ALMA MATER STUDIORUM
UNIVERSITA' DI BOLOGNA

DOTTORATO DI RICERCA IN

INGEGNERIA ENERGETICA, NUCLEARE

E DEL CONTROLLO AMBIENTALE

ING-IND/10 Fisica Tecnica Industriale

XXI CICLO

MULTIOBJECTIVE GENETIC ALGORITHMS APPLIED TO HEAT TRANSFER PROBLEMS

Il Coordinatore del Corso di Dottorato

Chiar.mo Prof. Ing. Antonio Barletta

Il Relatore

Chiar.mo Prof. Ing. Giampietro Fabbri

Candidato

Ing. Diego Copiello

Abstract

In the present work, the multi-objective optimization by genetic algorithms is investigated and applied to heat transfer problems. Firstly, the work aims to compare different reproduction processes employed by genetic algorithms and two new promising processes are suggested. Secondly, in this work two heat transfer problems are studied under the multi-objective point of view. Specifically, the two cases studied are the wavy fins and the corrugated wall channel. Both these cases have already been studied by a single objective optimizer. Therefore, this work aims to extend the previous works in a more comprehensive study.

Nomenclature

μ	Dynamic viscosity
ρ	Fluid density
$\boldsymbol{x}$	Decision vector
$\boldsymbol{y}$	Objective vector
C	Generational distance
c	Specific heat
D	Density function
F	Spread vector parameter
f	Fitness function
h	Heat transfer coefficient
K	Convergence vector parameter
k	Thermal conductivity
N	Population size
Nu	Nusselt number
P	Parent population

p	Pressure
Pr	Prandtl number
Q	Children population
R	Raw fitness function
Re	Reynolds number
S	Strength function
X	Parameter space
Y	Objective space

Contents

Abstract **i**

Nomenclature **iii**

Table of contents **v**

1 Introduction **1**

1.1 Multi-objective optimization 2

1.1.1 Basic concepts and terminology 2

1.2 Traditional approaches 5

1.2.1 Weighting method 6

1.2.2 Constraint method 6

1.3 Evolutionary Algorithms 7

1.3.1 Elitism 9

1.4 Overview 10

2 Multi-Objective Genetic Algorithms **11**

2.1 Non-dominated Sorting
Genetic Algorithm II 12

2.1.1 Crowding distance assignment 13

3 Multi-Objective Differential Evolution **15**

3.1 Directional Information incorporated within a Multiobjective Differential Evolution Algorithm 16
3.2 Multiobjective Differential Evolution with Self-Adaption 18
3.3 Directional Crossover 19
3.4 Fitness-Based Crossover 20
3.5 Fitness-Based Differential Evolution 22
3.6 Performance assessment 23
3.7 Experiments 25
3.7.1 R1 test case 26
3.7.2 R2 test case 26
3.7.3 R3 test case 28
3.7.4 R4 test case 29
3.7.5 Results 29

4 Multi-objective optimization of longitudinal wavy fins 45
4.1 Governing Equations and Dimensioning Criteria 47
4.2 Results 52
4.2.1 Optimization without constraints 55
4.2.2 Optimization with constrained volume 57
4.3 Conclusions 58

5 Multi-objective optimization of corrugated wall channels 63
5.1 Governing equations and mathematical model 64
5.2 Results 68
5.3 Conclusions 74

6 Conclusions 77

Chapter 1

Introduction

Almost all engineering problems involve the simultaneous optimization of several and often conflicting objectives. In the case of single-objective optimization, the optimal solution is usually clearly defined. On the contrary, the solution of a multi-objective optimization problem consists on a set of trade-off alternatives, generally known as *Pareto-optimal* solutions. These solutions are optimal in the wider sense that no other solutions in the search space are superior to them when all objectives are considered.

This work aims to show the basic principles of multiobjective optimization. Moreover, since on author's opinion multiobjective genetic algorithms are not enough mature for industrial applications unless a large amount of computational resources is available, an analysis of promising reproduction processes employed by genetic algorithms and an improvement of them is presented since it is still an open issue in the research field. Finally, two problems involving the heat transfer are analyzed by means of the techniques developed by the author. Specifically, the multiobjective approach will help both to understand the physics of the problem considered or to identify particular geometries interesting from the application point of view.

In this chapter, the principles of multi-objective optimization are outlined by means of the formal definitions of its basic concepts. In the following, traditional approaches are discussed highlighting their drawbacks. Afterwards, a general description of evolutionary algorithms is presented. Finally, section 1.4 sketches the scopes of the present work and gives an overview of the remaining chapters.

1.1 Multi-objective optimization

1.1.1 Basic concepts and terminology

Multi-objective optimization problems are common in all the engineering fields. For example, consider the design of a complex heat exchanger device where a fluid cools heated solid surfaces. Usually the heat transfer rate is to be maximized, while the coolant power-pump is to be minimized. Moreover, both the cost of the device and its volume are to be minimized. These targets are clearly conflicting since an improvement of the heat transfer rate is accompanied by an increase of the pressure loss.

In general, a multi-objective optimization problem can be formally formulated as follows[1].

To describe a *multi-objective optimization problem* let us consider a vector function $\boldsymbol{f}$ which maps a vector variable $\boldsymbol{x}$ of parameters to a vector $\boldsymbol{y}$ of objectives. Formally, the problem of optimizing $\boldsymbol{y}$ can be written as

$$\min/\max\ \boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x})), f_2(\boldsymbol{x})), \ldots, f_n(\boldsymbol{x}))) \tag{1.1}$$

$$\text{subject to } \boldsymbol{g}(\boldsymbol{x}) = (g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \ldots, g_s(\boldsymbol{x})) \leq 0 \tag{1.2}$$

$$\text{where } \boldsymbol{x} = (x_1, x_2, \ldots, x_m) \in X \tag{1.3}$$

$$\boldsymbol{y} = (y_1, y_2, \ldots, y_n) \in Y \tag{1.4}$$

[1]The definitions and terms presented here correspond to mathematical formulations widespread in multi-objective literature, see, e.g., [36].

and $\boldsymbol{g}(\boldsymbol{x})$ indicates the constraints that the system is subject to and determines the set of feasible solutions, $\boldsymbol{x}$ is called the *decision vector*, X is the *parameter space*, $\boldsymbol{y}$ is the *objective vector*, and Y is the *objective space*.
A set of decision vectors constitutes a solution to a multi-criteria optimization problem if it agrees with the concept of *Pareto optimum* that was formulated by Vilfredo Pareto[23] and it constitutes the background of multi-objective optimization. Such a criterion states that a set of solutions of a multi-objective optimization problem consists of all decision vectors for which the components of the corresponding objective vectors cannot all improve simultaneously. In order to explain the concept of *Pareto optimality*, let us assume, without loss of generality, a maximization problem and let us consider two decision vectors $\boldsymbol{a}, \boldsymbol{b} \in X$. Then, $\boldsymbol{a}$ is said to *dominate* $\boldsymbol{b}$ (according to Zitzler and Thiele[30], written $\boldsymbol{a} \succ \boldsymbol{b}$) if and only if

$$\begin{aligned} &\forall i \in 1,2,\ldots,n : f_i(\boldsymbol{a}) \geq f_i(\boldsymbol{b}) \wedge \\ &\exists j \in 1,2,\ldots,n : f_j(\boldsymbol{a}) > f_j(\boldsymbol{b}) \end{aligned} \tag{1.5}$$

All decision vectors which are not dominated by any other decision vector of a given set are called *non-dominated* regarding to this set. Whereas, all decision vectors that are non-dominated within the entire search space are denoted as *Pareto optimal* and they constitute the so called *Pareto-optimal set*. The corresponding objectives instead constitute the *Pareto-optimal front* in the objective space. In order to illustrate these concepts, let us consider the following multi-criteria optimization problem:

$$\min f_1(\boldsymbol{x}) = (x_1^2 + x_2^2) - 2x_1 - 2x_2 \tag{1.6}$$

$$\max\ f_2(\boldsymbol{x}) = -\left((x_1^2 + x_2^2) + 2x_1 + 2x_2\right) \tag{1.7}$$

subject to the following constraints

$$x_1, x_2 \in [-\pi, \pi] \tag{1.8}$$

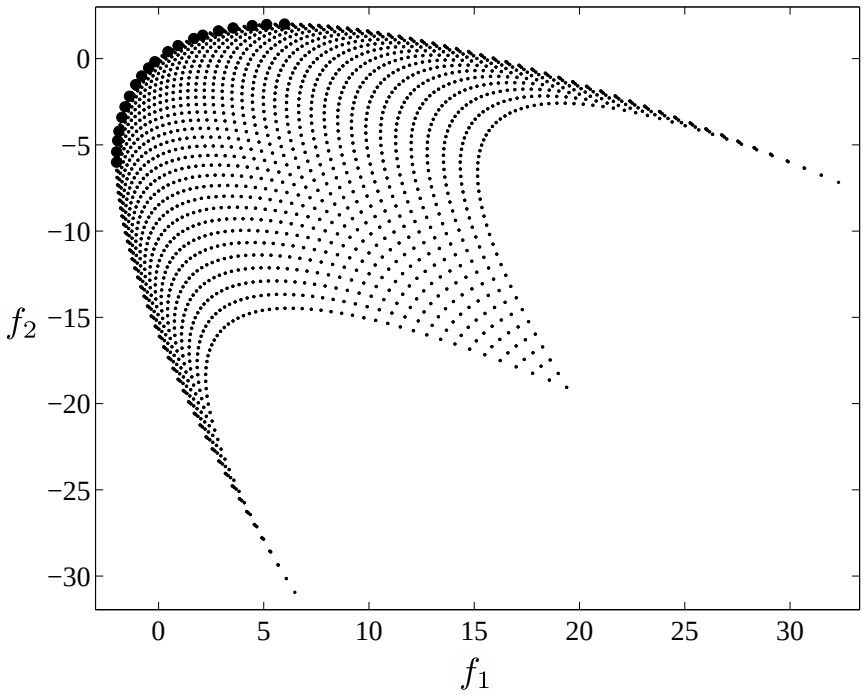


Figure 1.1: Objective example space.(·) Feasible region, (•) Pareto-optimal front.

In such a problem, both the minimum of function f_1 and the maximum of function f_2 are solutions of the problem. However, these two points do not coincide in the parameter space and therefore, both the two corresponding decision vectors are of interest. Moreover, between the f_1 minimum and f_2 maximum, a set of intermediate points can be found. All these points have the basic characteristic that none of the objectives can be improved without prejudicing the other. For example, let us consider the minimum of function f_1. In the objective space this point has the coordinates $f_1 = -2$ and $f_2 = -6$, a point with a higher value of f_2 can be found but the corresponding value of f_1 becomes inevitably higher since we are moving far away from the minimum of f_1. What makes a multi-objective optimization difficult to solve is the common situation when the individual optima corresponding to the distinct objective functions are sufficiently different. Then, the objectives are conflicting and they have to be optimized simultaneously. This simple example highlights that the single-objective optimality notion is not sufficient to solve such a problem. Instead, the notion of *Pareto-optimality* allows to define a set of solutions for a multi-objective problem. Figure 1.1 shows the objective space of this example and it highlights the Pareto-optimal front. It can be easily seen that the individuals belonging to the Pareto-front dominate all the other points in the objective space.

1.2 Traditional approaches

Classical methods for generating the Pareto-optimal front are typically based on the idea that the objective functions can be translated into a single objective function to be optimized. Such a function, for example, can result form a weighted summation of all the objective functions. Another way that can be employed to detect the Pareto front is to consider just a function to be optimized whereas the others can be handled as constraints that have to

be subsequently varied. These two methods are explained in the following together with their drawbacks.

1.2.1 Weighting method

The multi-criteria problem is converted into a single-objectives case by means of a linear combination of the objectives:

$$\text{optimize } y = f(\boldsymbol{x}) = w_1 f_1(\boldsymbol{x}) + w_2 f_2(\boldsymbol{x}) + \cdots + w_k f_k(\boldsymbol{x}) \tag{1.9}$$

$$\text{subject to } \boldsymbol{g}(\boldsymbol{x}) = (g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \ldots, g_s(\boldsymbol{x})) \leq 0 \tag{1.10}$$

w_i are known as weights and are typically normalized such that $\sum w_i = 1$. The solution of the problem can be achieved by means of a certain number of different weight combinations.
The main drawback of this technique is that it's not able to detect all solutions in the case of non-convex Pareto-fronts(see [36] for details).

1.2.2 Constraint method

In this method, one objective is optimized as usual whereas the other $k-1$ objectives are handled as constraints to be progressively changed. Formally, the problem can be reformulated as:

$$\text{maximize } y = f(\boldsymbol{x}) = f_h(\boldsymbol{x}) \tag{1.11}$$

$$\text{subject to } e_i(\boldsymbol{x}) = f_i(x) \geq \epsilon_i, \quad (1 \leq i \leq k, i \neq h) \tag{1.12}$$

The lower bounds, ϵ_i are the parameters that are varied by the optimizer in order to find multiple Pareto-optimal solutions. The main disadvantage of this method is that the values of the constraints ϵ_i must be known beforehand. As shown in figure 1.2, the function f_1 is to be maximized whereas function f_2 is handled as a constraint. However, if the value of f_2 fixed as a constraint is grater than $f_{2,max}$, the technique is not able to find any feasible solution. Therefore, $f_{2,max}$ must be known in advantage but this is not always possible.

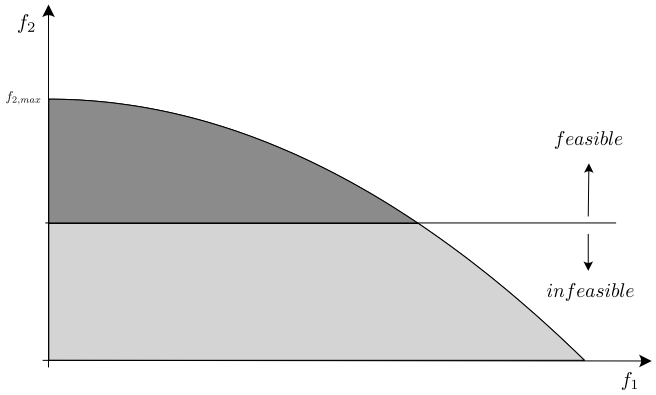


Figure 1.2: Example of the constraint method.

1.3 Evolutionary Algorithms

Evolutionary Algorithms (EA's) are a class of optimization technique which is inspired by natural evolution of species. In this section, ES's are briefly introduced with the aim to highlight their fundamentals that are based on Natural Selection[2].

Evolutionary Algorithms are an invaluable tool for the optimization and specifically they are successfully employed when:

- the objective functions have several local optima;
- no information about objective functions' derivatives are available;
- the optimization problem involves more than one objective function.

A generic EA starts from a population of candidate solutions (*individuals*) for a given problem. Individuals' objectives are evaluated and a quality performance is assigned to them. This measure is typically known as fitness and its definition depends on the algorithm considered. In some cases, objectives values are also referred to as fitness since in single-objective optimization the

[2]For further details on Natural Selection, the reader is referred to [12].

best elements of a population will be the individuals with better objective function value. In multi-objective optimization, the fitness in general will not coincide with the value of any of the objectives. Once the quality of the individuals is assigned, *better* elements are chosen in order to reproduce, once again the selection and the concept of *better individuals* depends on the specific algorithm employed. Regarding to the reproduction, offspring can be generated by means of two type of operations. The former is known as *crossover*. The latter is called *mutations*. In nature, the crossover is the process by which parents' genes are passed to their offspring and in an EA, the operation of generation of offspring from parents is therefore called crossover. The term *mutations* refers to a random change in an individual genes and they are employed to search unexplored zones of the search space. The processes of quality-evaluation, selection, and reproduction generate a new and improved population. This procedure continues until a stopping criteria is satisfied.

The generic EA above exposed highlights the key features of multi-objective optimization. In fact, the definition of *better individuals* must be addressed by means of the *fitness* function definition. In general, the individual *fitness* is typically defined as its degree of domination (i.e. non-dominated solutions have fitness equal to zero, individuals dominated by one solution have fitness equal to one, and so on). However, more complex definitions are provided in order to discriminate between equally ranked individuals. In addition to this, the following question arises:

- How better individuals are handled from one generation to another?

This question is answered by means of the *elitism* and this key feature is briefly exposed in the next subsection.

1.3.1 Elitism

Elitism is the strategy by which a certain number of best solutions are copied from the actual population into the population of the successive generation. In single objective optimization, De Jong[14] found that elitism can improve the performance of a single objective genetic algorithm on unimodal functions. On the contrary, in multimodal functions it may cause a premature convergence to a local optimum.
In multi-objective optimization, elitism plays a fundamental role and it has widely demonstrated that it improves the optimization performances[36][3]. However, the incorporation of elitism in a multi-objective algorithm is more complex than in single objective optimization. specifically, instead of one best individual, there is an elite set whose size can be considerable compared to the population. This fact, involves two questions which must be answered in a genetic algorithm definition:

- Which individuals are kept and for how long are retained in the elite set?

- How are elite members reinserted into the population?

