N° d'ordre :

Ecole Centrale de Lille

THESE

Présentée en vue
d'obtenir le grade de

DOCTEUR

en

Spécialité : Génie Electrique

par

Fouzia Moussouni-Messad

DOCTORAT DELIVRE PAR L'ECOLE CENTRALE DE LILLE

Titre de la thèse :

Méthodologie et algorithmes adaptés à l'optimisation multi-niveaux et multi-objectif de systèmes complexes

Soutenue le 08 juillet 2009 devant le jury d'examen :

Rapporteur	P. Di Barba	Professeur	Université de Pavia, Italie
Rapporteur	L. Gerbaud	Professeur	ENSE3 Grenoble INP, France
Examinateur	L. Krahenbuhl	Directeur de recherche	CNRS, France
Directeur de thèse	P. Brochet	Professeur	L2EP, Lille, France
Co-encadrant	S. Brisset	Maître de conférences HDR	L2EP, Lille, France
Invité	L. Nicod	Upstream activities manager	Alstom Transport, France
Invité	F. Gillon	Maître de conférences	L2EP, Lille, France

Thèse préparée dans le Laboratoire L2EP à l'Ecole Centrale de Lille

Ecole Doctorale SPI 072 (Lille I, Lille III, Artois, ULCO, UVHC, EC Lille)

École doctorale Sciences Pour l'Ingénieur SPI 072

N° d'ordre: **École Centrale de Lille**

Multi-level and multi-objective design optimization tools for handling complex systems

THÈSE

A dissertation in partial fulfillment of the degree of Philosophy July 08th 2009

in

Ecole Centrale de Lille

(Speciality Electrical engineering)

by

Fouzia Moussouni-Messad

Doctoral Commitee

Reporters:	P. Di Barba	Professor, University of Pavia, Italy
	L. Gerbaud	Professor, ENSE3 Grenoble INP , France
Examiners:	F. Gillon	Assistant professor, L2EP, France
	L. Krahenbuhl	Directeur de recherche CNRS, France
Supervisor:	P. Brochet	Professor, L2EP, France
	S. Brisset	Assistant professor, L2EP, France
Guests:	L. Nicod	Upstream activities manager, Alstom Transport

Laboratoire d'Electrotechnique et d'Electronique de Puissance de Lille

Résumé

La conception d'un système électrique est une tâche très complexe qui relève d'expertises dans différents domaines de compétence. Dans un contexte compétitif où l'avance technologique est un facteur déterminant, l'industrie cherche à réduire les temps d'étude et à fiabiliser les solutions trouvées par une approche méthodologique rigoureuse fournissant une solution optimale systémique. Il est alors nécessaire de construire des modèles et de mettre au point des méthodes d'optimisation compatibles avec ces préoccupations. En effet, l'optimisation unitaire de sous-systèmes (ou composants) sans prendre en compte les interactions ne permet pas d'obtenir un système optimal. Plus le système est complexe plus le travail est difficile et le temps de développement est important car il est difficile pour le concepteur d'appréhender le système dans toute sa globalité. Il est donc nécessaire d'intégrer la conception des composants dans une démarche systémique et globale qui prenne en compte à la fois les spécificités dun composant et ses relations avec le système qui l'emploie.

Analytical Target Cascading est une méthode d'optimisation multi niveaux de systèmes complexes. Cette approche hiérarchique consiste à décomposer un système complexe en sous-systèmes, jusqu'au niveau composant dont la conception relève d'algorithmes d'optimisation classiques. La solution optimale est alors trouvée par une technique de coordination qui assure la cohérence de tous les sous-systèmes.

Une première partie est consacrée à l'optimisation de composants électriques (moteur roue sans balais, transformateur de sécurité, moteur de traction) qui servent de cas test pour les différents algorithmes d'optimisation multi-objectif et/ou à variables mixtes étudiés et proposés dans cette étude. L'optimisation multi niveaux de systèmes complexes est étudiée dans la deuxième partie de la thèse, où une chaîne de traction électrique est choisie comme exemple de système complexe. L'approche de décomposition hiérarchique de la méthode *Analytical Target Cascading* se révèle bien adaptée à la démarche de conception optimale des systèmes complexes, tout en respectant l'organisation par produit de l'entreprise.

Mots-clés

Optimisation multi niveaux
Optimisation multidisciplinaire
Optimisation multi objectif
Optimisation à variables mixtes
Chaîne de traction ferroviaire

Abstract

The design of an electrical system is a very complex task which needs experts from various fields of competence. In a competitive environment, where technological advance is a key factor, industry seeks to reduce study time and to make solutions reliable by way of a rigorous methodology providing a systemic solution.

Then, it is necessary to build models and to develop optimization methods which are suitable with these concerns. Indeed, the optimization of sub-systems (or components) without taking into account the interaction does not allow to achieve an optimal system. More complex the system is more the work is difficult and the development time is important because it is difficult for the designer to understand and deal with the system in its complexity. Therefore, it is necessary to integrate the design components in a systemic and holistic approach to take into account, in the same time, the characteristics of a component and its relationship with the system it belongs to.

Analytical Target Cascading is a multi-level optimization method for handling complex systems. This hierarchical approach consists on the breaking-down of a complex system into sub-systems, and component where their optimal design is ensured by way of classical optimization algorithms. The optimal solution of the system must be composed of the component's solutions. Then a coordination strategy is needed to ensure consistency of all sub-systems.

First, the studied and proposed optimization algorithms for handling multi-objective optimization problems dealing with mixed-integer design variables are tested and compared on the optimization of electrical components (like brushless DC wheel motor, safety transformer, and traction motor). The second part of the thesis focuses on the multi-level optimization of complex systems. The optimization design of railway traction system is taken as a benchmark.

The hierarchical breaking-down approach proposed by Analytical Target Cascading method is well suited to ensure optimal design of complex systems, with respect of the company's product breaking-down structure.

Keywords

Multi-level optimization
Multidisciplinary design optimization
Mixed-integer optimization
Multi-objective optimization
Railway traction system

To my other half, my husband

To my parents

Contents

List of Figures

List of Tables

Background

1

Background

This work has been carried out as a part of a research project FuturElec 3, in the Laboratoire d'Electrotechnique et d'Electronique de Puissance de Lille" (L2EP) at Ecole Centrale de Lille with the OPTIMIZATION team.

The OPTIMIZATION team develops design methodologies of electric machines as well as the computing tools which collect them. The scientific objective is thus the computer aided engineering of electric machines. The axis structuring this research is the use of optimization, i.e. the formulation of optimization problems and their treatment.

This research project is initiated by CNRT "Réseaux et Machines Electriques du Futur" leading to design electrical machines with low environmental impact for the transport of the future. There is two key components to this project, i.e. two sub-projects are achieved:

1. Modelling and optimal design in railway applications traction system by [Kreuawan, 2008].
2. Multi-level and multi-objective design optimization tools for handling complex systems.

The second one represents the focus of this thesis.

1.1 Introduction

Modern engineering products are becoming increasingly complex, particularly in industries such as railway, aerospace and automotive.

Complex Systems term has many interpretations and emerges in many disciplines and domains. The complex nature of technological and engineering problems is manifested in different ways. To tackle the new challenges of Complex System Problems it requires a new research, a synergy of many disciplines, and an integration of new concept, methodologies and tools. The cornerstones for such developments are Modelling, Systems Methodologies, Decision Making/Control and Optimization.

Methodologies for complex system optimization are commonly developed based on the needs of a particular industry. Today improved techniques for the optimization of complex engineering problems are emerging from universities and research laboratories.

In this chapter three important points are introduced. The first point gives a brief analysis of a complex system where we tray to answer the question: why to break-down a system?

Thereafter the concept of the engineering design optimization is described. Then the new issues that this thesis will answer are introduced. To conclude, the thesis overview is given.

1.2 Complex system analysis

Words like engineering, model, optimization and system are widely used. Some definitions are given to clarify these terms for their use in this thesis.

Engineering

"Engineering: The creative application of scientific principles to design or develop structures, machines, apparatus, or manufacturing processes, or works utilizing them singly or in combination; or to construct or operate the same with full cognizance of their design; or to forecast their behaviour under specific operating conditions; all as respects an intended function, economics of operation and safety to life and property." (American Engineers'Council for Professional Development - ECPD).

