%	0.67%	0.92%	0.40%	73	1.83%	1.27%	1.45%	1.56%
27	1.49%	0.19%	0.64%	1.14%	74	0.96%	0.70%	0.61%	0.61%
28	-0.12%	-0.22%	-0.52%	-0.10%	75	1.63%	1.86%	1.83%	1.84%
29	-0.02%	0.21%	0.29%	0.46%	76	0.05%	0.47%	-0.01%	0.38%
30	-0.04%	0.05%	0.12%	0.05%	77	-2.08%	-1.65%	-2.05%	-2.00%
31	0.58%	0.89%	0.45%	0.56%	78	-0.76%	-1.04%	-1.20%	-1.11%
32	0.15%	0.82%	1.01%	0.96%	79	0.55%	0.57%	0.66%	0.82%
33	-1.69%	-1.07%	-1.44%	-1.75%	80	0.96%	0.48%	0.99%	0.33%
34	0.34%	0.32%	0.42%	0.46%	81	1.37%	0.98%	0.75%	1.26%
35	-0.03%	-0.40%	-0.10%	0.90%	82	-0.45%	-0.43%	-0.77%	-0.75%
36	0.26%	0.17%	0.12%	1.09%	83	0.42%	-0.14%	0.23%	0.25%
37	0.75%	0.83%	0.95%	1.09%	84	0.43%	0.45%	-0.03%	-0.35%
38	0.65%	1.07%	1.15%	1.21%	85	1.82%	1.12%	1.31%	1.30%
39	0.99%	1.71%	1.98%	1.37%	86	-0.01%	-0.31%	-0.02%	0.02%
40	-0.78%	-0.18%	-0.71%	-0.22%	87	0.75%	0.80%	0.91%	0.46%
41	1.07%	1.30%	1.57%	1.33%	88	0.05%	-1.11%	-0.88%	-0.77%
42	-0.97%	-0.66%	-0.70%	-0.69%	89	0.22%	1.17%	1.06%	0.86%
43	-0.05%	-0.65%	-0.29%	-0.27%	90	0.23%	-0.08%	-0.26%	-0.33%
44	0.66%	-0.31%	0.28%	0.19%	91	1.13%	0.46%	0.53%	0.72%
45	-0.59%	-0.46%	-0.20%	-0.01%	92	0.34%	0.47%	0.15%	-0.14%
46	-0.04%	0.65%	0.54%	0.15%	93	0.04%	-0.17%	0.04%	0.05%
47	0.09%	0.27%	0.59%	0.52%	94	-1.10%	-1.11%	-1.16%	-1.12%

Table 6.5: Table of comparison of return of IPyC, the solution with minimal risk, a medium risk and maximal risk.

day	IPyC	min. risk	med. risk	max. risk	day	IPyC	min. risk	med. risk	max. risk
0	1.000	1.000	1.000	1.000					
1	0.998	1.005	1.002	0.998	48	1.100	1.152	1.172	1.174
2	0.998	1.012	1.006	1.000	49	1.102	1.148	1.170	1.168
3	0.982	1.004	0.996	0.987	50	1.115	1.158	1.176	1.171
4	0.987	1.009	1.000	0.990	51	1.120	1.162	1.185	1.182
5	0.991	1.010	1.001	0.990	52	1.121	1.162	1.183	1.176
6	0.984	0.995	0.986	0.974	53	1.123	1.159	1.180	1.175
7	0.978	0.989	0.978	0.970	54	1.122	1.162	1.178	1.170
8	0.988	0.997	0.987	0.975	55	1.134	1.171	1.191	1.177
9	0.994	1.000	0.991	0.983	56	1.140	1.177	1.205	1.192
10	0.991	0.998	0.991	0.983	57	1.143	1.180	1.211	1.200
11	0.996	1.010	1.002	0.997	58	1.148	1.183	1.214	1.201
12	1.004	1.020	1.014	1.008	59	1.149	1.183	1.219	1.211
13	1.009	1.023	1.020	1.011	60	1.158	1.193	1.225	1.217
14	1.005	1.026	1.019	1.013	61	1.168	1.200	1.236	1.227
15	1.019	1.039	1.040	1.033	62	1.163	1.193	1.229	1.219
16	1.035	1.045	1.041	1.037	63	1.158	1.190	1.226	1.217
17	1.028	1.044	1.034	1.023	64	1.167	1.186	1.231	1.219
18	1.039	1.052	1.043	1.037	65	1.145	1.166	1.210	1.189
19	1.044	1.056	1.048	1.036	66	1.128	1.154	1.193	1.170
20	1.043	1.056	1.047	1.035	67	1.138	1.155	1.196	1.185
21	1.057	1.067	1.064	1.050	68	1.115	1.141	1.183	1.154
22	1.066	1.075	1.073	1.060	69	1.115	1.136	1.178	1.148
23	1.059	1.069	1.070	1.054	70	1.094	1.130	1.173	1.142
24	1.056	1.069	1.070	1.045	71	1.104	1.136	1.182	1.151
25	1.059	1.081	1.071	1.050	**72**	**1.115**	**1.152**	**1.196**	**1.165**
26	1.059	1.088	1.081	1.054	73	1.136	1.166	1.214	1.183
27	1.075	1.090	1.088	1.066	74	1.147	1.175	1.221	1.190
28	1.074	1.087	1.082	1.065	75	1.165	1.196	1.244	1.212
29	1.073	1.090	1.086	1.070	76	1.166	1.202	1.243	1.217
30	1.073	1.090	1.087	1.071	77	1.142	1.182	1.218	1.192
31	1.079	1.100	1.092	1.077	78	1.133	1.170	1.203	1.179
32	1.081	1.109	1.103	1.087	79	1.139	1.177	1.211	1.189
33	1.063	1.097	1.087	1.068	80	1.150	1.182	1.223	1.193
34	1.066	1.101	1.092	1.073	81	1.166	1.194	1.232	1.208
35	1.066	1.096	1.090	1.082	82	1.161	1.189	1.223	1.199
36	1.069	1.098	1.092	1.094	83	1.165	1.187	1.226	1.202
37	1.077	1.107	1.102	1.106	84	1.170	1.192	1.225	1.198
38	1.084	1.119	1.115	1.120	85	1.192	1.206	1.241	1.213
39	1.094	1.138	1.137	1.135	86	1.192	1.202	1.241	1.214
40	1.086	1.136	1.129	1.132	87	1.201	1.212	1.252	1.219
41	1.098	1.151	1.147	1.147	88	1.201	1.198	1.241	1.210
42	1.087	1.143	1.138	1.139	89	1.204	1.212	1.255	1.220
43	1.086	1.136	1.135	1.136	90	1.207	1.211	1.251	1.216
44	1.094	1.132	1.138	1.139	91	1.220	1.217	1.258	1.225
45	1.087	1.127	1.136	1.138	92	1.224	1.222	1.260	1.223
46	1.087	1.135	1.142	1.140	93	1.225	1.220	1.260	1.224
47	1.088	1.138	1.149	1.146	94	1.211	1.207	1.246	1.210

Table 6.6: Table of comparison of investment of IPyC, the solution with minimal risk, a medium risk and maximal risk.

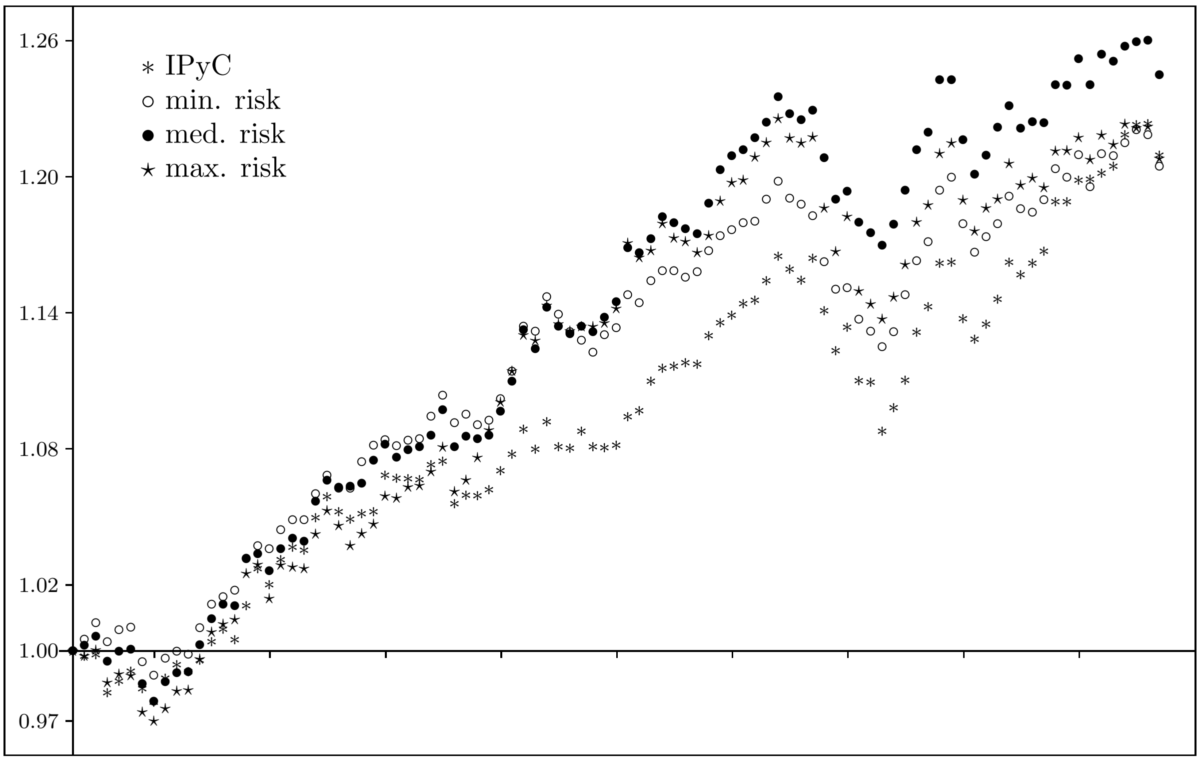


Figure 6.3: Graphic comparison of the IPyC and the solutions with minimal risk, a medium risk and maximal risk.

6.4 Conclusions and Future Work

In this chapter we applied our ST-MOPSO algorithm to the Markowitz' portfolio selection problem. As shown in Section 6.3 our results seem to be quite good. But of course before reaching any conclusions we need to do more experimental work. For instance, we do not really know how good are our 5-day moving averages. It would be interesting (and important!) to determine how sensitive our results are to the length of the moving averages.

Conclusions and Future Work

We have presented convergence proofs of three meta-heuristics that have been used for solving MOPs: simulated annealing, an artificial immune system (based on clonal selection theory), and a general evolutionary algorithm.

It is worth noting that in the case of the general MhAs, our convergence proof extends previous proofs of convergence of genetic algorithms for single-objective optimization (e.g., [40]). Actually, our proof is valid for a more general class of MhAs that use uniform mutation.

Regarding the artificial immune system, the proofs included here together with some of our previous work [57] constitute the only attempts currently known to prove convergence of such metaheuristic.

Finally, regarding simulated annealing, our proof relies on previous work by Laarhoven, Aarts and Korst [1, 25], but it constitutes (to the best of our knowledge) the first proof of convergence of simulated annealing in multiobjective optimization problems.

As part of our future work, we intend to extend these results to a more general case in which not even uniform mutation is required. We also plan to analyze other types of heuristics used for multiobjective optimization, and to try to determine bounds of convergence for such algorithms.

In the final section of each of the Chapters 2 to 6 we already mentioned other possible problems for further research.

Bibliography

[1] E.H. Aarts and J.H. Korst. *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing.* John Wiley & Sons, Chichester, UK, 1989.

[2] C. A. Coello Coello, D.A. Van Veldhuizen, and G.B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems.* Kluwer Academic Publishers, New York, May 2002. ISBN 0-3064-6762-3.

[3] C.A. Coello Coello and N. Cruz Cortés. Solving Multiobjective Optimization Problems using an Artificial Immune System. *Genetic Programming and Evolvable Machines*, 6(2):163–190, June 2005.

[4] C.A. Coello Coello, G. Toscano Pulido, and M. Salazar Lechuga. Handling Multiple Objectives With Particle Swarm Optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):256–279, June 2004.

[5] N. Cruz Cortés and C.A. Coello Coello. Multiobjective Optimization Using Ideas from the Clonal Selection Principle. In E. Cantú-Paz et al., editor, *Genetic and Evolutionary Computation—GECCO 2003. Proceedings, Part I*, pages 158–170. Springer. Lecture Notes in Computer Science Vol. 2723, July 2003.

[6] D. Dasgupta, editor. *Artificial Immune Systems and Their Applications.* Springer-Verlag, Berlin, 1999.

[7] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms.* John Wiley & Sons, Chichester, UK, 2001. ISBN 0-471-87339-X.

[8] K. Deb, M. Mohan, and S. Mishra. Towards a Quick Computation of Well-Spread Pareto-Optimal Solutions. In *EMO 2003*, pages 222–236, Faro, Portugal, April 2003. LNCS 2632.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.

[10] J.L. Doob. *Stochastic Processes.* Wiley, New York, 1953.

[11] K.A. Dowsland. Simulated Annealing. In C.R. Reeves, editor, *Modern Heuristic Techniques for Combinatorial Problems*, chapter 2, pages 20–69. John Wiley & Sons, 1993.

[12] R.C. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pages 39–43, Nagoya, Japan, October 1995.

[13] D.B. Fogel. *Evolutionary Computation. Toward a New Philosophy of Machine Intelligence.* The Institute of Electrical and Electronic Engineers, New York, 1995.

[14] L.J. Fogel. *Artificial Intelligence through Simulated Evolution.* Wiley, New York, 1966.

[15] L.J. Fogel. *Artificial Intelligence through Simulated Evolution. Forty Years of Evolutionary Programming.* Wiley, New York, 1999.

[16] C.M. Fonseca and P.J. Fleming. Multiobjective Optimization. In T. Bck, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, volume 1, pages C4.5:1–C4.5:9. Oxford University Press, 1997.

[17] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, Reading, MA, 1989.

[18] P. Hajela and J. Sun Yoo. Immune Network Modelling in Design Optimization. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 167–183. McGraw-Hill, 1999.

[19] O. Hernández-Lerma. *Adaptive Markov Control Processes.* Springer-Verlag, New York, 1989.

[20] X. Hui, R.C. Eberhart, and Y. Shi. Particle Swarm with Extended Memory for Multiobjective Optimization. In *2003 IEEE Swarm Intelligence Symposium Proceedings*, pages 193–197, Indianapolis, Indiana, USA, April 2003. IEEE Service Center.

[21] M. Iosifescu. *Finite Markov Processes and Their Applications.* Wiley, Chichester, UK, 1980.

[22] J. Kennedy and R.C. Eberhart. Particle swarm optimization. In *Proc. IEEE Int. Conf. Neural Networks*, pages 1942–1948, Piscataway, NJ, USA, 1995.

[23] S. Kirkpatrick, C.D. Gellatt, and M.P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983.

[24] R. Korn and E. Korn. *Option Pricing and Portfolio Optimization: Modern Methods of Financial Mathematics*, volume 31 of *Graduate Studies in Mathematics.* American Mathematical Society, Providence, Rhode Island, 2000.

[25] P. Laarhoven and E.H. Aarts. *Simulated Annealing: Theory and Applications*. D. Reidel, Boston, Massachusetts, 1987.

[26] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. *Evolutionary Computation*, 10(3):263–282, Fall 2002.

[27] G.F. Lawler. *Introduction to Stochastic Processes*. Chapman & Hall/CRC, Boca Raton, Florida, 1995.

[28] X. Li. Better Spread and Convergence: Particle Swarm Multiobjective Optimization Using the Maximin Fitness Function. In K. Deb et al., editor, *Genetic and Evolutionary Computation–GECCO 2004. Proceedings of the Genetic and Evolutionary Computation Conference. Part I*, pages 117–128, Seattle, Washington, USA, June 2004. Springer-Verlag, Lecture Notes in Computer Science Vol. 3102.

[29] M. Mahfouf, M.Y. Chen, and D.A. Linkens. Adaptive Weighted Particle Swarm Optimisation for Multi-objective Optimal Design of Alloy Steels. In *Parallel Problem Solving from Nature - PPSN VIII*, pages 762–771, Birmingham, UK, September 2004. Springer-Verlag. Lecture Notes in Computer Science Vol. 3242.

[30] H.M. Markowitz. Portfolio selection. *Journal of Finance*, 7(1):77–91, 1952.

[31] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[32] K.M. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, Massachusetts, 1998.

[33] S. Mostaghim and J. Teich. The role of ε-dominance in multi objective particle swarm optimization methods. In *CEC'2003*, volume 3, pages 1764–1771. IEEE Press, December 2003.

[34] S. Mostaghim and J. Teich. Strategies for Finding Good Local Guides in Multi-objective Particle Swarm Optimization (MOPSO). In *Proceedings of IEEE 2003 Swarm Intelligence Symposium*, pages 26–33, Indianapolis, IN, April 2003. IEEE Press.

[35] S. Mostaghim and J. Teich. Covering Pareto-optimal Fronts by Subswarms in Multi-objective Particle Swarm Optimization. In *2004 Congress on Evolutionary Computation (CEC'2004)*, volume 2, pages 1404–1411, Portland, Oregon, USA, June 2004. IEEE Service Center.

[36] L. Nunes de Castro and F. J. Von Zuben. Learning and Optimization Using the Clonal Selection Principle. *IEEE Transactions on Evolutionary Computation*, 6(3):239–251, 2002.

[37] Leandro Nunes de Castro and Jonathan Timmis. *An Introduction to Artificial Immune Systems: A New Computational Intelligence Paradigm*. Springer-Verlag, 2002.

[38] K.E. Parsopoulos, D.K. Tasoulis, and M.N. Vrahatis. Multiobjective Optimization Using Parallel Vector Evaluated Particle Swarm Optimization. In *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2004)*, volume 2, pages 823–828, Innsbruck, Austria, February 2004. ACTA Press.

[39] V. Černy. A Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, 1985.

[40] G. Rudolph. Convergence Analysis of Canonical Genetic Algorithms. *IEEE Transactions on Neural Networks*, 5:96–101, 1994.

[41] G. Rudolph. On a Multi-Objective Evolutionary Algorithm and Its Convergence to the Pareto Set. In *Proceedings of the 5th IEEE Conference on Evolutionary Computation*, pages 511–516, Piscataway, New Jersey, 1998. IEEE Press.

[42] G. Rudolph and A. Agapie. Convergence Properties of Some Multi-objective Evolutionary Algorithms. In *Proceedings of the 2000 Conference on Evolutionary Computation*, volume 2, pages 1010–1016, Piscataway, NJ, July 2000. IEEE Press.

[43] R. Saab, R. Monroy, and F. Godínez. Towards a model for an immune system. In C.A. Coello Coello, A. de Albornoz, L.E. Sucar, and O Cairó Battistutti, editors, *MICAI 2002: Advances in Artificial Intelligence. Second Mexican International Conference on Artificial Intelligence*, pages 401–410, Mérida, Yucatán, México, April 2002. Springer. Lecture Notes in Artificial Intelligence. Volume 2313.

[44] S.M. Sait and H. Youssef. *Iterative Computer Algorithms with Applications in Engineering. Solving Combinatorial Optimization Problems.* IEEE Computer Society, Los Alamitos, California, 1999.

[45] H.P. Schwefel. *Evolution and Optimum Seeking*. Wiley, New York, 1995.

[46] E. Seneta. *Non-Negative Matrices and Markov Chains*. Springer-Verlag, New York, second edition, 1981.

[47] P. Serafini. Simulated Annealing for Multiple Objective Optimization Problems. In G.H. Tzeng, H.F. Wang, U.P. Wen, and P.L. Yu, editors, *Proceedings of the Tenth International Conference on Multiple Criteria Decision Making: Expand and Enrich the Domains of Thinking and Application*, volume 1, pages 283–292, Berlin, 1994. Springer-Verlag.

[48] D. Srinivasan and T. Hou Seow. Particle Swarm Inspired Evolutionary Algorithm (PS-EA) for Multiobjective Optimization Problem. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, volume 4, pages 2292–2297, Canberra, Australia, December 2003. IEEE Press.

[49] A. Tarakanov and D. Dasgupta. A formal model of an artificial immune system. *BioSystems*, 55:151–158, 2000.