In general, two main approaches can be found in literature. One strategy is to copy a certain number of non-dominated solutions to the new population. Another strategy consists on copying non-dominated solutions to an external population. Thereafter, at each generation a certain percentage of the population is filled up or replaced by members of the external population

[3]It is worth noting that premature convergence of a genetic algorithm is also important in multi-objective optimization. However, this can be handled by employing appropriate mating schemes and/or reproduction operators.

1.4 Overview

The present document is organized as follows: in chapter 2 the basis of the genetic algorithm employed are highlighted, in chapter 3 the multi-objective differential evolution is described and several reproduction processes are evaluated. In chapters 4 and 5 two basic problems involving the heat transfer are studied. Specifically, the cases of wavy fins and corrugated channels are analyzed from a multi-objective point of view. Finally, brief conclusions are enlisted in chapter 6.

Chapter 2

Multi-Objective Genetic Algorithms

In the present chapter, a description of a multi-objective genetic algorithms is presented: the Non-dominated Sorting Genetic Algorithm II (NSGA-II). This algorithm represents the state of the art in multi-objective optimization. In the present chapter the main framework of the algorithm is described whereas, the following chapter deals with the reproduction processes found in literature or suggested by the author.

It is worth noting that in literature other interesting multi-objective genetic algorithms can be found such as the Strength of Pareto Evolutionary Algorithm 2 (SPEA2). In fact, this algorithm has been successfully used by the author to a case where the heat transfer was involved[6]. However, since this document aims to highlight the differences between promising reproduction methods, the main framework of the genetic algorithm was kept the same for all the cases tested. Moreover, the NSGA-II was chosen since almost all the reproduction methods found in literature were proposed on the basis of the NSGA-II.

2.1 Non-dominated Sorting Genetic Algorithm II

The NSGA-II is an elitist multi-objective genetic algorithm developed by Deb et al.[21]. The NSGAII starts from an initial population P_0 of N individuals. Once all the elements are evaluated, the population is sorted on the basis of the non-domination level of its individuals. Thereafter, each solution is assigned a *fitness* or *rank* equal to its non-domination level (1 is the best, 2 is the next level and so on). Thus, minimization of fitness is assumed. At first, the usual binary tournament selection, recombination and mutation operators are used to create a children population Q_0 of size N. Since elitism is introduced by comparing current population with previous-found best non-dominated solutions, the procedure is different after the initial generation and it is enlisted in the following:

- combine parent with children population:

$$R_t = P_t \cup Q_t; \tag{2.1}$$

- sort the individuals in R_t according to their non-domination level;
- set the next-generation parent population P_{t+1} to an empty archive;
- fill P_{t+1} with non-dominated individuals of population P_t according also to their crowding distance (cfr sec. 2.1.1). If there are more individuals than required in the last front than select from it less crowded individuals (the size of P_{t+1} is equal to N);
- generate offspring population Q_{t+1} from P_{t+1}
- end if a stopping criteria is satisfied otherwise return to step 1.

2.1.1 Crowding distance assignment

Along with convergence to the Pareto-optimal set, it is also desired that an evolutionary algorithm maintains a good spread of solutions in the obtained set of solutions. To this aim, when the algorithm is required to chose a certain maximum number of solutions in a front, it must be able to discriminate too crowded elements from poorly crowded solutions. Therefore, to get an estimate of the density of solutions surrounding a particular individual in the population, the average distance of two points in the front on either side of the point of interest along each of the objective is calculated. Such a quantity, labelled as i_{dist} and called crowding-distance, serves as an estimate of the size of the largest cuboid enclosing the point i without including any other point in the front (see Fig. 2.1). In order to compute the crowding

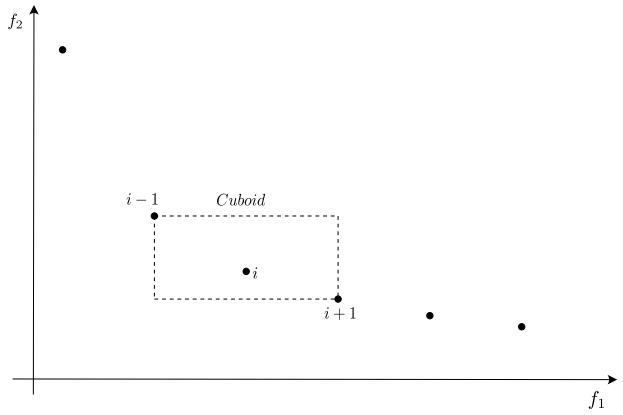


Figure 2.1: Crowding distance computation.

distance of an individual i, it is required to sort the population according the distance from i in each the objective function value and in their ascending order. Thereafter, for each objective function, the boundary solutions (i.e. solutions with smallest and largest function values) are assigned an infinite value. All other intermediate solutions are assigned a distance value equal to the absolute difference in the function values of two adjacent solutions.

This calculation is continued with the other objective functions. The overall crowding distance value of a solution is computed as the sum of its distances calculated for each objective.

Chapter 3

Multi-Objective Differential Evolution

The Differential Evolution (DE) is a population based optimization algorithm that showed important advantages in single objective optimization since it demonstrated to have globally and locally correlated step-sizes. Specifically, the step-sizes self-adapt over time in relation to the location of the population of individuals in the search space, resulting in an extremely efficient search. Moreover, DE has demonstrated to be invariant under a rotation of the decision space[25]. Therefore, it can manage problems influenced by non-separable parameters. It is worth noting that many real-world problems have such parameter interactions. As a result, DE is an ideal choice for optimizing engineering problems. In the last decade, researchers has successfully applied DE to multi-objective problems [1, 2, 28, 22, 34, 35], but the application of the DE scheme in these cases was the same of the one typically employed in single-objective optimization.

More recently, some researchers proposed different variants of the original DE leveraging aspects of the multiobjective domain[17][33]. Specifically Iorio and Li proposed three variants of the DE and identified the best performance al-

gorithm as a scheme where the DE computation was given by two terms: the former controls the convergence and the latter influences the spread of the population. The DE parameter setting was fixed and resulting from considerations found in literature and related to single-objective optimization. Zamuda et al. modified DE by considering control parameters that adapt at each step by means of information provided by parents that take part in the generation process.
In this work, an analysis of the effect of different fixed parameters settings has been performed. Moreover, it has been considered a procedure with self-adapting parameters similar to that of Zamuda et al. and it was compared with previous cases. In addition to this, a comparison with other reproduction techniques has been performed. Specifically, it has been employed the Directional Crossover[24] and two new crossover schemes, called Fitness-Based crossover, has been suggested and used alone either within the scheme of Iorio and Li.

3.1 Directional Information incorporated within a Multiobjective Differential Evolution Algorithm

The idea of incorporating directional information within differential evolution in a multiobjective genetic algorithm was introduced by Iorio and Li in 2006. In their paper, three type of Differential Evolution Algorithm was tested with the NSGA-II as evolutionary algorithm[21]. In the best performing scheme (NSDE-DCS), the DE calculation was composed by two terms: the former controls the convergence and the latter influences the spread of the population. Specifically, the generation of offspring were done by means the following steps:

1. $i = 1$;

2. If there exists an r_1 and r_2 such that the ranks of x_{r1} and x_{r2} are equal, and $r_1 \neq r_2 \neq i$, where $r_1, r_2 \in \{1, 2, \ldots, N\}$, select x_{r1} and x_{r2} for the DE calculation. If there exists an r_3 such that the rank of x_{r3} is less than the rank of x_i, and $r_3 \neq r_1 \neq r_2 \neq i$, where $r_3 \in \{1, 2, \ldots, N\}$, select r_3 for the DE calculation.

3. Generate offspring u_i as

$$u_i = x_i + K \cdot (x_{r3} - x_i) + F \cdot (x_{r1} - x_{r2}) \,; \tag{3.1}$$

4. $i = i + 1$;

5. Repeat from step 2 until $i = N$.

where the rank of an individual corresponds to its level of dominance (i.e. non-dominated individuals have rank equal to one and so on). The K-term generates *convergence vectors* which point towards regions where better ranked individuals are located. Whereas, the F-term is a differential *spread vector*, and should contribute to the generation of offspring which are spread out across a non-dominated front. K and F are scaling factors responsible for the magnitude of *convergence vectors* and *spread vectors* respectively. In their paper, Iorio and Li suggested values of $K = 0.4$ and $F = 0.8$. Such values are fixed and resulted from considerations found in literature and related to single-objective optimization[25]. therefore, a sensitivity analysis of the effects of these parameters is mandatory is order to asses the performance of this crossover.

3.2 Multiobjective Differential Evolution with Self-Adaption

In 2007, Zamuda et al.[33] introduced the concept of self-adaption of the of two factors employed during the reproduction process of the Differential Evolution Multiobjective Algorithm(DEMO)[11]. The first factor is an amplification term that controls the differential computation, whereas, the latter controls the selection procedure of the variable to be mutated. For each individual i in the current population G, the vector $v_{i,G+1}$ is created as follows:

$$v_{i,G+1} = x_{r1,G} + F_{i,G+1} \cdot (x_{r2,G} - x_{r3,G}) \tag{3.2}$$

where $i \neq r_1 \neq r_2 \neq r_3$, and the amplification factor $F_{i,G+1}$ is computed as follows

$$F_{i,G+1} = \langle F_G \rangle_i e^{\tau N(0,1)} \tag{3.3}$$

where τ is a learning factor and it is equal to $1/8\sqrt{2D}$, D being a dimension of the problem. $N(0,1)$ is a random number with a Gauss distribution. The $\langle F_G \rangle_i$ denotes averaging the parameter F of the individuals chosen:

$$\langle F_G \rangle_i = \frac{F_{i,G} + F_{r1,G} + F_{r2,G} + F_{r3,G}}{4} \tag{3.4}$$

Similarly, a parameter $CR_{i,G+1}$ is computed by means of the follwing relations:

$$CR_{i,G+1} = \langle CR_G \rangle_i e^{\tau N(0,1)} \tag{3.5}$$

$$\langle CR_G \rangle_i = \frac{CR_{i,G} + CR_{r1,G} + CR_{r2,G} + CR_{r3,G}}{4} \tag{3.6}$$

The i-th offspring $u_{i,G+1}$ is computed by

$$u_{i,j,G+1} = \begin{cases} v_{i,j,G+1} & \text{if } rand(0,1) \leq CR_{i,G+1} \\ x_{i,j,G} & \text{otherwise} \end{cases} \tag{3.7}$$

where $j \in [1, D]$ denotes the j-th parameter, and $rand(0,1) \in [0,1]$ denotes an uniformly distributed random number. Zamuda et al. bounded the parameters F and CR as

$$0.1 \leq F \leq 0.9 \tag{3.8}$$

$$1/24 \leq CR \leq 0.3 \tag{3.9}$$

Their initialization was $F = 0.5$ and $CR = 0.3$.

3.3 Directional Crossover

The basic idea that lies beneath the Directional Crossover (DC) consists in the concept that offspring should be guided toward the Pareto frontier considering information about objectives. Specifically, at each generation and each individual i, two elements $\boldsymbol{x}_{i1}$ and $\boldsymbol{x}_{i2}$ are selected and the offspring is generated as:

$$\begin{aligned} \boldsymbol{x} = \boldsymbol{x}_i + S \cdot sign(F - F_1) \cdot (\boldsymbol{x}_i - \boldsymbol{x}_{i1}) + \\ + T \cdot sign(F - F_2) \cdot (\boldsymbol{x}_i - \boldsymbol{x}_{i2}) \end{aligned} \tag{3.10}$$

where T and S are random numbers between 0 and 1, $F = \vec{F}_i \times \vec{F}_i$, $F_1 = \vec{F}_i \times \vec{F}_{i1}$ and $F_2 = \vec{F}_i \times \vec{F}_{i2}$. Moreover, $\vec{F}_j$ is a vector whose components are the objectives of the individual j. For further details about the meaning of Eq. 3.10, the reader is referred to [24], however it is worth to note that in this way, global information about the objectives landscape is retained and employed in the generation step. Individuals $\boldsymbol{x}_{i1}$ and $\boldsymbol{x}_{i2}$ can be selected by means of any selection scheme and in the this work a tournament selection scheme has been chosen.

3.4 Fitness-Based Crossover

In this section, the Fitness-Based Crossover (FBC) developed by the author is exposed. The basic concept consists in the idea that, during the reproduction process, populations should migrate towards regions poorly populated. In fact, individuals belonging to areas with high population density shares with more individuals the available resources and therefore, such individuals should be penalized in the generation process. A similar concept was used in the niching techniques named *fitness sharing*[4]. Moreover, we aim to guide the reproduction process to generate individuals as close the Pareto front as possible. Before showing how offspring are generated, the concept of fitness of an individual as described by Zitzler et al.[31] is explained. In order to compute the fitness, three values for each individual i must be first calculated: the strength $S(i)$, the raw fitness $R(i)$ and the density $D(i)$. Firstly, Zitzler et al. define the strength of the i-th individual $S(i)$ as the number of solutions it dominates:

$$S(i) = \left| \left\{ j \mid j \in \boldsymbol{P}_t + \bar{\boldsymbol{P}}_t \wedge i \succ j \right\} \right| \tag{3.11}$$

where $|\cdot|$ indicates the cardinality of the set, $+$ stands for the multiset union and the symbol $\succ$ corresponds to the Pareto dominance relation. Once the strength is evaluated, the value of the raw fitness of the individual i can be computed as the sum of the strengths of its dominators:

$$R(i) = \sum_{j \in \boldsymbol{P}_t + \bar{\boldsymbol{P}}_t, j \succ i} S(j) \tag{3.12}$$

In order to illustrate how the strength and the raw fitness assignments work, in figure 3.1 these values are reported close to each element of a population. The problem depicted consists in maximizing both the cost function F_1 and the cost function F_2. Since elements lying near the Pareto front should be more probably preferred during reproduction, then elements with lower raw fitness values will be chosen with a higher probability than individuals with

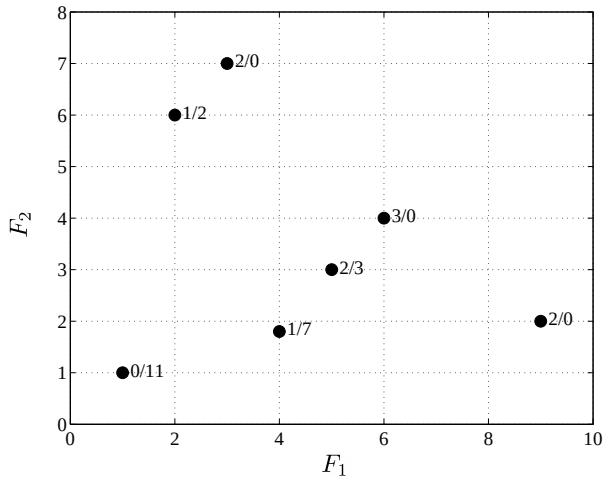


Figure 3.1: Strength $S(i)$ and Raw $R(i)$ fitness assignment. Both objectives F_1 and F_2 are to be maximized. Points are plotted with $S(i)/R(i)$ near the i-th individual.

higher values of $R(i)$. However, further information must be also included in order to discriminate between individuals with the same values of R. This can be done including informations about individual density in the objective space to the raw fitness value. The density estimation technique provided Zitzler et al. is an adaption of the k-th nearest neighbor method [29]. Specifically, for each element i the distance between it and all the other individuals in the objective space is stored in a list. After having sorted this list in ascending order, the k-th distance σ_i^k is chosen. The value of k is equal to the square root of the sample size:

$$k = \sqrt{2N} \tag{3.13}$$

where N is the population size. The density of the i-th individual is then defined by the following relation

$$D(i) = \frac{1}{\sigma_i^k + 2} \tag{3.14}$$

Finally, the fitness value of the i-th individual can be computed as:

$$f(i) = R(i) + D(i) \tag{3.15}$$

From the basic concepts above outlined, the genetic operator described in the following has been formulated:

1. Select elements $\boldsymbol{i}_1$ and $\boldsymbol{i}_2$ with fitness functions values f_{i1} and f_{i2} respectively.

2. From the two individuals selected, set $\boldsymbol{p}_1$ as the parent with the lower value of the fitness function (labelled as f_1) whereas, set $\boldsymbol{p}_2$ as the parent with the higher value of the fitness function (labelled as f_2).

3. Compute offspring as follows

$$\boldsymbol{o}_1 = \lambda\left(\frac{f_2}{f_1}\boldsymbol{p}_1 + \frac{f_1}{f_2}\boldsymbol{p}_2\right) \tag{3.16}$$

$$\boldsymbol{o}_2 = \lambda\left(\frac{f_2}{f_1}\boldsymbol{p}_1 - \frac{f_1}{f_2}\boldsymbol{p}_2\right) \tag{3.17}$$

   where λ is a random number between 0 and 1.

It can be seen that the FBC is able to reproduce towards areas poorly populated. In fact, for two parents with equal ranks (i.e. with equal raw fitness), the individual with higher density value is penalized with respect to the other one since the resulting offspring are nearer to $\boldsymbol{p}_1$ than $\boldsymbol{p}_2$ as depicted in figure 3.2. Moreover, in the case that two individuals with different ranks reproduce, the crossover generate offspring that will be mainly in the direction of the less dominated parent.