Model

"Model: A pattern of something to be made". This word is first recorded in 1575 (Merriam-Webster [1]).
Models are validated only when they have been verified by observation and measurement under controlled conditions.

Optimization

"Optimization: An act, process, or methodology of making something (as a design, system, or decision) as fully perfect, functional, or effective as possible; specifically: the mathematical procedures (as finding the maximum of a function) involved in this". This word is first recorded in 1857 (Merriam-Webster).

System

"System: A regularly interacting or interdependent group of items forming a unified whole". This word is first recorded in 1720 (Merriam-Webster).
A system can be decomposed on other systems that are called **sub-systems**. For example, a system can be a rail train, an auto-mobile, an aircraft, a collection of a great number of sub-systems and components.

In systems design we need to decide about: what components may be used and what each component may do?

1.2.1 Complex system

Many definitions are considered in the literature. A complex system can be defined as a system consisting of sub-systems and interactions [Oliver *et al.*, 1997].

According to [Papalambros & Michelena, 2000] a complex system is one in which there are multiple interactions between many components. Where the term system is used in contrast to the component.

For [Allison, 2004], a complex system is an assembly of interacting members that is difficult to understand as a whole.

As will be shown in this thesis, a system may qualify as complex due to its large scale (large number of components or inputs), or due to strong interactions.

1.2.2 Why to break-down a system?

As the complexity of the engineering system increases, experience and knowledge of only one expert are not enough to understand and deal with the full problem. An approach to overcome these challenges is to decompose the whole problem into smaller ones that are easier to be solved. But the system solution must be composed of the component's solutions.

[1] Merriam-Webster is America's foremost publisher of language-related reference works. `http://www.merriam-webster.com`

This is why the complex system handling is not only a breaking-down of a large scale problem into components but also a coordination strategy to ensure the consistency between these components. where the consistency means the system and its components are connected in a coherent manner.

Partitioning: breaking a problem down to pieces

In practical, a complex engineering system can be partitioned according to objects or disciplines. Object decomposition involves dividing a system into components. In general use, the system is decomposed into sub-systems, the sub-systems are decomposed into components, the components into elements, and so on.

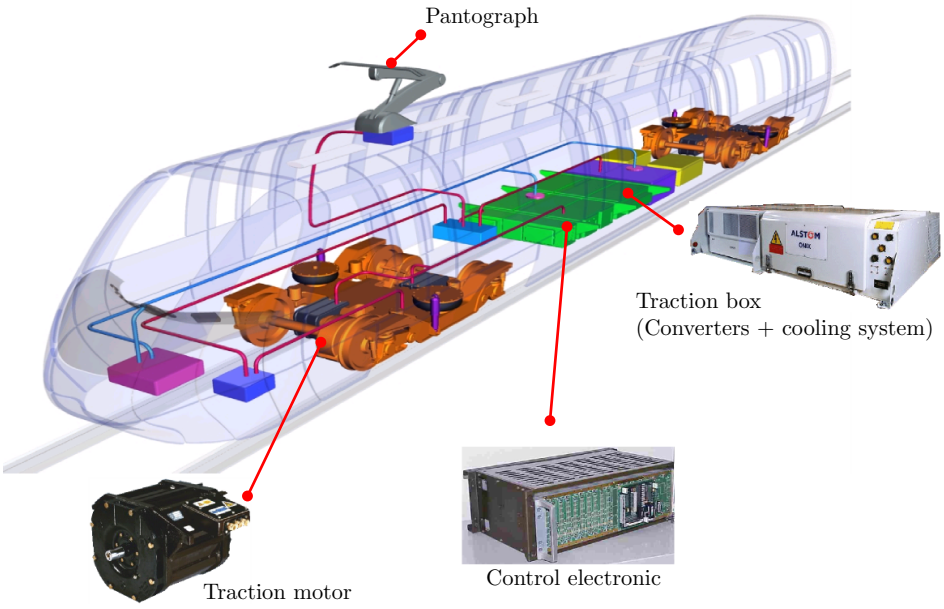


Figure 1.1: Decomposition of a rolling stock system according to objects.

For example in railway industry, Figure 1.1 shows the product breakdown of a train. It is decomposed into a bogie, a carriage and a traction system. The traction system can be again decomposed into sub-systems such as the traction box, the motor, and the transformer, etc. By this way, the system forms a multi-level hierarchical structure (Figure 1.3).

Discipline decomposition involves dividing a system according to disciplines. The same train system might be partitioned by discipline into mechanical, electrical, and thermal one (Figure 1.2).

The design of such complex engineering system is governed by mutual interactions between distinct disciplines. Then, using multi-disciplinary design optimization methodologies (MDO), these disciplines should be considered simultaneously to ensure an efficient design of the engineering system [Allison *et al.*, 2007].

Much work [Krishnamachari & Papalambros, 1995] and [Wagner & Papalambros, 1993] has already been done in the area of the system partitioning methodology. As shown in [Kreuawan, 2008] a complex system can broken down to components and each component may be made up of distinct disiplines analyses. This thesis will focus on the multi-level partitioning according to objects.

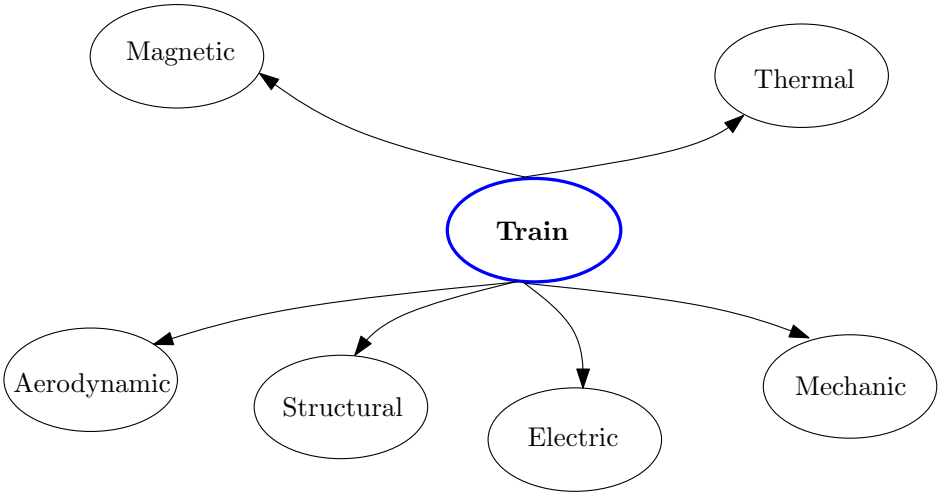


Figure 1.2: Multidisciplinary analysis.

Coordination: putting the pieces back together

Coordination strategies based on engineering intuition are usually posed in an empirical manner, for example in the project review meetings, and cannot guarantee that they converge to the same solution set as that of the undecomposed problem.
In coordination the challenge is to prove **convergence** of the strategy. Convergence formally would mean that the non-partitioned problem and the partitioned one solved with coordination have the same solution set. This is often hard or impossible to prove for practical non-linear problems.