[50] G. Toscano Pulido and C.A. Coello Coello. Using Clustering Techniques to Improve the Performance of a Particle Swarm Optimizer. In K. Deb et al., editor, *Genetic and Evolutionary Computation–GECCO 2004. Proceedings of the Genetic and Evolutionary Computation Conference. Part I*, pages 225–237, Seattle, Washington, USA, June 2004. Springer-Verlag, Lecture Notes in Computer Science Vol. 3102.

[51] E.L. Ulungu. *Optimisation combinatoire multicritere: Determination de l'ensemble des solutions efficaces et methodes interactives*. PhD thesis, Faculté des Sciences, Université de Mons-Hainaut, Mons, Belgium, 1993.

[52] E.L. Ulungu, J. Teghem, and Ph. Fortemps. Heuristics for Multi-Objective Combinatorial Optimization by Simulated Annealing. In J. Gu, G. Chen, Q. Wei, and S. Wang, editors, *Multiple Criteria Decision Making: Theory and Applications. Proceedings of the 6th National Conference on Multiple Criteria Decision Making*, pages 228–238, Windsor, UK, 1995. Sci-Tech.

[53] E.L. Ulungu, J. Teghem, Ph. Fortemps, and D. Tuyttens. MOSA Method: A Tool for Solving Multiobjective Combinatorial Optimization Problems. *Journal of Multi-Criteria Decision Analysis*, 8(4):221–236, 1999.

[54] E.L. Ulungu, J. Teghem, and Ch. Ost. Efficiency of Interactive Multi-Objective Simulated Annealing Through a Case Study. *Journal of the Operational Research Society*, 49:1044–1050, 1998.

[55] D. A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, May 1999.

[56] D.A. Van Veldhuizen and G.B. Lamont. On Measuring Multiobjective Evolutionary Algorithm Performance. In *2000 Congress on Evolutionary Computation*, volume 1, pages 204–211, Piscataway, New Jersey, July 2000. IEEE Service Center.

[57] M. Villalobos-Arias, C. A. Coello Coello, and O. Hernández-Lerma. Convergence Analysis of a Multiobjective Artificial Immune System Algorithm. In *Artificial Immune Systems: Third International Conference (ICARIS 2004). Proceedings*, pages 226–235, Catania, Sicily, Italy, September 2004. Springer. Lecture Notes in Computer Science Vol. 3239.

[58] E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November 1999.

[59] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.

Appendix A

Tables of data for Chapter 6

This appendix contains the tables of all data considered in Chapter 6. These data were obtained from the "Mexican Stock Market"(Bolsa Mexicana de Valores). The values represent the closing prices, in mexican pesos (see Chapter 6 for details).

date	Alfa A	AmTel A1	Amxl	Bimbo A	Cemex CPO	Elektra	Femsaubd	Gcarso A1	Gfinburo	Gfnorte
28/09/2004	42.090	23.800	22.027	24.752	63.800	76.200	50.200	51.843	18.860	50.082
29/09/2004	42.880	24.420	22.226	24.655	64.720	76.790	50.620	52.787	19.300	52.799
30/09/2004	43.060	24.600	22.206	24.439	64.090	76.480	50.300	51.992	19.520	52.750
01/10/2004	43.480	24.890	22.756	25.203	64.800	76.750	50.830	52.250	19.820	53.319
04/10/2004	43.280	25.250	23.185	25.350	65.760	76.400	50.870	52.558	20.000	52.888
05/10/2004	43.100	24.600	22.956	25.340	65.470	76.610	51.040	52.648	19.600	52.780
06/10/2004	42.860	24.300	22.526	25.144	67.140	76.690	51.140	52.518	19.080	52.721
07/10/2004	42.990	24.310	22.506	25.291	66.580	78.000	51.010	52.379	19.360	52.966
08/10/2004	42.150	23.810	22.007	24.214	65.120	79.500	50.920	52.131	19.000	51.760
11/10/2004	42.610	24.100	22.057	24.567	65.000	81.000	50.910	52.210	19.200	51.877
12/10/2004	42.210	24.160	22.157	24.322	64.560	81.030	50.410	52.677	19.230	51.701
13/10/2004	41.460	24.400	22.127	24.576	62.960	81.120	50.430	51.982	19.300	51.230
14/10/2004	40.640	24.200	21.927	24.175	64.000	81.010	50.700	51.962	19.100	51.620
15/10/2004	41.440	24.400	22.386	24.420	64.640	82.000	50.700	51.445	19.000	51.200
18/10/2004	41.590	24.950	22.616	24.371	64.100	82.520	50.500	52.180	19.270	50.750
19/10/2004	41.260	25.150	22.606	24.449	63.430	83.490	49.990	52.578	19.200	50.980
20/10/2004	41.600	25.460	22.896	24.478	62.980	85.000	50.200	52.727	19.300	51.310
21/10/2004	42.320	25.440	22.676	24.469	64.780	86.400	50.400	53.463	19.380	52.820
22/10/2004	42.540	25.510	22.636	24.958	65.740	88.620	50.400	53.671	19.620	53.060
25/10/2004	43.280	25.260	22.396	24.801	65.080	89.990	50.600	53.771	19.800	53.780
26/10/2004	43.390	26.490	23.665	24.890	66.100	92.760	50.500	54.019	19.750	54.030
27/10/2004	43.440	27.750	25.073	24.860	66.880	92.770	50.410	53.473	19.800	53.500
28/10/2004	44.300	27.330	24.694	25.350	66.080	90.700	50.560	53.174	19.400	53.930
29/10/2004	44.850	27.890	25.443	25.438	66.680	90.920	50.980	54.546	19.290	54.200
01/11/2004	45.010	27.970	25.523	26.456	66.820	91.500	50.720	53.373	19.400	54.490
02/11/2004	44.890	27.930	25.093	26.827	66.570	91.850	50.440	53.254	19.330	54.400
03/11/2004	46.140	28.310	25.683	27.385	67.500	92.650	50.990	52.926	19.500	54.950
04/11/2004	48.540	28.700	25.992	27.434	68.130	92.500	52.020	54.616	19.520	54.850
05/11/2004	48.520	28.350	25.413	27.336	67.930	92.700	51.970	53.801	19.590	55.090
08/11/2004	48.310	27.980	25.083	27.307	68.800	92.660	52.180	54.039	19.570	55.040
09/11/2004	49.000	27.870	25.263	27.591	68.630	92.230	52.180	55.510	19.550	54.860
10/11/2004	49.490	27.890	25.123	27.806	67.470	92.200	52.560	55.749	19.650	56.730

Table A.1: Table of prices.

date	Alfa A	AmTel A1	Amxl	Bimbo A	Cemex CPO	Elektra	Femsaubd	Gcarso A1	Gfinburo	Gfnorte
11/11/2004	50.600	28.240	25.483	27.591	68.910	92.270	52.790	56.454	20.200	58.200
12/11/2004	50.600	28.050	25.333	27.630	69.230	92.000	52.350	56.653	20.490	57.220
15/11/2004	50.120	27.860	25.133	27.990	70.100	92.250	51.880	57.150	20.730	58.340
16/11/2004	49.850	27.620	25.113	27.750	69.990	92.600	52.000	57.490	20.890	58.150
17/11/2004	50.990	27.900	25.133	27.800	70.620	92.750	51.700	56.790	21.000	60.250
18/11/2004	51.790	27.600	25.233	27.690	70.540	93.990	51.610	57.400	21.000	60.650
19/11/2004	51.410	27.100	24.594	27.340	70.180	93.500	51.130	56.990	20.680	58.360
22/11/2004	51.500	27.210	24.774	27.500	70.120	94.110	51.400	56.850	20.700	58.740
23/11/2004	53.760	27.020	24.604	27.270	71.080	95.040	51.360	55.790	20.690	59.710
24/11/2004	55.740	27.090	24.744	27.250	71.180	95.300	51.900	55.900	20.700	59.250
25/11/2004	56.870	27.000	25.043	27.330	71.910	96.380	51.900	56.200	20.940	59.980
26/11/2004	57.210	27.150	25.373	27.480	71.870	98.500	52.110	56.690	20.980	60.000
29/11/2004	57.040	27.870	26.052	27.650	71.600	104.200	53.590	55.790	21.070	59.870
30/11/2004	55.620	27.900	26.082	27.010	72.230	105.740	53.700	54.950	20.950	63.020
01/12/2004	57.050	28.430	26.322	27.020	74.710	108.420	54.130	55.200	20.990	62.200
02/12/2004	56.570	28.040	25.883	27.560	74.240	108.020	55.200	54.490	20.830	62.600
03/12/2004	55.830	28.360	26.192	27.920	73.900	107.700	54.900	53.900	20.800	62.530
06/12/2004	54.860	28.590	26.991	28.390	74.600	107.300	55.090	53.900	20.800	61.550
07/12/2004	53.610	28.800	26.931	28.400	74.500	107.000	55.400	53.370	20.200	61.110
08/12/2004	55.110	28.500	26.692	28.400	74.970	105.040	55.900	53.410	20.000	62.780
09/12/2004	55.690	28.500	26.422	28.350	75.020	103.130	56.440	54.420	20.720	62.780
10/12/2004	57.230	28.810	26.872	28.390	76.990	105.000	56.820	54.590	21.000	63.950
13/12/2004	57.350	28.990	27.101	28.390	76.170	105.850	56.500	55.500	20.640	63.880
14/12/2004	57.640	29.700	28.140	28.530	76.290	105.090	56.860	56.200	20.750	63.280
15/12/2004	57.560	29.670	28.100	28.500	77.200	105.900	57.680	56.000	20.960	64.850
16/12/2004	57.340	29.760	28.060	28.500	78.240	105.910	59.020	56.080	20.810	65.800
17/12/2004	57.060	29.950	28.540	28.600	78.160	104.500	58.360	56.000	20.970	65.670
20/12/2004	56.700	29.810	28.330	28.720	78.610	105.000	58.230	56.000	20.840	66.200
21/12/2004	56.450	30.100	28.730	28.780	79.960	104.890	58.610	56.890	20.990	66.250
22/12/2004	56.530	30.260	28.730	28.710	80.100	104.300	58.900	57.320	20.750	68.200
23/12/2004	57.690	30.300	28.780	28.520	80.850	103.910	58.910	57.360	20.600	68.460
24/12/2004	57.800	30.500	28.990	28.460	81.040	105.440	58.900	57.500	20.900	68.500
27/12/2004	57.590	30.600	29.000	28.280	80.620	103.560	58.850	59.000	20.990	69.810
28/12/2004	57.640	31.000	29.190	28.200	81.490	103.310	58.280	58.990	21.000	69.790
29/12/2004	57.540	32.470	29.460	28.800	81.910	103.490	59.600	59.950	20.980	69.270
30/12/2004	57.170	32.010	29.300	28.590	81.490	103.400	59.170	60.030	20.570	70.240
31/12/2004	57.000	31.920	29.120	28.160	81.230	103.500	58.510	60.000	20.510	70.140
03/01/2005	57.390	32.600	30.050	28.300	81.390	100.770	59.250	59.920	20.950	69.070
04/01/2005	56.410	31.140	28.970	28.390	81.320	95.810	59.880	59.460	20.800	67.110
05/01/2005	55.140	30.420	28.400	28.490	79.260	94.280	59.270	58.160	20.470	65.950
06/01/2005	55.780	30.950	28.590	28.740	79.760	95.280	58.870	59.280	21.130	66.070
07/01/2005	54.880	29.910	27.900	28.860	78.190	95.510	57.700	58.000	20.580	65.460
10/01/2005	53.460	29.940	27.570	28.010	79.300	93.660	57.980	57.930	20.510	64.930
11/01/2005	51.540	28.820	26.820	28.190	78.870	92.000	57.020	56.680	20.420	64.000
12/01/2005	51.900	29.270	27.030	28.680	80.230	91.000	57.230	56.940	20.670	64.630
13/01/2005	53.390	29.510	27.510	29.810	80.810	91.940	57.700	57.470	20.650	67.870
14/01/2005	55.070	30.500	28.350	29.750	83.110	94.300	59.350	57.460	20.850	69.620
17/01/2005	56.250	30.980	28.600	29.800	84.680	95.940	59.300	58.700	21.000	70.130
18/01/2005	56.900	32.180	29.020	31.200	84.200	96.580	60.090	61.190	21.950	70.000
19/01/2005	56.380	32.970	29.160	30.920	82.900	96.590	60.500	64.000	22.710	69.980
20/01/2005	55.080	31.750	28.090	30.250	81.650	95.960	59.890	62.500	22.390	69.500
21/01/2005	53.770	31.200	27.830	29.720	81.300	96.450	59.500	60.520	22.500	68.760
24/01/2005	53.850	31.750	27.730	30.720	80.610	97.060	59.250	60.500	23.090	68.150
25/01/2005	55.040	32.290	28.460	30.700	81.360	96.250	58.930	61.330	22.850	68.330
26/01/2005	56.140	33.090	29.210	30.790	82.550	95.380	59.250	61.740	23.040	69.970
27/01/2005	55.680	32.530	28.810	30.710	83.660	95.500	59.390	61.310	22.580	69.960
28/01/2005	56.910	32.750	29.520	30.700	83.540	95.400	59.460	61.210	22.360	70.660
31/01/2005	57.570	32.650	29.600	31.250	83.890	95.450	60.140	62.590	22.460	71.970
01/02/2005	58.450	33.130	30.320	32.340	86.160	95.790	61.500	62.780	22.620	72.330
02/02/2005	58.150	33.010	30.550	33.520	86.550	95.460	61.450	61.940	22.750	71.740
03/02/2005	58.800	33.630	30.900	34.600	85.940	95.800	61.780	62.170	23.330	72.110
04/02/2005	58.140	33.850	31.220	33.550	86.020	96.000	62.710	62.000	22.850	71.680
07/02/2005	59.460	33.920	31.440	34.470	85.960	94.850	62.870	61.920	22.730	71.500
08/02/2005	58.400	34.050	31.140	33.950	86.960	94.120	63.080	61.440	22.360	71.980
09/02/2005	59.070	34.680	31.840	33.810	86.820	94.620	62.890	61.510	22.400	71.450
10/02/2005	62.440	34.770	31.310	33.900	86.710	94.940	63.500	61.550	22.310	71.900
11/02/2005	62.130	34.510	31.270	33.950	87.180	96.280	63.390	61.540	22.260	71.920
14/02/2005	60.860	33.850	31.050	33.420	86.990	95.790	62.940	61.330	22.050	71.470

Table A.2: Table of prices (cont.).

	Gmexico	Gmodelo	Kimber	Peñoles	Soriana	Telecom	Telmex	Televisa	Vitro	Walmex
date	B	C	A		B	A1		CPO	A	V
28/09/2004	45.960	27.490	32.628	49.580	37.000	16.650	18.152	29.850	10.640	38.390
29/09/2004	45.900	27.580	32.667	51.420	37.000	17.010	18.320	30.150	11.270	38.800
30/09/2004	45.950	27.670	32.521	53.000	36.910	16.910	18.221	29.940	11.070	38.660
01/10/2004	46.960	28.370	32.688	52.740	37.900	16.870	18.271	30.430	10.920	38.650
04/10/2004	47.660	28.700	33.093	53.490	38.290	17.040	18.340	30.590	10.600	38.960
05/10/2004	47.160	28.910	33.044	53.000	38.510	16.870	18.301	30.750	10.550	38.750
06/10/2004	47.390	28.870	33.034	53.510	38.570	16.800	18.241	31.200	10.460	38.610
07/10/2004	47.400	28.800	32.995	54.890	38.690	16.700	18.152	31.050	10.630	38.890
08/10/2004	48.190	28.610	32.935	53.630	39.080	16.330	17.983	30.240	10.510	38.120
11/10/2004	48.510	28.800	32.955	53.000	39.000	16.400	18.073	30.450	10.530	38.340
12/10/2004	48.000	28.790	32.856	52.700	39.700	16.630	18.182	30.720	10.480	39.050
13/10/2004	45.720	28.850	32.590	51.470	39.500	16.400	18.231	30.510	10.450	38.630
14/10/2004	45.520	28.850	32.560	51.500	39.000	16.350	18.201	30.330	10.310	37.870
15/10/2004	46.260	29.070	32.570	51.500	39.200	16.510	18.489	30.600	10.500	38.570
18/10/2004	45.990	29.150	32.787	51.500	39.260	16.990	18.588	30.650	10.350	38.680
19/10/2004	45.750	29.020	32.985	52.000	38.580	16.930	18.519	30.700	10.350	38.430
20/10/2004	45.610	29.180	32.886	51.990	37.940	17.000	18.876	30.570	10.330	38.180
21/10/2004	46.110	29.270	32.935	51.710	38.310	17.540	19.193	31.230	10.770	38.090
22/10/2004	48.600	29.220	32.748	51.530	38.700	17.680	19.223	31.570	10.550	38.020
25/10/2004	49.210	29.170	32.629	52.400	38.690	17.500	18.985	31.010	10.500	38.140
26/10/2004	49.450	29.200	32.886	51.310	39.080	17.700	19.242	31.130	10.280	37.600
27/10/2004	48.570	29.680	33.083	51.920	38.700	17.750	19.460	31.470	10.210	37.910
28/10/2004	47.430	29.600	33.330	52.850	37.290	17.650	19.510	31.400	10.250	37.660
29/10/2004	47.660	29.570	34.091	52.710	37.070	17.750	19.619	31.770	10.300	37.750
01/11/2004	48.260	29.590	34.792	51.670	37.940	17.880	19.946	32.070	10.250	37.740
02/11/2004	47.760	30.010	34.861	51.350	38.160	17.990	19.946	32.090	10.390	38.100
03/11/2004	47.660	29.950	34.851	51.220	38.400	18.480	20.244	32.300	10.450	38.690
04/11/2004	48.820	30.230	34.525	52.850	38.240	18.600	20.244	32.490	10.390	38.700
05/11/2004	49.880	30.120	34.486	53.510	38.100	18.760	20.263	32.030	10.060	38.370
08/11/2004	50.550	30.000	34.644	53.850	38.120	18.600	20.045	31.440	10.200	38.600
09/11/2004	51.000	30.050	35.049	53.600	37.800	18.500	20.006	31.960	10.460	38.630
10/11/2004	51.990	30.090	35.197	53.710	37.800	18.620	20.016	31.900	10.670	38.700
11/11/2004	52.560	30.260	35.305	53.520	38.090	19.000	20.501	32.810	10.830	39.100
12/11/2004	52.180	30.170	35.157	53.500	38.240	18.840	20.392	32.960	10.610	39.550
15/11/2004	52.390	30.080	35.256	53.680	38.400	18.970	20.422	33.120	10.570	38.950
16/11/2004	52.460	29.910	35.207	54.440	39.090	18.900	20.353	33.150	10.490	38.830
17/11/2004	53.870	29.890	35.157	56.350	39.650	19.000	20.382	33.100	10.360	38.990
18/11/2004	53.970	29.570	35.256	58.940	39.600	19.000	20.293	33.510	10.290	38.890
19/11/2004	52.990	29.010	34.871	59.000	38.500	18.480	19.867	33.440	10.100	38.470
22/11/2004	53.360	29.100	35.207	59.000	39.150	18.480	19.857	33.470	10.150	38.590
23/11/2004	52.950	29.450	35.296	58.000	38.700	18.000	19.788	33.740	10.190	38.790
24/11/2004	53.870	29.550	35.839	58.580	38.430	18.090	19.480	33.910	10.220	38.820
25/11/2004	54.170	29.750	35.997	59.180	38.640	18.280	19.510	34.020	10.540	38.980
26/11/2004	54.360	29.580	36.342	59.270	39.010	18.390	19.629	34.860	10.430	39.100
29/11/2004	54.700	29.580	37.034	60.600	39.160	18.640	19.679	34.910	10.620	39.330
30/11/2004	54.260	29.430	37.034	58.620	38.600	17.810	19.441	34.870	10.350	38.530
01/12/2004	54.600	29.600	37.320	60.750	38.900	18.220	19.609	34.860	10.580	38.780
02/12/2004	52.640	29.180	37.330	59.120	38.240	18.040	19.540	34.240	10.540	38.100
03/12/2004	52.520	29.180	37.300	59.580	38.300	17.810	19.431	33.550	10.450	38.120
06/12/2004	52.280	29.240	37.700	58.500	38.100	17.840	19.679	33.420	10.600	38.100
07/12/2004	51.540	29.450	37.870	57.070	37.630	17.330	19.351	33.560	10.570	38.040

Table A.3: Table of prices (cont.).