3.5 Fitness-Based Differential Evolution

The Fitness-Based Differential Evolution (FB-DE) conjugate the Fitness-Based crossover with the differential evolutionary scheme of Iorio and Li

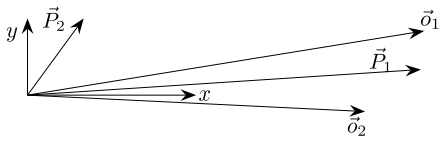


Figure 3.2: Offsprings $\vec{o}_1$ and $\vec{o}_2$ generated by two parents $\vec{P}_1$ and $\vec{P}_2$ in the case of f_1 lower than f_2.

[17]. Specifically, the FB-DE crossover steps are equal to those enumerated in section 3.1, except for equation 3.1 that is replaced with the following relations in order to create two offspring u_1 and u_2:

$$u_1 = K\left(\frac{f_{r3}}{f_i}\boldsymbol{x}_i + \frac{f_i}{f_{r3}}\boldsymbol{x}_{r3}\right) + F\left(\boldsymbol{x}_{r1} - \boldsymbol{x}_{r2}\right) \tag{3.18}$$

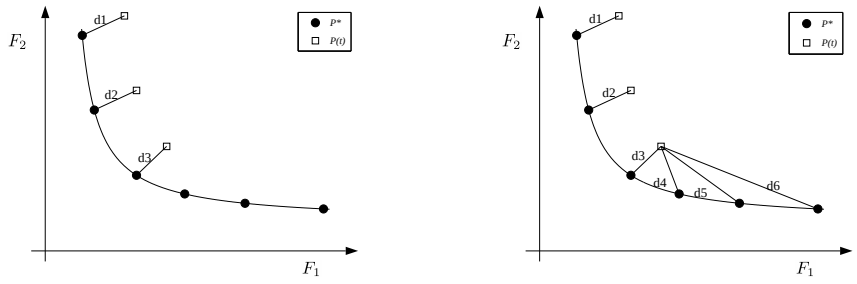
$$u_2 = K\left(\frac{f_{r3}}{f_i}\boldsymbol{x}_i - \frac{f_i}{f_{r3}}\boldsymbol{x}_{r3}\right) + F\left(\boldsymbol{x}_{r1} - \boldsymbol{x}_{r2}\right) \tag{3.19}$$

Similarly to the NSDE-DCS, each offspring is constituted by a *convergence vector* (controlled by parameter K) and by a *spread vector* (controlled by parameter F).

3.6 Performance assessment

A set of solutions generated by a multiobjective evolutionary algorithm must be evaluated using at least two performance measures. In fact, a solution set can rapidly converge to the Pareto optimal set but, on the other hand, the coverage of the front can be poor (see Fig. 3.3). Up to now, a large number of metrics has been suggested and an interesting comparative study was performed by Zitzler et al.[3].

In the performance assessment, we employed the metrics suggested by Iorio



(a) $C(P(t), P^*)$ (b) $C(P^*, P(t))$

Figure 3.3: Metrics performance. Example of a solution set that poorly covers the Pareto front ($C(P^*, P(t)) > C(P(t), P^*)$)

and Li[17] for the convergence evaluation and briefly recalled in the following. Given a target set of points P^* and the population at each generation $P(t)$, the smallest normalized Euclidean distance to P^* for each point i in $P(t)$ is

$$d_i = \min_{j=1}^{|P^*|} \sqrt{\sum_{k=1}^{M} \left(F_{k,i} - F^*_{k,j}\right)^2} \tag{3.20}$$

where M is the number of objectives involved and $F_{k,i}$ is the k-th objective of the i-th individual. Therefore, the convergence metric can be calculated as

$$C\left(P(t), P^*\right) = \frac{1}{|P(t)|} \sum_{i=1}^{|P(t)|} d_i \tag{3.21}$$

Eq.3.21 is also referred to as *generational distance*. Iorio and Li used the *generational distance* since it can be also used for the assessment of the Pareto optimal front coverage. This can be done by swapping the reference set P^* with the set $P(t)$ and the related equations are:

$$d_i = \min_{j=1}^{|P(t)|} \sqrt{\sum_{k=1}^{M} \left(F^*_{k,i} - F_{k,j}\right)^2} \tag{3.22}$$

$$C\left(P^*, P(t)\right) = \frac{1}{|P^*|} \sum_{i=1}^{|P^*|} d_i \tag{3.23}$$

With the metric $C(P^*, P(t))$, the distance of the solutions in the Pareto optimal set P^* to a non dominated set $P(t)$ can be measured. Figure 3.3 depicts how the two metrics work in the case of a population that is near the Pareto front but badly covers it. The optimization of a multiobjective problem will be acceptable only if both the two metrics tends to zero. However, it is worth noting that the two metrics becomes zero if they exactly match the reference set, but this condition is almost impossible.

Another characteristic that a multi-objective algorithm must satisfy is obviously the robustness. We may regard to robustness as the effective capability of the algorithm to reach the optimal Pareto front within a maximum number of generations. Therefore, if an algorithm is robust, the two metrics previously described have to be almost zero at the end of the simulation.

To evaluate the performance of the algorithm employed in this work, for each of the two generational distances, the mean and the standard deviation are computed. Specifically, for each test case and for each algorithm, 25 runs are performed and the statistics are calculated on the resulting data. Moreover, to assess the robustness performance, the last values of the mean histories of the metrics are also provided.

3.7 Experiments

The DE schemes previously exposed have been tested on four rotated test problems developed by Iorio and Li[18] in order to compare performances of the algorithms considered. Aiming to limit differences between the optimization procedures only to the generation process, all the procedures have been used in the NSGA-II[21] framework. However, the original DEMOwSA employed the SPEA2 algorithm[31]. In order to avoid confusion between the original DEMOwSA and the current configuration, the self-adaption procedure employed here is referred to as NSDEwSA.

3.7.1 R1 test case

Problem $R1$ is characterized by a valley in objective f_2. The Pareto-optimal set is situated along the length of this valley as well, and when the problem is subject to a rotation the valley can trap a non-rotationally invariant search from progressing along it. The function f_1 is linear and f_2 is non-linear. The $R1$ problem is described as follows

$$f_1(\boldsymbol{y}) = y_1 \tag{3.24}$$

$$f_2(\boldsymbol{y}) = g(\boldsymbol{y}) \cdot h(f_1(\boldsymbol{y}), g(\boldsymbol{y})) \tag{3.25}$$

$$h(f_1(\boldsymbol{y}), g(\boldsymbol{y})) = exp\left(\frac{-f_1(\boldsymbol{y})}{g(\boldsymbol{y})}\right) \tag{3.26}$$

$$g(\boldsymbol{y}) = 1 + 10(m-1) + \sum_{i=2}^{m}\left[y_i^2 - 10cos(4\pi y_i)\right] \tag{3.27}$$

$$\boldsymbol{y} = \boldsymbol{R}\boldsymbol{x}, -0.3 \leq x_i \leq 0.3, \; for\, i = 1, 2, \ldots, m \tag{3.28}$$

$$|f_1| \leq 0.3 \tag{3.29}$$

where $\boldsymbol{R}$ is a rotation matrix. Specifically, a uniformly distributed random rotation is used to introduce parameter interdependencies into a problem by rotating the parameter vector $\boldsymbol{x}$. This transformation does not change the objective landscape of the problem domain since it is an isometry transformation, therefore, it preserves distances between points and angles. Figure 3.4 shows the Pareto-optimal front of problem R1.

3.7.2 R2 test case

Problem $R2$ is similar to the ZTD3 problem[21], and has a Pareto optimal front which is not continuous. $R2$ represents a difficulty to an optimization algorithm because it has to locate a number of discontinuous Pareto-optimal fronts, and maintain solutions in each of those fronts. When $R2$ is rotated an optimization algorithm which only searches independently along the principle coordinate axes will generate non-dominated solutions which skew away

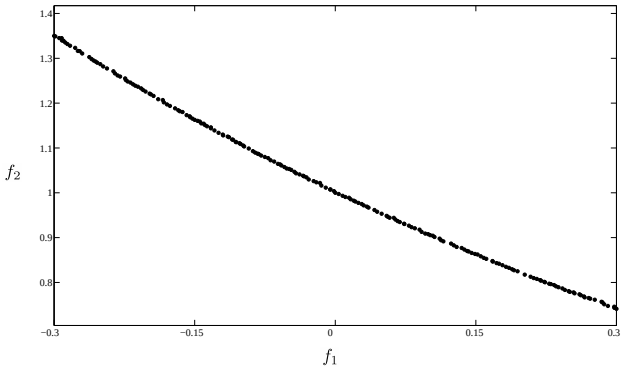


Figure 3.4: Pareto-optimal front of R1 problem.

significantly from the Pareto-optimal front. The reason for this behavior is that perturbed solutions have to travel quite far along the principle coordinate axes before an independent perturbation can generate a solution which dominates the current non-dominated set. The function f_1 is linear and f_2 is non-linear.

$$f_1(\boldsymbol{y}) = y_1 \tag{3.30}$$

$$f_2(\boldsymbol{y}) = g(\boldsymbol{y}) \cdot h(f_1(\boldsymbol{y}), g(\boldsymbol{y})) \tag{3.31}$$

$$\begin{aligned} h(f_1(\boldsymbol{y}), g(\boldsymbol{y})) = 1.0 \quad &+ \quad exp\left(\frac{-f_1(\boldsymbol{y})}{g(\boldsymbol{y})}\right) \\ &+ \quad \left(\frac{f_1(\boldsymbol{y}) + 1.0}{g(\boldsymbol{y})}\right) sin(5\pi f_1(\boldsymbol{y})) \end{aligned} \tag{3.32}$$

$$g(\boldsymbol{y}) = 1 + 10(m-1) + \sum_{i=2}^{m} \left[y_i^2 - 10cos(\pi y_i)\right] \tag{3.33}$$

$$\boldsymbol{y} = \boldsymbol{R}\boldsymbol{x}, -1.0 \leq x_i \leq 1.0, \; for\, i = 1, 2, \ldots, m \tag{3.34}$$

$$|f_1| \leq 1.0 \tag{3.35}$$

where $\boldsymbol{R}$ is uniformly distributed random rotation matrix. In figure 3.5 the Pareto-optimal front of problem $R2$ is depicted.

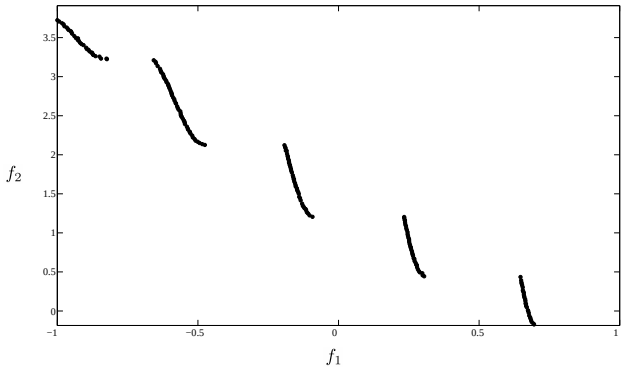


Figure 3.5: Optimal front of R2 problem.

3.7.3 R3 test case

In test case $R3$, decision space variables increment at a regular interval, evaluate with non-regular intervals in the objective space making it hard to find a uniform distribution along the Pareto-optimal front. Problem R3 is similar to the ZTD6 problem[21]. f_1 and f_2 are non-linear functions. The following relations describe the R3 test case.

$$f_1(\boldsymbol{y}) = 1.0 - exp(2.0y_1)sin^6(6\pi y_1)/9.0 \tag{3.36}$$

$$f_2(\boldsymbol{y}) = g(\boldsymbol{y}) \cdot h(f_1(\boldsymbol{y}), g(\boldsymbol{y})) \tag{3.37}$$

$$h(f_1(\boldsymbol{y}), g(\boldsymbol{y})) = 1.0 - exp\left(\frac{f_1(\boldsymbol{y})}{g(\boldsymbol{y})}\right)^2 \tag{3.38}$$

$$g(\boldsymbol{y}) = 1 + 10(m-1) + \sum_{i=2}^{m}\left[y_i^2 - 10cos(\pi y_i)\right] \tag{3.39}$$

$$\boldsymbol{y} = \boldsymbol{R}\boldsymbol{x}, -1.0 \le x_i \le 1.0, \; for\, i = 1, 2, \ldots, m \tag{3.40}$$

$$0.3 \le f_1 \le 1.0 \tag{3.41}$$

Figure 3.6 shows the Pareto-optimal front of the $R3$ test case.

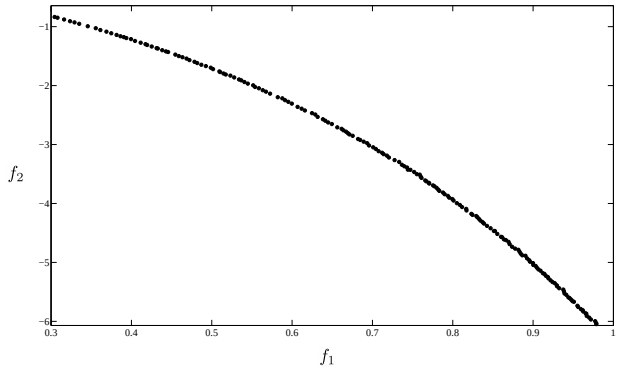


Figure 3.6: Optimal front of R3 problem.

3.7.4 R4 test case

Problem *R4* is based on the Schwefel function[21], where local front is located far from the global minimum. Points are easily trapped by this deceptive front. *R4* is difficult, and objective f_2 is characterized by a number of valleys, including the highly deceptive valleys far from the true Pareto-optimal front. These valleys correspond to the modalities generated by function $g(\boldsymbol{y})$. Each of these valleys can trap points in a sub-optimal non-dominated front.

$$f_1(\boldsymbol{y}) = y_1 \tag{3.42}$$

$$f_2(\boldsymbol{y}) = g(\boldsymbol{y}) \cdot h(f_1(\boldsymbol{y}), g(\boldsymbol{y})) \tag{3.43}$$

$$h(f_1(\boldsymbol{y}), g(\boldsymbol{y})) = exp\left(\frac{-f_1(\boldsymbol{y})}{g(\boldsymbol{y})}\right) \tag{3.44}$$

$$g(\boldsymbol{y}) = 1 + 0.015578(m-1) + \sum_{i=2}^{m}\left[y_i^2 - 0.25\left(y_i sin\left(32.0\sqrt{|y_i|}\right)\right)\right] \tag{3.45}$$

$$\boldsymbol{y} = \boldsymbol{R}\boldsymbol{x}, -1.0 \le x_i \le 1.0, \; for\, i = 1, 2, \ldots, m \tag{3.46}$$

$$|f_1| \le 1.0 \tag{3.47}$$

In figure 3.7 the Pareto-optimal front of problem *R4* is depicted.

3.7.5 Results

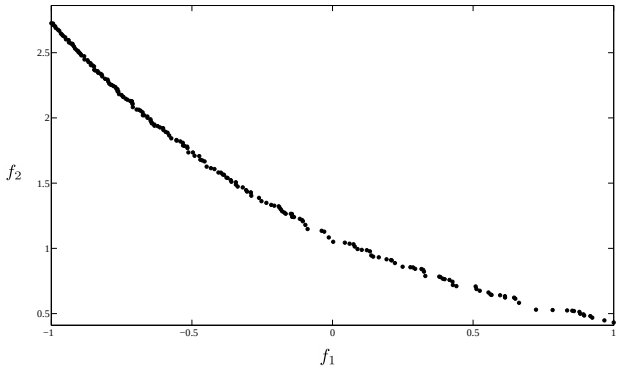


Figure 3.7: Optimal front of R4 problem.

Table 3.1: R1 test case, NSDE-DCS($K \in [0.2, 0.8]$,$F = 0.8$).

	$C(P, P^*)$			$C(P^*, P)$		
	μ	σ	$t = 300$	μ	σ	$t = 300$
K = 0.2	6.7330	15.6211	0.2819	2.5036	5.7349	0.3009
K = 0.25	6.2051	14.9554	0.4694	2.2261	5.2216	0.3675
K = 0.3	5.4523	14.0844	0.4460	1.8047	4.8026	0.2467
K = 0.35	4.7786	13.9632	0.0485	1.5495	4.8266	0.0711
K = 0.4	4.6866	13.2385	0.4295	1.5407	4.4296	0.2382
K = 0.45	4.2981	13.1422	0.2350	1.3478	4.1984	0.1376
K = 0.5	3.9433	12.7422	0.0918	1.1730	4.1709	0.0374
K = 0.55	3.7042	12.3530	0.1015	1.1644	3.9794	0.0704
K = 0.6	3.8811	12.2421	0.2700	1.2891	3.8834	0.2565
K = 0.65	3.9037	12.7062	0.0980	1.2911	4.3210	0.1035
K = 0.7	3.7082	12.3721	0.1840	1.2545	4.2227	0.1098
K = 0.75	3.7478	12.4476	0.1748	1.4078	4.3956	0.1852
K = 0.8	4.1164	12.4342	0.5294	1.5656	4.5033	0.3297

Table 3.2: R2 test case, NSDE-DCS($K \in [0.2, 0.8]$,$F = 0.8$).