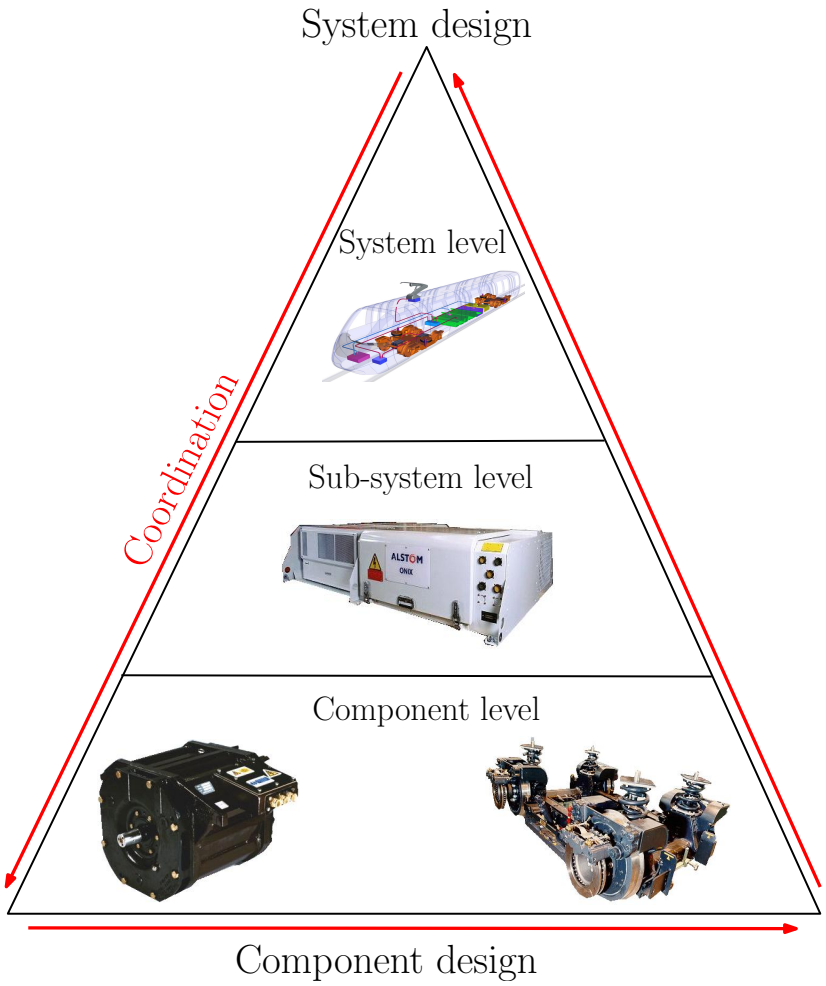


Figure 1.3: Rolling stock system decomposition and coordination.

1.3 Engineering design optimization

The goal of the design process may range from providing a practicable solution to a problem where none is previously known to improving on or replacing an existing design. In many design problems there are several possible alternative design concepts and design variables which specify, for example, the dimensions, proportions, and other details of the item [Fox, 1971]. The optimum design aspect arises because it is assumed that the values of these design variables are to be chosen in such a way that the design will be the one that satisfies all the limitations and the restrictions placed on it and it is best in some sense. Then, such a final design represents what it is called **optimum design**.

In mathematics and computer science, an optimization problem is the problem of finding the best solution from all feasible solutions. The formal mathematical model of a mono-objective optimization problem is a statement of the form:

$$\begin{cases} \textit{minimize} & f(X) \\ \textit{subject to} & h_i(X) = 0,\ i = 1, \ldots, l \\ & g_j(X) \leq 0,\ j = 1, \ldots, m \\ & X = [x_1, x_2, ..., x_n] \in S \end{cases} \tag{1.1}$$

where $f(X)$ is the scalar-value of objective function, to be minimized, that is function of the design variables X. $g(X)$ and $h(X)$ are the inequality and equality vector-valued of the constraint functions. S represents the research space.

In applied mathematical terminology, design variables are the unknowns of the optimization problem (1.1) to be solved. For example design variables can be associated with the object's size like its length and height or they may represent the number of items. According to previous work in the design optimization domain, the choice of design variables is the responsibility of the designer. Indeed, it can be guided for example by the design of experiment methodology, the fractional factorial design, and Taguchi design which are able to determine what are the more significant design variables influencing the design.

In practical, design problems often include two or more conflicting objectives, leading to the multi-objective optimization problems:

$$\begin{cases} \textit{minimize} & F(X) \\ \textit{subject to} & g(X) \leq 0 \\ & h(X) = 0 \\ & X = [x_1, x_2, ..., x_n] \in S \end{cases} \tag{1.2}$$

where $F(X)$ is the vector-valued of the objective functions of the problem.

An other type of optimization problems, deriving from engineering situations, are the mixed-integer optimization problems that deal with various kinds of variables: integer, discrete, zero-one and continuous.

Design optimization is the application of numerical algorithms and techniques to engineering systems in order to assist the engineers and ensure that the finished design will have the high performance, high reliability, low weight, and low cost.

Typical engineering systems are described by very large numbers of design variables, and the engineers have for task to specify suitable values for these variables. Because of the size and complexity of the engineering design problems, even the most skilled engineers are unable to take into account all of the variables simultaneously. This is why the recent appearance of decomposition-based optimization tools, with the ability to control the entire design optimization process, represent a significant advance. Using this versatile new capability the design can be tackled in a more comprehensive way, taking into account latest advances in multi-level, multidisciplinary analysis, and multi-objective optimization.

1.4 New issues

In this thesis some most known optimization algorithms are studied, implemented, modified and compared on the optimization of several practical test cases. But beside the parameter tuning of these algorithms, and their comparison to each other, the objectives of this thesis is to propose new features in order to answer the following engineering issues:

1.4.1 Mixed-integer design optimization

One way to solve mixed-integer optimization problems, is to consider all design variables as continuous. Thereafter, an appropriate continuous-variable optimization method is applied. Then, at the end, the optimal solution is truncated (rounded off). However, the obtained solution maybe infeasible or inaccurate [Cao & Wu, 1997, Kim *et al.*, 1998].

According to the state of the art, the evolutionary algorithms (EA)can be modified to be able to handle optimization problems which contain integer, discrete, zero-one and continuous variables. The modified EA provides an improvement in global search and convergence performance in a mixed-variable space.

A suitable approach to handle problems with various kinds of variables and constraints represent the first issue that will be discussed in this thesis.

1.4.2 Multi-objective design optimization

In the Engineering design a lot of problems deal with conflicting objectives. The main difficulty of such hard problems is to choose the suitable optimization method
[Alighanbari *et al.*, 2005], [Besharati *et al.*, 2006], [Chung *et al.*, 1996], [Erbas *et al.*, 2006], [Kim & De Weck, 2005], [Kim *et al.*, 2006a], [Qin & Zhao, 2006], [Tappeta & Renaud, 2001], and [Zeljkovic & Maksimovic, 2006].

Indeed, an ideal method gives a Pareto optimal set having a uniform distribution and a better coverage of the objective space, and even finds solutions in non-convex region in a short time. Unfortunately such a method does not exist.

In this thesis the focus will be on some multi-objective optimization algorithms: the weighting methods [Kim & De Weck, 2005, Kim & De Weck, 2006], and the multi-objective evolutionary algorithms [Deb *et al.*, 2002, Zitzler *et al.*, 2001].

1.4.3 Multi-level design optimization

The design of a complex system is a very delicate task. For example in rolling stock industry, optimization techniques are used in the design process to improve the performances of components. But without taking into account the interaction between sub-systems, some design leads to sub-optimal system. In order to design a system or even a sub-system, it is essential to consider the system as a whole [Papalambros & Michelena, 2000]. Thus this leads to a large scale problem with a large number of inputs and strong interactions.

As explained before, one approach to overcome these challenges is to decompose the whole problem into smaller ones that are easier to be solved.

Analytical target cascading (ATC) is a methodology that decompose a whole system into sub-systems to form a multilevel hierarchical structure. ATC gives special importance to the interactions between components of the same system that are frequently more difficult to understand than the components itself.

A well characterization of these interactions and providing an opportunity to exploit the synergy between components represent an other issue for this thesis. Furthermore, it will be emphasized that ATC has never been used, as we known, in electrical engineering.

1.5 Thesis overview

This chapter introduced the basic concepts of a complex system and how the engineers can lead to an optimal design. This thesis has to answer for three new issues:

- the proposition of a suitable approach in order to handle various kinds of variables and constraints for the electrical optimization problems;
- give of suitable multi-objective optimization algorithms in order to fast build a Pareto front with accuracy;
- to qualify ATC as efficient way to design electrical devices and systems;
- the improvement of the convergence of ATC which is a powerful strategy for the complex system optimization.

Mainly, the thesis is organized into two parts:

- Part I: State of the art.
- Part II: Improvement of algorithms.

Part I, State of the art, offers a critical review of the single-level and the multi-level optimization. In chapter 2, mono-objective, mixed-integer and multi-objective optimization are presented briefly. Chapter 3 introduces formal approaches of complex systems optimization and explains the concept of the analytical target cascading (ATC). Knowledge of ATC enables a detailed understanding of the most fundamental multi-level optimization and multidisciplinary design optimization (MDO) approaches.