date	Gmexico B	Gmodelo C	Kimber A	Peñoles	Soriana B	Telecom A1	Telmex	Televisa CPO	Vitro A	Walmex V
08/12/2004	50.640	29.710	37.400	55.440	37.890	17.400	19.322	33.450	11.000	38.010
09/12/2004	50.330	29.750	37.210	56.620	37.710	17.420	19.441	32.880	11.320	38.170
10/12/2004	51.890	30.150	36.470	58.380	38.500	17.540	19.480	32.710	12.450	38.380
13/12/2004	53.000	30.140	36.510	58.300	38.410	17.650	19.510	33.180	12.330	38.330
14/12/2004	53.220	30.000	36.610	58.000	38.500	18.100	19.590	33.570	12.250	38.860
15/12/2004	54.190	29.830	36.540	60.120	39.240	18.200	19.670	33.440	12.900	38.880
16/12/2004	54.340	29.960	36.550	59.000	38.800	18.490	19.800	33.320	12.780	38.340
17/12/2004	54.090	30.200	36.540	59.000	38.850	18.280	19.740	33.140	12.700	38.210
20/12/2004	54.710	30.090	36.530	58.770	38.950	18.320	19.760	33.000	12.350	38.460
21/12/2004	56.000	30.200	36.750	60.250	38.340	18.610	20.320	33.110	11.880	38.700
22/12/2004	55.930	30.130	37.500	60.500	38.300	19.010	20.940	32.960	12.000	38.500
23/12/2004	56.010	30.530	37.470	59.510	38.450	19.250	21.080	33.060	12.000	38.330
24/12/2004	56.190	30.550	37.750	60.240	38.830	19.200	21.080	33.190	12.100	38.390
27/12/2004	56.510	30.790	37.500	60.000	38.980	19.430	21.290	33.090	11.990	38.150
28/12/2004	56.440	30.790	37.810	60.140	39.490	19.710	21.540	34.170	11.900	38.250
29/12/2004	56.270	31.180	38.400	59.900	40.300	19.910	21.620	34.240	11.800	38.410
30/12/2004	56.720	31.000	38.390	59.900	40.020	19.920	21.570	33.750	11.850	38.250
31/12/2004	56.220	30.660	38.500	60.000	39.980	19.850	21.420	33.640	11.620	38.290
03/01/2005	56.700	31.030	38.470	59.610	39.790	19.910	21.420	33.710	11.750	38.240
04/01/2005	53.920	31.000	37.810	57.740	38.850	19.540	21.230	33.040	10.930	38.260
05/01/2005	54.000	31.220	37.030	58.480	39.250	19.050	20.830	32.770	10.890	38.080
06/01/2005	54.610	31.100	36.900	57.860	39.200	19.560	20.870	33.070	11.210	38.290
07/01/2005	54.500	30.070	36.400	57.330	38.370	18.760	20.510	32.430	11.130	38.050
10/01/2005	54.610	30.170	36.020	57.880	38.500	18.900	20.640	32.620	11.020	38.060
11/01/2005	54.130	29.710	35.500	57.710	37.520	18.350	20.280	31.670	10.800	37.860
12/01/2005	54.150	30.180	36.810	57.710	38.000	18.470	20.380	32.240	11.120	37.960
13/01/2005	54.180	30.170	36.520	57.590	38.330	18.650	20.630	32.100	11.290	38.370
14/01/2005	54.710	30.310	36.390	57.750	38.720	19.050	20.760	33.040	11.480	38.780
17/01/2005	54.540	30.320	36.400	57.750	38.750	19.360	20.870	33.400	11.400	38.940
18/01/2005	56.000	30.450	36.390	58.960	39.240	19.850	21.110	33.950	11.480	39.400
19/01/2005	55.190	30.400	36.200	57.730	38.760	20.120	20.830	33.920	11.350	38.920
20/01/2005	53.770	30.250	35.700	55.360	38.590	19.800	20.720	33.180	11.240	38.210
21/01/2005	54.030	30.190	35.880	56.000	38.360	19.800	20.470	32.490	11.520	38.020
24/01/2005	54.800	29.880	36.150	56.990	38.930	20.150	20.680	32.550	11.300	38.030
25/01/2005	55.210	30.120	36.500	57.620	39.100	20.230	20.930	32.640	12.300	38.380
26/01/2005	56.490	30.100	36.500	58.170	39.250	20.420	20.970	33.490	12.930	38.750
27/01/2005	56.580	30.010	36.340	59.000	39.040	20.390	20.980	33.220	12.220	38.370
28/01/2005	55.990	30.460	36.240	59.480	39.150	20.370	20.960	32.760	12.400	38.220
31/01/2005	56.390	30.210	36.310	58.700	39.580	20.380	20.890	33.010	12.210	38.620
01/02/2005	57.190	30.280	36.190	57.720	39.560	20.620	21.270	34.040	12.150	39.270
02/02/2005	57.480	30.060	36.090	57.180	39.700	20.320	21.200	34.200	12.190	38.920
03/02/2005	56.840	30.340	36.010	57.580	39.600	20.700	21.480	34.250	12.220	39.380
04/02/2005	56.480	30.670	36.270	57.040	39.000	20.220	21.490	34.630	12.240	39.400
07/02/2005	57.000	30.410	36.180	56.200	39.200	20.150	21.520	35.030	11.890	39.050
08/02/2005	56.820	30.740	36.270	55.920	39.580	20.290	21.570	35.080	12.230	40.360
09/02/2005	56.770	30.820	37.350	56.500	41.300	20.740	22.360	34.940	12.030	40.870
10/02/2005	57.400	31.000	37.230	58.990	42.490	20.800	22.370	35.580	12.150	41.290
11/02/2005	57.140	31.600	37.330	60.430	42.800	20.880	22.490	35.490	12.130	41.190
14/02/2005	56.590	31.900	37.240	58.020	41.470	20.490	22.430	34.940	12.010	40.800

Table A.4: Table of prices (cont.).

	Alfa	AmTel	Amxl	Bimbo	Cemex	Elektra	Femsaubd	Gcarso	Gfinburo	Gfnorte
date	A	A1		A	CPO			A1		
29/09/2004	1.877	2.605	0.907	-0.395	1.442	0.774	0.837	1.821	2.333	5.425
30/09/2004	0.420	0.737	-0.090	-0.873	-0.973	-0.404	-0.632	-1.506	1.140	-0.093
01/10/2004	0.975	1.179	2.474	3.124	1.108	0.353	1.054	0.497	1.537	1.078
04/10/2004	-0.460	1.446	1.888	0.583	1.481	-0.456	0.079	0.590	0.908	-0.809
05/10/2004	-0.416	-2.574	-0.991	-0.039	-0.441	0.275	0.334	0.170	-2.000	-0.204
06/10/2004	-0.557	-1.220	-1.871	-0.772	2.551	0.104	0.196	-0.245	-2.653	-0.112
07/10/2004	0.303	0.041	-0.089	0.584	-0.834	1.708	-0.254	-0.265	1.468	0.465
08/10/2004	-1.954	-2.057	-2.219	-4.257	-2.193	1.923	-0.176	-0.474	-1.860	-2.278
11/10/2004	1.091	1.218	0.227	1.455	-0.184	1.887	-0.020	0.153	1.053	0.227
12/10/2004	-0.939	0.249	0.453	-0.996	-0.677	0.037	-0.982	0.895	0.156	-0.340
13/10/2004	-1.777	0.993	-0.135	1.046	-2.478	0.111	0.040	-1.321	0.364	-0.911
14/10/2004	-1.978	-0.820	-0.903	-1.633	1.652	-0.136	0.535	-0.038	-1.036	0.761
15/10/2004	1.969	0.826	2.096	1.012	1.000	1.222	0.000	-0.995	-0.524	-0.814
18/10/2004	0.362	2.254	1.026	-0.200	-0.835	0.634	-0.394	1.430	1.421	-0.879
19/10/2004	-0.793	0.802	-0.044	0.321	-1.045	1.175	-1.010	0.762	-0.363	0.453
20/10/2004	0.824	1.233	1.281	0.120	-0.709	1.809	0.420	0.284	0.521	0.647
21/10/2004	1.731	-0.079	-0.960	-0.040	2.858	1.647	0.398	1.395	0.415	2.943
22/10/2004	0.520	0.275	-0.176	2.000	1.482	2.569	0.000	0.390	1.238	0.454
25/10/2004	1.740	-0.980	-1.059	-0.627	-1.004	1.546	0.397	0.185	0.917	1.357
26/10/2004	0.254	4.869	5.665	0.355	1.567	3.078	-0.198	0.462	-0.253	0.465
27/10/2004	0.115	4.757	5.952	-0.118	1.180	0.011	-0.178	-1.012	0.253	-0.981
28/10/2004	1.980	-1.514	-1.514	1.968	-1.196	-2.231	0.298	-0.558	-2.020	0.804
29/10/2004	1.242	2.049	3.034	0.348	0.908	0.243	0.831	2.579	-0.567	0.501
01/11/2004	0.357	0.287	0.314	4.002	0.210	0.638	-0.510	-2.150	0.570	0.535
02/11/2004	-0.267	-0.143	-1.683	1.406	-0.374	0.383	-0.552	-0.223	-0.361	-0.165
03/11/2004	2.785	1.361	2.349	2.080	1.397	0.871	1.090	-0.616	0.879	1.011
04/11/2004	5.202	1.378	1.206	0.179	0.933	-0.162	2.020	3.192	0.103	-0.182
05/11/2004	-0.041	-1.220	-2.229	-0.357	-0.294	0.216	-0.096	-1.492	0.359	0.438
08/11/2004	-0.433	-1.305	-1.297	-0.107	1.281	-0.043	0.404	0.443	-0.102	-0.091
09/11/2004	1.428	-0.393	0.717	1.039	-0.247	-0.464	0.000	2.722	-0.102	-0.327
10/11/2004	1.000	0.072	-0.554	0.780	-1.690	-0.033	0.728	0.430	0.512	3.409
11/11/2004	2.243	1.255	1.431	-0.774	2.134	0.076	0.438	1.266	2.799	2.591
12/11/2004	0.000	-0.673	-0.588	0.142	0.464	-0.293	-0.833	0.352	1.436	-1.684
15/11/2004	-0.949	-0.677	-0.789	1.303	1.257	0.272	-0.898	0.877	1.171	1.957
16/11/2004	-0.539	-0.861	-0.079	-0.857	-0.157	0.379	0.231	0.595	0.772	-0.326
17/11/2004	2.287	1.014	0.080	0.180	0.900	0.162	-0.577	-1.218	0.527	3.611
18/11/2004	1.569	-1.075	0.397	-0.396	-0.113	1.337	-0.174	1.074	0.000	0.664
19/11/2004	-0.734	-1.812	-2.534	-1.264	-0.510	-0.521	-0.930	-0.714	-1.524	-3.776
22/11/2004	0.175	0.406	0.731	0.585	-0.085	0.652	0.528	-0.246	0.097	0.651
23/11/2004	4.388	-0.698	-0.685	-0.836	1.369	0.988	-0.078	-1.865	-0.048	1.651
24/11/2004	3.683	0.259	0.568	-0.073	0.141	0.274	1.051	0.197	0.048	-0.770
25/11/2004	2.027	-0.332	1.211	0.294	1.026	1.133	0.000	0.537	1.159	1.232
26/11/2004	0.598	0.556	1.316	0.549	-0.056	2.200	0.405	0.872	0.191	0.033
29/11/2004	-0.297	2.652	2.677	0.619	-0.376	5.787	2.840	-1.588	0.429	-0.217
30/11/2004	-2.489	0.108	0.115	-2.315	0.880	1.478	0.205	-1.506	-0.570	5.261
01/12/2004	2.571	1.900	0.919	0.037	3.433	2.535	0.801	0.455	0.191	-1.301
02/12/2004	-0.841	-1.372	-1.670	1.999	-0.629	-0.369	1.977	-1.286	-0.762	0.643
03/12/2004	-1.308	1.141	1.196	1.306	-0.458	-0.296	-0.543	-1.083	-0.144	-0.112
06/12/2004	-1.737	0.811	3.051	1.683	0.947	-0.371	0.346	0.000	0.000	-1.567
07/12/2004	-2.279	0.735	-0.222	0.035	-0.134	-0.280	0.563	-0.983	-2.885	-0.715
08/12/2004	2.798	-1.042	-0.890	0.000	0.631	-1.832	0.903	0.075	-0.990	2.733
09/12/2004	1.052	0.000	-1.010	-0.176	0.067	-1.818	0.966	1.891	3.600	0.000

Table A.5: Table of returns

date	Alfa A	AmTel A1	Amxl	Bimbo A	Cemex CPO	Elektra	Femsaubd	Gcarso A1	Gfinburo	Gfnorte
10/12/2004	2.765	1.088	1.701	0.141	2.626	1.813	0.673	0.312	1.351	1.864
13/12/2004	0.210	0.625	0.855	0.000	-1.065	0.810	-0.563	1.667	-1.714	-0.109
14/12/2004	0.506	2.449	3.833	0.493	0.158	-0.718	0.637	1.261	0.533	-0.939
15/12/2004	-0.139	-0.101	-0.142	-0.105	1.193	0.771	1.442	-0.356	1.012	2.481
16/12/2004	-0.382	0.303	-0.142	0.000	1.347	0.009	2.323	0.143	-0.716	1.465
17/12/2004	-0.488	0.638	1.709	0.351	-0.102	-1.331	-1.118	-0.143	0.769	-0.198
20/12/2004	-0.631	-0.467	-0.735	0.420	0.576	0.478	-0.223	0.000	-0.620	0.807
21/12/2004	-0.441	0.973	1.412	0.209	1.717	-0.105	0.653	1.589	0.720	0.076
22/12/2004	0.142	0.532	0.000	-0.243	0.175	-0.562	0.495	0.756	-1.143	2.943
23/12/2004	2.052	0.132	0.174	-0.662	0.936	-0.374	0.017	0.070	-0.723	0.381
24/12/2004	0.191	0.660	0.730	-0.210	0.235	1.472	-0.017	0.244	1.456	0.058
27/12/2004	-0.363	0.328	0.034	-0.632	-0.518	-1.783	-0.085	2.609	0.431	1.912
28/12/2004	0.087	1.307	0.655	-0.283	1.079	-0.241	-0.969	-0.017	0.048	-0.029
29/12/2004	-0.173	4.742	0.925	2.128	0.515	0.174	2.265	1.627	-0.095	-0.745
30/12/2004	-0.643	-1.417	-0.543	-0.729	-0.513	-0.087	-0.721	0.133	-1.954	1.400
31/12/2004	-0.297	-0.281	-0.614	-1.504	-0.319	0.097	-1.115	-0.050	-0.292	-0.142
03/01/2005	0.684	2.130	3.194	0.497	0.197	-2.638	1.265	-0.133	2.145	-1.526
04/01/2005	-1.708	-4.479	-3.594	0.318	-0.086	-4.922	1.063	-0.768	-0.716	-2.838
05/01/2005	-2.251	-2.312	-1.968	0.352	-2.533	-1.597	-1.019	-2.186	-1.587	-1.729
06/01/2005	1.161	1.742	0.669	0.878	0.631	1.061	-0.675	1.926	3.224	0.182
07/01/2005	-1.613	-3.360	-2.413	0.418	-1.968	0.241	-1.987	-2.159	-2.603	-0.923
10/01/2005	-2.587	0.100	-1.183	-2.945	1.420	-1.937	0.485	-0.121	-0.340	-0.810
11/01/2005	-3.591	-3.741	-2.720	0.643	-0.542	-1.772	-1.656	-2.158	-0.439	-1.432
12/01/2005	0.698	1.561	0.783	1.738	1.724	-1.087	0.368	0.459	1.224	0.984
13/01/2005	2.871	0.820	1.776	3.940	0.723	1.033	0.821	0.931	-0.097	5.013
14/01/2005	3.147	3.355	3.053	-0.201	2.846	2.567	2.860	-0.017	0.969	2.578
17/01/2005	2.143	1.574	0.882	0.168	1.889	1.739	-0.084	2.158	0.719	0.733
18/01/2005	1.156	3.873	1.469	4.698	-0.567	0.667	1.332	4.242	4.524	-0.185
19/01/2005	-0.914	2.455	0.482	-0.897	-1.544	0.010	0.682	4.592	3.462	-0.029
20/01/2005	-2.306	-3.700	-3.669	-2.167	-1.508	-0.652	-1.008	-2.344	-1.409	-0.686
21/01/2005	-2.378	-1.732	-0.926	-1.752	-0.429	0.511	-0.651	-3.168	0.491	-1.065
24/01/2005	0.149	1.763	-0.359	3.365	-0.849	0.632	-0.420	-0.033	2.622	-0.887
25/01/2005	2.210	1.701	2.633	-0.065	0.930	-0.835	-0.540	1.372	-1.039	0.264
26/01/2005	1.999	2.478	2.635	0.293	1.463	-0.904	0.543	0.669	0.832	2.400
27/01/2005	-0.819	-1.692	-1.369	-0.260	1.345	0.126	0.236	-0.696	-1.997	-0.014
28/01/2005	2.209	0.676	2.464	-0.033	-0.143	-0.105	0.118	-0.163	-0.974	1.001
31/01/2005	1.160	-0.305	0.271	1.792	0.419	0.052	1.144	2.255	0.447	1.854
01/02/2005	1.529	1.470	2.432	3.488	2.706	0.356	2.261	0.304	0.712	0.500
02/02/2005	-0.513	-0.362	0.759	3.649	0.453	-0.345	-0.081	-1.338	0.575	-0.816
03/02/2005	1.118	1.878	1.146	3.222	-0.705	0.356	0.537	0.371	2.549	0.516
04/02/2005	-1.122	0.654	1.036	-3.035	0.093	0.209	1.505	-0.273	-2.057	-0.596
07/02/2005	2.270	0.207	0.705	2.742	-0.070	-1.198	0.255	-0.129	-0.525	-0.251
08/02/2005	-1.783	0.383	-0.954	-1.509	1.163	-0.770	0.334	-0.775	-1.628	0.671
09/02/2005	1.147	1.850	2.248	-0.412	-0.161	0.531	-0.301	0.114	0.179	-0.736
10/02/2005	5.705	0.260	-1.665	0.266	-0.127	0.338	0.970	0.065	-0.402	0.630
11/02/2005	-0.496	-0.748	-0.128	0.147	0.542	1.411	-0.173	-0.016	-0.224	0.028
14/02/2005	-2.044	-1.912	-0.704	-1.561	-0.218	-0.509	-0.710	-0.341	-0.943	-0.626

Table A.6: Table of returns (cont.).