	$C(P, P^*)$			$C(P^*, P)$		
	μ	σ	$t = 300$	μ	σ	$t = 300$
K = 0.2	8.7835	26.4037	0.0066	3.7442	9.4755	0.9354
K = 0.25	7.8083	25.0597	0.0066	3.6233	9.1037	1.1408
K = 0.3	6.5936	23.0856	0.0066	2.6802	7.8621	0.7303
K = 0.35	6.1601	22.4660	0.0065	2.5807	7.1522	0.8842
K = 0.4	5.7934	21.8310	0.0066	2.2544	6.8276	0.6789
K = 0.45	5.4270	21.3832	0.0066	2.3978	6.7948	0.9355
K = 0.5	5.1210	20.7892	0.0065	2.2943	6.4482	0.9354
K = 0.55	5.0155	20.7638	0.0066	2.4280	6.4557	1.1408
K = 0.6	4.9671	20.5871	0.0065	2.4053	6.1576	1.1405
K = 0.65	4.8201	20.2092	0.0066	2.1995	6.3439	0.8839
K = 0.7	4.7930	20.3403	0.0066	2.2242	6.7029	0.8843
K = 0.75	4.8040	20.2180	0.0065	2.3160	6.6728	0.9354
K = 0.8	5.0437	20.8894	0.0065	2.2593	7.2011	0.6790

Table 3.3: R3 test case, NSDE-DCS($K \in [0.2, 0.8]$,$F = 0.8$).

	$C(P, P^*)$			$C(P^*, P)$		
	μ	σ	$t = 300$	μ	σ	$t = 300$
K = 0.2	0.1342	0.0821	0.0195	0.9220	0.6485	0.0727
K = 0.25	0.1338	0.0865	0.0104	0.9283	0.7037	0.0194
K = 0.3	0.1290	0.0883	0.0097	0.8798	0.6986	0.0187
K = 0.35	0.1228	0.0881	0.0095	0.8352	0.6950	0.0187
K = 0.4	0.1242	0.0895	0.0092	0.8514	0.6975	0.0182
K = 0.45	0.1206	0.0841	0.0106	0.8078	0.6601	0.0200
K = 0.5	0.1211	0.0855	0.0162	0.8079	0.6584	0.0672
K = 0.55	0.1121	0.0911	0.0088	0.7368	0.6993	0.0179
K = 0.6	0.1097	0.0905	0.0090	0.7286	0.6947	0.0180
K = 0.65	0.1155	0.0896	0.0091	0.7825	0.6968	0.0182
K = 0.7	0.1193	0.0849	0.0158	0.8036	0.6641	0.0662
K = 0.75	0.1216	0.0829	0.0225	0.8338	0.6531	0.1146
K = 0.8	0.1213	0.0858	0.0183	0.8207	0.6732	0.0719

Table 3.4: R4 test case, NSDE-DCS($K \in [0.2, 0.8]$,$F = 0.8$).

	$C(P, P^*)$			$C(P^*, P)$		
	μ	σ	$t = 300$	μ	σ	$t = 300$
K = 0.2	0.0803	0.2521	0.0123	0.0473	0.1082	0.0136
K = 0.25	0.0733	0.2375	0.0125	0.0432	0.0982	0.0138
K = 0.3	0.0668	0.2277	0.0122	0.0398	0.0937	0.0135
K = 0.35	0.0623	0.2201	0.0122	0.0369	0.0849	0.0136
K = 0.4	0.0594	0.2140	0.0122	0.0350	0.0828	0.0136
K = 0.45	0.0560	0.2062	0.0121	0.0338	0.0798	0.0135
K = 0.5	0.0549	0.2064	0.0121	0.0334	0.0795	0.0135
K = 0.55	0.0541	0.2030	0.0121	0.0330	0.0765	0.0134
K = 0.6	0.0529	0.2027	0.0121	0.0322	0.0770	0.0134
K = 0.65	0.0528	0.1993	0.0121	0.0323	0.0764	0.0135
K = 0.7	0.0539	0.2044	0.0121	0.0327	0.0784	0.0135
K = 0.75	0.0555	0.2077	0.0120	0.0341	0.0816	0.0135
K = 0.8	0.0571	0.2098	0.0120	0.0352	0.0809	0.0134

Table 3.5: R1 test case, NSDE-DCS($K = 0.4$, $F \in [0.5, 1.1]$).

	$C(P, P^*)$			$C(P^*, P)$		
	μ	σ	$t = 300$	μ	σ	$t = 300$
F = 0.5	2.0261	9.7081	0.0242	0.8432	3.4214	0.2666
F = 0.55	2.2631	10.1888	0.0227	0.8695	3.6195	0.1805
F = 0.6	2.6035	10.8217	0.0097	0.9191	3.6714	0.1061
F = 0.65	2.9662	11.4562	0.0509	0.9736	3.8292	0.0553
F = 0.7	3.6378	12.3148	0.1509	1.2680	4.1103	0.1861
F = 0.75	4.0204	13.0216	0.0457	1.3001	4.4476	0.0290
F = 0.8	4.5694	13.5330	0.1296	1.5294	4.5482	0.1546
F = 0.85	5.4123	14.2094	0.2713	1.8086	4.7451	0.2523
F = 0.9	5.4206	14.4633	0.1876	1.6607	4.5413	0.1519
F = 0.95	6.6948	14.9699	0.8701	2.4590	4.8384	0.6926
F = 1	6.5413	14.7900	0.7929	1.9997	4.6881	0.3044
F = 1.05	7.2939	15.6223	0.6907	2.3849	5.0012	0.4225
F = 1.1	7.6971	15.9375	0.5405	2.5908	4.9727	0.5075

Table 3.6: R2 test case, NSDE-DCS($K = 0.4$, $F \in [0.5, 1.1]$).

	$C(P, P^*)$			$C(P^*, P)$		
	μ	σ	$t = 300$	μ	σ	$t = 300$
F = 0.5	3.0514	16.1872	0.0126	2.0805	5.7608	1.2386
F = 0.55	3.2672	16.6975	0.0069	2.0365	5.4545	1.1689
F = 0.6	3.7457	17.9934	0.0067	2.1376	6.0887	1.0540
F = 0.65	4.0725	18.5655	0.0067	2.2252	6.2126	1.0583
F = 0.7	4.7282	19.9705	0.0065	2.2964	6.5828	0.9355
F = 0.75	5.2167	20.8979	0.0066	2.5837	6.6372	1.1816
F = 0.8	5.5819	21.5312	0.0065	2.4272	6.7250	0.9353
F = 0.85	6.1783	22.6765	0.0066	2.5595	7.0596	0.9357
F = 0.9	6.5032	23.1246	0.0069	2.3513	7.2378	0.5666
F = 0.95	7.6617	24.6580	0.0071	2.9483	7.6980	0.7513
F = 1	8.0205	25.3231	0.0071	3.0552	7.8755	0.6955
F = 1.05	9.2900	25.4965	0.6990	3.7656	7.7575	1.1755
F = 1.1	9.3010	26.7166	0.0081	3.1543	8.2380	0.2678

Table 3.7: R3 test case, NSDE-DCS($K = 0.4$, $F \in [0.5, 1.1]$).

	$C(P, P^*)$			$C(P^*, P)$		
	μ	σ	$t = 300$	μ	σ	$t = 300$
F = 0.5	0.0395	0.0713	0.0087	0.2116	0.4728	0.0186
F = 0.55	0.0438	0.0752	0.0083	0.2504	0.5101	0.0188
F = 0.6	0.0533	0.0809	0.0083	0.3201	0.5693	0.0186
F = 0.65	0.0642	0.0862	0.0084	0.4080	0.6272	0.0182
F = 0.7	0.0816	0.0904	0.0086	0.5277	0.6695	0.0181
F = 0.75	0.0984	0.0936	0.0087	0.6581	0.7116	0.0180
F = 0.8	0.1259	0.0889	0.0093	0.8693	0.7007	0.0183
F = 0.85	0.1496	0.0757	0.0287	1.0429	0.6076	0.1401
F = 0.9	0.1544	0.0717	0.0273	1.0660	0.5904	0.0710
F = 0.95	0.1795	0.0535	0.0808	1.2805	0.4409	0.3522
F = 1	0.1859	0.0480	0.0985	1.3473	0.3884	0.5057
F = 1.05	0.1982	0.0401	0.1302	1.4560	0.2871	0.7034
F = 1.1	0.1964	0.0402	0.1402	1.4485	0.2607	0.8983

Table 3.8: R4 test case, NSDE-DCS($K = 0.4$, $F \in [0.5, 1.1]$).

	$C(P, P^*)$			$C(P^*, P)$		
	μ	σ	$t = 300$	μ	σ	$t = 300$
F = 0.5	0.0352	0.1592	0.0117	0.0248	0.0655	0.0136
F = 0.55	0.0372	0.1647	0.0119	0.0254	0.0685	0.0137
F = 0.6	0.0407	0.1732	0.0115	0.0271	0.0707	0.0132
F = 0.65	0.0441	0.1820	0.0117	0.0286	0.0746	0.0133
F = 0.7	0.0482	0.1903	0.0116	0.0302	0.0741	0.0133
F = 0.75	0.0533	0.2011	0.0119	0.0328	0.0800	0.0133
F = 0.8	0.0580	0.2093	0.0120	0.0348	0.0805	0.0134
F = 0.85	0.0630	0.2195	0.0125	0.0373	0.0834	0.0138
F = 0.9	0.0713	0.2373	0.0129	0.0402	0.0848	0.0141
F = 0.95	0.0757	0.2402	0.0135	0.0434	0.0900	0.0147
F = 1	0.0848	0.2581	0.0136	0.0472	0.0936	0.0149
F = 1.05	0.0892	0.2689	0.0142	0.0477	0.0923	0.0151
F = 1.1	0.0938	0.2793	0.0149	0.0495	0.0944	0.0157

Table 3.9: R1 test case, NSDE-FB($K \in [0.2, 8]$, $F = 0.6$).

	$C(P, P^*)$			$C(P^*, P)$		
	μ	σ	$t = 300$	μ	σ	$t = 300$
K = 0.2	5.7168	8.3850	3.1666	0.2643	2.1347	0.1169
K = 0.25	5.1658	8.7859	1.1489	0.2496	2.1413	0.0872
K = 0.3	3.7785	9.0765	0.6732	0.2513	2.1429	0.1000
K = 0.35	3.5380	9.1101	0.3069	0.2386	2.0795	0.0855
K = 0.4	2.8081	9.1069	0.1055	0.2234	2.0241	0.0727
K = 0.45	2.7775	9.1684	0.1357	0.2172	1.9859	0.0648
K = 0.5	2.6923	9.5998	0.0670	0.2181	2.1040	0.0580
K = 0.55	2.5565	9.5839	0.0709	0.2133	2.1177	0.0556
K = 0.6	2.5029	9.7026	0.0403	0.2020	2.1133	0.0374
K = 0.65	2.3528	9.5418	0.0412	0.1876	1.9143	0.0377
K = 0.7	2.6705	10.2523	0.0285	0.1929	2.1333	0.0275
K = 0.75	2.5188	9.9678	0.0285	0.1817	2.0331	0.0256
K = 0.8	2.5507	10.1276	0.0171	0.1795	2.0239	0.0180

Table 3.10: R2 test case, NSDE-FB($K \in [0.2, 8]$, $F = 0.6$).

	$C(P, P^*)$			$C(P^*, P)$		
	μ	σ	$t = 300$	μ	σ	$t = 300$
K = 0.2	13.9635	14.7472	1.4663	0.6018	3.6554	0.0974
K = 0.25	10.8887	16.0953	0.2538	0.5733	3.9676	0.0791
K = 0.3	7.0711	15.8544	0.0439	0.4871	3.8390	0.0435
K = 0.35	5.2257	15.7588	0.0530	0.4374	3.8057	0.0459
K = 0.4	4.4424	15.7082	0.0355	0.4104	3.8302	0.0366
K = 0.45	4.4524	16.4564	0.0238	0.3994	3.9507	0.0266
K = 0.5	4.1875	16.6140	0.0209	0.3970	4.2003	0.0246
K = 0.55	3.9992	16.8057	0.0214	0.3692	3.9981	0.0246
K = 0.6	4.0634	17.0609	0.0185	0.3526	3.7828	0.0220
K = 0.65	3.7982	16.9240	0.0162	0.3548	3.9704	0.0200
K = 0.7	3.9897	17.3611	0.0114	0.3433	3.9074	0.0164
K = 0.75	3.8969	17.5378	0.0104	0.3232	3.6787	0.0153
K = 0.8	3.9467	17.6460	0.0087	0.3411	4.0216	0.0142

Table 3.11: R3 test case, NSDE-FB($K \in [0.2, 8]$, $F = 0.6$).

	$C(P, P^*)$			$C(P^*, P)$		
	μ	σ	$t = 300$	μ	σ	$t = 300$
K = 0.2	0.0910	0.0437	0.0788	0.1916	0.1050	0.1716
K = 0.25	0.0940	0.0472	0.0778	0.1718	0.1089	0.1461
K = 0.3	0.0971	0.0495	0.0766	0.1665	0.1123	0.1333
K = 0.35	0.1015	0.0518	0.0732	0.1532	0.1168	0.1135
K = 0.4	0.1005	0.0532	0.0648	0.1417	0.1208	0.0932
K = 0.45	0.0989	0.0549	0.0585	0.1368	0.1252	0.0814
K = 0.5	0.0995	0.0591	0.0535	0.1418	0.1320	0.0752
K = 0.55	0.0915	0.0581	0.0449	0.1309	0.1305	0.0604
K = 0.6	0.0941	0.0599	0.0406	0.1396	0.1393	0.0569
K = 0.65	0.0960	0.0605	0.0402	0.1419	0.1396	0.0551
K = 0.7	0.0959	0.0639	0.0338	0.1491	0.1517	0.0466
K = 0.75	0.0967	0.0627	0.0331	0.1549	0.1556	0.0457
K = 0.8	0.1029	0.0623	0.0360	0.1598	0.1534	0.0503

Table 3.12: R4 test case, NSDE-FB($K \in [0.2, 8]$, $F = 0.6$).

	$C(P, P^*)$			$C(P^*, P)$		
	μ	σ	$t = 300$	μ	σ	$t = 300$
K = 0.2	0.0408	0.1551	0.0125	0.0345	0.0616	0.0178
K = 0.25	0.0396	0.1611	0.0130	0.0296	0.0605	0.0158
K = 0.3	0.0383	0.1607	0.0123	0.0272	0.0604	0.0145
K = 0.35	0.0396	0.1649	0.0125	0.0268	0.0590	0.0142
K = 0.4	0.0400	0.1701	0.0123	0.0268	0.0602	0.0143
K = 0.45	0.0394	0.1703	0.0121	0.0268	0.0616	0.0142
K = 0.5	0.0391	0.1750	0.0116	0.0247	0.0611	0.0137
K = 0.55	0.0394	0.1764	0.0117	0.0245	0.0615	0.0135
K = 0.6	0.0421	0.1855	0.0117	0.0253	0.0628	0.0134
K = 0.65	0.0430	0.1883	0.0118	0.0258	0.0657	0.0134
K = 0.7	0.0437	0.1949	0.0117	0.0253	0.0664	0.0133
K = 0.75	0.0451	0.1984	0.0117	0.0260	0.0649	0.0133
K = 0.8	0.0454	0.2005	0.0115	0.0254	0.0645	0.0132

Table 3.13: R1 test case, NSDE-FB($K = 0.6$, $F \in [0.4, 1]$).

	$C(P, P^*)$			$C(P^*, P)$		
	μ	σ	$t = 300$	μ	σ	$t = 300$
F = 0.4	2.1230	8.2040	0.1095	0.2287	2.1060	0.0723
F = 0.45	2.1840	8.6365	0.0862	0.2233	2.1401	0.0619
F = 0.5	2.1872	8.8506	0.0804	0.2095	1.9941	0.0594
F = 0.55	2.4016	9.2966	0.0645	0.2057	2.0208	0.0515
F = 0.6	2.6311	9.8009	0.0681	0.2096	2.1459	0.0511
F = 0.65	2.6127	9.9969	0.0312	0.1874	2.0037	0.0291
F = 0.7	2.6740	10.2788	0.0160	0.1857	2.2015	0.0161
F = 0.75	2.9275	10.5640	0.0057	0.1788	2.1268	0.0067
F = 0.8	2.9893	10.7879	0.0022	0.1730	2.0264	0.0033
F = 0.85	3.3713	11.3007	0.0022	0.1769	2.0872	0.0033
F = 0.9	3.3973	11.0299	0.0022	0.1697	1.9867	0.0033
F = 0.95	3.8732	11.3708	0.1362	0.1910	2.1414	0.0104
F = 1	3.9056	11.7658	0.0026	0.1770	1.9601	0.0035

Table 3.14: R2 test case, NSDE-FB($K = 0.6$, $F \in [0.4, 1]$).