Part II is devoted to present and discuss the contribution of the thesis. In the case of a mono-level optimization, the contribution consists on the modification, improvement and adaptation of some algorithms that are selected from the literature.

In chapter 4 the parameters tuning of the selected optimization algorithms itself is studied and discussed. Therafter, by means of the tuning results, several mono-objective optimization algorithms are compared in chapter 5. Several improvements are proposed.

How to assess the performance of some powerful multi-objective optimization algorithms is the focus of chapter 6, where the constraints handling is discussed too. In one hand, improved algorithms for handling constrained multi-objective optimization problems are proposed. In an other hand, an hybrid algorithm is developed in the aim to reduce the computational time when dealing with 3 dimensional finite elements (3D FE) models.

Chapter 7 is devoted to propose an improved algorithm dealing in the same time with multi-objective and mixed-integer optimization problems.

As explained before, ATC has several shortcomings especially how to ensure the consistency between components of the same system. The improvement of the ATC convergence is presented in chapter 8, where a new formulation is proposed.

Throughout part II, Improvement of algorithms, practical benchmarks from electrical engineering domain are employed in order to establish the important concepts introduced before. The proposed algorithms are compared and tested on the optimization design of:

1. Railway traction system components:
    - Brushless DC wheel motor, using a mono-objective and a multi-objective optimization;

- Safety isolating transformer in the case of a combinatorial, and a multi-objective optimization using 3D finite element model;
- Traction motor by applying a mixed-integer multi-objective optimization.

2. Railway traction system with the multi-level optimization approach ATC.

Part I

State of the art

2

Mono-level optimization

This part, State of the art, offers a critical review of the mono-level and the multi-level optimization. In chapter 2, mono-objective, mixed-integer and multi-objective optimization are presented briefly. Chapter 3 introduces formal approaches of complex systems optimization and explains the concept of the analytical target cascading (ATC).

In this study, mono-level optimization domain includes all kinds of optimization that deal with problems having a mono-level structure that are mono-objective optimization and multi-objective optimization. Furthermore, both kinds of optimization may deal with various types of design variables (integer, discrete, zero-one and continuous) leading to a mixed-integer optimization.

A classification is needed to surround the mono-objective optimization domain. Indeed, according to the state of the art, mainly there are two classes of mono-objective optimization (Figure 2.1):

1. combinatorial optimization,
2. and continuous optimization.

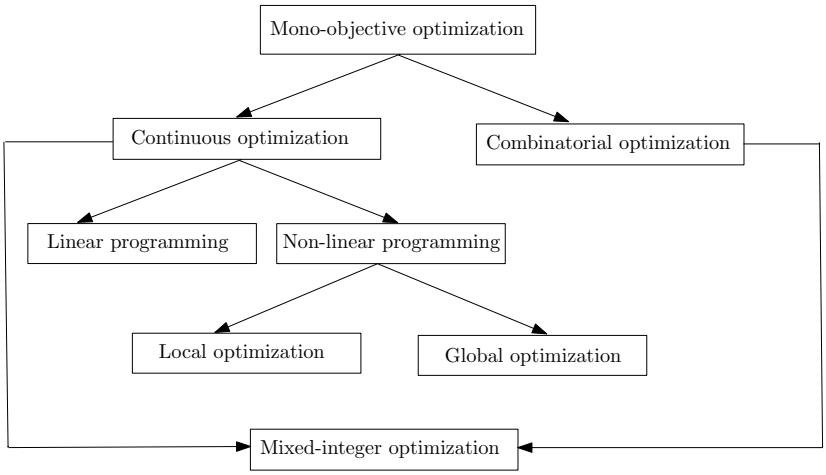


Figure 2.1: Classification of mono-objective optimization problems.

Continuous optimization
As its name indicated, with the continuous optimization the design variables are all continu-

ous. Generally, the linear programming (LP) is separated from the non-linear programming (NLP) which is considered as a hard optimization.

Linear programming
This kind of optimization treats problems with objective and constraints functions that are only linear. However in practice, especially in electrical engineering, these functions are mostly non-linear or they can not be expressed analytically.

Non-linear programming
Optimization problems whose mathematical models have non-linear equations are called Non-Linear Programming (NLP). Note that, the presence of only one non-linear equation in the model is sufficient to classify the problem as a NLP problem. In general, with NLP two kinds of optimizations are highlighted:

- local optimization,
- and global optimization.

Local optimization
Local search needs generally a starting point X_0 in order to move around the neighbourhood of X_0 and gives the best solution found in this region, i.e. a local optimum. But, in several practical cases, it is necessary to find the best solution, i.e. a global solution, in the whole search space.

Global optimization
The emerging field of Global Optimization (GO) deals with mathematical programming problems, in the presence of multiple local optima, and seeks to find the best solution (or global solution). Unfortunately, looking for a global solution is an optimization problem which is intrinsically very difficult to solve. Indeed, based on both theoretical analysis (introduced below) and practical experience, the time required to solve a GO problem increases rapidly, really exponentially, with the number of variables and constraints [Minoux, 1983], [Adeli, 1994], [Horst & Tuy, 1995], and [Venkataraman, 2001].

In other term, global optimization problems belong to the complexity class of **NP-hard** problems. This implies natural limits on the resolvability of such problems in practice as it is illustrated in section 2.1.2.

This thesis does not deal with LP problems but focuses on the NLP ones, especially the global optimization problems.

Combinatorial optimization
Contrary to a continuous optimization, combinatorial optimization deals with discrete, integer and zero-one (binary) design variables. The number of possible solutions is limited but generally very large: a combinatorial number which is about $n!$, $2^n, \ldots$, where n is the number of the design variables. Then it is impossible to explore all the solutions (different combinations of the variables values) in a moderate time.

Mixed-integer optimization
As its name indicated, this kind of optimization deals with continuous, integer, discrete and zero-one design variables in the same time. In this thesis it will be shown how to modify global optimization algorithm like genetic algorithms to solve optimization problems with several kind of variables.

First mono-objective optimization is discussed followed by a brief review of a mixed integer optimization where the design variables that define the function to be optimized can be continuous, discrete and have zero-one values. Thereafter, a multi-objective optimization is introduced.

2.1 Mono-objective optimization

In the following, first we discuss the mathematical formulation of a mono-objective optimization problem. Thereafter some elements of a global optimization and the analytical conditions of a convergence of a local search are given. Then, we tray to give a classification and list the most important optimization method to solve mono-objective optimization problems.

2.1.1 Optimization problem formulation

Mathematical formulation of an optimum design in engineering is usually as:

$$\begin{cases} \min f(X) & \\ \textit{subject to} & h_i(X) = 0,\ i = 1, \ldots, l \\ & g_j(X) \leq 0,\ j = 1, \ldots, m \\ & X = [x_1, x_2, ..., x_n] \in S \end{cases} \tag{2.1}$$

In this formulation X is the vector of problem variables, $f(X)$ is the objective function, $h_i(X),\ i = 1, \ldots, l$ and $g_j(X),\ j = 1, \ldots, m$ are the set of equality and inequality constraints of the problem respectively.

Before to discuss the general classification of optimization problems and how they can be solved effectively, some definitions and responses to what is a global and local optimum, and answer to what is convexity, convex system and convex function are given first.

Convexity

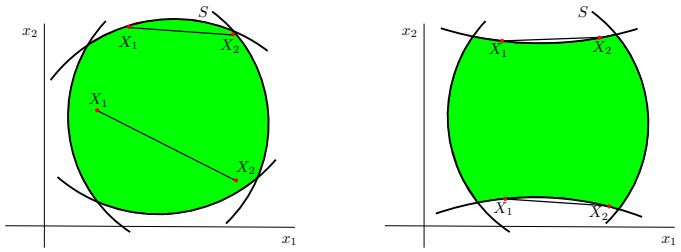
A minimization problem is called convex if the objective function $f(X)$ is convex and the set of constraints $g_j(X)$ and $h_i(X)$ bounds a convex domain, otherwise the problem is non-convex [Adeli, 1994, Venkataraman, 2001]. In accordance with that, linear programming (LP) problems are always convex [Adeli, 1994]. However non-linear programming (NLP) problems, in most cases, are non-convex then hard to be solved.