date	Gmexico B	Gmodelo C	Kimber A	Penñoles	Soriana B	Telecom A1	Telmex	Televisa CPO	Vitro A	Walmex V
29/09/2004	-0.131	0.327	0.119	3.711	0.000	2.162	0.929	1.005	5.921	1.068
30/09/2004	0.109	0.326	-0.447	3.073	-0.243	-0.588	-0.541	-0.697	-1.775	-0.361
01/10/2004	2.198	2.530	0.516	-0.491	2.682	-0.237	0.272	1.637	-1.355	-0.026
04/10/2004	1.491	1.163	1.239	1.422	1.029	1.008	0.380	0.526	-2.930	0.802
05/10/2004	-1.049	0.732	-0.149	-0.916	0.575	-0.998	-0.216	0.523	-0.472	-0.539
06/10/2004	0.488	-0.138	-0.030	0.962	0.156	-0.415	-0.325	1.463	-0.853	-0.361
07/10/2004	0.021	-0.242	-0.120	2.579	0.311	-0.595	-0.489	-0.481	1.625	0.725
08/10/2004	1.667	-0.660	-0.180	-2.296	1.008	-2.216	-0.928	-2.609	-1.129	-1.980
11/10/2004	0.664	0.664	0.060	-1.175	-0.205	0.429	0.496	0.694	0.190	0.577
12/10/2004	-1.051	-0.035	-0.300	-0.566	1.795	1.402	0.603	0.887	-0.475	1.852
13/10/2004	-4.750	0.208	-0.812	-2.334	-0.504	-1.383	0.273	-0.684	-0.286	-1.076
14/10/2004	-0.437	0.000	-0.091	0.058	-1.266	-0.305	-0.163	-0.590	-1.340	-1.967
15/10/2004	1.626	0.763	0.030	0.000	0.513	0.979	1.580	0.890	1.843	1.848
18/10/2004	-0.584	0.275	0.667	0.000	0.153	2.907	0.536	0.163	-1.429	0.285
19/10/2004	-0.522	-0.446	0.602	0.971	-1.732	-0.353	-0.373	0.163	0.000	-0.646
20/10/2004	-0.306	0.551	-0.299	-0.019	-1.659	0.413	1.927	-0.423	-0.193	-0.651
21/10/2004	1.096	0.308	0.150	-0.539	0.975	3.176	1.681	2.159	4.259	-0.236
22/10/2004	5.400	-0.171	-0.570	-0.348	1.018	0.798	0.155	1.089	-2.043	-0.184
25/10/2004	1.255	-0.171	-0.362	1.688	-0.026	-1.018	-1.238	-1.774	-0.474	0.316
26/10/2004	0.488	0.103	0.787	-2.080	1.008	1.143	1.358	0.387	-2.095	-1.416
27/10/2004	-1.780	1.644	0.601	1.189	-0.972	0.282	1.133	1.092	-0.681	0.824
28/10/2004	-2.347	-0.270	0.746	1.791	-3.643	-0.563	0.255	-0.222	0.392	-0.659
29/10/2004	0.485	-0.101	2.281	-0.265	-0.590	0.567	0.559	1.178	0.488	0.239
01/11/2004	1.259	0.068	2.057	-1.973	2.347	0.732	1.668	0.944	-0.485	-0.026
02/11/2004	-1.036	1.419	0.199	-0.619	0.580	0.615	0.000	0.062	1.366	0.954
03/11/2004	-0.209	-0.200	-0.028	-0.253	0.629	2.724	1.491	0.654	0.577	1.549
04/11/2004	2.434	0.935	-0.935	3.182	-0.417	0.649	0.000	0.588	-0.574	0.026
05/11/2004	2.171	-0.364	-0.114	1.249	-0.366	0.860	0.098	-1.416	-3.176	-0.853
08/11/2004	1.343	-0.398	0.458	0.635	0.052	-0.853	-1.076	-1.842	1.392	0.599
09/11/2004	0.890	0.167	1.169	-0.464	-0.839	-0.538	-0.198	1.654	2.549	0.078
10/11/2004	1.941	0.133	0.423	0.205	0.000	0.649	0.050	-0.188	2.008	0.181
11/11/2004	1.096	0.565	0.309	-0.354	0.767	2.041	2.427	2.853	1.500	1.034
12/11/2004	-0.723	-0.297	-0.420	-0.037	0.394	-0.842	-0.532	0.457	-2.031	1.151
15/11/2004	0.402	-0.298	0.281	0.336	0.418	0.690	0.146	0.485	-0.377	-1.517
16/11/2004	0.134	-0.565	-0.140	1.416	1.797	-0.369	-0.340	0.091	-0.757	-0.308
17/11/2004	2.688	-0.067	-0.140	3.508	1.433	0.529	0.146	-0.151	-1.239	0.412
18/11/2004	0.186	-1.071	0.281	4.596	-0.126	0.000	-0.438	1.239	-0.676	-0.256
19/11/2004	-1.816	-1.894	-1.092	0.102	-2.778	-2.737	-2.101	-0.209	-1.846	-1.080
22/11/2004	0.698	0.310	0.963	0.000	1.688	0.000	-0.050	0.090	0.495	0.312
23/11/2004	-0.768	1.203	0.252	-1.695	-1.149	-2.597	-0.349	0.807	0.394	0.518
24/11/2004	1.737	0.340	1.539	1.000	-0.698	0.500	-1.553	0.504	0.294	0.077
25/11/2004	0.557	0.677	0.441	1.024	0.546	1.050	0.153	0.324	3.131	0.412
26/11/2004	0.351	-0.571	0.960	0.152	0.958	0.602	0.610	2.469	-1.044	0.308
29/11/2004	0.625	0.000	1.902	2.244	0.385	1.359	0.253	0.143	1.822	0.588
30/11/2004	-0.804	-0.507	0.000	-3.267	-1.430	-4.453	-1.209	-0.115	-2.542	-2.034
01/12/2004	0.627	0.578	0.773	3.634	0.777	2.302	0.867	-0.029	2.222	0.649
02/12/2004	-3.590	-1.419	0.026	-2.683	-1.697	-0.988	-0.354	-1.779	-0.378	-1.753
03/12/2004	-0.228	0.000	-0.080	0.778	0.157	-1.275	-0.558	-2.015	-0.854	0.052
06/12/2004	-0.457	0.206	1.072	-1.813	-0.522	0.168	1.276	-0.387	1.435	-0.052
07/12/2004	-1.415	0.718	0.451	-2.444	-1.234	-2.859	-1.662	0.419	-0.283	-0.157
08/12/2004	-1.746	0.883	-1.241	-2.856	0.691	0.404	-0.154	-0.328	4.068	-0.079
09/12/2004	-0.612	0.135	-0.508	2.128	-0.475	0.115	0.616	-1.704	2.909	0.421

Table A.7: Table of returns (cont.).

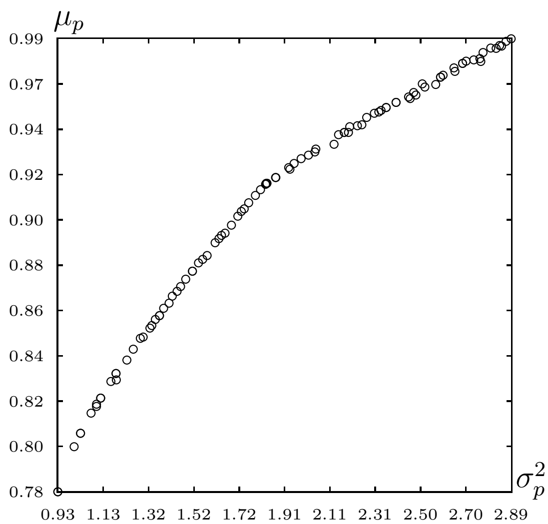
date	Gmexico B	Gmodelo C	Kimber A	Penñoles	Soriana B	Telecom A1	Telmex	Televisa CPO	Vitro A	Walmex V
10/12/2004	3.100	1.345	-1.989	3.108	2.095	0.689	0.204	-0.517	9.982	0.550
13/12/2004	2.139	-0.033	0.110	-0.137	-0.234	0.627	0.153	1.437	-0.964	-0.130
14/12/2004	0.415	-0.464	0.274	-0.515	0.234	2.550	0.410	1.175	-0.649	1.383
15/12/2004	1.823	-0.567	-0.191	3.655	1.922	0.552	0.408	-0.387	5.306	0.051
16/12/2004	0.277	0.436	0.027	-1.863	-1.121	1.593	0.661	-0.359	-0.930	-1.389
17/12/2004	-0.460	0.801	-0.027	0.000	0.129	-1.136	-0.303	-0.540	-0.626	-0.339
20/12/2004	1.146	-0.364	-0.027	-0.390	0.257	0.219	0.101	-0.422	-2.756	0.654
21/12/2004	2.358	0.366	0.602	2.518	-1.566	1.583	2.834	0.333	-3.806	0.624
22/12/2004	-0.125	-0.232	2.041	0.415	-0.104	2.149	3.051	-0.453	1.010	-0.517
23/12/2004	0.143	1.328	-0.080	-1.636	0.392	1.262	0.669	0.303	0.000	-0.442
24/12/2004	0.321	0.066	0.747	1.227	0.988	-0.260	0.000	0.393	0.833	0.157
27/12/2004	0.569	0.786	-0.662	-0.398	0.386	1.198	0.996	-0.301	-0.909	-0.625
28/12/2004	-0.124	0.000	0.827	0.233	1.308	1.441	1.174	3.264	-0.751	0.262
29/12/2004	-0.301	1.267	1.560	-0.399	2.051	1.015	0.371	0.205	-0.840	0.418
30/12/2004	0.800	-0.577	-0.026	0.000	-0.695	0.050	-0.231	-1.431	0.424	-0.417
31/12/2004	-0.882	-1.097	0.287	0.167	-0.100	-0.351	-0.695	-0.326	-1.941	0.105
03/01/2005	0.854	1.207	-0.078	-0.650	-0.475	0.302	0.000	0.208	1.119	-0.131
04/01/2005	-4.903	-0.097	-1.716	-3.137	-2.362	-1.858	-0.887	-1.988	-6.979	0.052
05/01/2005	0.148	0.710	-2.063	1.282	1.030	-2.508	-1.884	-0.817	-0.366	-0.470
06/01/2005	1.130	-0.384	-0.351	-1.060	-0.127	2.677	0.192	0.915	2.938	0.551
07/01/2005	-0.201	-3.312	-1.355	-0.916	-2.117	-4.090	-1.725	-1.935	-0.714	-0.627
10/01/2005	0.202	0.333	-1.044	0.959	0.339	0.746	0.634	0.586	-0.988	0.026
11/01/2005	-0.879	-1.525	-1.444	-0.294	-2.545	-2.910	-1.744	-2.912	-1.996	-0.525
12/01/2005	0.037	1.582	3.690	0.000	1.279	0.654	0.493	1.800	2.963	0.264
13/01/2005	0.055	-0.033	-0.788	-0.208	0.868	0.975	1.227	-0.434	1.529	1.080
14/01/2005	0.978	0.464	-0.356	0.278	1.017	2.145	0.630	2.928	1.683	1.069
17/01/2005	-0.311	0.033	0.027	0.000	0.077	1.627	0.530	1.090	-0.697	0.413
18/01/2005	2.677	0.429	-0.027	2.095	1.265	2.531	1.150	1.647	0.702	1.181
19/01/2005	-1.446	-0.164	-0.522	-2.086	-1.223	1.360	-1.326	-0.088	-1.132	-1.218
20/01/2005	-2.573	-0.493	-1.381	-4.105	-0.439	-1.590	-0.528	-2.182	-0.969	-1.824
21/01/2005	0.484	-0.198	0.504	1.156	-0.596	0.000	-1.207	-2.080	2.491	-0.497
24/01/2005	1.425	-1.027	0.753	1.768	1.486	1.768	1.026	0.185	-1.910	0.026
25/01/2005	0.748	0.803	0.968	1.105	0.437	0.397	1.209	0.276	8.850	0.920
26/01/2005	2.318	-0.066	0.000	0.955	0.384	0.939	0.191	2.604	5.122	0.964
27/01/2005	0.159	-0.299	-0.438	1.427	-0.535	-0.147	0.048	-0.806	-5.491	-0.981
28/01/2005	-1.043	1.500	-0.275	0.814	0.282	-0.098	-0.095	-1.385	1.473	-0.391
31/01/2005	0.714	-0.821	0.193	-1.311	1.098	0.049	-0.334	0.763	-1.532	1.047
01/02/2005	1.419	0.232	-0.330	-1.670	-0.051	1.178	1.819	3.120	-0.491	1.683
02/02/2005	0.507	-0.727	-0.276	-0.936	0.354	-1.455	-0.329	0.470	0.329	-0.891
03/02/2005	-1.113	0.931	-0.222	0.700	-0.252	1.870	1.321	0.146	0.246	1.182
04/02/2005	-0.633	1.088	0.722	-0.938	-1.515	-2.319	0.047	1.109	0.164	0.051
07/02/2005	0.921	-0.848	-0.248	-1.473	0.513	-0.346	0.140	1.155	-2.859	-0.888
08/02/2005	-0.316	1.085	0.249	-0.498	0.969	0.695	0.232	0.143	2.860	3.355
09/02/2005	-0.088	0.260	2.978	1.037	4.346	2.218	3.662	-0.399	-1.635	1.264
10/02/2005	1.110	0.584	-0.321	4.407	2.881	0.289	0.045	1.832	0.998	1.028
11/02/2005	-0.453	1.935	0.269	2.441	0.730	0.385	0.536	-0.253	-0.165	-0.242
14/02/2005	-0.963	0.949	-0.241	-3.988	-3.107	-1.868	-0.267	-1.550	-0.989	-0.947

Table A.8: Table of returns (cont.).

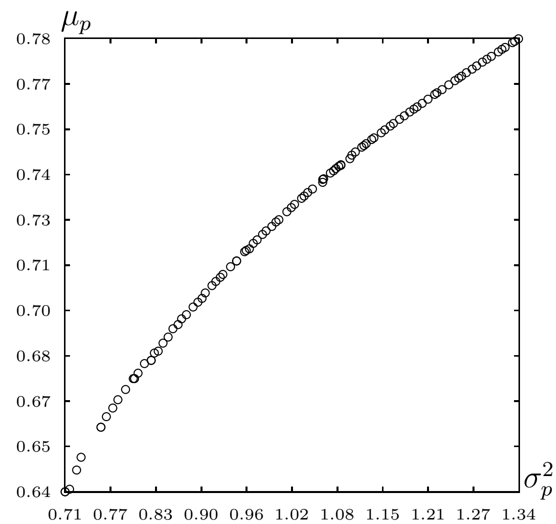
Appendix B

Graphs of the solutions

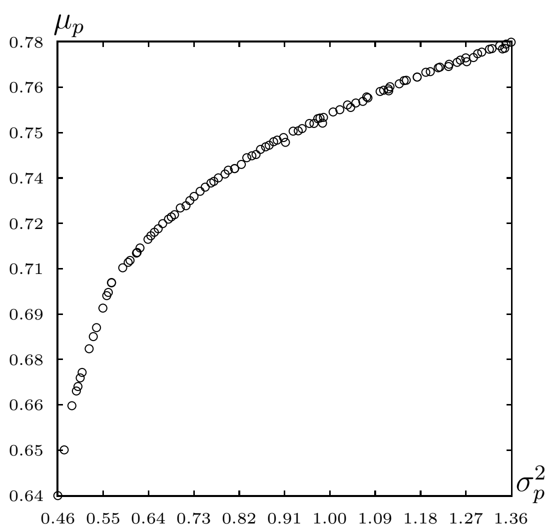
This appendix includes the graphs of the solutions obtained in chapter 6 (see this chapter for details).



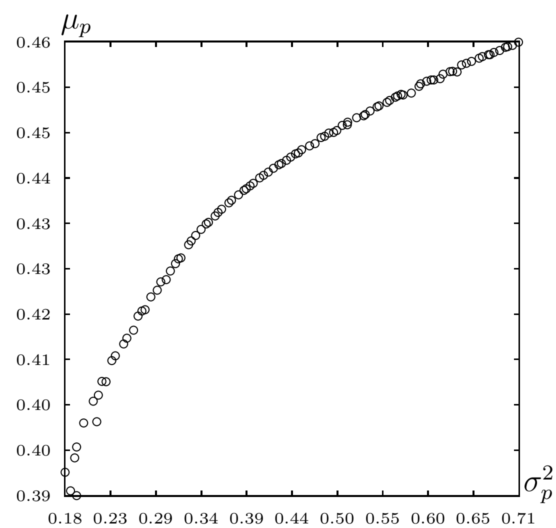
0



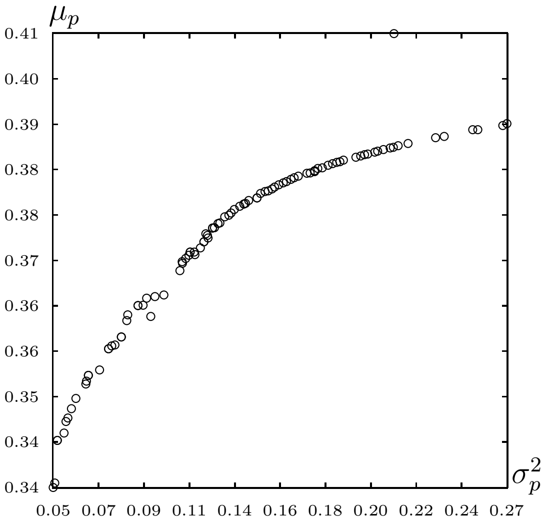
1



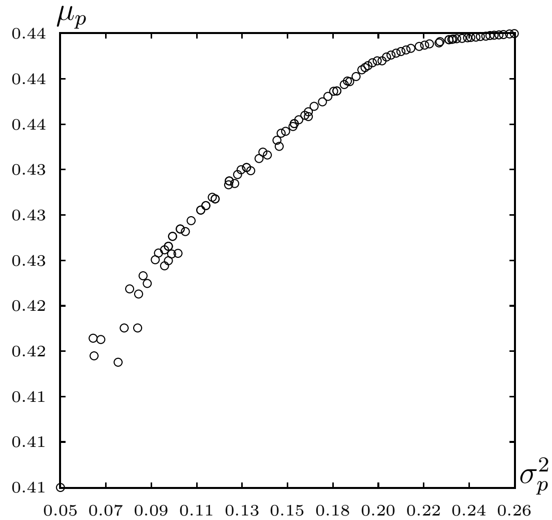
2



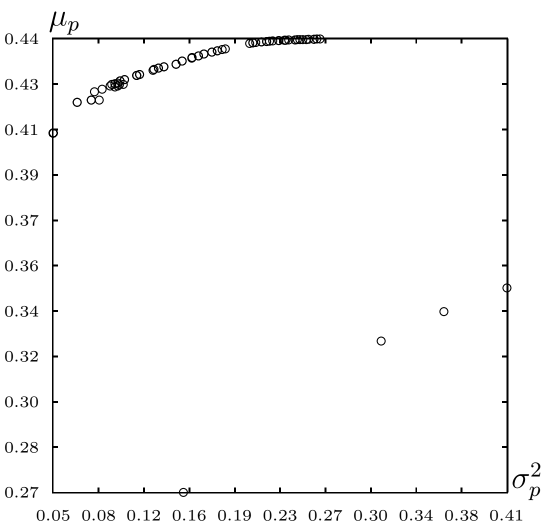
3


μp
0.41
0.40
0.39
0.38
0.38
0.37
0.36
0.36
0.35
0.34
0.34
σp2
0.05 0.07 0.09 0.11 0.14 0.16 0.18 0.20 0.22 0.24 0.27

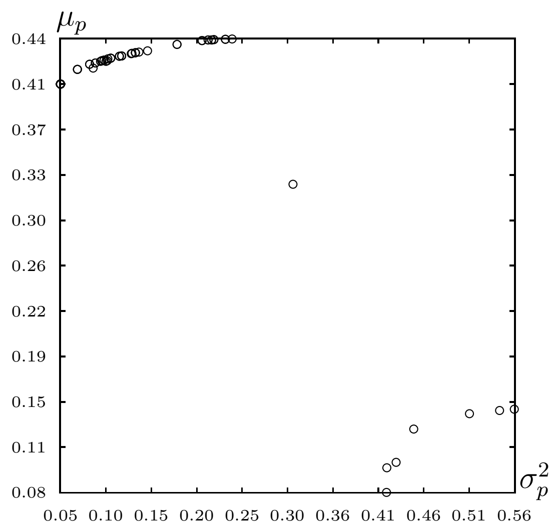
4


μp
0.44
0.44
0.44
0.43
0.43
0.43
0.42
0.42
0.41
0.41
0.41
σp2
0.05 0.07 0.09 0.11 0.13 0.15 0.18 0.20 0.22 0.24 0.26

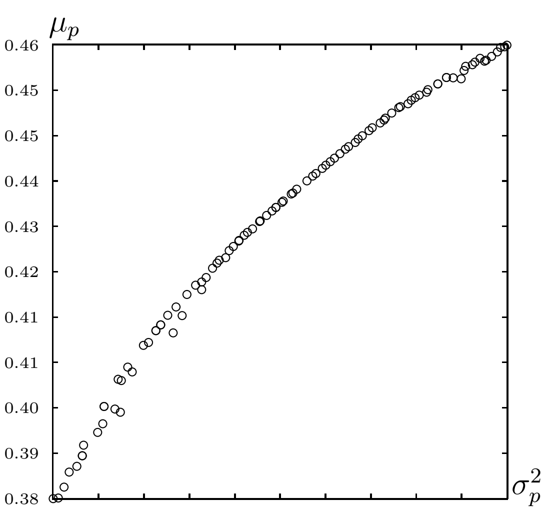
5


μp
0.44
0.43
0.41
0.39
0.37
0.36
0.34
0.32
0.30
0.28
0.27
σp2
0.05 0.08 0.12 0.16 0.19 0.23 0.27 0.30 0.34 0.38 0.41