	$C(P, P^*)$			$C(P^*, P)$		
	μ	σ	$t = 300$	μ	σ	$t = 300$
F = 0.4	3.3025	14.4045	0.0361	0.3911	3.8048	0.0359
F = 0.45	3.3836	15.0769	0.0294	0.3706	3.8460	0.0309
F = 0.5	3.4360	15.4162	0.0311	0.3815	4.0567	0.0320
F = 0.55	3.6166	15.9994	0.0217	0.3699	4.0205	0.0249
F = 0.6	3.7779	16.7219	0.0146	0.3434	3.8100	0.0191
F = 0.65	3.9876	17.2713	0.0135	0.3490	3.9209	0.0182
F = 0.7	4.3981	17.9771	0.0086	0.3532	3.9301	0.0142
F = 0.75	4.8385	19.0057	0.0075	0.3679	4.0981	0.0131
F = 0.8	5.3287	19.9263	0.0066	0.3647	3.8808	0.0125
F = 0.85	5.4343	20.1920	0.0066	0.3748	4.0662	0.0123
F = 0.9	5.4312	20.5329	0.0066	0.3560	3.7363	0.0123
F = 0.95	5.6537	20.6825	0.0070	0.3714	3.8296	0.0126
F = 1	6.6094	22.1664	0.0077	0.3779	3.6687	0.0130

Table 3.15: R3 test case, NSDE-FB($K = 0.6$, $F \in [0.4, 1]$).

	$C(P, P^*)$			$C(P^*, P)$		
	μ	σ	$t = 300$	μ	σ	$t = 300$
F = 0.4	0.0903	0.0531	0.0508	0.1241	0.1214	0.0732
F = 0.45	0.0893	0.0557	0.0478	0.1251	0.1271	0.0651
F = 0.5	0.0971	0.0546	0.0508	0.1360	0.1266	0.0727
F = 0.55	0.0968	0.0581	0.0463	0.1375	0.1306	0.0644
F = 0.6	0.0981	0.0595	0.0450	0.1469	0.1397	0.0616
F = 0.65	0.1006	0.0604	0.0432	0.1485	0.1438	0.0567
F = 0.7	0.1072	0.0603	0.0423	0.1618	0.1435	0.0575
F = 0.75	0.1050	0.0603	0.0352	0.1583	0.1442	0.0512
F = 0.8	0.1185	0.0584	0.0427	0.1797	0.1482	0.0611
F = 0.85	0.1177	0.0591	0.0424	0.1856	0.1518	0.0567
F = 0.9	0.1283	0.0567	0.0476	0.1998	0.1504	0.0670
F = 0.95	0.1341	0.0541	0.0542	0.2098	0.1458	0.0718
F = 1	0.1345	0.0541	0.0536	0.2168	0.1504	0.0737

Table 3.16: R4 test case, NSDE-FB($K = 0.4$, $F \in [0.4, 1]$).

	$C(P, P^*)$			$C(P^*, P)$		
	μ	σ	$t = 300$	μ	σ	$t = 300$
F = 0.4	0.0369	0.1586	0.0125	0.0259	0.0566	0.0142
F = 0.45	0.0380	0.1637	0.0123	0.0262	0.0567	0.0146
F = 0.5	0.0374	0.1645	0.0121	0.0243	0.0558	0.0138
F = 0.55	0.0404	0.1774	0.0122	0.0257	0.0604	0.0138
F = 0.6	0.0427	0.1866	0.0119	0.0261	0.0646	0.0135
F = 0.65	0.0454	0.1978	0.0117	0.0266	0.0648	0.0135
F = 0.7	0.0480	0.2077	0.0116	0.0271	0.0685	0.0133
F = 0.75	0.0498	0.2166	0.0117	0.0269	0.0687	0.0133
F = 0.8	0.0533	0.2253	0.0117	0.0286	0.0721	0.0132
F = 0.85	0.0581	0.2452	0.0119	0.0299	0.0773	0.0133
F = 0.9	0.0590	0.2417	0.0121	0.0307	0.0747	0.0134
F = 0.95	0.0671	0.2700	0.0124	0.0334	0.0812	0.0137
F = 1	0.0713	0.2856	0.0128	0.0347	0.0865	0.0140

Table 3.17: R1 test case.

	$C(P,P^*)$			$C(P^*,P)$		
	μ	σ	$t = 300$	μ	σ	$t = 300$
FB	2.4911	9.6598	0.0130	0.1776	1.9513	0.0136
DIR	4.4178	8.8670	1.3186	0.1844	2.0600	0.0253
NSDEwSA	9.5052	17.1937	0.3989	3.7528	6.4780	0.5275
NSDE-DCS	1.8127	9.1876	0.0242	0.7821	3.1994	0.2705
NSDE-FB	3.7395	12.2289	0.0020	0.1688	2.1821	0.0031

Table 3.18: R2 test case.

	$C(P,P^*)$			$C(P^*,P)$		
	μ	σ	$t = 300$	μ	σ	$t = 300$
FB	4.5442	18.5590	0.0097	0.3988	3.6440	0.0149
DIR	5.9082	16.4762	0.0145	0.4197	4.0826	0.0213
NSDEwSA	8.4487	25.6996	0.0065	4.3173	9.4732	1.475378
NSDE-DCS	2.7975	15.4704	0.0143	1.8967	4.9974	1.189710
NSDE-FB	5.9085	22.1947	0.0067	0.3715	3.6335	0.012502

In this work, first a comparison of the performances of the reproduction scheme proposed by Iorio and Li[17] has been carried out. In fact, their crossover is influenced by two parameters. The former controls the convergence speed towards the Pareto-optimal front (parameter K), the latter controls the population spread (parameter F). Iorio and Li suggested the parameters setting of $K = 0.4$ and $F = 0.8$. This configuration was based on considerations found in literature and therefore in author's opinion, there is a lack in the analysis of the effects of different values of such parameters.
In the analysis conducted here, one of the two parameters has been assumed to be fixed, the other is let to vary near the setting employed in [17].

Table 3.19: R3 test case.

	$C(P,P^*)$			$C(P^*,P)$		
	μ	σ	$t = 300$	μ	σ	$t = 300$
FB	0.1418	0.0542	0.0678	0.2637	0.1838	0.0985
DIR	0.0575	0.0602	0.0188	0.1224	0.1739	0.0297
NSDEwSA	0.0774	0.0909	0.0094	0.4911	0.6642	0.0180
NSDE-DCS	0.0378	0.0688	0.0085	0.2061	0.4622	0.0188
NSDE-FB	0.1158	0.0681	0.0247	0.2354	0.1947	0.0367

Table 3.20: R4 test case.

	$C(P,P^*)$			$C(P^*,P)$		
	μ	σ	$t = 300$	μ	σ	$t = 300$
FB	0.0678	0.2882	0.0130	0.0334	0.0745	0.0139
DIR	0.0474	0.1772	1.3186	0.0294	0.0587	0.0186
NSDEwSA	0.0680	0.2250	0.3989	0.0446	0.1024	0.0143
NSDE-DCS	0.0334	0.1538	0.0242	0.0244	0.0628	0.0155
NSDE-FB	0.0704	0.2698	0.0020	0.0357	0.0855	0.0136

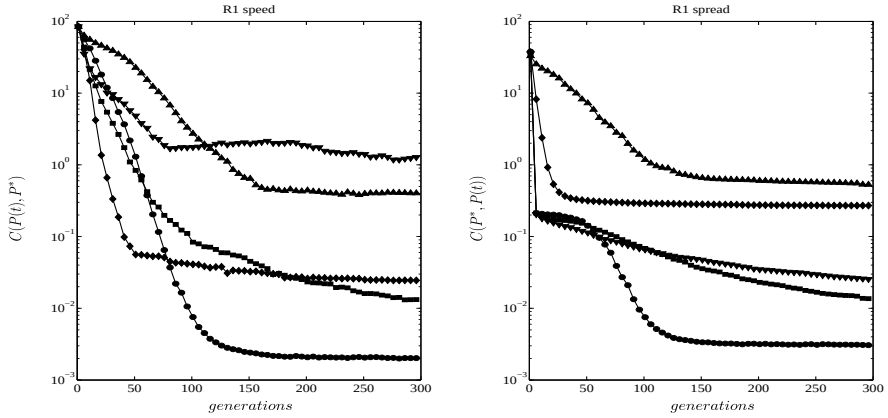


Figure 3.8: R1 test case. (■) Fitness-Based Crossover, (▼) Directional Crossover, (▲) NSDEwSA, (♦) NSDE-DCS, (●) NSDE-FB

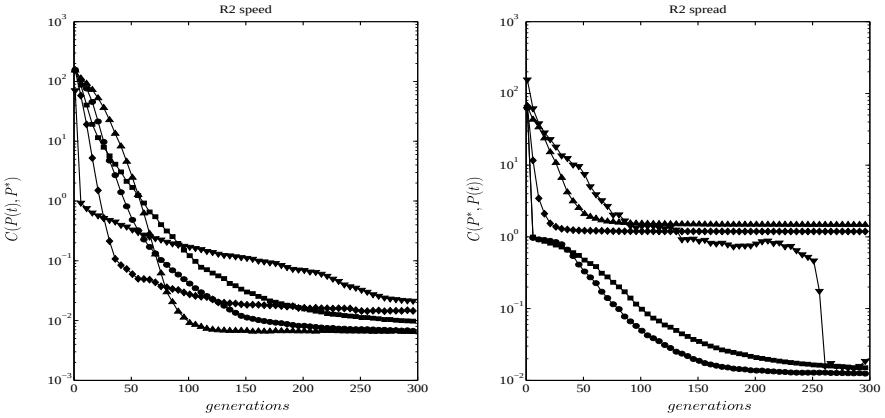


Figure 3.9: R2 test case. (■) Fitness-Based Crossover, (▼) Directional Crossover, (▲) NSDEwSA, (♦) NSDE-DCS, (●) NSDE-FB

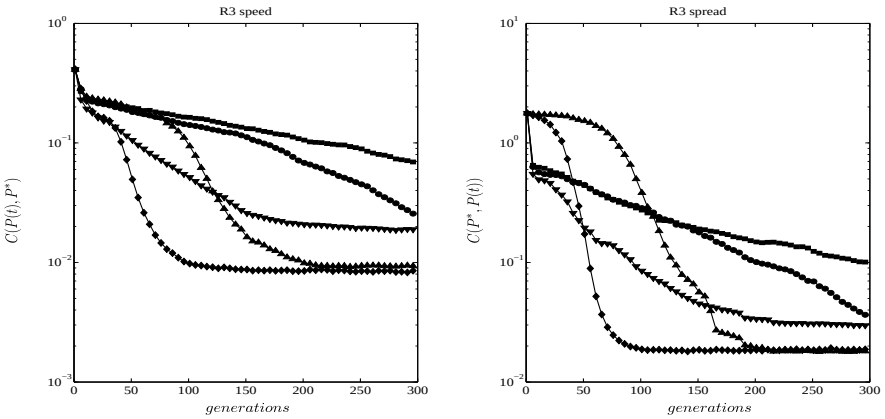


Figure 3.10: R3 test case. (■) Fitness-Based Crossover, (▼) Directional Crossover, (▲) NSDEwSA, (♦) NSDE-DCS, (●) NSDE-FB

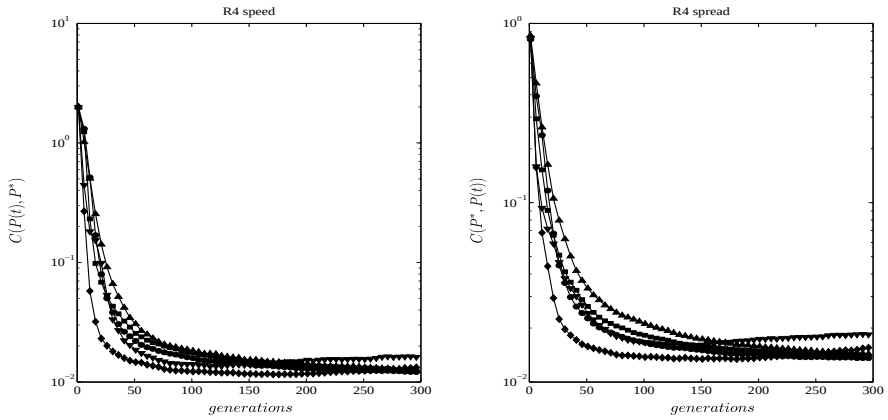


Figure 3.11: R4 test case. (■) Fitness-Based Crossover, (▼) Directional Crossover, (▲) NSDEwSA, (⧫) NSDE-DCS, (●) NSDE-FB

Tables 3.1, 3.2, 3.3 and 3.4 report the simulation results about the effects of different values of the parameter K for the test cases previously described. It can be seen that larger values of K allows to obtain a faster convergence toward the Pareto-front with an increased robustness. Specifically, a parameter setting of about $K = 0.6$ allows the best performances in terms of convergence speed, population spread and robustness of the solutions.

Tables 3.5, 3.6, 3.7 and 3.8 report the simulation results regarding the effect of different values of the population spread parameter F. It can be deduced that a small value of F allows a fast convergence and a better population spread. However, the lowest values of the metrics employed at generation $t = 300$ can be found for a value of F not too small. Therefore, a population spread parameter of $F = 0.55$ allows best performance in terms of convergence speed, population spread and robustness of the solutions.

Similarly to the previous comparison, we performed several tests to detect the best performances of the NSDE-FB algorithm. Tables 3.9, 3.10, 3.11 and 3.12 report the simulation results about the effect of different values of parameter K for the Iorio and Li[18] test cases. It can be seen that the optimal value of K is equal to 0.65. In fact, such a value of the convergence vector

parameter allows a good balance of performances expressed by the metrics considered.

Tables 3.13, 3.14, 3.15 and 3.16 report the simulation results regarding the effect of the spread vector parameter F. Specifically, a setting of $F = 0.55$ allows the best performance of the algorithm.

It is worth noting that a variation of F in NSDE-DCS and in NSDE-FB affect not only the population spread measure, but also the convergence speed of the algorithms and their robustness. This is due to the fact that well distributed populations detect faster the Pareto-optimal front since the decision space is searched more efficiently.

Finally, a comparison between different methodologies has been carried out on the rotated test problems previously described.

Considering the Fitness-Based crossover, it can be seen from figure 3.10 that it is unable to correctly solve the $R3$ problem.

Within the NSGA-II framework, the Directional crossover performs well in all the test cases except for the $R1$ problem where its convergence speed is unable to be less than 10^0 before the maximum allowed generations. Also in the $R2$ problem, Directional crossover performs worse than the others algorithms.

Similarly, also NSDEwSA is unable to correctly solve the $R1$ test case, and in the $R3$ problem, NSDEwSA is unable to successfully cover the Pareto front. NSDE-DCS has a behavior similar to NSDEwSA in the $R2$ case, whereas in the other problems, it exhibits good performances.

Finally, NSDE-FB shows outstanding performances in almost all the test cases, except for $R3$ problem where it converges slowly toward the Pareto-optimal solution. However, by increasing the maximum number of generations, the NSDE-FB is expected correctly solve also the $R3$ problem since its performance metrics are not constant in the last generations. Therefore, this scheme will be employed in the application showed in the next chapters.

Chapter 4

Multi-objective optimization of longitudinal wavy fins

In many engineering fields, finned dissipators are an invaluable tool to remove heat in particular where high fluxes must be transferred. A typical use of such a heat exchanger can be found in the electronic industry where components must be cooled in order to ensure both performance and reliability. Moreover, the temperature of an electronic component must be lowered considering also the exigence of reducing the energy consumption of cooling fans and reducing the volume of the device. For example, the former exigence is connected with the customer requirement of long battery endurance. Whereas, the latter exigence is related to the weight and to the volume of the device. In recent years in fact, the electronic industry has developed smaller and smaller components and the heat dissipators for them must consequently be scaled. Another reason that lead to the reduction of the finned dissipator volume is the necessity of lowering the amount of material used in the device. In fact, as reported by the LCA Committee of the Japanese aluminum industry [26] and recalled by Bar-Choen and Iyengar [13], the estimated amount of aluminium used for cooling electronic devices was of about 10 Million-kg

in 2001. Although, we have focused only in the electronic field so far, similar considerations can be easily made in all fields were heat dissipators play a fundamental role.

The problem of the optimization of longitudinal fins has been studied by many researchers in the last century [15, 16, 5, 20]. In particular several fin profiles have been suggested and undulated fins have shown the best performances in terms of heat transfer [19, 10]. However, the optimum fin shape has been solved only partially. In fact, a criterion for detecting the optimal geometry has been proposed by Fabbri [7, 8] where a genetic algorithm has been employed to detect the geometry that ensures the highest heat transfer. However, as just pointed out the optimum geometry itself may not be usefull since limits in the hydraulic resistance or in the volume of the fins may also arise. Although, the genetic algorithm previously used can be employed considering constraints in the hydraulic resistance, the value of such limits is typically unknown *a priori*. Hence, a different approach should be used and in this case *multi-objective optimization* techniques are suitable methods since their application together with genetic algorithms permits to detect rapidly the set of solutions that optimize all the objectives of interest.

Although, these techniques have been developed since the mid-1980s, their usage in the heat transfer field is recent and not common. In particular, Hilbert et al. [32] performed a multi-objective shape design optimization of a tube bank heat exchanger whereas, Nobile et al. [27] studied convective periodic channels.