Furthermore, if a minimization problem is convex then there is only one minimum which is the solution [Adeli, 1994]. On contrary, in non-convex problems several local minima may exist and it is more difficult to find the global optimum.

This study focuses on how to find the global solution of non-convex NLP problems. Then, it is obvious that some terms like convexity, convex set, convex and non-convex function must be clarified. For this purpose, few definitions are given below.

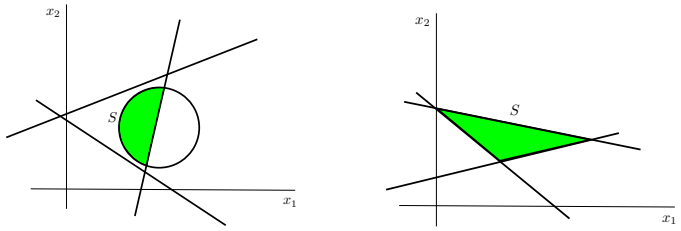
Definition 2.1.1 *(Convexity [Venkataraman, 2001]) If an optimization problem can be shown to be convex, then the local minimum will also be a global minimum. A convex optimization problem requires the objective function to be convex. If constraints are present, then all of the inequality constraints must be convex, while the equality constraints must be linear.*

Definition 2.1.2 *(Convex Set [Venkataraman, 2001]) This usually relates to the design vector (or design point) of the problem. Consider two design points X_1 and X_2. As before, the set S defines a design region and by implication both X_1 and X_2 belong to S. The line joining X_1 and X_2 contains a whole series of new design points. If all of these points also belong to S, then S is a convex set. A closed set S requires that the points X_1 and X_2 can also be*



(a) Convex system.

(b) Nonconvex system.



(c) Intersection of convex systems is a convex system.

(d) Systems bounded by linear constraints are convex.

Figure 2.2: Convex and nonconvex system [Venkataraman, 2001].

on the boundary. If, however, the line joining X_1 and X_2 contains some points that do not belong to S, then S is not a convex set.

In other words, the domain S is convex if for $0 < \alpha < 1$ any point $X = \alpha X_1 + (1 - \alpha)X_2$ is also inside the domain. For a two-dimensional design problem, an example of convex and non-convex domains are shown in Figure 2.2(a) and 2.2(b) respectively. As in Figure 2.2(c), the intersection of convex systems is also convex. Systems bounded by straight lines, as the example in Figure 2.2(d), are always convex. Convex system may be bounded or unbounded.

Definition 2.1.3 *(Convex Function [Venkataraman, 2001]) A convex function is defined only for a convex set. Once again let S be a convex set from which two design points X_1 and X_2 are selected. Let f be the function whose convexity is to be established. Consider any point X_a on a line joining X_1 and X_2. The function f is convex if $f(X_a)$ is less than the value of the corresponding point on the line joining X_1 and X_2.*

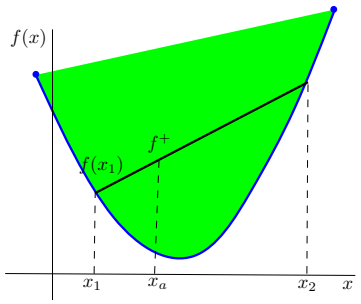


Figure 2.3: Convex function.

In n dimensions, convexity of a function is not easy to see then a one-dimension representation is illustrated in Figure 2.3. The condition for convexity in this figure is:

$$f(X_a) \leq f^+ \tag{2.2}$$

In other term, a mathematical condition of convexity is that the Hessian matrix of the function f must be positive semi-infinite at all points in the set S. This corresponds to the **KT conditions** (presented below on page 26) for unconstrained minimization problem.

As for convexity concept, global and local optimum theory should be defined due to their large use in this thesis.

Global versus local solution

Following [Horst & Tuy, 1995], global optimum of an optimization problem is defined as follows:

Definition 2.1.4 *(Global minimum) Given a non-empty closed set $S \subset R^n$, and a continuous function $f : A \longrightarrow R$, where $A \subset R^n$ is a suitable set containning S. The point X^* satisfying:*

$$f(X^*) \leq f(X),\ \forall X \in S$$

*is called a **global solution** of f over S. The corresponding value of f is called a **global minimum** of f over S.*

Then an efficient optimization method has to find at least one point $X^* \in S$ satisfying $f(X^*) \leq f(X)$ for all $X \in S$ or show that such a point does not exist.
According to the literature [Fox, 1971], [Minoux, 1983], [Adeli, 1994], [Horst & Tuy, 1995], [Venkataraman, 2001], a global minimum for the function $f(X)$ exists if the function is continuous on the non-empty feasible set S that is closed and bounded. According to [Venkataraman, 2001] and [Minoux, 1983], the conditions for the existence of $X^* \in S$ are attributed to the well-known Theorem of **Weiersteass**.

However, it is important to note that while the global minimum is guaranteed if the conditions are satisfied, they do not negate the existence of the global solution if they are not satisfied. For the sake of simplicity and clarity, throughout this thesis, it will be assumed that a solution $X^* \in S$ exists. Moreover, it is essential to realize that global optimum may still exist even if the convexity conditions are not met.

When the feasible region or the objective function of an optimization problem does not present convexity, there may be several local minima, where a local minimum X^* is defined as follow:

Definition 2.1.5 *(Local minimum) [Horst & Tuy, 1995] Let $\|.\|$ denote the Euclidean norm in R^n and let $\epsilon > 0$ be a real number. Then an ϵ-neighbourhood of a point $X^* \in R^n$ is defined as the open ball*

$$N(X^*, \epsilon) = \{X \in R^n, \|X - X^*\| < \epsilon\}$$

centred at X^* with radius ϵ.

A real-valued function f defined on the real line R is said to have a local minimum point at the point X^*, if there exists some $\epsilon > 0$, such that

$$f(X^*) < f(X),\ \forall X \in N(X^*, \epsilon)$$

To put it simply, a local minimum X^* is defined as a point for which there exists some $\epsilon > 0$, so that for all X such that $\|X - X^*\| \leq \epsilon$ then $f(X^*) \leq f(X)$. This holds to say, on some region around X^* all of the function values are greater than or equal to the value at that point.

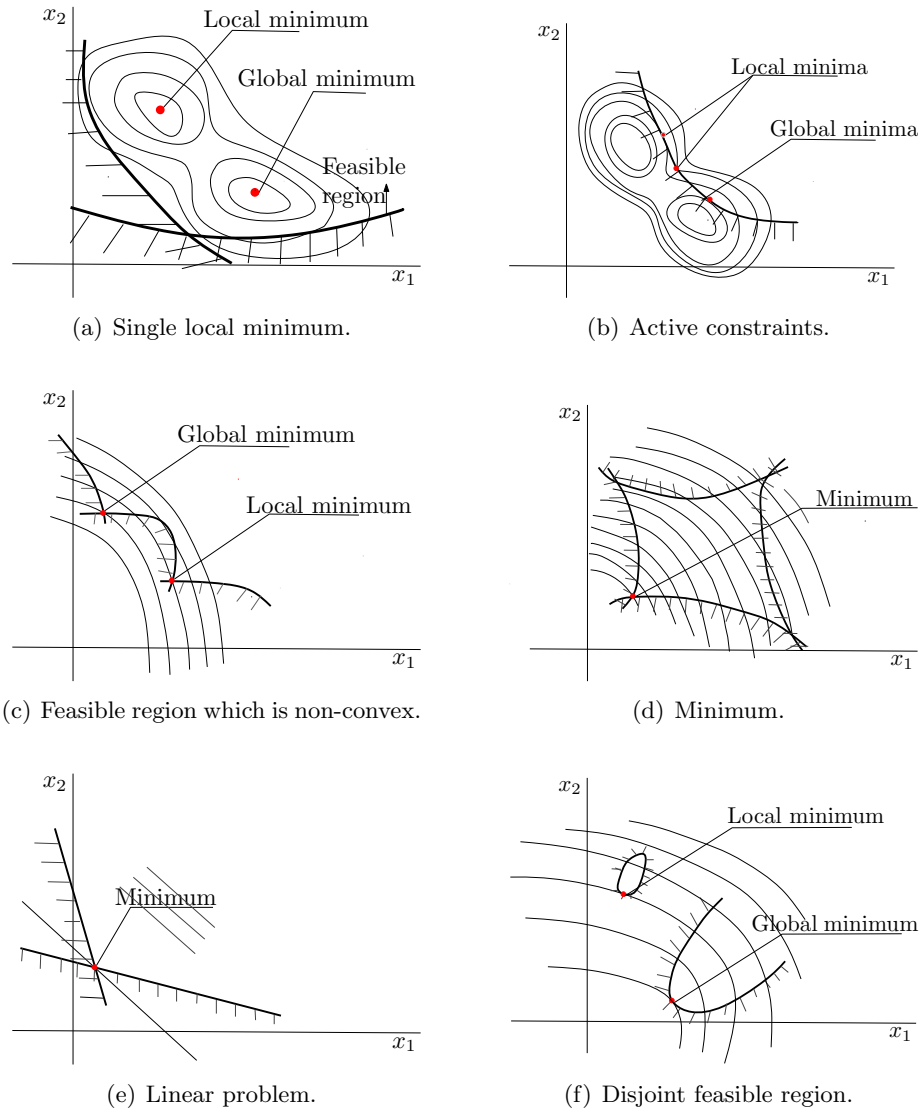
To conclude, note that since $\max f(X) = -\min(-f(X));\ X \in S$, global maximization problems are also included in equation (2.1) presented on page 21. Then global and local maxima are also concerned by these definitions.