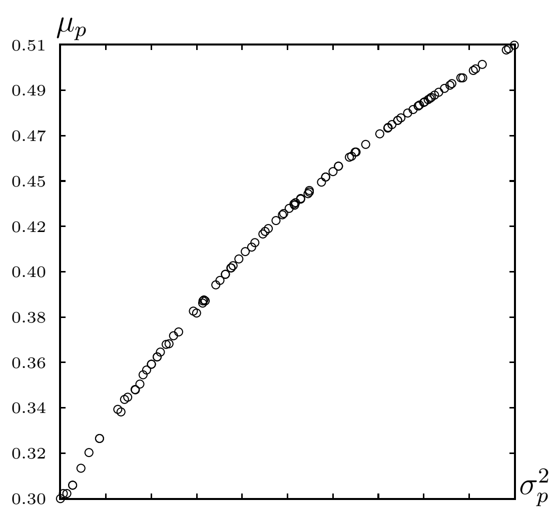
6


μp
0.44
0.41
0.37
0.33
0.30
0.26
0.22
0.19
0.15
0.11
0.08
σp2
0.05 0.10 0.15 0.20 0.25 0.30 0.36 0.41 0.46 0.51 0.56

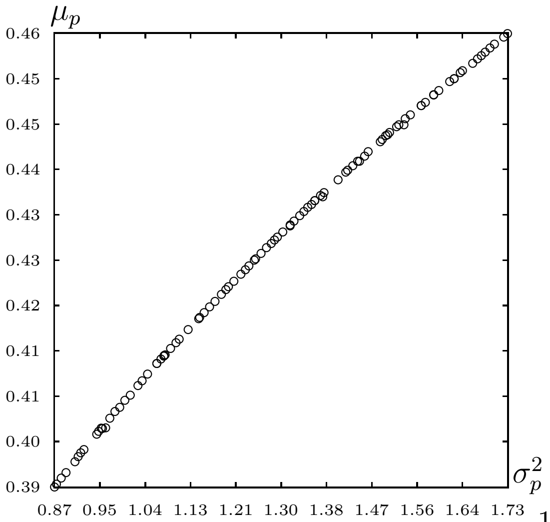
7


μp
0.46
0.45
0.45
0.44
0.43
0.42
0.41
0.41
0.40
0.39
0.38
σp2
0.85 0.93 1.02 1.10 1.18 1.26 1.34 1.42 1.50 1.58 1.66

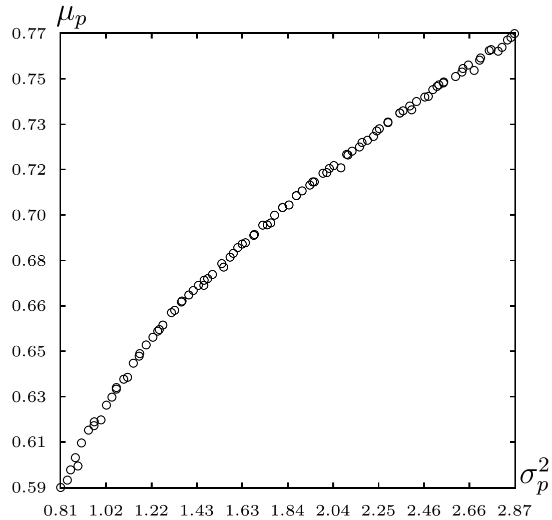
8


μp
0.51
0.49
0.47
0.45
0.42
0.40
0.38
0.36
0.34
0.32
0.30
σp2
0.48 0.61 0.74 0.87 1.00 1.13 1.26 1.39 1.52 1.65 1.78