In the present work, the optimization of heat transfer through finned dissipators by means of the multi-objective genetic algorithm described in previous chapters is investigated. The two objectives considered here are the maximization of the heat transfer and, at the same time, the minimization of the hydraulic resistance. Since the two objectives are conflicting, the result of the genetic algorithm is a set of optimal solutions each of them lying

on the Pareto front trade-off curve. The same methodology is then applied considering different constraints in the volume of the fins.

4.1 Governing Equations and Dimensioning Criteria

Let us consider a conduit composed by a finned plate opposite to a flat surface, both of them infinitely long and wide. The flat surface is thermally insulated while a constant heat flux q'' is applied to the finned plate bottom. The coolant flows in the conduit parallel to the fins. All conduit fins are identical and the sections orthogonal to the flow direction have two symmetry axes (see Fig. 4.1). Since the present work considers an optimization

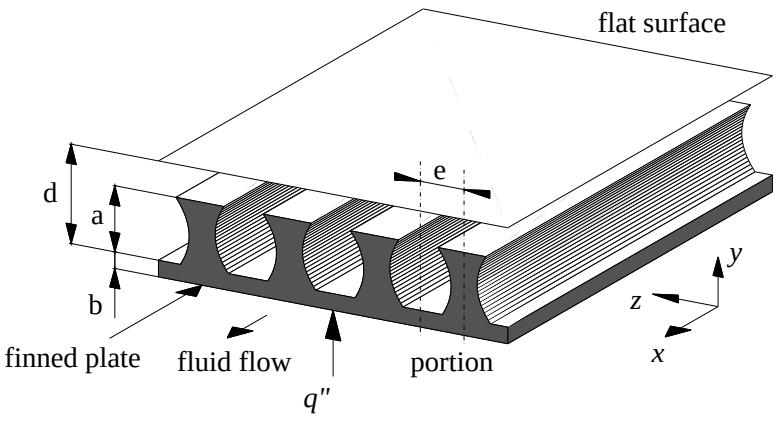


Figure 4.1: Finned Conduit

technique that employs a large number of individuals in order to correctly solve the problem, the simplicity of the model is mandatory. In fact, for the test cases considered a population of 100 individuals that reproduces for 100 generations have to be considered in order to detect the real Pareto

front. Therefore, a total number of 10000 cost functions evaluations is required. For such a large number of individuals, the usage of time consuming solvers is not allowed in order to obtain results within a reasonable time. For this reason, in the present work we introduce the following hypotheses and considerations:

- the fluid flow is laminar and incompressible;
- the system is in steady state;
- velocity and temperature are fully developed;
- fluid and solid properties are uniform and fluid independent;
- natural convection is negligible in regard to the forced convection;
- viscous dissipation is negligible.

The heat transfer performance of the system can be studied considering only a portion of it which is delimited by the finned plate bottom, the top flat surface, and two symmetry axes (see Fig. 4.2). Let us choose an orthogonal coordinate system where the x axis is parallel to the streamwise direction, the y axis is perpendicular to the flat plate while the z direction is parallel to it. As shown in Fig. 4.2, let a be the fin height, b the fin base thickness, e the distance between the two symmetry axes, and d the distance between the fin base and the flat surface.

Under such conditions, only the x component u of the fluid velocity is non zero and it can be determined solving the corresponding component of the momentum equation:

$$\frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = \frac{1}{\mu}\frac{\partial p}{\partial x} \tag{4.1}$$

where μ is the dynamic viscosity and p the pressure. Two kind of boundary conditions are needed to solve equation (4.1). The former is the no-slip

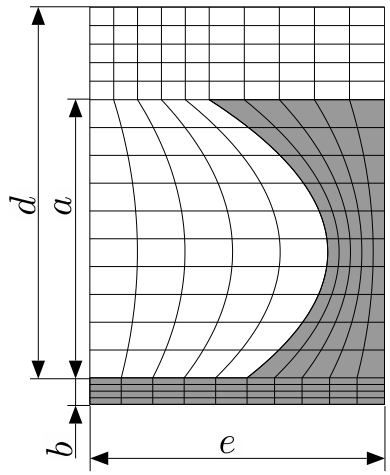


Figure 4.2: Computational Domain.

condition considered at the walls(i.e., $u = 0$). The latter consists of the zero velocity partial derivative at the symmetry axes along the normal direction(i.e., $\frac{\partial u}{\partial n} = 0$).

Since the applied heat flux is uniform and the thermal profile is fully developed, the temperatures of the fluid and the solid change linearly with the x coordinate. The conductive heat flux is constant and can be neglected in an overall energy balance. Therefore the temperature distribution in the coolant is described by the following relationship:

$$\rho c_p u \frac{\partial T_c}{\partial x} = k_c \left[\frac{\partial^2 T_c}{\partial y^2} + \frac{\partial^2 T_c}{\partial z^2} \right] \tag{4.2}$$

where ρ is the density, c_p the specific heat, and k_c the thermal conductivity of the coolant.

The temperature distribution inside the finned plate is instead described by the Laplace equation:

$$\frac{\partial^2 T_f}{\partial y^2} + \frac{\partial^2 T_f}{\partial z^2} = 0 \tag{4.3}$$

Equations (4.2) and (4.3) must be integrated imposing the following boundary conditions:

- on the contact surface, the temperature is the same in the solid and in the fluid;
- on the contact surface, the heat flux in the normal direction is the same in the solid and in the fluid;
- on the symmetry axes and on the insulated wall, the heat flux in the normal direction must be zero;
- on the finned plate bottom surface, the heat flux must be equal to q'' and parallel to the y direction.

Lastly, the value of the temperature in one point of the section is needed. Equations (4.1),(4.2) and (4.3) are here solved numerically by means of the finite element code previously used and tested by Fabbri [7].
In order to evaluate the heat-transfer performance of the finned conduit some definitions are given here. For the system described, the following global heat transfer coefficient can be defined as

$$h = \frac{q''}{T_{max} - T_b} \tag{4.4}$$

where T_{max} is the maximum temperature on the surface on which q'' is imposed and T_b is the bulk temperature of the fluid. The temperatures T_{max} and T_b must be calculated at the same value of the x coordinate. Moreover, it is possible to define an equivalent Nusselt number:

$$Nu_e = \frac{2\,h\,d}{k_c} \tag{4.5}$$

which corresponds to the Nusselt number which would be calculated if the same heat flux q'' were dissipated through a flat surface with zero thickness at the distance d from the insulated surface.

Moreover, the normalized hydraulic resistance is considered in the optimization:

$$\zeta = \frac{-dp/dx}{w_t/e} \Big/ \frac{12\mu}{d^3} \tag{4.6}$$

which indicates how many times the hydraulic resistance per unit of length and width of the conduit increases due to the presence of the fins.
The computational domain geometry is described by the parameters a, b, d, e and the fin profile function $g(y)$. Taking d as reference length, we obtain the following dimensionless variables:

$$\alpha = \frac{a}{d};\ \beta = \frac{b}{d};\ \epsilon = \frac{e}{d};\ \phi(\eta) = \frac{g(y)}{d};\ \eta = \frac{y}{d} \tag{4.7}$$

Let us assign a polynomial form to the profile function ϕ:

$$\phi(\eta) = \sum_{i=0}^{n} \psi_i \eta^i \tag{4.8}$$

the function $\phi(\eta)$ is univocally determined by the $n+1$ parameters ψ_i or, alternatively, by $n+1$ values of ϕ in equidistant points on the η axis, namely:

$$\phi_i = \phi\left(\frac{i}{n}\alpha\right) \tag{4.9}$$

Since changes in ϕ_i induce in $\phi(\eta)$ variations of more comparable entity then do changes in ψ_i, the first ones are preferable as fin profile describing parameters instead of the latter ones. Moreover, the average half-width of the fin $\bar{\phi}$ results

$$\bar{\phi} = \sum_{i=0}^{n} \frac{\psi_i(\phi_0, \ldots, \phi_n)}{i+1} \alpha^i \tag{4.10}$$

Hence, we can express the average thickness of the finned plate $\bar{\sigma}$ as:

$$\bar{\sigma} = \beta + \frac{\alpha\bar{\phi}}{\epsilon} \tag{4.11}$$

This parameter is representative of the volume and the weight of the finned plate.

From these definitions, the optimization of a finned plate can be addressed aiming to maximize the Nusselt number and, at the same time, minimizing the hydraulic resistance or minimizing the entropy generated. Moreover, the average thickness can be constrained to an established value $\bar{\sigma}_0$ in order to limit the volume of the fins. To this aim the parameters ϵ and ϕ_i can be reproduced by means of opportune genetic operations whereas, the value of β can be forced to assume the value

$$\beta = \bar{\sigma}_0 - \frac{\alpha}{\epsilon}\bar{\phi} \tag{4.12}$$

In this way, β can be negative or too small. If that occurs, ϕ_i can be resized or the parameter combination can be rejected by assigning a null value to the Nusselt number and an infinite value to the hydraulic resistance.

4.2 Results

The NSGA-II algorithm with the differential evolutionary reproduction operators described in previous chapters (NSDE-FB) has been employed to optimize the geometry previously studied by Fabbri [8]. Only three values of the polynomial order n of the fin profile function have been studied here, namely

$$n = 0,\ 2,\ 4. \tag{4.13}$$

The parameters set up of the algorithm used to optimize the fins is listed below:

- population size, $N = 100$;
- maximum number of generation, $t_{max} = 100$.

For the finite element method, a grid with 20×80 elements (21×81 nodes) has been used. More finer grids have been tested without finding any significant variation in Nu_e. In the test cases, a grid of 35×80 elements has

caused changes in Nu_e of less than 0.5 % whereas, a grid of 20×110 elements produced alterations in Nu_e of less than 0.7 %. Moreover, the normalized fin height α has been considered equal to 0.75 and the thermal conductivity ratio γ equal to 300. Such a value of γ corresponds to the case of a copper finned plate cooled by water. The remaining geometrical parameters has been allowed to vary between the following values:

$$0.001 \leqslant \beta \leqslant 0.2; \tag{4.14}$$

$$0.05 \leqslant \epsilon \leqslant 0.6; \tag{4.15}$$

$$0.001 \leqslant \phi_i \leqslant \epsilon. \tag{4.16}$$

Table 4.1: Geometries that ensure a reduction of about 20 % of the hydraulic resistance of maximum heat transfer geometries.

α	n	$\bar{\sigma}$	β	ϵ	ϕ_0	ϕ_1	ϕ_2	ϕ_3	ϕ_4	Nu_e	ζ	$\Delta Nu_e\,(\%)$	$\Delta\zeta\,(\%))$
0.75	0	0.256	0.0861	0.2085	0.0472	-	-	-	-	57.66	12.31	4.1	23
0.75	2	0.322	0.1304	0.2036	0.0815	0.0272	0.1225	-	-	77.46	15.50	5.8	24
0.75	4	0.276	0.1043	0.1958	0.0915	0.0375	0.0322	0.0254	0.1426	82.93	16.00	4.7	20
0.75	0	0.200	0.0710	0.1923	0.0331	-	-	-	-	56.03	11.96	4.5	22
0.75	2	0.200	0.0380	0.1982	0.0751	0.0191	0.1055	-	-	72.50	14.31	5.6	23
0.75	4	0.200	0.0472	0.1978	0.0943	0.0337	0.0350	0.0152	0.1405	79.97	15.17	6.7	23
0.75	0	0.100	0.0244	0.1805	0.0182	-	-	-	-	48.80	10.83	4	21
0.75	2	0.100	0.0144	0.1874	0.0371	0.0118	0.04	-	-	52.70	11.04	5	23
0.75	4	0.100	0.0158	0.1855	0.0469	0.0129	0.0143	0.0103	0.0903	59.74	12.37	5.3	22

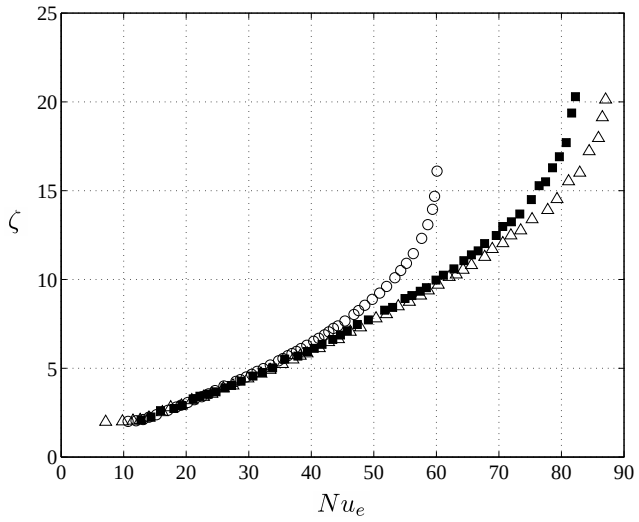


Figure 4.3: Pareto fronts of the unconstrained cases.(○) rectangular fins, (■) parabolic fins, (△) 4-th order fins.

4.2.1 Optimization without constraints

Fig. 4.3 shows the Pareto fronts related to the three cases here analyzed. It can be seen that for all the polynomial orders considered, the maximum Nusselt number agrees with data reported by Fabbri [8]. Moreover, all the cases tested has detected the Pareto front in much fewer generations than t_{max}. Specifically, the $4-th$ polynomial order has been the most complex case to solve since the resulting number of variables was equal to seven. In that case, the Pareto front has been detected and well sampled within 40 generations. Whereas, the rectangular fin shape and the parabolic fin profile have been solved within 20 and 30 generations respectively. The analysis has been limited to the detection of the maximum Nusselt number since Fabbri [8] demonstrated that those geometries that maximize the fin efficiency also maximize Nu_e. Therefore, the fin efficiency has not been considered here.
A first important consideration that can be made on the basis of the present results concerns with global optimum geometries. In fact, it can be clearly

seen that there are profiles that provide good performance in terms of heat exchange but, at the same time, are also able to provide a sensible reduction of the hydraulic resistance. In Table 4.1, profiles that provide the maximum heat transfer with a reduction of the hydraulic resistance of at least the 20 % are listed. Parameters on the left of the vertical line have been imposed, while those on the right have been found by the genetic optimization algorithm. For all the unconstrained geometries, the Nusselt number decrease ranges between 4.1 % and 5.8 % whereas, the ζ has a reduction that ranges between 20 % and 24 %.

If it is of interest the maximum heat transfer, Fabbri [8] pointed out that optimum geometries are based on a compromise between two exigencies. The first consists in having, in the cavities between the fins, velocities which are comparable with those at fin tip or higher. In fact, higher velocity near the fin tip causes higher thermal gradients in this region, which would lower the bulk temperature without enhancing the heat transfer from the plate base and the lateral surface on the fin. The second exigence consists in maintaining the maximum velocity as close to the dissipator as possible in order to relatively increase the thermal gradient. As a consequence of the first exigence, fins cannot be too closely spaced, for the second one they cannot be too sparse. When dealing with Pareto optimal solutions, the balance between the two exigencies progressively change. In particular, since here we are interested in minimizing the hydraulic resistance, the individuals belonging to the Pareto front exhibit the tendency of having larger cavities between the fins as the parameter ζ decreases. This can be deduced from Fig. 4.6 where four cases belonging to the Pareto front of $n = 4$ are depicted. From this figure, it can be seen that the fin spacing progressively decreases as the Nusselt number increases. Also the fin profile changes since the fin tip tends to extend toward the center of the channel between the fins. In fact, higher order profiles that maximize the Nusselt number force the maximum velocity to occur between

the fins. Such a behavior induces higher thermal gradients near the finned plate base and the fin lateral surface. At low hydraulic resistance, the fin tips are thinner and the maximum velocity progressively occurs at higher values of the normalized coordinate η (see Fig. 4.4). As consequence of having thinner fin tips and larger cavities, the performance of wavy fins are not much different from those obtained with rectangular fins and the Pareto fronts overlap at low ζ. In other words, if a designer is interested in cases where the constraint imposed in ζ is particularly stringent, the employment of corrugated fins is not usefull to improve the heat transfer coefficient. Such an effect can be clearly seen in Fig. 4.3 where the curves related to the geometries analyzed here overlap if $\zeta < 6$. Moreover, parabolic and 4-th order fins provide similar performance if $\zeta \lesssim 10$. This is of fundamental role if also manufacturing exigencies are taken into account since rectangular fins are easier to realize. On the other hand, complex fin profiles are more expensive to realize than rectangular devices but the manufacturing cost of these complex fins can be easily gained thanks to lower usage costs.

4.2.2 Optimization with constrained volume

In practice, in the optimization of a particular geometry, some constraints may also arise. In the application presented here, a typical constraint can be the solid volume of the device. In the present analysis, we set up two different values of the average thickness of the finned plate, namely $\bar{\sigma}_0 = 0.2$ and $\bar{\sigma}_0 = 0.1$.