Feasibility versus optimality

In most cases, the design space S is often specified by a set of constraints, equalities or inequalities, that the design points $X \in S$ have to satisfy.

In the n-dimensional Cartesian space, the point X corresponds to a vector of length $n = |X|$, where $|X|$ denotes the number of design variables. The boundaries of the region are part of the set S (closed): the lower and upper bounds are notated X^{Low} and X^{Up}. The length of the vector X should be a finite number.

All points $X = [x_1, x_2, ..., x_n] \in S$ which satisfy the constraints $h_i(X),\ i = 1, \ldots, l$ and $g_j(X),\ j = 1, \ldots, m$ are called feasible design and form the feasible region. Because of the optimal solution must be feasible, then feasibility is more important that optimality [Venkataraman, 2001], [Adeli, 1994], [Fox, 1971], and [Minoux, 1983].



(a) Single local minimum. (b) Active constraints.

(c) Feasible region which is non-convex. (d) Minimum.

(e) Linear problem. (f) Disjoint feasible region.

Figure 2.4: Shape of a feasible region in minimization problems [Adeli, 1994].

Figure 5.3 shows several shapes of feasible region for two-variables design problems and the problem solution in each case. In the case of Figure 2.4(a), several local minima exist inside the feasible region. All constraints are passive and the problem should be considered as an unconstrained optimization problem. In Figure 2.4(b) there are local minima due to the shape of the objective function, and some inequality constraints $g_j(X) \leq 0$ are active. Local minima appear in Figure 2.4(c) because of the geometry of constraints. The optimum is on

the boundary with some $g_j(X) = 0$. Figure 2.4(d) represents a non-linear problem having a unique constrained minimum as a solution. A linear problem which has only one minimum is illustrated in Figure 2.4(e). Figure 2.4(f) presents a problem with disjoint feasible regions, where each of the many of the aforementioned cases can occur. This kind of problem is considered as a hard optimization. An active constraint is just the opposite of a passive one, and it means the solution is on its boundary. Note that contrary to passive constraints, the active ones are sizeable constraints in an optimization design problem.

Classification of optimization problems

The variables that define the function to be optimized can be continuous and/or discrete and, in addition, they often have to satisfy certain constraints.

Table 2.1 gives a brief review of a classification of continuous optimization problems, where all variables and all functions are continuous.

Table 2.1: Optimization problem classification.

Model-class	Properties of the objective function $f(X)$	Properties of the constraint functions $h_i(X)$ and $g_j(X)$
- Lipschitz optimization problems	- Lipschitz-continuous function	- Lipschitz-continuous functions
- Quadratic optimization problems	- Indefinite-quadratic function	- Linear or quadratic functions
- Linear optimization problems (LP)	- Linear function	- Linear functions
- Non-linear optimization problems (NLP)	- Non-linear function	- At least one of the constraints is a non-linear function
- Convex programming problems	- Convex function	- Equality constraint functions, - Linear inequality constraints' function

After these brief definitions and the optimization problem classification, the theory of the global optimization and the local optimization are presented. First some issues regarding the global optimization are given. Then some important elements of the stochastic global optimization are discussed, followed by the presentation of the most known optimization methods that can be used for finding the optimal solutions of hard optimization problems.

2.1.2 Global optimization

With the optimization problem formulated in equation (2.1), it is necessary to compute the global optimum (or a likely approximation) of the objective function.

In practice, this objective function is non-convex and complex. Then it is not possible to obtain an exact solution to such hard problem. Indeed, in most cases of practical interest, the number of local optima is unknown and it can be quite large since it increases, on the average, exponentially with the number of variables of the problem.

In the presence of such structure, optimization problems can be extremely difficult. For illustration, see Figure 2.5 which displays a relatively simple composition of a trigonometric function in just two variables (denoted by x_1 and x_2) [Spall, 2003]. The objective function is defined over the search space $S = [0, 10]^2$, and formulated as follow: $f(X) = x_1 sin(4x_1) + 1.1\, x_2 sin(2x_2)$, $X = [x_1, x_2]$. This function has many local minima, with one global minimum at $X^* = [9.039, 8.668]$, corresponding to $f(X) = -18.555$. Under these circumstances, as shown in Figure 2.5, it is essential to use a good global optimization method. Furthermore,

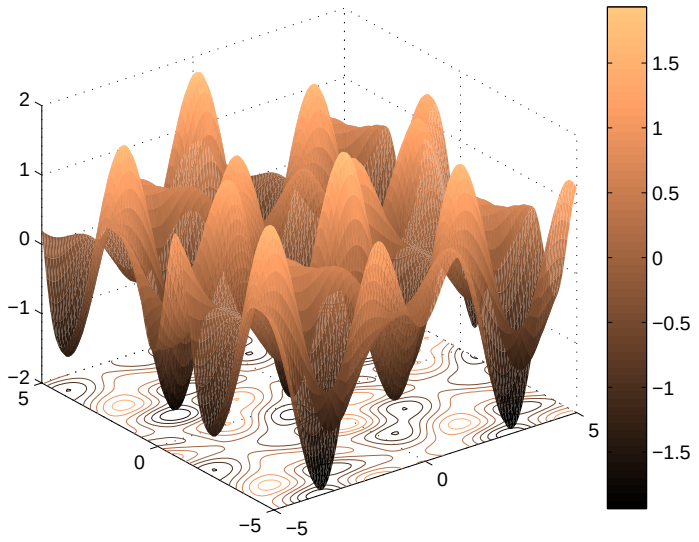


Figure 2.5: Many-minima continuous problem.

instead of exact solutions, diverse numerical approximations to the global optimal solution can be accepted.

In the past ten years, global optimization has received a lot of attention due to the success of some algorithms like evolutionary algorithms especially genetic algorithms (GA), ant colony optimization (ACO) and particle swarm optimization (PSO) which are widely studied in this work.

Unlike global optimization methods, most of the traditional approaches like local optimization methods fail to escape from a local optimum in order to continue the search for the global solution. However, as will be shown in the second part of this study, local search techniques can be used especially to improve the convergence of the global optimization algorithms. Then it seems useful to present briefly the concept of local search.

In the following, Kuhn-Tucker conditions which are used to prove the convergence of local search methods, exactly for non-linear and non-convex optimization algorithms, are discussed. Thereafter, some elements of the stochastic global optimization are introduced. Then, in the next section, suitable algorithms to solve global optimization problem are listed and discussed briefly.

Kuhn-Tucker conditions

An analytical foundation is an essential component of the numerical techniques especially to understand the conditions that the optimal solutions will have to satisfy and what is the stopping criteria.