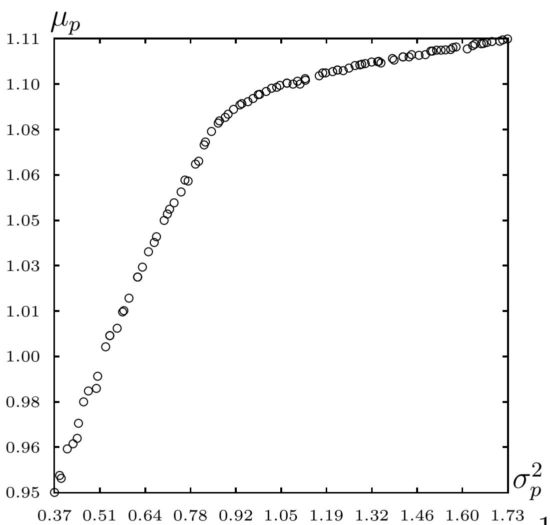
9



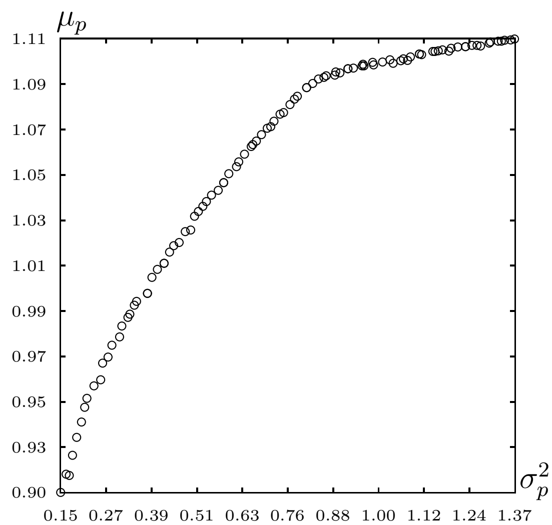
10



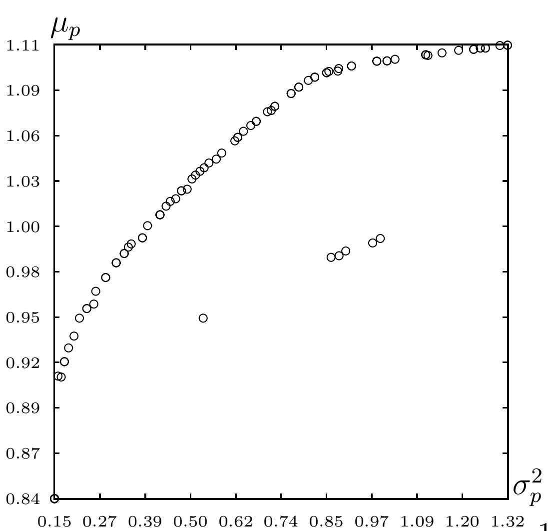
11



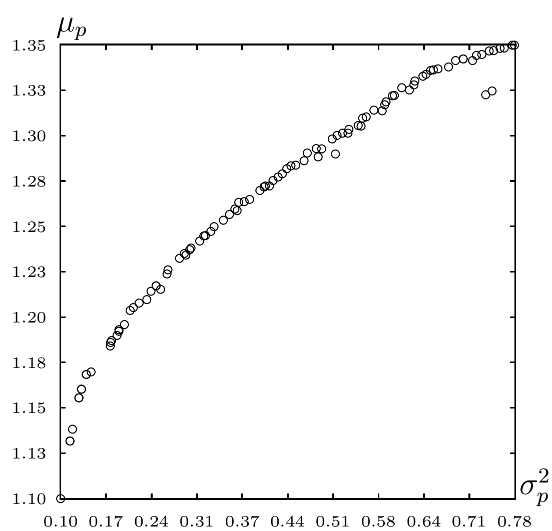
12



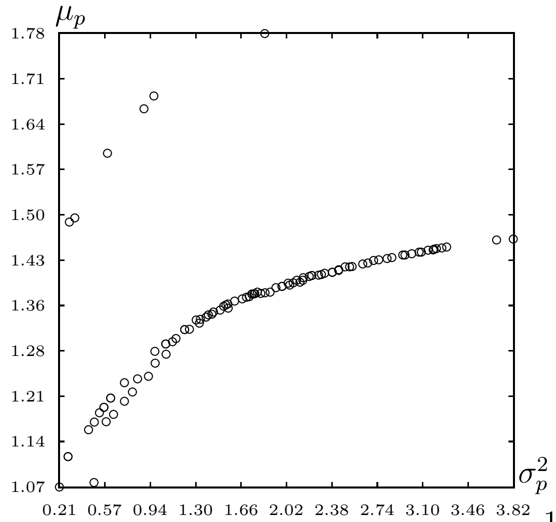
13



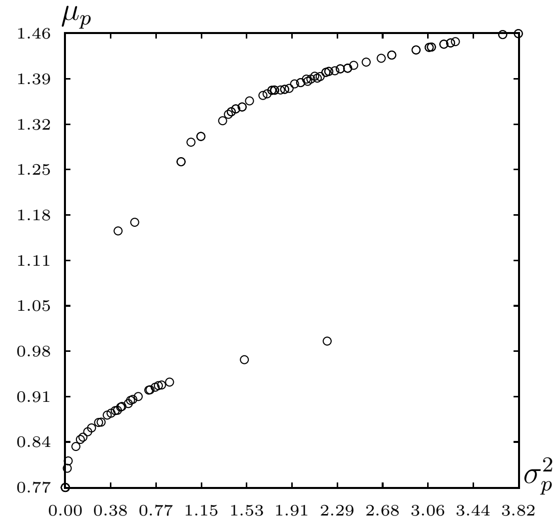
14



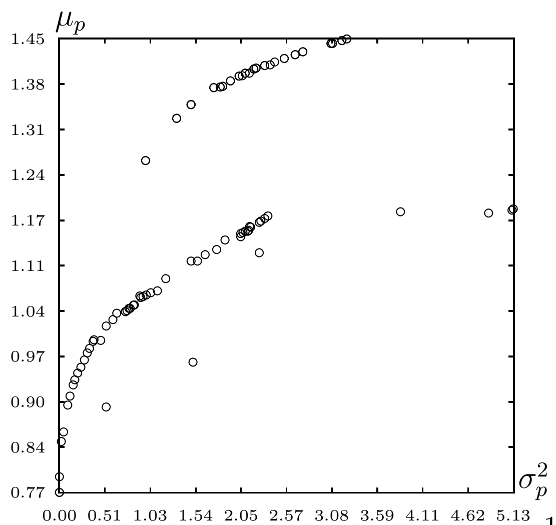
15



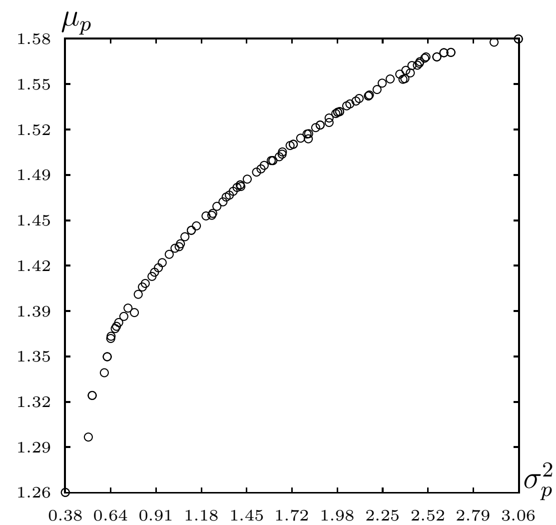
16



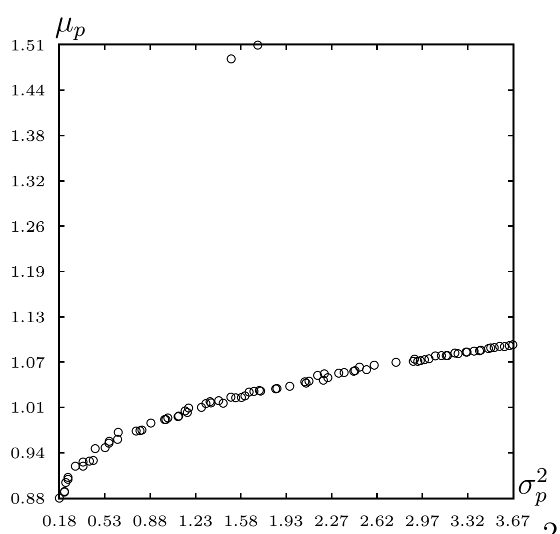
17



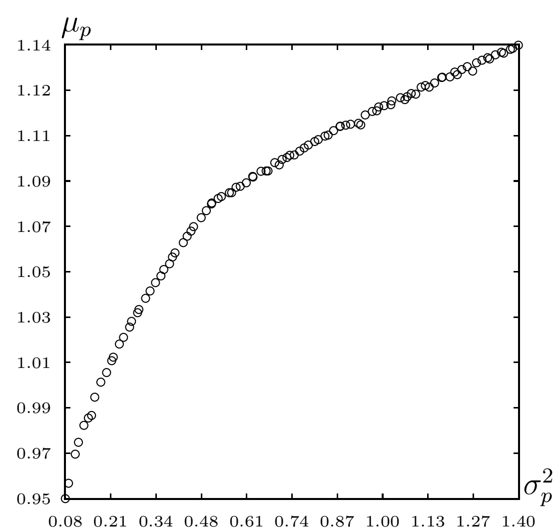
18



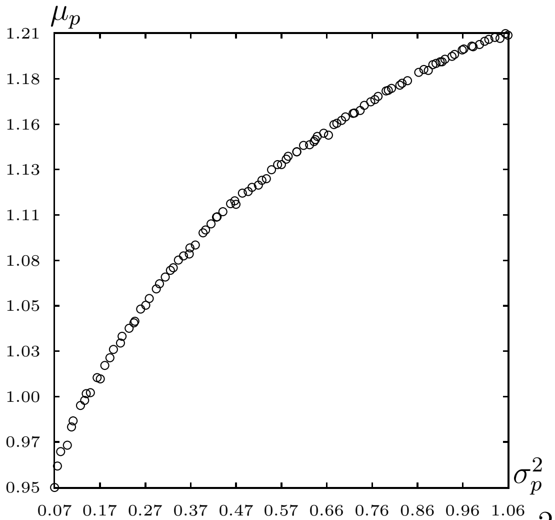
19



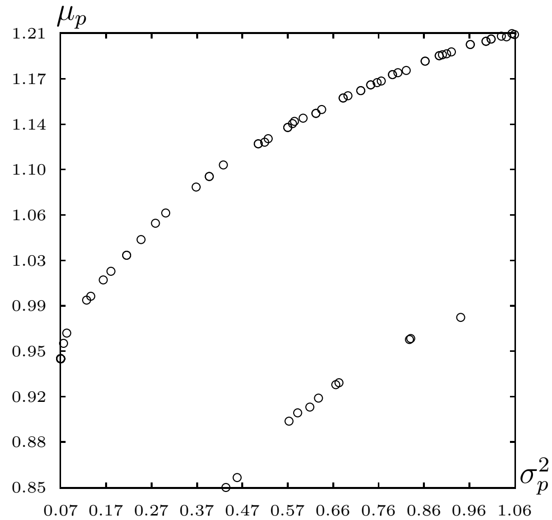
20



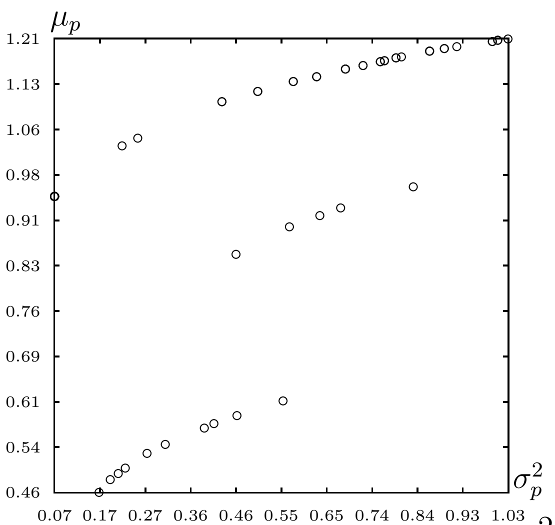
21



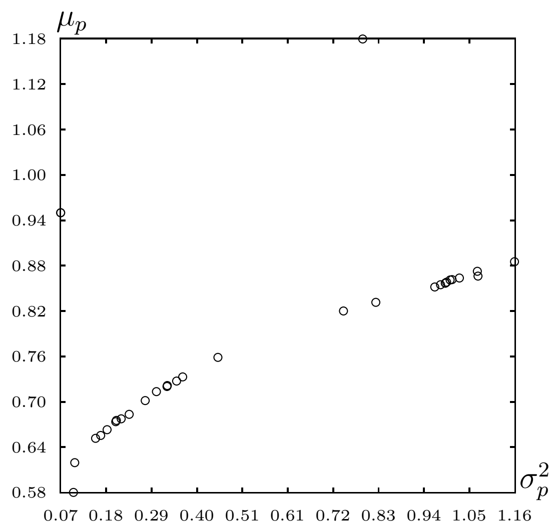
22



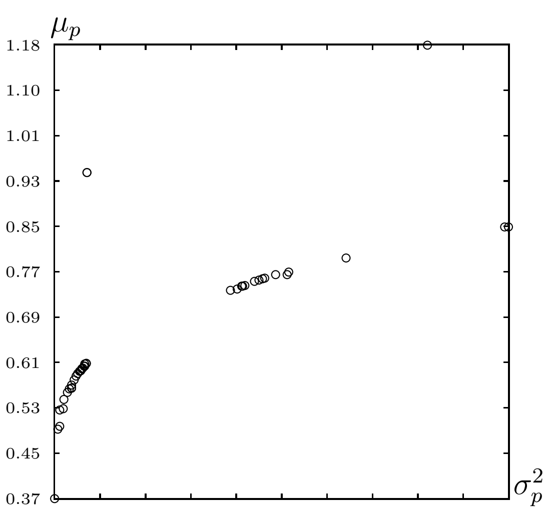
23



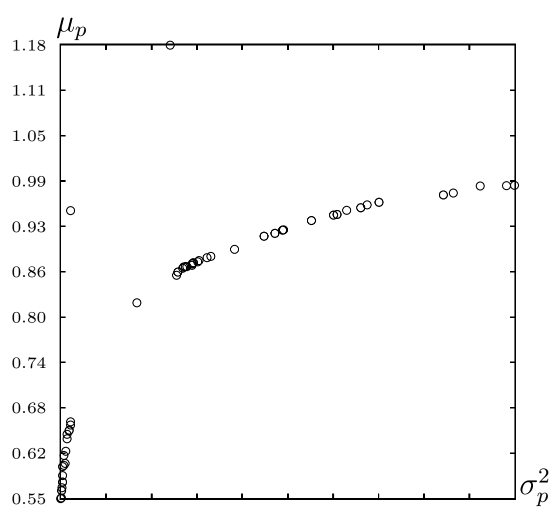
24



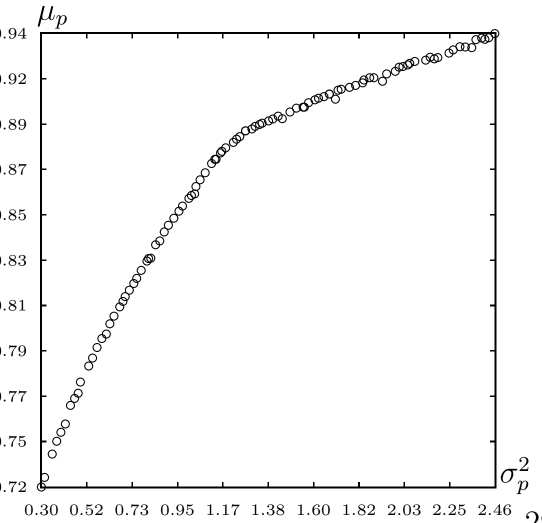
25



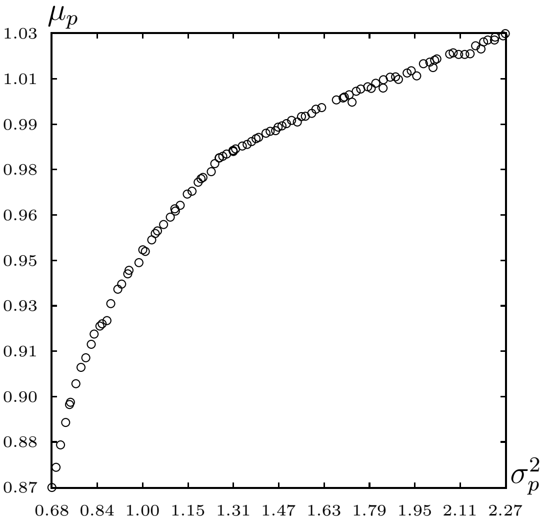
26



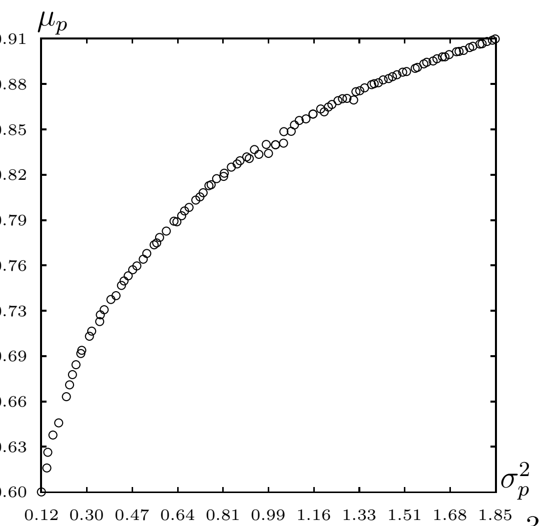
27



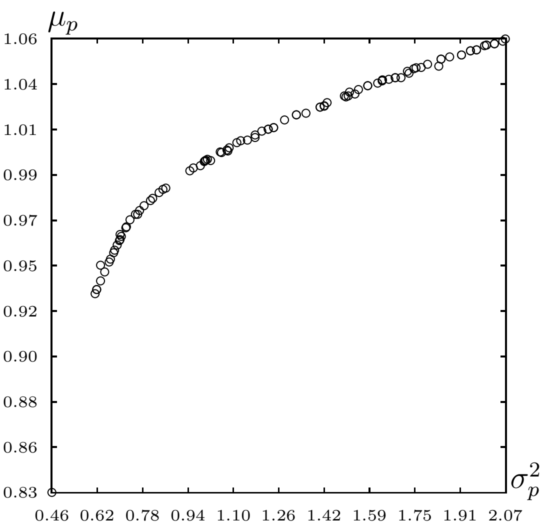
28



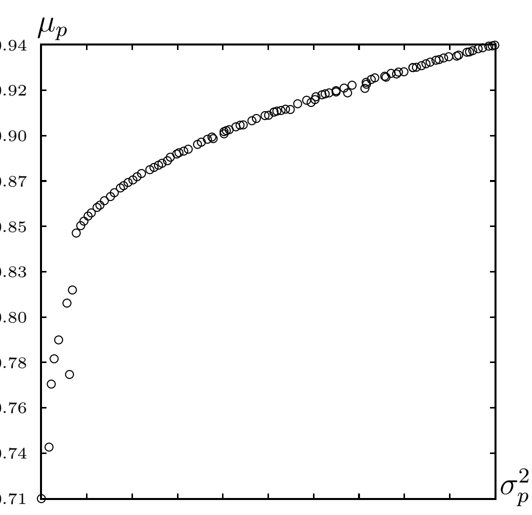
29



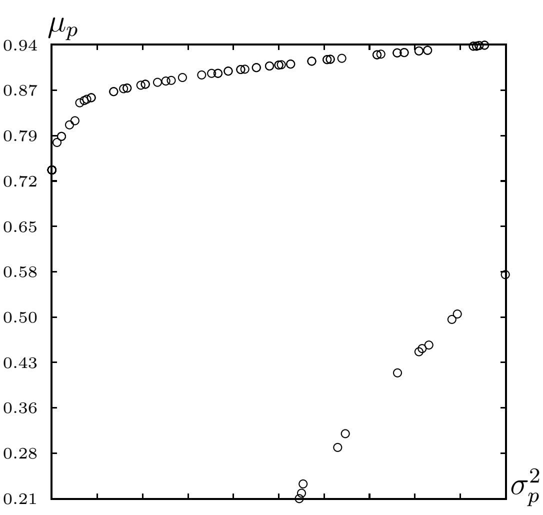
30



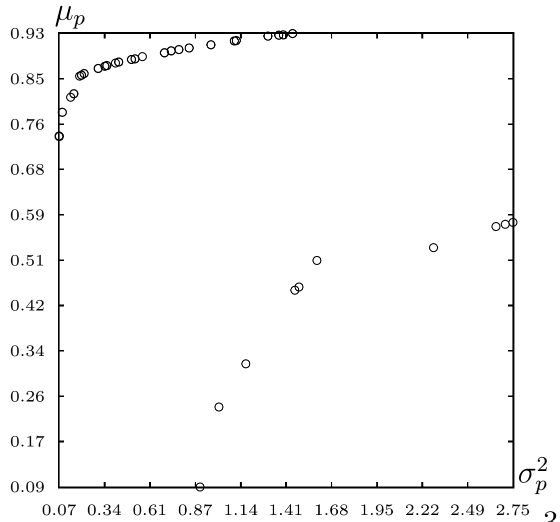
31



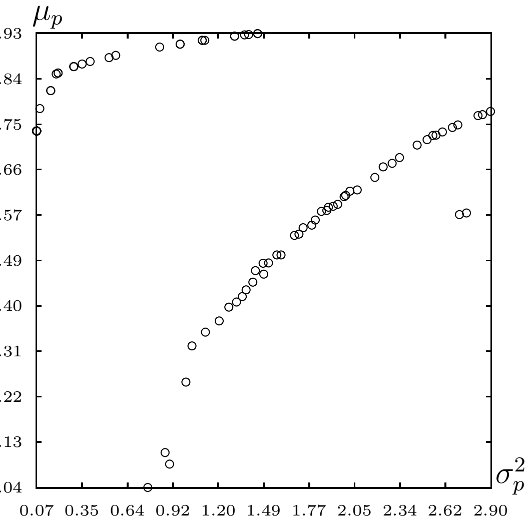
32



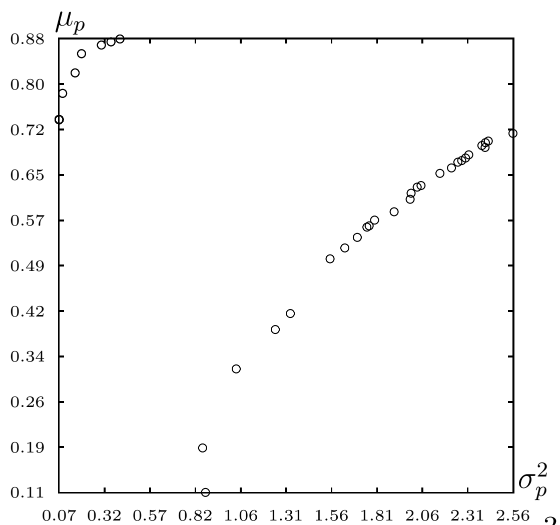
33



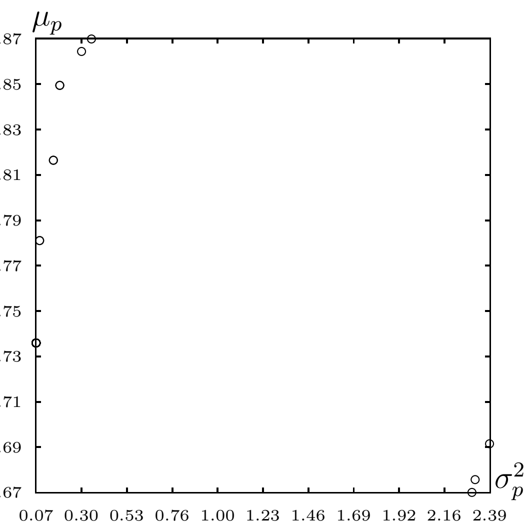
34



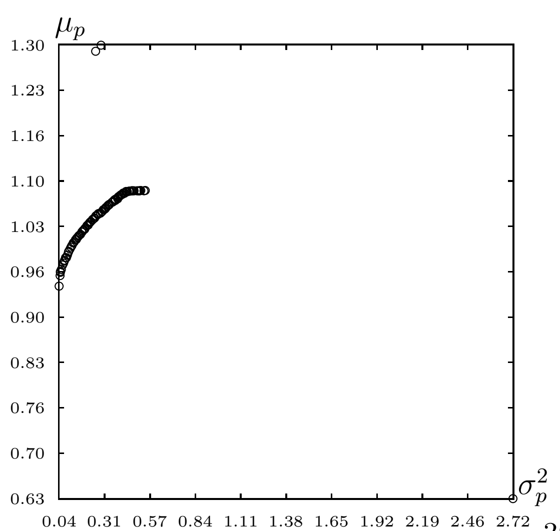
35



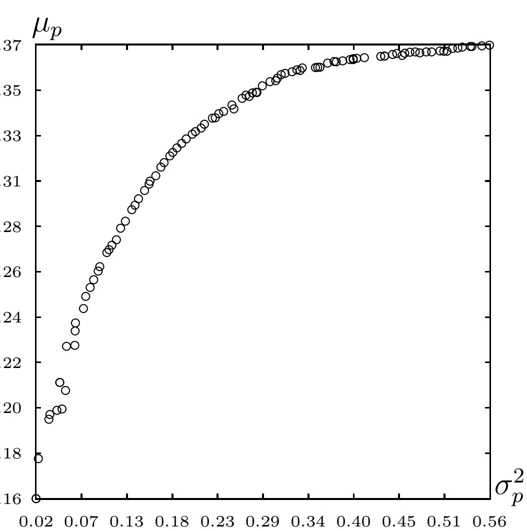
36



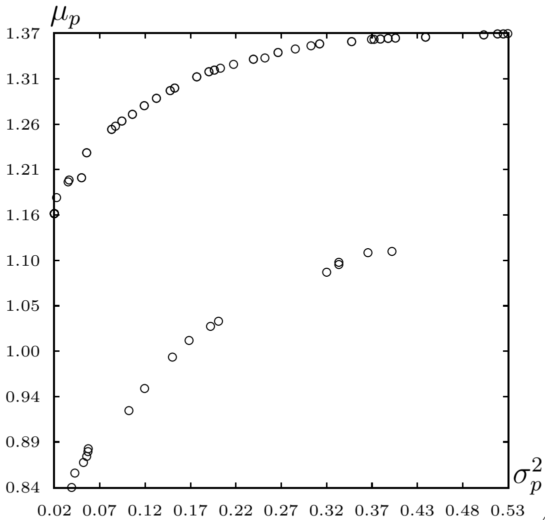
37



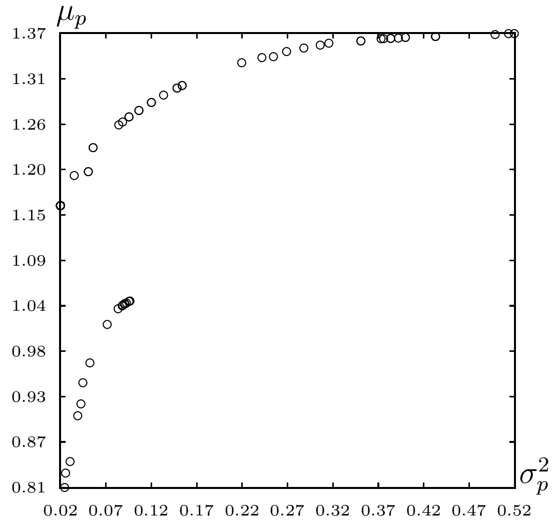