Fig.s 4.7 and 4.8 show the Pareto fronts for the two limits considered and for the three geometries optimized here. In all cases the constraint entail a reduction of the heat transfer as already pointed out by Fabbri [8]. However, the same consideration of the unconstrained study can be sentenced here. In fact, in all cases there are geometries that provide good performances in terms of heat transfer and, at the same time, provide a significant reduction

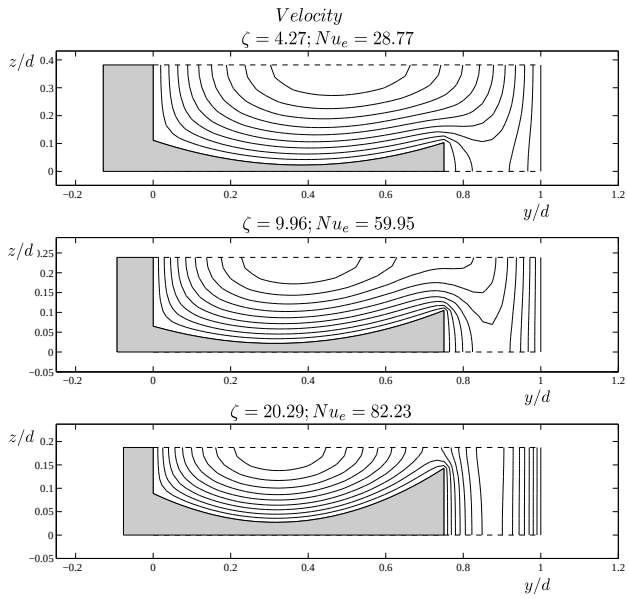


Figure 4.4: Velocity distribution in the transversal section for $n = 2$, $\alpha = 0.75$. Curves are drawn every 10% of the maximum velocity.

of the parameter ζ (see Table 4.1). Moreover, in all cases, constrained and unconstrained, the three Pareto fronts overlap when Nu_e is lower than 40.

4.3 Conclusions

In the present chapter, the multi-objective approach was applied to the optimization of the heat transfer through finned heat dissipator cooled by laminar flow. In particular, the optimal geometry of wavy fins described by polynomial functions was studied and the parameter combinations that aim to maximize the heat transfer and, at the same time, aim to minimize the hydraulic resistance were searched. The results are a set of solutions lying on a curve called Pareto front and the analysis of this curve allowed to find geometries where the Nusselt number is slightly reduced but the hydraulic resistance is appreciably lower than the one computed in the best heat ex-

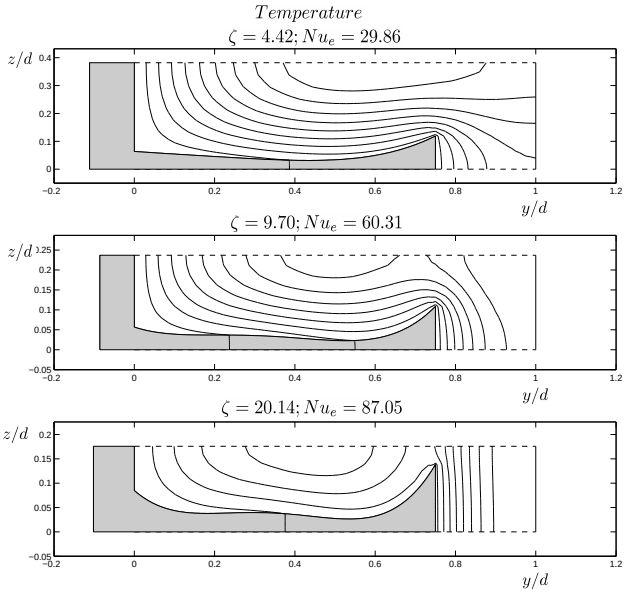


Figure 4.5: Temperature distribution in the transversal section for $n = 4$, $\alpha = 0.75$. Curves are drawn every 10% of the difference between the maximum and the minimum temperature.

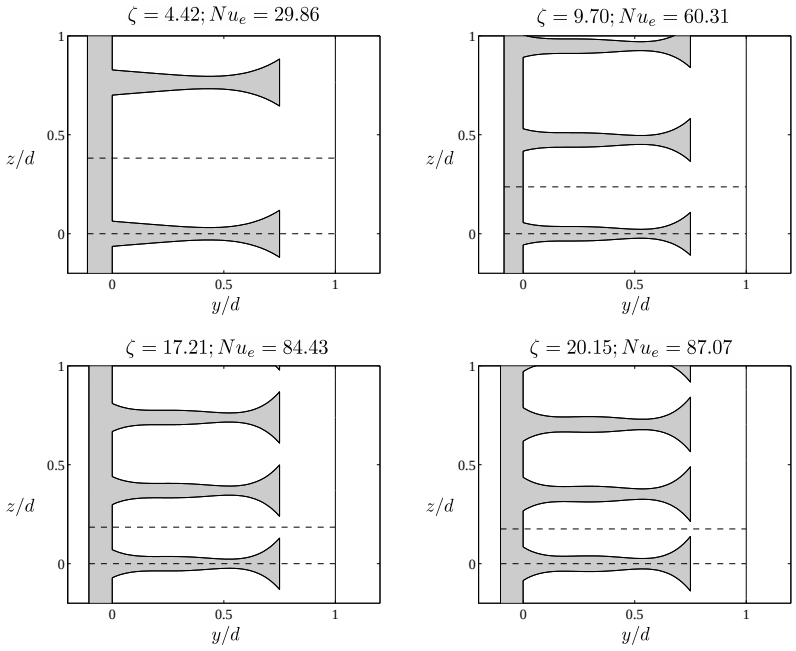


Figure 4.6: Optimum geometries for $n = 4$.

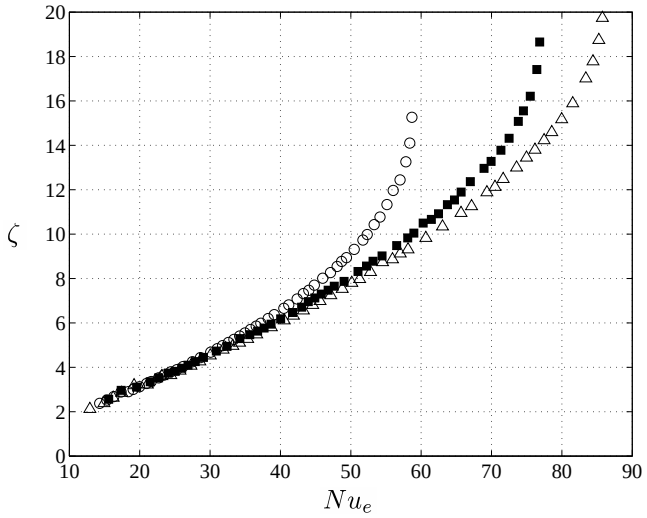


Figure 4.7: Pareto fronts in case of constrained solid volume $\bar{\sigma}_0 = 0.2$.(○) rectangular fins, (■) parabolic fins, (△) 4-th order fins.

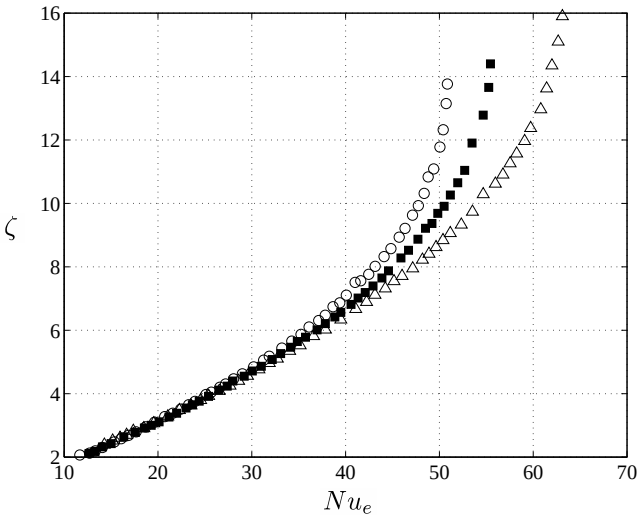


Figure 4.8: Pareto fronts in case of constrained solid volume $\bar{\sigma}_0 = 0.1$.(○) rectangular fins, (■) parabolic fins, (△) 4-th order fins.

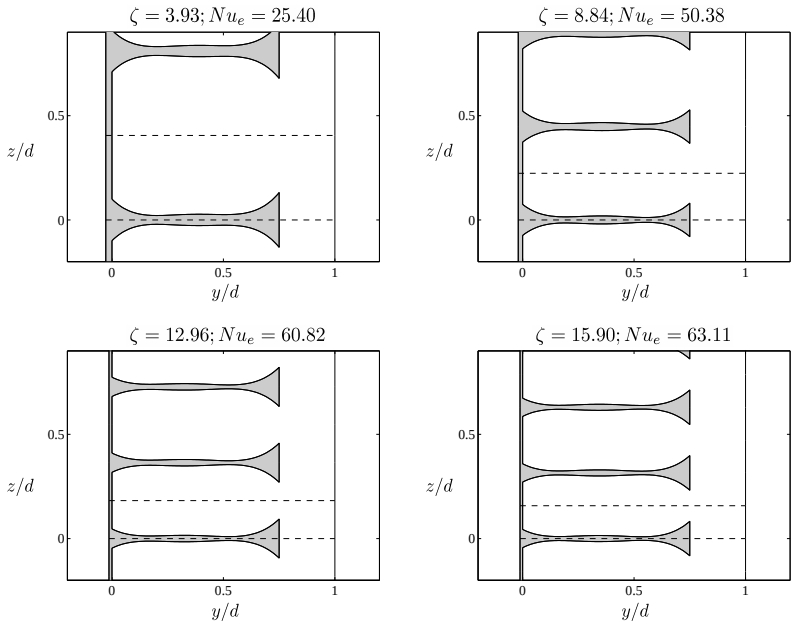


Figure 4.9: Optimum geometries for $n = 4$ with a constrained solid volume $\bar{\sigma}_0 = 0.1$.

changer profile. Moreover, if the hydraulic resistance is limited to a particular stringent value, the adoption of wavy fins do not improve the heat transfer of the device here analyzed.

In the present chapter also the effect of a constrained solid volume of the fins was studied. The main effect, as expected, consists in a reduction of the heat transfer. As in the unconstrained case, there are geometries that provide a great reduction of the hydraulic resistance without reducing the heat transfer. It must also be remarked that the present analysis has been limited to cases where the fluid flow has been considered laminar and fully developed. Therefore, the results presented here can be usefully employed only in cases where the entrance region is short in respect to the total length and where the Reynolds number is low.

Chapter 5

Multi-objective optimization of corrugated wall channels

As pointed out in the previous chapter, high performance heat exchangers are commonly required and the adoption of complex geometries is a typical way to increase the thermal characteristics of the overall devices. Specifically, complex shape configurations are employed to increase the heat exchange surface maintaining at the same time reduced volume and weight of the dissipator. Moreover, the heat transfer is influenced by the velocity field caused by the particular geometry employed.

Instead of wavy fins studied in chapter 4, another solution frequently employed consists on the usage of wavy wall surfaces, and typical configurations are V-shaped or sinusoidal corrugations. Only recently the interest has been focused to other kind of heat exchange surface profiles. Specifically, Fabbri[9] demonstrated that a fifth-order polynomial profile performs better than a sinusoidal corrugation. Nobile et al.[27] performed simulations on a wavy channel composed by two surfaces described by a NURBS parameterization. Moreover, in the paper of Nobile et al. an extensive literature review concerning the shape optimization of convective problems can be found and

from it two relevant remarks follow. The first is that most of the problems concerning with heat exchange were optimized with problem-specific algorithms and a lack of generality and robustness can arise from the methods used. The second remark is relative to the fact that, in all works they cited, it was considered a single-objective optimization task, whereas conflicting objectives can all play a fundamental role in the problem studied. As mentioned, in a heat exchanger, it is important to maximize the heat transfer but, at the same time, it is also of fundamental role to minimize the pressure drop. To adjust these remarks, Nobile et al. used a multiobjective commercial optimizer. In the work published by Fabbri[9] both of the above remarks can be formulated and in particular, the wavy wall heat exchanger were optimized by only maximizing the Nusselt number whereas the pressure drop was handled as a single value constraint instead as an objective to minimized. Similar considerations but formulated in more general terms are exposed also in the first chapter of this work where traditional methods are enlisted together their drawbacks.
In the present chapter, the heat transfer from a corrugated wall channel described by a fifth order polynomial profile is studied under a multiobjective point of view by means of the multiobjective differential evolution. Such a methodology allows to fix all remarks pointed out by other researchers since the same code employed by Fabbri has been used for consistency with previous works.

5.1 Governing equations and mathematical model

A channel composed by two parallel flat walls has been assumed. The upper wall is a zero thickness thermally insulated surface whereas, the lower wall is corrugated and crossed by a heat flux q'' which is uniformly imposed on

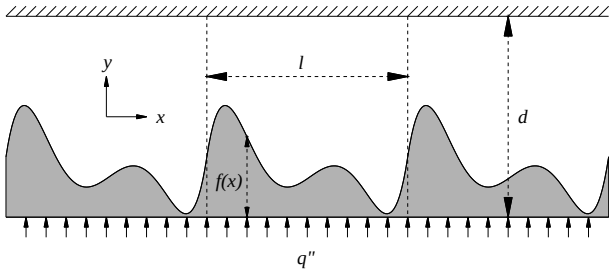


Figure 5.1: Wavy channel geometry.

its external surface (see Fig.5.1). The thickness of the corrugated wall is a periodic function f of the longitudinal coordinate x whose period is l. Between the two walls, a coolant fluid passes through in laminar flow in the x direction.
The following hypotheses have been introduced:

1. the system is in steady state;

2. the velocity and temperature profile are fully developed;

3. the fluid is incompressible and its properties are uniform;

4. the viscous dissipation within the fluid is negligible.

Due to the homogeneity in the normal direction to plane (x, y), the heat transfer performance of the channel can be studied by only determining the distribution of the velocity and the temperature on plane (x, y). Moreover, due to the periodicity of the channel shape, the study can be limited to a portion whose length is equal to l.
The velocity distribution must satisfy the mass balance and the momentum equations, which for a steady-state incompressible flow can be reduced to the

following form:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{5.1}$$

$$\rho\left(\frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y}\right) = -\frac{\partial p}{\partial x} + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{5.2}$$

$$\rho\left(\frac{\partial vu}{\partial x} + \frac{\partial v^2}{\partial y}\right) = -\frac{\partial p}{\partial y} + \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \tag{5.3}$$

where ρ and μ are the density and the dynamic viscosity of the fluid, respectively, u and v are the velocity components in the $x-$ and $y-$ direction, respectively, and p is the sum of the pressure and the gravity potential contribution.

The temperature distribution in the fluid must satisfy the following energy balance equation:

$$\rho c\left(\frac{\partial t_f u}{\partial x} + \frac{\partial t_f v}{\partial y}\right) = k\left(\frac{\partial^2 t_f}{\partial x^2} + \frac{\partial^2 t_f}{\partial y^2}\right) \tag{5.4}$$

where c and k are the specific heat and the thermal conductivity of the fluid, respectively, and t_f is the fluid temperature. In the solid the Laplace equation holds:

$$\left(\frac{\partial^2 t_s}{\partial x^2} + \frac{\partial^2 t_s}{\partial y^2}\right) = 0 \tag{5.5}$$

t_s being the solid temperature.

Equations 5.1-5.5 can be solved numerically by means of the control volume finite element code previously tested and employed by Fabbri[9]. Moreover, the channel performances have been evaluated considering the same parameters described by Fabbri for consistency with previous works. Specifically, Fabbri compared the wavy channel performances with those of a reference channel composed by an insulated flat surface and a flat surface crossed by a uniform heat flux equal to q''. To this aim, let us suppose that the inlet bulk temperature is the same for both channels. Under this condition, let $t_{b0}(x)$ be the bulk temperature and let us suppose that the temperature distribution

on the external surface of the corrugated wall $t_s(x,0)$ occurs on the surface of the reference channel. In this case, the local heat transfer coefficient in the reference channel would be:

$$h_x = \frac{q''}{t_s(x,0) - t_b(x)}. \tag{5.6}$$

The local Nusselt number would then result

$$Nu_x = \frac{h_x 2d}{k} \tag{5.7}$$

where d is the distance between the external surfaces of the channel. Referring to dimensionless entities, Nu_x can be calculated as follows

$$Nu_x = \frac{RePr}{L} \frac{1}{T_s(x,0) - T_b(x)} \tag{5.8}$$

where Re and Pr are the Reynolds and Prandtl numbers respectively and they are calculated as

$$Re = \frac{\rho w_0 2d}{\mu} \tag{5.9}$$

$$Pr = \frac{\mu c}{k_f} \tag{5.10}$$

being w_0 the average velocity. Moreover, in eq. 5.8, the dimensionless temperatures have been normalized as

$$T_s = \frac{t_s}{\Delta t} \tag{5.11}$$

$$T_b = \frac{t_b}{\Delta t} \tag{5.12}$$

where Δt is the coolant fluid bulk temperature drop which is the same in the reference and the corrugated channel. Lastly, L is the normalized period:

$$L = \frac{l}{d} \tag{5.13}$$

An equivalent Nusselt number for the corrugated channel can then be defined as the average of Nu_x:

$$Nu_e = \frac{1}{L} \int_0^L Nu_x dx \tag{5.14}$$

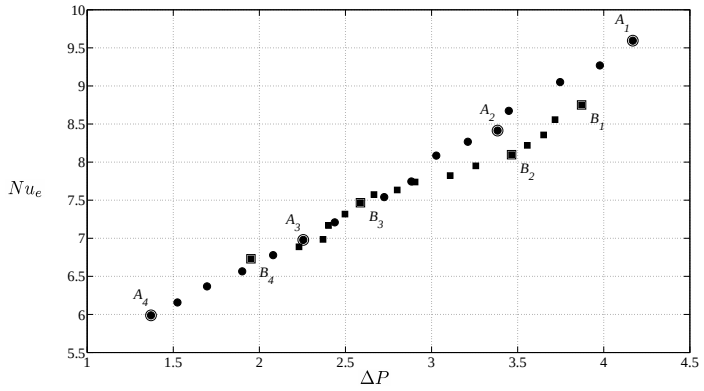


Figure 5.2: Polynomial profile Pareto front. (•) unconstrained case; (▪) constrained case: $\bar{f} = 0.2$.