In mathematics, the solution is obtained by satisfying the **necessary** and **sufficient** conditions related to the class of the solved problems [Venkataraman, 2001], [Adeli, 1994], [Fox, 1971], and [Minoux, 1983]. The necessary conditions are those relations that a candidate point for the optimum solution must satisfy. If it does, then, and this is important, it **may be** an optimal solution. To qualify a design vector X as an optimum, it must satisfy additional conditions called the *sufficient* conditions. Therefore, an optimum solution must satisfy both necessary and sufficient conditions [Venkataraman, 2001], [Adeli, 1994], [Fox, 1971], and [Minoux, 1983].

Traditionally, there is a bottom-up presentation of material for non-linear optimization. Unconstrained problems are discussed first followed by the constrained ones. For constrained problems the case of equality constrained problems is discussed first.

Unconstrained problems

Let $X^* = [x_1^*, \ldots, x_n^*]^T$ be a minimum solution of the function $f(X)$ defined in the interval $[X^{Low},\ X^{Up}]$.

For clarification, the discussion of optimality conditions is illustrated using two-variables problems because the solution and the ideas can also be expressed graphically as illustrated in Figure 2.6.

First-order condition (FOC): necessary condition

Using the first-order Taylor series expansion (see Annex A) to approximate the value of the function $f(X)$ at the point $X^* + \Delta X$ we obtain:

$$f(X^* + \Delta X) \cong f(X^*) + \frac{df}{dX} \mid_{X^*} (\Delta X) \quad (2.3)$$

In other terms, for infinitesimal displacement $dX = [dx_1, \ldots, dx_n]$ about the optimum solution X^*, the function $f(X)$ can be approximated by its plane tangent at the solution X^*. As shown in Figure 2.6(a), moving to any point $(X^* + dX),\ dX = [dx_1, \ldots, dx_n] \neq [0, \ldots, 0]$ in the plane from the optimum $[x_1^*, \ldots, x_n^*]$ will not change the value of the function $f(X)$, that involves:

$$\nabla f(X^*) \Delta X = 0 \quad (2.4)$$

Therefore, the gradient of $f(X)$ at the optimum solution must be zero. That is:

$$\nabla f(X^*) = 0 \quad (2.5)$$

Equation (2.5) expresses the first-order conditions (FOC) for an unconstrained optimization problem. FOC is a necessary and not a sufficient condition. Indeed, FOC is used to identify solutions which can be candidates for an optimum solution, but FOC can not ensure the optimality.

For example, let consider the convex function $f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^2 - x_1 x_2$ defined over the space search $[0, 4]^2$. The point $x_1^* = 2$, $x_2^* = 2$ gives the optimum $f(x_1^*, x_2^*) = -2$, where the derivative $f'(x_1^*, x_2^*) = 0$. In contrary, let consider the non-convex function $f(x) = x^3,\ x \in [-3, +3]$ where its derivative is expressed as follow $f'(x) = 2x^2$. Even if $f'(0) = 0$, $x = 0$ is not an optimal solution as shown in 2.6(b).

Then to ensure optimality, additional considerations or sufficient conditions are necessary.

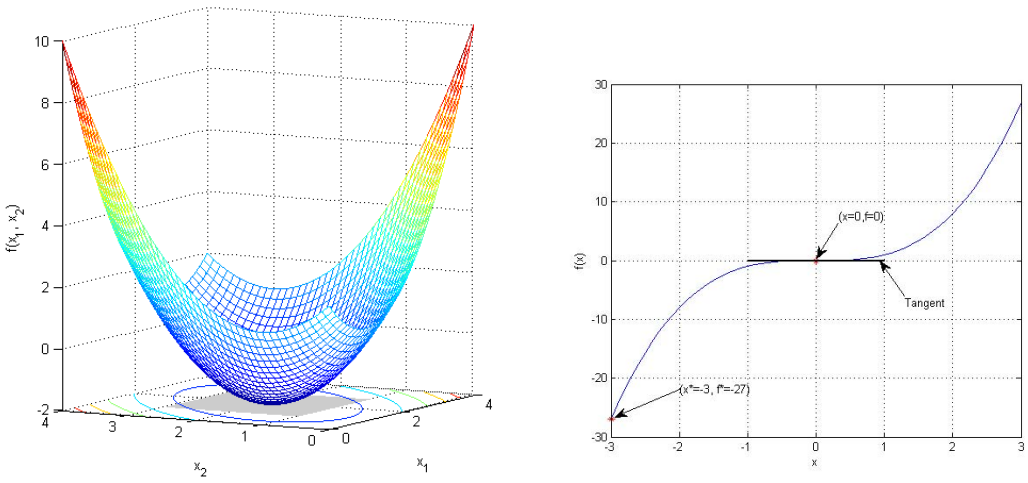
Second-order condition (SOC): sufficient condition

The second-order conditions, as its name indicates, involves second-order derivatives of the functions. The SOC is obtained through the Taylor expansion of the function to a second order at the optimal solution X^*. In the neighbours of X^* the function can be replaced by its quadratic approximation:

$$f(X^* + \Delta X) \cong f(X^*) + \nabla f(X^*)^T \Delta X + \frac{1}{2}\, \Delta X^T H(X^*) \Delta X \quad (2.6)$$

where $H(X^*)$ is the Hessian [2] of f at the solution X^*. As for FOC, ΔX is the displacement from X^* in any direction.

[2] Hessian of a function f is a matrix of the second derivatives of f function of several variables.



(a) Convex function. (b) Non-convex function.

Figure 2.6: FOC conditions for an unconstrained optimization problem.

Since $X^* = [x_1^*, \ldots, x_n^*]$ is the lowest point of the convex surface representing f, if any or all values of $[x_1, \ldots, x_n]$ change from the optimal values $[x_1^*, \ldots, x_n^*]$ in any direction, the value of the function will certainly increase. Then, it is clear that the optimal solution must be a point that satisfies:

$$\Delta f = f(X^* + \Delta X) - f(X^*) = \nabla f(X^*)^T \Delta X + \frac{1}{2!} \Delta X^T H(X^*) \Delta X > 0, \text{ for all } \Delta X \quad (2.7)$$

where Δf is the change in the function value.

Using FOC at the solution: $\nabla f(X^*) = 0$. This leads to:

$$\Delta f = \frac{1}{2} \Delta X^T H(X^*) \Delta X > 0, \text{ for all } \Delta X \quad (2.8)$$

Then at the optimal solution, the Hessian matrix $H(X^*)$ of f must be positive definite.

In the next section, FOC and SOC are generalized for the constrained optimization problems.

Constrained problems

First, the generalization of FOC and SOC for equality constrained problems is discussed followed by inequality constrained problems.

Equality constrained problems

Let consider the optimization problem described in equation (2.1) with only the l equality constraints $h_j(X)$, $j = 1, \ldots l$. If this equality constrained problem can be transformed into an unconstrained optimization one, then following the previous section, the analytical conditions (FOC and SOC) can be established in order to ensure and find the optimal solution.

By using the Lagrangian function (see Annex A), the problem (2.1) without the inequality constraints $g_k(X)$, $k = 1, \ldots m$ is transformed into an unconstrained problem as follow:

$$\begin{cases} \min L(X, \lambda) = f(X) + \sum_{j=1}^{l} \lambda_j h_j(X) \\ X = [x_1, x_2, ..., x_n] \in S \end{cases} \quad (2.9)$$

Thereafter, using the necessary condition FOC for an unconstrained problem, $(n + l)$ equations with $(n + l)$ variables are obtained in order to determine the values for $x_1^*, \ldots, x_n^*$ and $\lambda_1^*, \ldots, \lambda_n^*$:

$$\begin{cases} \dfrac{\partial L(X,\lambda)}{\partial x_i} = \dfrac{\partial f(X)}{\partial x_i} + \displaystyle\sum_{j=1}^{l} \lambda_j \frac{\partial h_j(X)}{\partial x_j} = 0,\ i = 1, \ldots, n \\ \dfrac{\partial L(X,\lambda)}{\partial \lambda_j} = \dfrac{\partial(\lambda_j h_j(X))}{\partial \lambda_j} = h_j(X) = 0,\ j = 1, \ldots, l \\ X^{Low} \leqslant X \leqslant X^{Up}, \qquad X^{Low}, X, X^{Up} \in R^n \end{cases} \tag{2.10}$$

Equations of (2.10) express the necessary FOC for an equality constrained optimization problem. The last equation in the set is the equality constraints. This ensures the solution is feasible. The first n equations can be assembled in a vector form as follow:

$$\nabla f(X^*) + \sum_{j=1}^{l} \lambda_j \nabla h_j(X^*) = 0 \tag{2.11}$$

Equation (2.11) explains that at the optimal solution the vector $-\nabla f$ is a linear combination of the gradients of the equality constraints $\nabla h_j,\ j = 1, \ldots, l$.