38



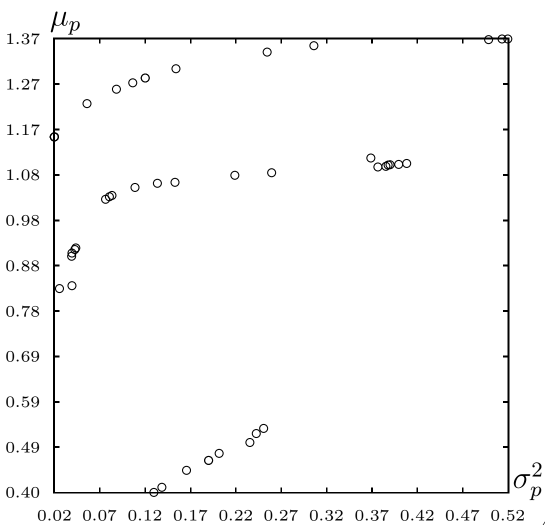
39



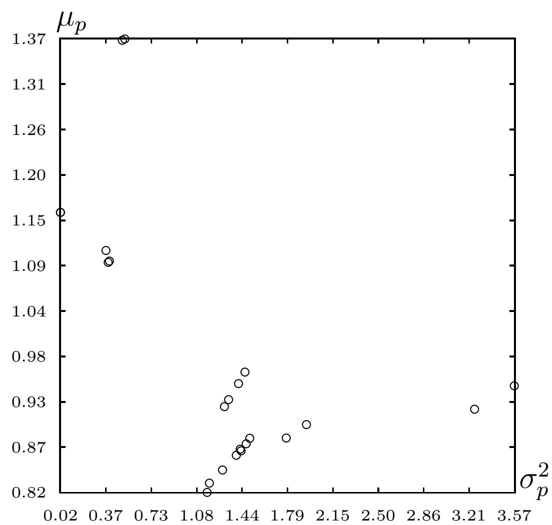
40



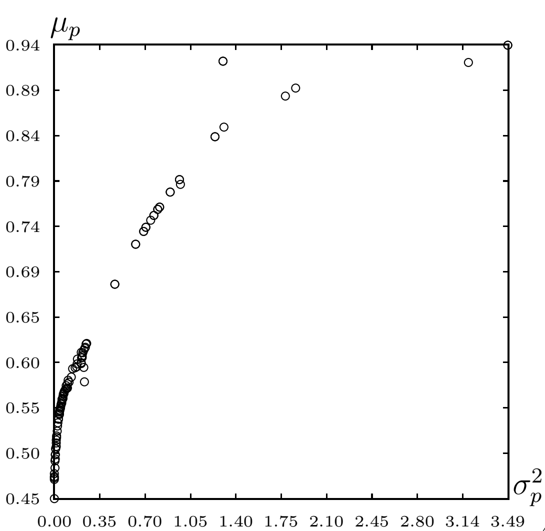
41



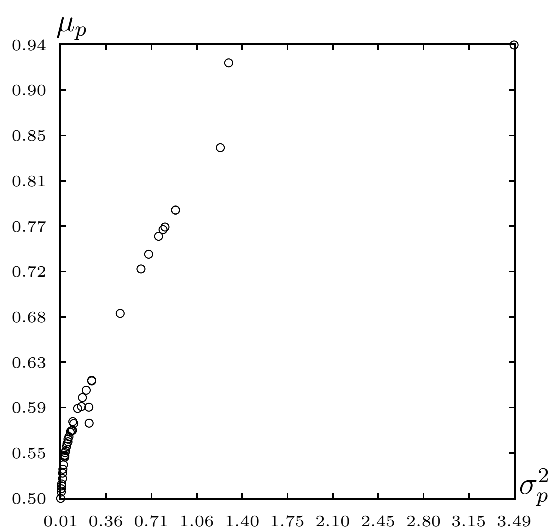
42



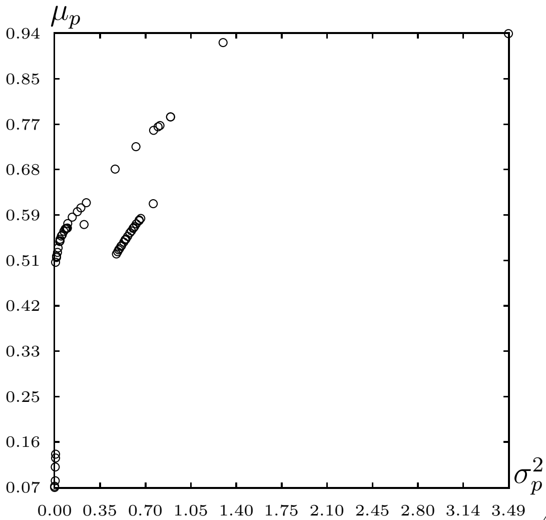
43



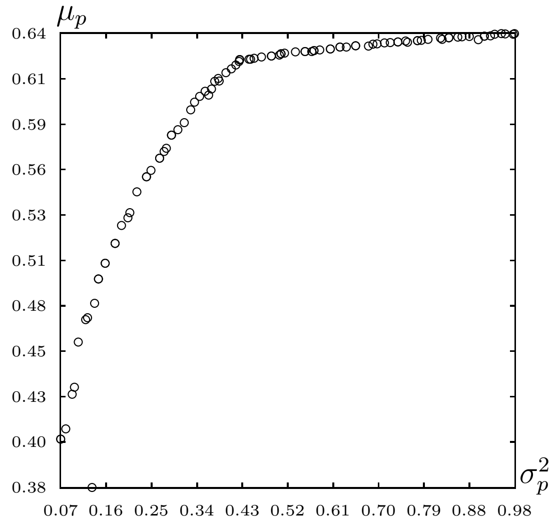
44



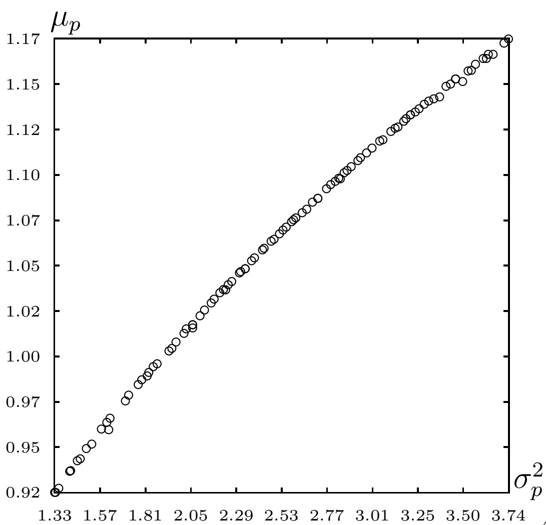
45



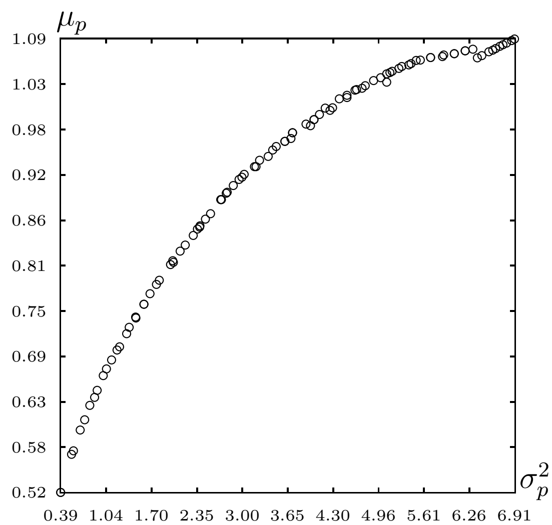
46



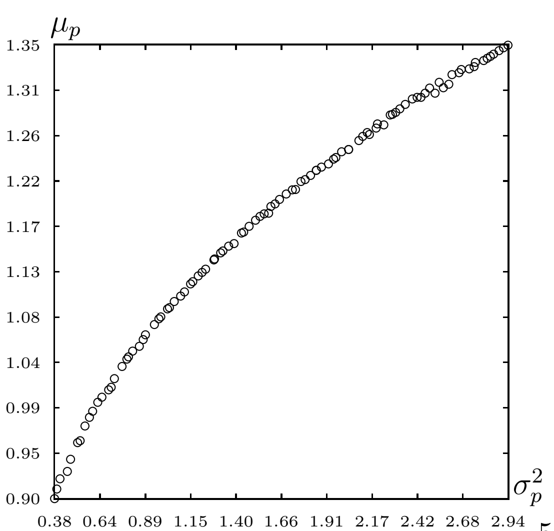
47



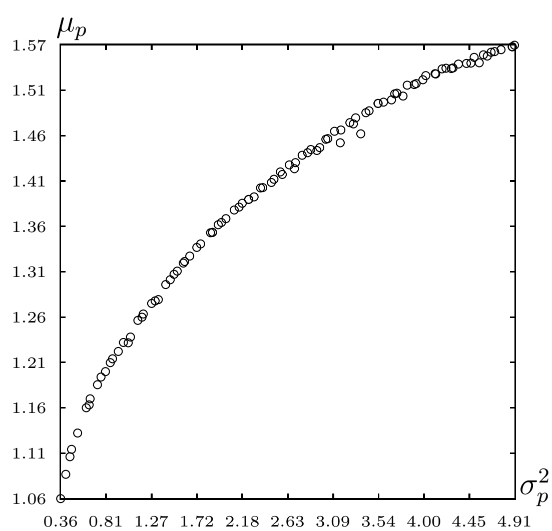
48



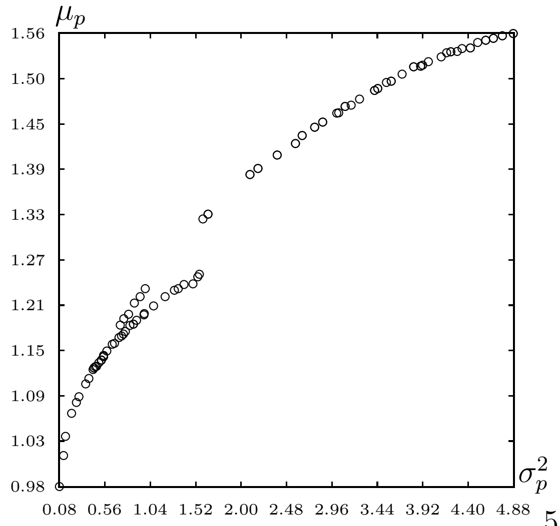
49



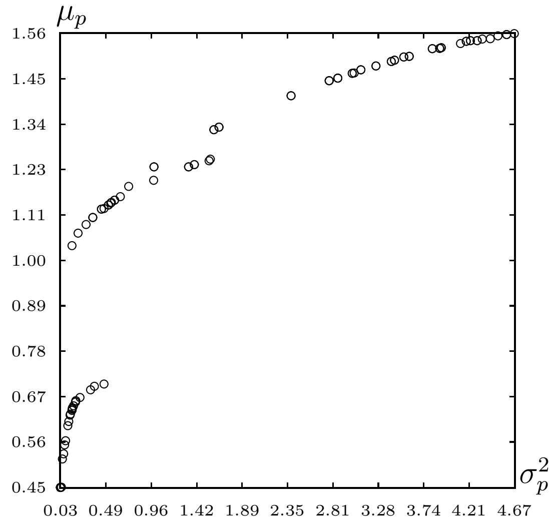
50



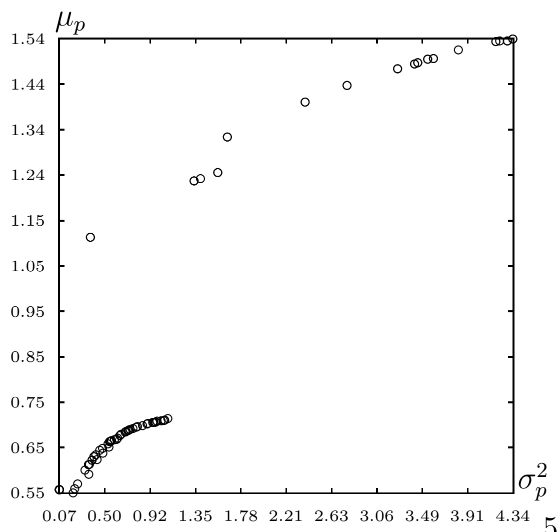
51



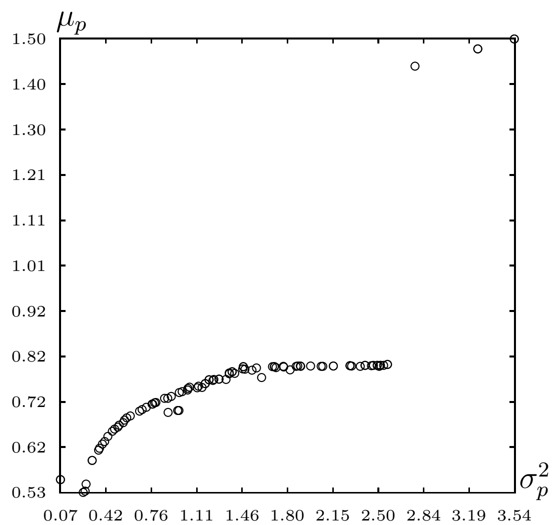
52



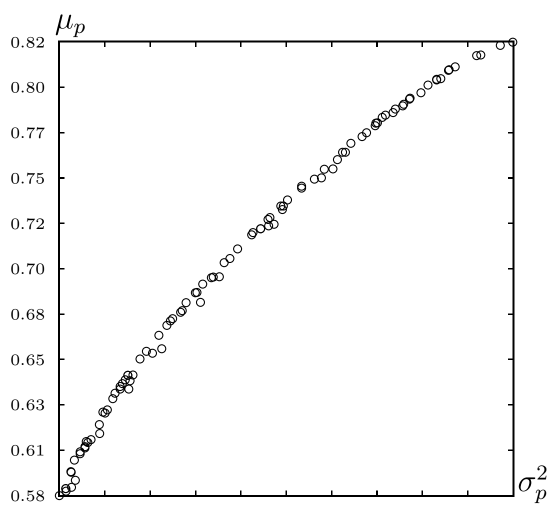
53



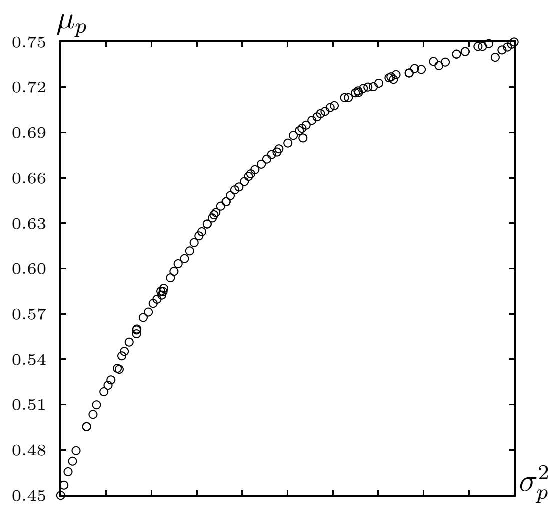
54



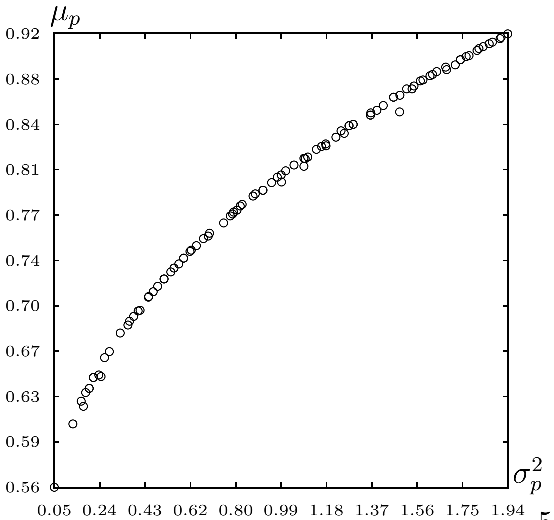
55



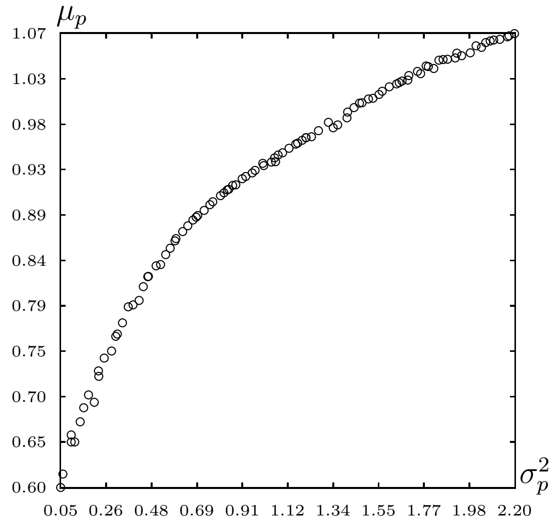
56



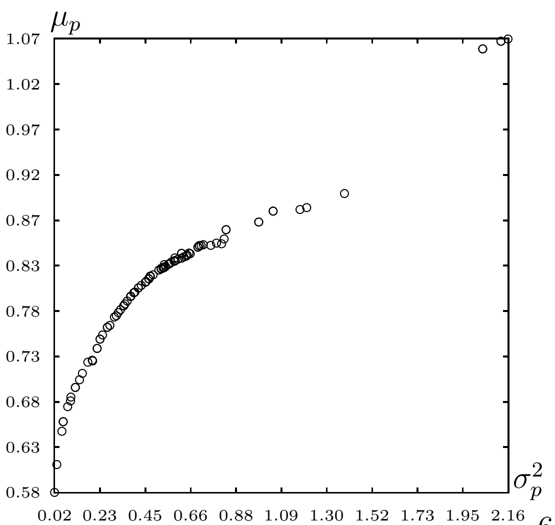
57



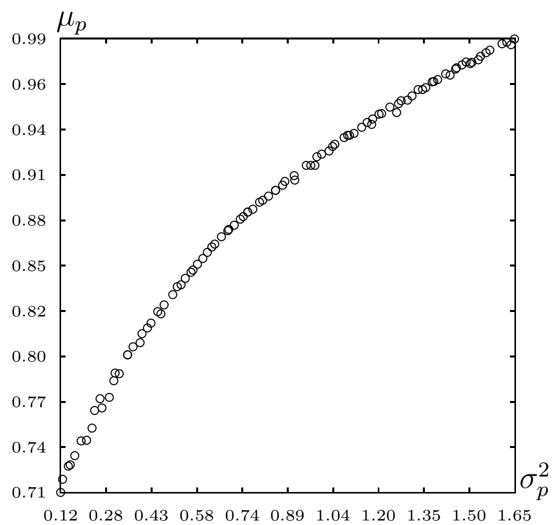
58



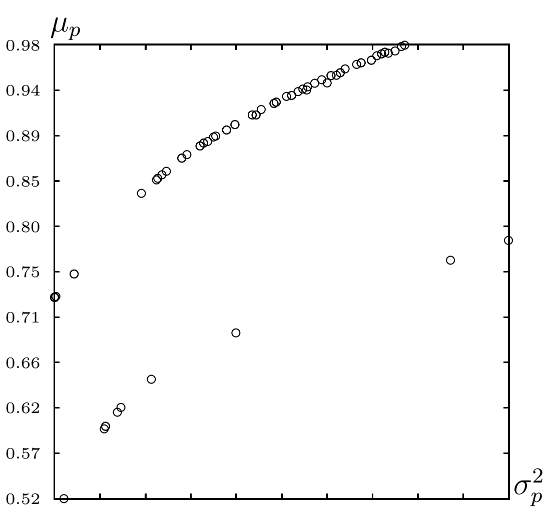
59



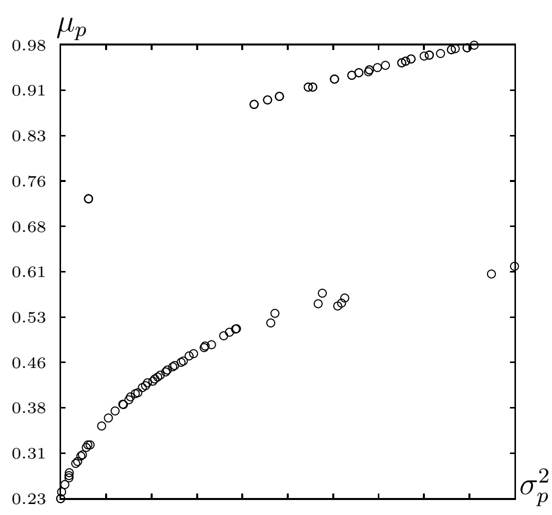
60



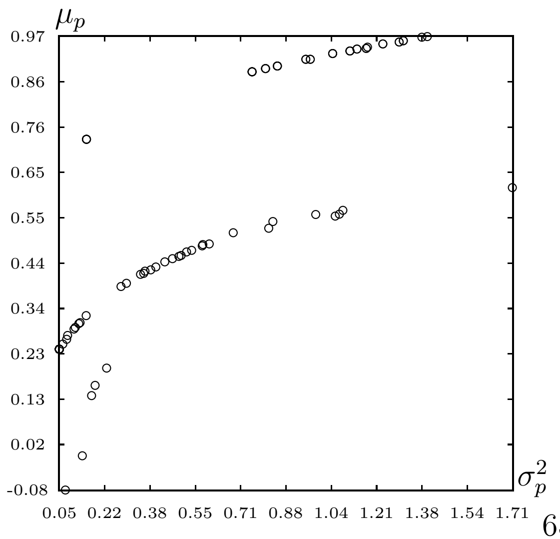
61



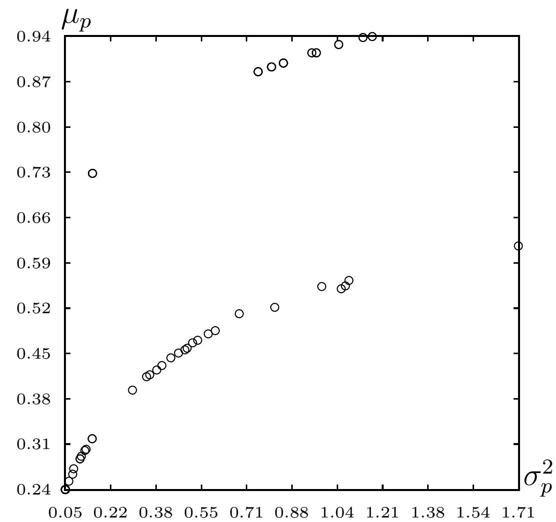
62



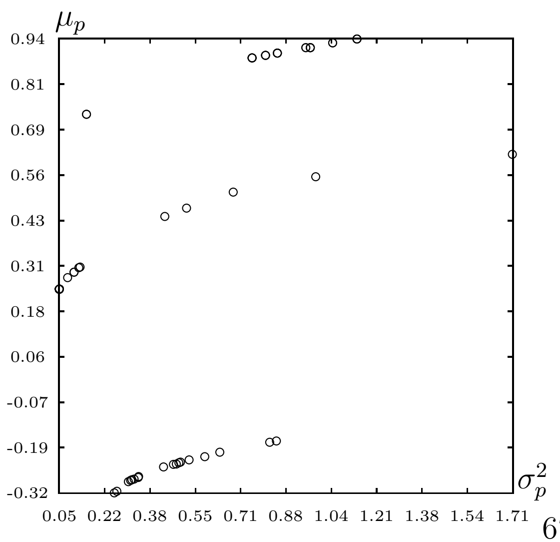
63



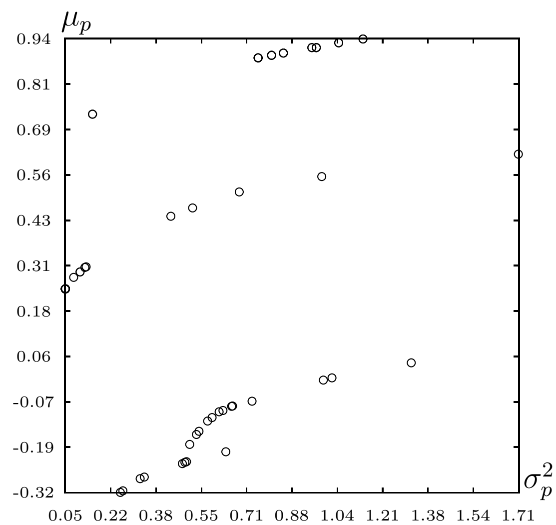
64



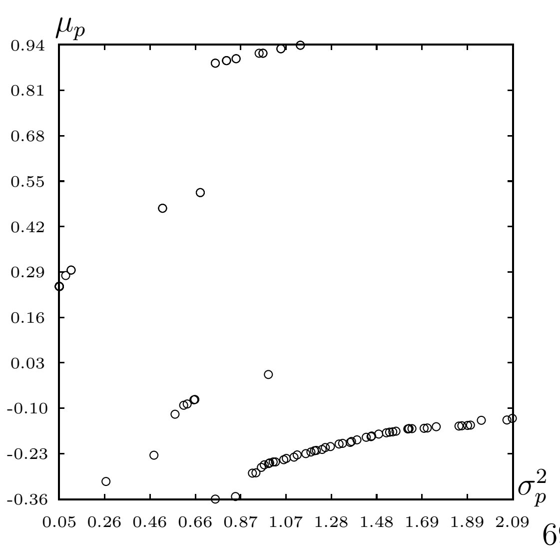
65



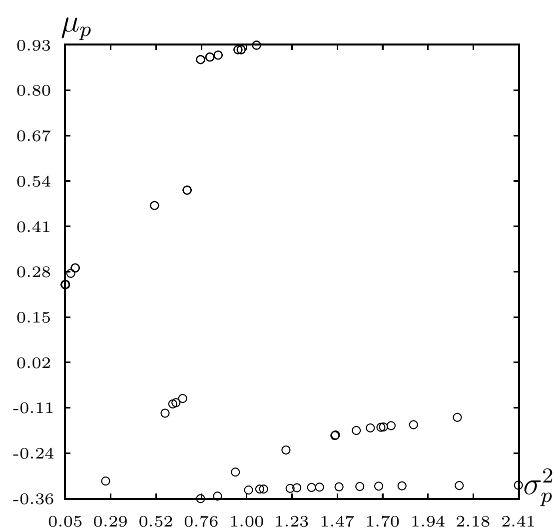
66



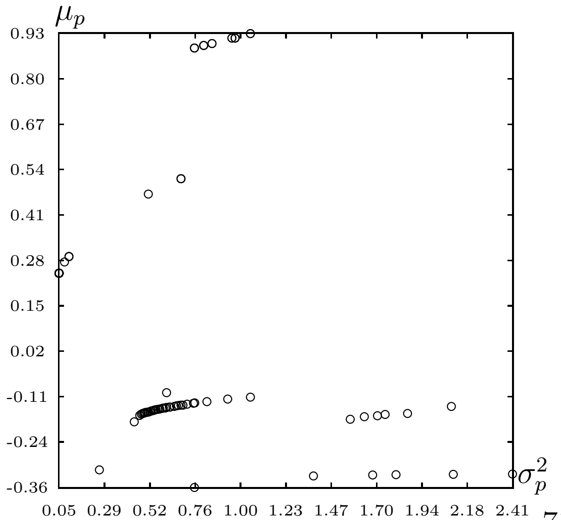
67



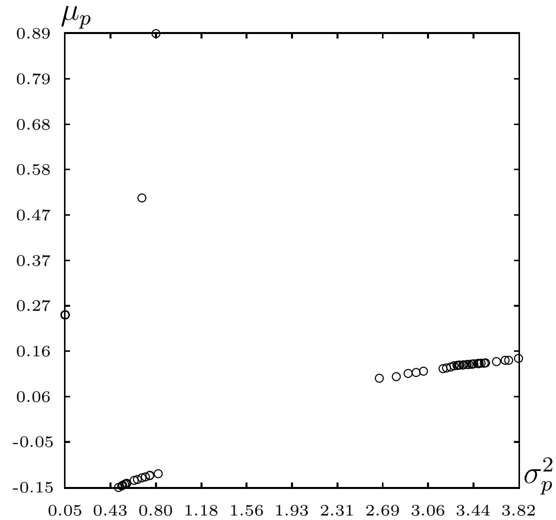