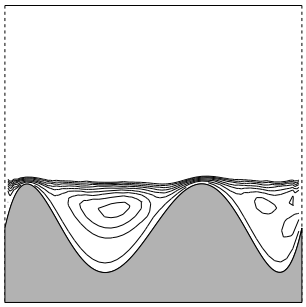
Such an equivalent Nusselt number corresponds to the average Nusselt number which would be calculated for the reference channel if on its heat flux crossed surface occurs the same temperature distribution as on the corrugated channel, for given heat flux and inlet bulk temperature. Moreover, it depends on the channel geometry, the Reynolds number, the Prandtl number, and on the ratio γ between the thermal conductivity of the solid and that of the fluid. Moreover, the normalized pressure P has been defined as:

$$P = \frac{p}{\Delta p_0} \tag{5.15}$$

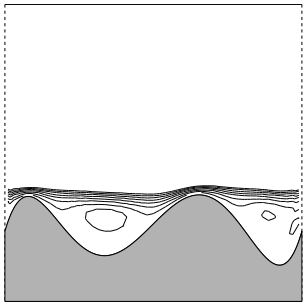
where Δp_0 is the pressure drop in a portion of the reference channel whose length is equal to l.

5.2 Results

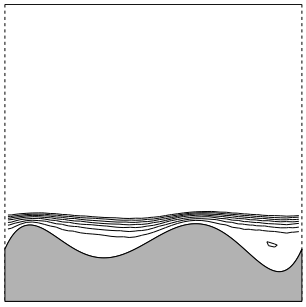
The in-house code previously developed Fabbri[9] has been utilized in the NSDE-FB multiobjective algorithm described in chapter 3 to determine the corrugation profiles that provide the best efficiency in terms of both the heat transfer and the pressure drop.



(a) A_1 stream lines.

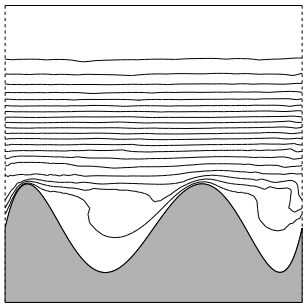


(b) A_2 stream lines.

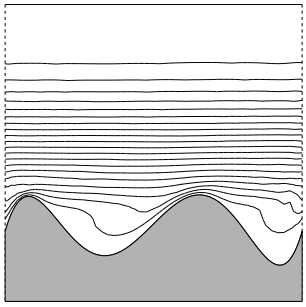


(c) A_3 stream lines.

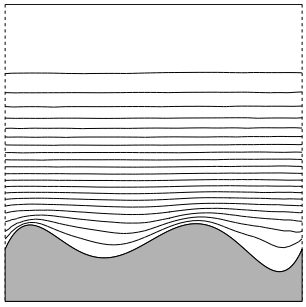
Figure 5.3: Unconstrained case stream lines.



(a) A_1 temperature isocurves.

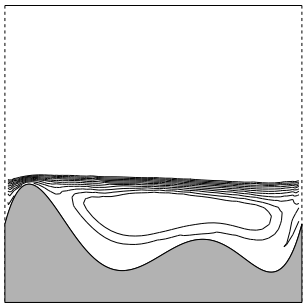


(b) A_2 temperature isocurves.

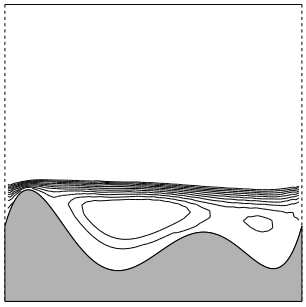


(c) A_3 temperature isocurves.

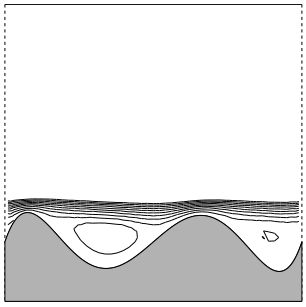
Figure 5.4: Unconstrained case temperature isocurves.



(a) B_1 stream lines.

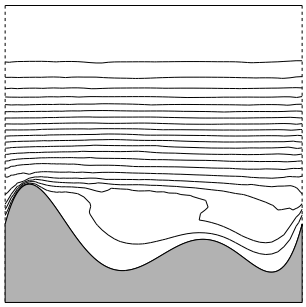


(b) B_2 stream lines.

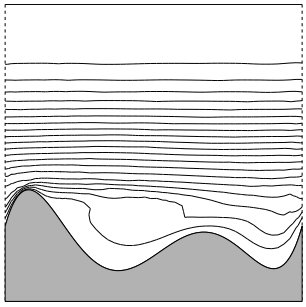


(c) B_3 stream lines.

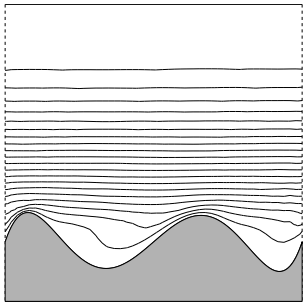
Figure 5.5: Constrained case stream lines($\bar{f} = 0.2$).



(a) B_1 temperature isocurves.



(b) B_2 temperature isocurves.



(c) B_3 temperature isocurves.

Figure 5.6: Constrained case temperature isocurves($\bar{f} = 0.2$).

For the corrugation profile $f(x)$, a fifth-order polynomial profile as in Fabbri has been adopted for consistency with his previous works. The function and its first derivative assume the same value in $x = 0$ and $x = l$. Moreover, the corrugation profile is univocally determined by four variables and the values f_i assumed by $f(x)$ in the first four of six equidistant points between 0 and l have been assumed.

Constraints have been considered for the minimum and maximum values of $f(x)$, and the corrugated wall volume. Specifically, to prevent the channel from being too narrow and the corrugated wall from being too thick, the following constraints have been imposed:

$$\max f(x), \leqslant 0.4 \tag{5.16}$$

$$\min f(x), \geqslant 0.1 \tag{5.17}$$

Moreover, the analysis has been limited to the case where L is equal to 1, γ to 500, and Reynolds and Prandtl numbers equal to 500 and 5 respectively. Figure 5.2 reports the Pareto front relative to the polynomial interpolation. Considering the geometry that maximizes the Nusselt number (labelled with A_1, see Fig. 5.3(a) and Fig. 5.4(a)), the corrugation profile is composed by two peaks alternated by two cavities. Near the peaks the heat is directly transferred to the main flow due to conduction, so that the isothermal lines are denser. In the cavities, the heat removed from the wall is transferred to the main flow through eddies. This convective heat transfer mechanism causes a reduction of the heat transfer coefficient with respect to the case of a maximum thickness flat wall channel.

By analyzing the geometries lying on the Pareto front (from A_1 to A_3, see Fig. 5.3 and Fig. 5.4), it can be seen that the two peaks y-coordinates remain at the same level and they progressively decrease causing a reduction of both the Nusselt number and the pressure drop. Specifically, the Nusselt number is lower due to the reduction of the local heat transfer coefficient since the

isothermal curves are less dense than the case of maximum heat transferred. Whereas, the pressure drop decreases since the main flow is less accelerated. Moreover, it can be seen that the bottom of the first cavity changes its position whereas, the y-coordinate of the second cavity bottom remains equal to the lower limit causing a reduction of the strength of the vortex in the first cavity. Such a behavior was expected since, in the case of maximum Nu_e, a strong vortex is present in the first cavity whereas two smaller vortices appear in the latter one. The changes in the vortical structures cause a reduction of both the heat transferred and the pressure drop.
Similar considerations can be formulated analyzing the Pareto-front of those geometries with a constrained solid volume(labelled as B_1, B_2, and B_3, see Fig. 5.5 and Fig. 5.6).

5.3 Conclusions

In the present chapter, the heat transfer from a wavy channel cooled by a laminar flow under conditions of forced convection and imposed heat flux was investigated. Specifically, a multiobjective optimization of a fifth order corrugation profile aiming to maximize the heat transfer and to minimize the pressure drop simultaneously was performed. Moreover, the optimization process considered two different cases: the former related to the case of a profile function constrained in its minimum and maximum values; the latter took into account also a constrained solid volume.
The simulations performed allowed to detect the Pareto fronts for the two cases and the analysis of the shape configurations on these fronts was carried out. In fact, the evaluation of the Pareto front geometries allowed to detect the main heat transfer mechanisms as well as the principal causes of pressure drop.
Finally, some considerations can be made comparing the Pareto fronts of

the way fins studied in the previous chapter and the Pareto front of a wavy channel. Although the two problems are different, it is worth noting that in the wavy channel the Nusselt number is a linear function of the normalized pressure drop. Therefore, it is not possible to identify geometries that provide a high reduction of the pressure drop but, at the same time, provide a small reduction of the Nusselt number as done for the wavy fins in the previous chapter.

Chapter 6

Conclusions

In the present work, the multi-objective optimization by genetic algorithms was investigated and applied to heat transfer problems. However, the direct application of genetic algorithms to the cases of interest was not a practical way due to a large amount of computational resources required. To address this issue, an intensive study on the reproduction processes employed by genetic algorithms was carried out since it was identified that this area could significantly improve. The analysis performed considered several reproduction methodologies available in literature and aimed to identify the more promising one. Moreover, on the basis of the results obtained, two new crossover techniques were suggested and compared with the available operators. Specifically, one of the two operators revealed to significantly improve the performances of a multiobjective algorithm in almost all the test cases analyzed.

From the application point of view, two problems where the heat transfer is involved were studied.
In the first case, the optimal geometries of wavy fins described by polynomial functions were investigated. Specifically, the parameter combinations that aim to maximize the heat transfer and, at the same time, aim to mini-

mize the hydraulic resistance. The result of this study consists in the Pareto front and two main considerations derive from its analysis:

- The *Pareto* front trend highlights that exist some geometries which allow to reduce the hydraulic resistance with a Nusselt number slightly reduced if compared with the best heat exchanger profile.
- The adoption of wavy fins do not always improve the heat transfer. Specifically, this is evident when the hydraulic resistance is limited to a stringent value.

In the latter case, the optimal profile shapes of corrugated wall channels were investigated. In this application, a linear Pareto front was detected and therefore, it was not possible to identify geometries that provide a reduction of the pressure drop but, at the same time, provide a small reduction of the Nusselt number as obtained in the wavy fins case.
In both the application analyzed, the multi-objective approach revealed to be an invaluable tool in order to identify the main heat transfer mechanisms as well as the principal causes of pressure drop.

Finally, it is worth noting that part of this work has been recently published by the author in the following article:

D. Copiello; G. Fabbri. Multi-objective genetic optimization of the heat transfer from longitudinal wavy fins. *International Journal of Heat and Mass Transfer*, 53:1167-1176, 2009.

On the contrary, the chapter that analyzes the corrugated channel relates to a study that is still in progress. Future works will focus on more complex corrugation profiles as well as the study of the entrance region effects.

Bibliography

[1] H.A. Abbass. A memetic pareto evolutionary approach to artifcial neural networks. In *Proc. of the Australian Joint Conference on Artifcial Intelligence*, volume 2256.

[2] H.A. Abbass. The self-adaptive pareto differential evolution algorithm. In *Proc. of the CEC'2002*, volume 1, pages 831–836, 2002.

[3] E. Zitzler; L. Thiele; M. Laumanns; C.M. Fonseca; V.G. da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, April 2003.

[4] J. Richardson D.E. Goldberg. Genetic algorithms with sharing for multimodal function optimization. In *Proc. of the Second International Conference on Genetic Algorithms and their application*, pages 41–49, 1987.

[5] R.J. Duffin. A variational problem relating to cooling fins. *J Math Mech*, 8:47–56, 1959.

[6] D. Copiello; G. Fabbri. Multi-objective genetic optimization of the heat transfer from longitudinal wavy fins. *International Journal of Heat and Mass Transfer*, 53:1167–1176, 2009.

[7] G. Fabbri. A genetic algorithm for fin profile optimization. *International Journal of Heat and Mass Transfer*, 40:2165–2172, 1997.

[8] G. Fabbri. Optimization of heat transfer through finned dissipators cooled by laminar flow. *International Journal of Heat and Fluid Flow*, 19:644–654, 1998.

[9] G. Fabbri. Heat transfer optimization in corrugated wall channels. 43:4299–4310, 2000.

[10] M. Spiga; G. Fabbri. Effcienza di dissipatori a profilo sinusoidale. Proc 12th UIT National Congress, L'Aquila, Italy, pages 197–204.

[11] T. Robič; B. Filipič. Demo: Differential evolution for multiobjective optimization. In Springer, editor, *Proc. of the EMO 2005*, volume 3410, pages 520–533, 2005.

[12] R.L. Haupt; S.E. Haupt. *Practical Genetic Algorithms.* Wiley-Interscience, 2004.

[13] A. Bar-Choen; M. Iyengar. Design optimization of air cooled heat sinks for sustainable development. IEEE Transactions on Components and Packaging Technologies, 2002.

[14] K.A. De Jong. *An analysis of the behavior of a class of genetic adaptive systems.* PhD thesis, University of Michigan, 1975.

[15] A.D. Snider; A.D. Kraus. The quest for the optimum longitudinal fin profile. *Heat Transfer Eng*, 8:19–25, 1987.

[16] A.D. Snider; A.D. Kraus; S. Graff; M. Rodriguez; A.G Kusmierczyk. Optimal fin profiles. classical and modern.

[17] A. Iorio; X. Li. Incorporating directional information within a differential evolution algorithm for multi-objective optimization. In *In Proc. GECCO'06*, volume 1, pages 691–697.

[18] A. Iorio; X. Li. Rotated test problems for assessing the performance of multi-objective optimization algorithms. In *In Proc. GECCO'06*, volume 1, pages 683–689, 2006.

[19] G. Fabbri; G. Lorenzini. Analisi numerica bidimensionale di dissipatori a profilo sinusoidale. Proc 13th UIT National Congress, Bologna, Italy, pages 491–499, 1995.

[20] C.J. Maday. The minimum weight one dimensional straight fin. *ASME J Eng Ind*, 96:161–165, 1974.

[21] K. Deb; S. Agrawal; A. Pratab; T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.

[22] H.A. Abbass; R. Sarker; C. Newton. Pde: A pareto-frontier differential evolution approach for multi-objective optimization problems. In *Proc. of the CEC'2001*, volume 2, pages 971–978, 2001.

[23] Vilfredo Pareto. *Cours D'économie politique*, volume I and II. F. Rouge, 1896.

[24] C. Poloni; A. Giurgevich; L. Onesti; V. Pediroda. Hybridization of a multi-objective genetic algorithm, a neural network and a classical optimizer for a complex design problem in fluid dynamics. *Computer Methods in App. Mech. and Eng.*, 186:403–402, 2000.

[25] K. Price. *Differential evolution*. 1999.

[26] LCA (Life Cycle Assessment Committee Report). Sumary of inventory data. Technical report, Japan Aluminium Association, 1999.

[27] E. Nobile; F. Pinto; G. Rizzetto. Geometric parametrization and multi-objective shape optimization of convective periodic channels. *Numerical Heat Transfer, Part B*, (50):425–453, 2006.

[28] H.A. Abbass; R. Sarker. The pareto differential evolution algorithm. *Int. J. on Artifcial Intelligence Tools*, 11:531–552, 2002.

[29] B.W. Silverman. *Density estimation for statistics and data analysis*. London: Chapman and Hall, 1986.

[30] E. Zitzler; 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.

[31] E. Zitzler; M. Laumanns; L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Laboratory, ETH Zurich, May 2001.

[32] R. Hilbert; G. Janiga; R. Baron; D. Thévenin. Multi-objective shape optimization of a heat exchanger using parallel genetic algorithms. *International Journal of Heat and Mass Transfer*, (49):2567–2577, 2006.

[33] A. Zamuda; J. Brest; B. Bošković; V. Žumer. Differential evolution for multiobjective optimization with self adaption. In *Proc. of the CEC'2007*, pages 3617–3624, 2007.

[34] C.S. Chang; D.Y. Xu. Differential evolution of fuzzy automatic train operation for mass rapid transit system. In *IEEE Proc. of Electric Power Applications*, volume 3, pages 206–212, 2000.

[35] C.S. Chang; D.Y. Xu. Differential evolution of fuzzy automatic train operation for mass rapid transit system. In *IEEE Proc. of Electric Power Applications*, volume 3, pages 206–212, 2000.

[36] E. Zitzler. *Evolutionary Algorithms for Multi-objective Optimization: Methods and Applications*. PhD thesis, ETH Zurich, 1999.