68



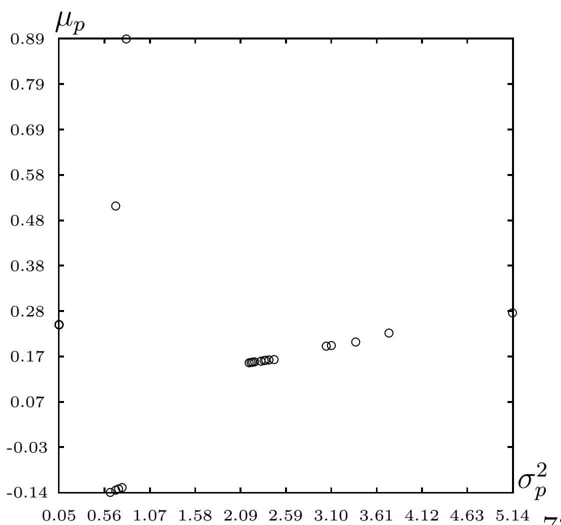
69



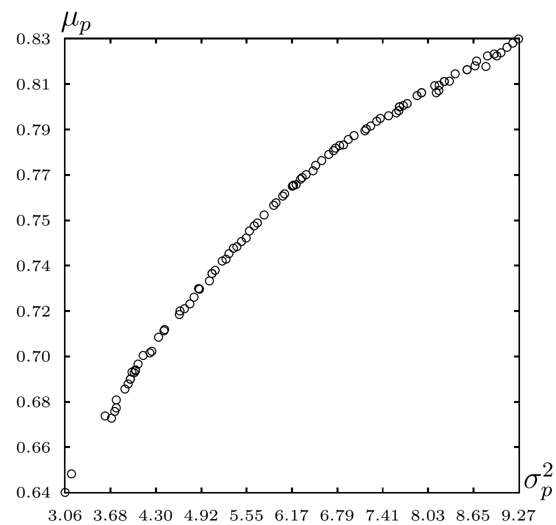
70



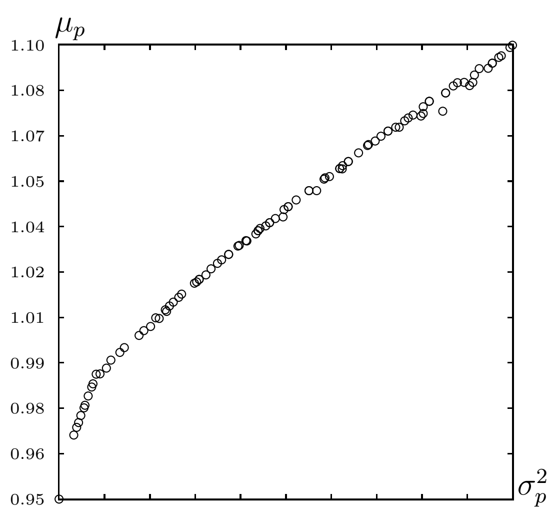
71



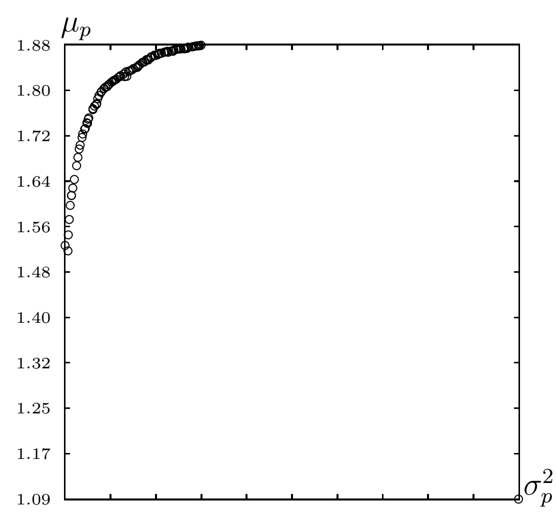
72



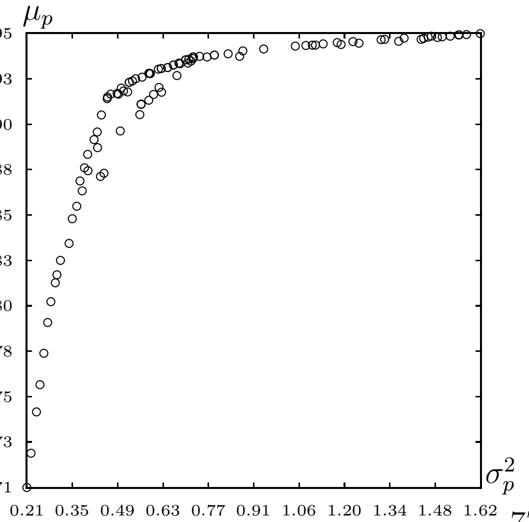
73



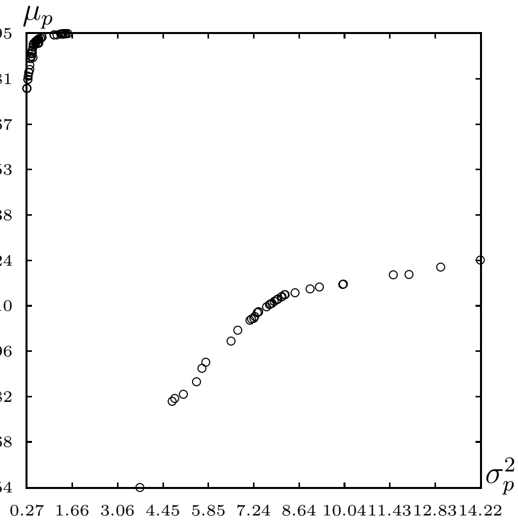
74



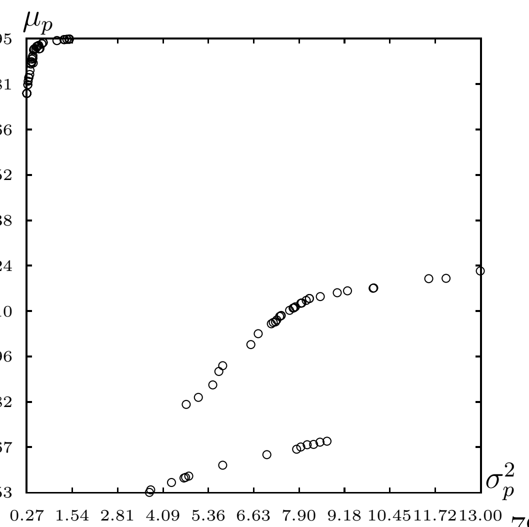
75



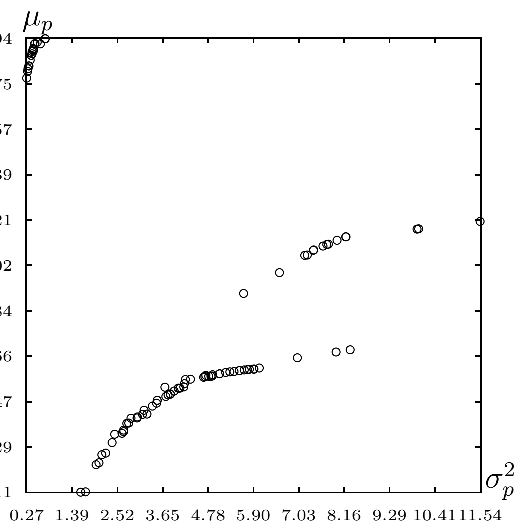
76



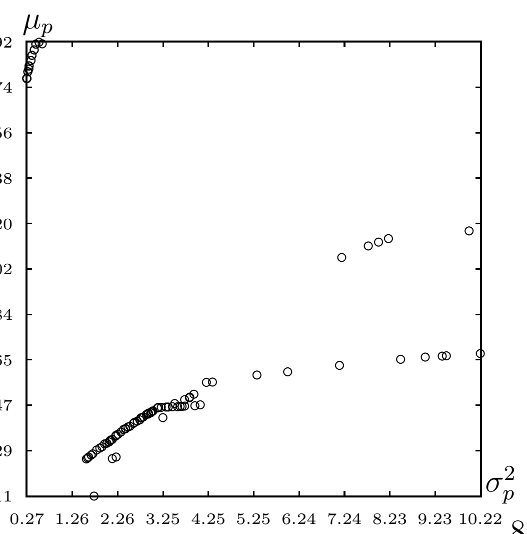
77



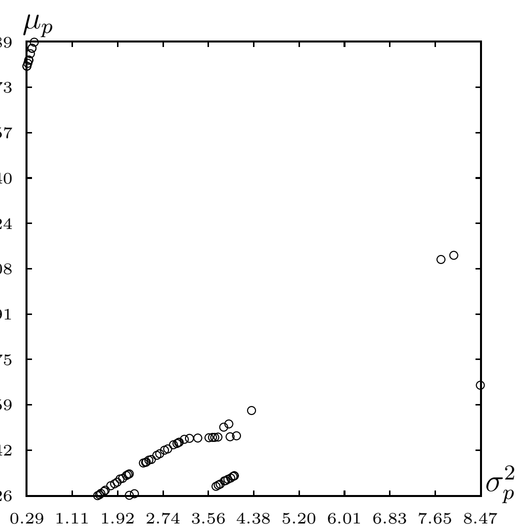
78



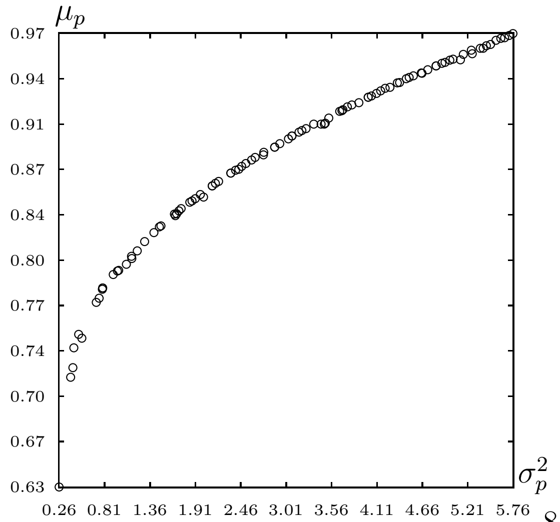
79



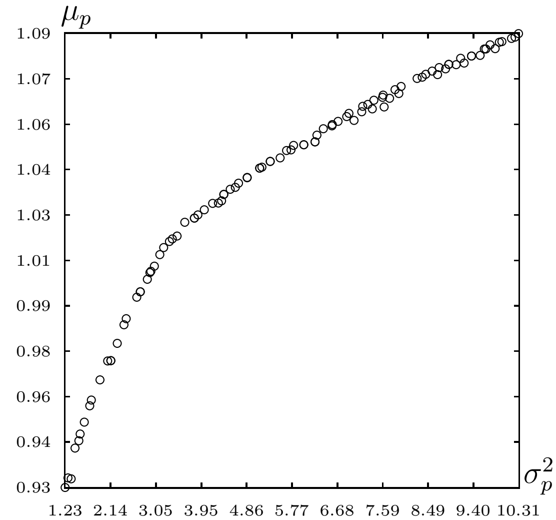
80



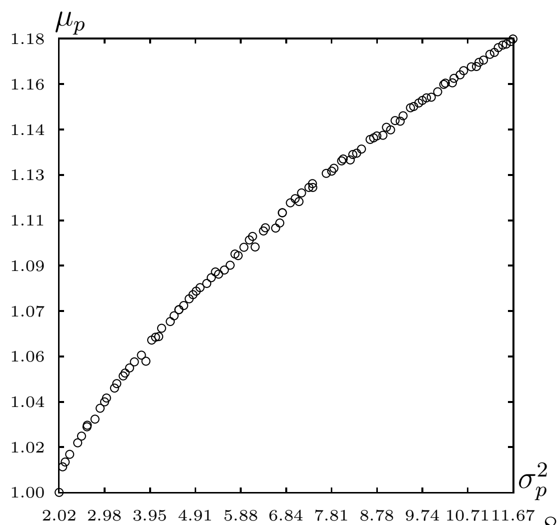
81



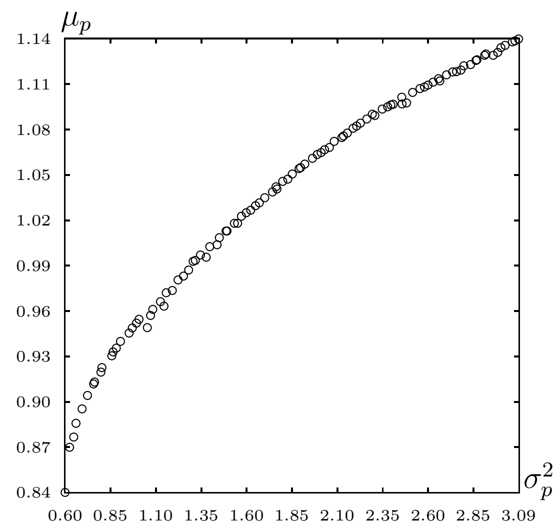
82



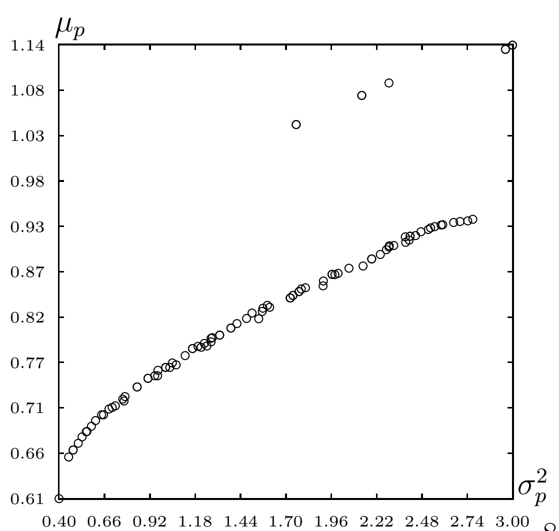
83



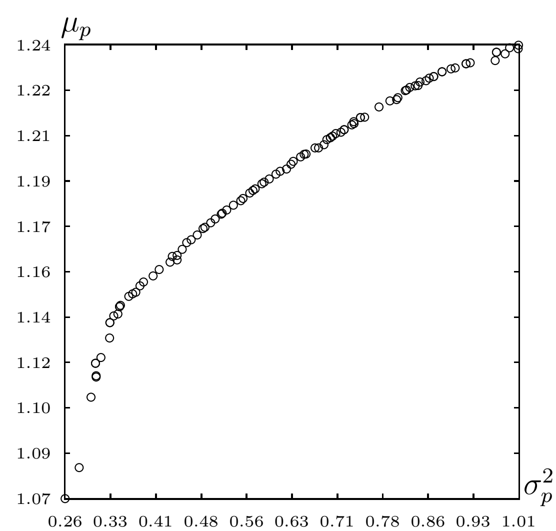
84



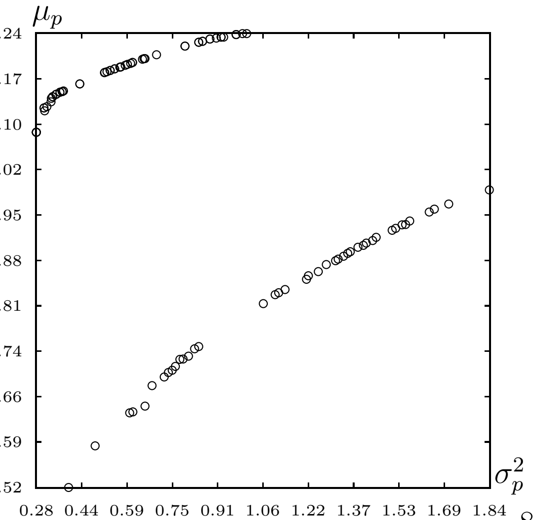
85



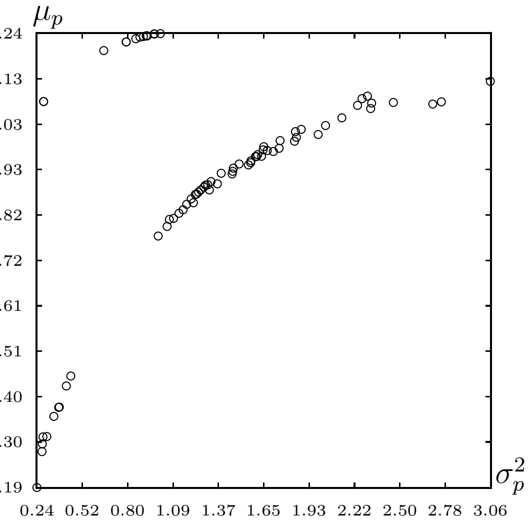
86



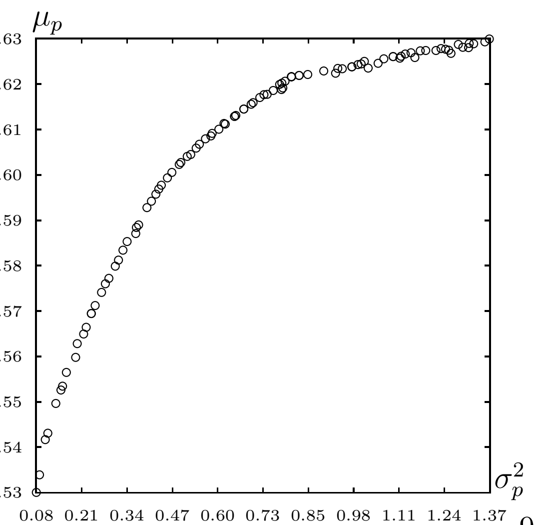
87



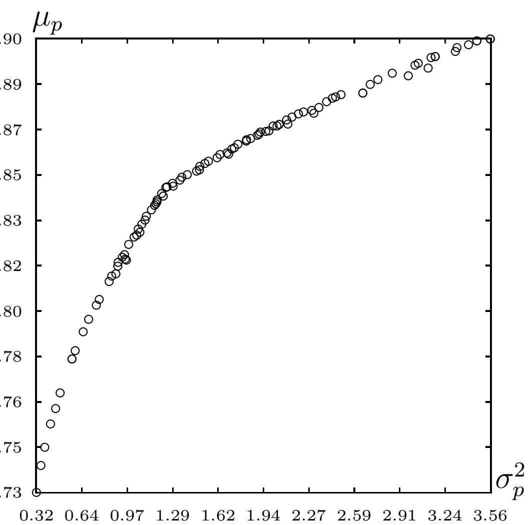
88



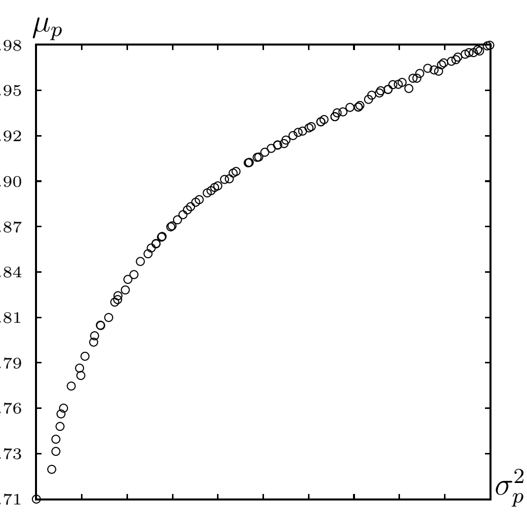
89



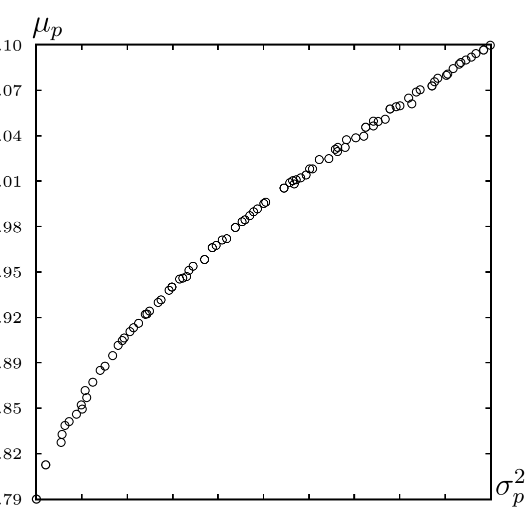
90



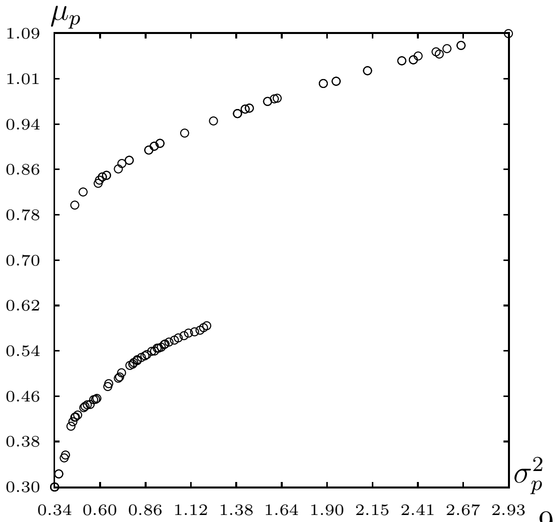
91



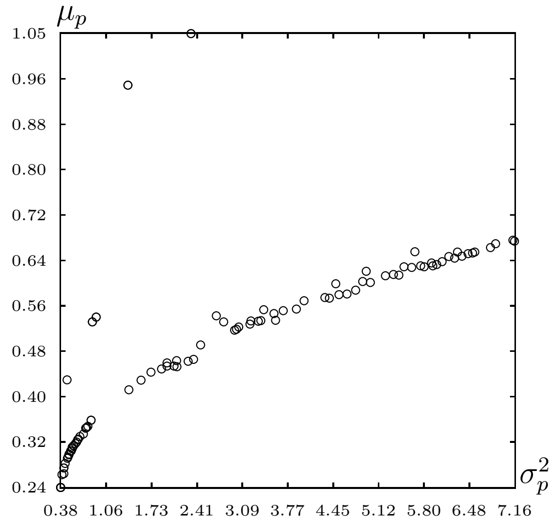
92



93


μ_p
1.09
1.01
0.94
0.86
0.78
0.70
0.62
0.54
0.46
0.38
0.30
σ_p^2
0.34 0.60 0.86 1.12 1.38 1.64 1.90 2.15 2.41 2.67 2.93

94


μ_p
1.05
0.96
0.88
0.80
0.72
0.64
0.56
0.48
0.40
0.32
0.24
σ_p^2
0.38 1.06 1.73 2.41 3.09 3.77 4.45 5.12 5.80 6.48 7.16

95