As explained before, the solution obtained by FOC is an optimum if and only if the SOC is met. Thus at the solution X^* the Lagrangian function should increase for any change in ΔX:

$$\begin{cases} \Delta L > 0, \text{ for all } \Delta X \\ where: \\ \quad \Delta L = L(X^* + \Delta X) - L(X^*) = \nabla L(X^*)^T \Delta X + \dfrac{1}{2}\, \Delta X^T H(X^*) \Delta X \end{cases} \tag{2.12}$$

where $H(X^*)$ is the Hessian of L at the solution X^*.

For the purpose of clarification, the FOC and SOC are illustrated using the following two-variables problems:

$$\begin{cases} \min f(x_1, x_2) = -x_1 x_2 \\ \text{subject to} \qquad h(x_1, x_2) : x_1^2/4 + x_2^2 - 1 = 0 \\ \qquad\qquad\qquad\ \ 0 \leqslant x_1 \leqslant 3;\ 0 \leqslant x_2 \leqslant 3 \end{cases} \tag{2.13}$$

By applying the FOC to problem (2.13) we have:

$$\nabla f(X^*) = -\lambda \nabla h(X^*) \tag{2.14}$$

where $X^* = [1.4142,\ 0.7071]$ is the optimal solution of this problem.

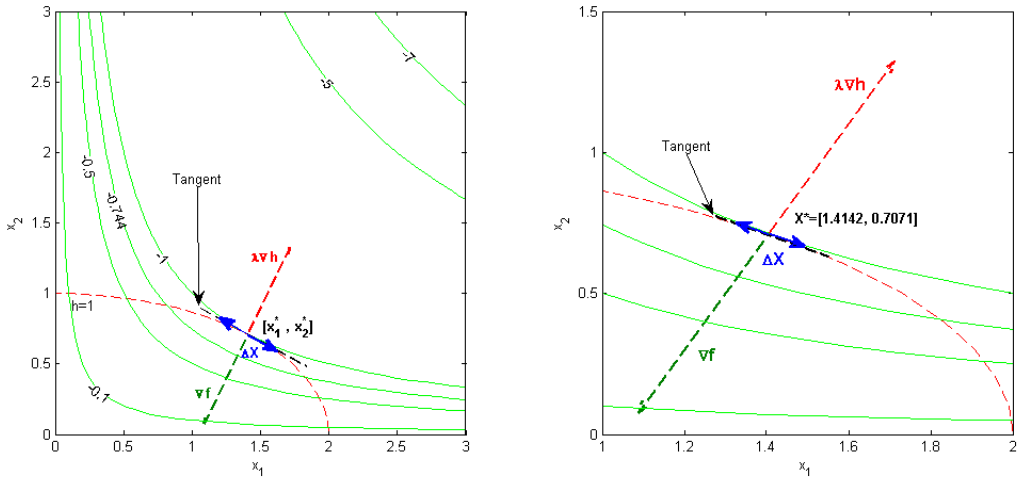
This means at the solution $X^* = [1.4142,\ 0.7071]$ the gradient of the objective function and the gradient of the equality constraint are parallel and oppositely directed (see Figure 2.7).

Furthermore, using the SOC:

$$\lambda \nabla h(X^*) \Delta X = 0$$

which means changes in ΔX are not arbitrary, but they have to satisfy a linearised equality constraint at the solution (see Figure 2.7).

Note that the sufficient conditions are applied in a small neighbourhood of the optimal solution.



(a) Analytical conditions.

(b) Zoom in the neighbourhood of the optimum.

Figure 2.7: Optimization problem with equality constraints.

Inequality constrained problems

Let consider the optimization problem described in equation (2.1) with only the m inequality constraints $g_j(X)$, $j = 1, \dots m$. As for the equality constrained problem, if it can be transformed into an equality constrained one, then it is possible to establish the analytical conditions (FOC and SOC) to ensure and find the optimal solution.

By adding a slack variable z_j for each inequality constraint $g_j(X)$, the inequality constrained problem is transformed to an equality constrained one as follow:

$$\begin{cases} \min f(X) \\ \text{subject to } g_j(X) + z_j^2 = 0,\ j = 1, \dots, m \\ \qquad\qquad X^{Low} \leqslant X \leqslant X^{Up}, \qquad X^{Low},\ X,\ X^{Up} \in R^n \end{cases} \tag{2.15}$$

As presented before and the Lagrangian function, the problem ((2.15)) is transformed into an unconstrained problem as follow:

$$\begin{cases} \min L(X, Z, \beta) = f(X) + \sum_{j=1}^{m} \beta_j [g_j(X) + z_j^2] \\ \qquad\qquad X^{Low} \leqslant X \leqslant X^{Up}, \qquad X^{Low},\ X,\ X^{Up} \in R^n \end{cases} \tag{2.16}$$

where β is the vector of the Lagrangian coefficients associated with this optimization problem.

Then, the FOC are:

$$\begin{cases} \dfrac{\partial L(X, Z, \beta)}{\partial x_i} = \dfrac{\partial f(X)}{\partial x_i} + \sum_{j=1}^{m} \beta_j \dfrac{\partial g_j(X)}{\partial x_i} = 0,\ i = 1, \dots, n \\ \dfrac{\partial L(X, Z, \beta)}{\partial z_j} = 2\beta_j z_j = 0,\ j = 1, \dots, m \\ \dfrac{\partial L(X, Z, \beta)}{\partial \beta_j} = \dfrac{\partial (\beta_j [g_j(X) + z_j^2])}{\partial \beta_j} = g_j(X) + z_j^2 = 0,\ j = 1, \dots, m \end{cases} \tag{2.17}$$

By simple recombination, the $2m$ last equations can be collapsed to m equations and the slack variables z_j will be eliminated from the problem. Then the FOC are restated as:

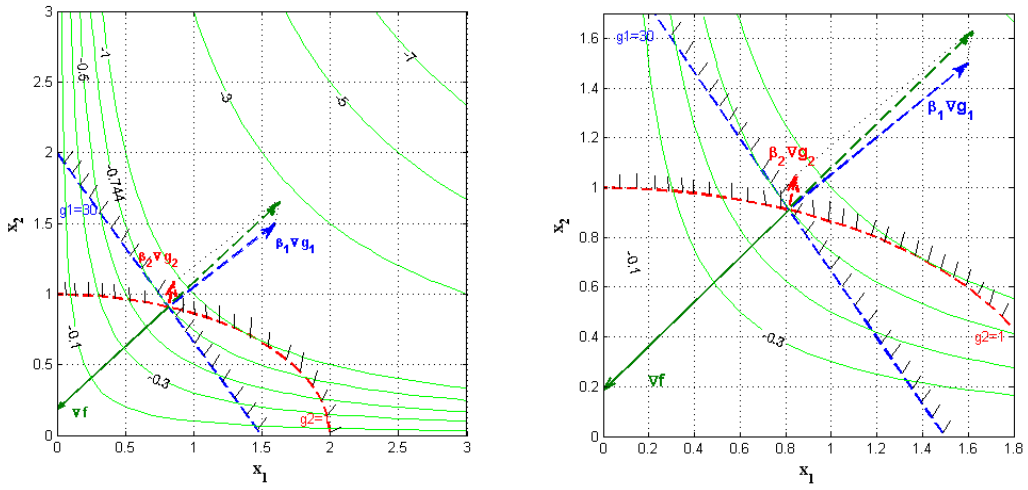
$$\begin{cases} \dfrac{\partial L(X,\beta)}{\partial x_i} = \dfrac{\partial f(X)}{\partial x_i} + \displaystyle\sum_{j=1}^{m} \beta_j \dfrac{\partial g_j(X)}{\partial x_i} = 0,\ i = 1,\ldots,n \\ \beta_j g_j(X) = 0,\ j = 1,\ldots,m \end{cases} \tag{2.18}$$

With the m last equations the nontrivial solution is: either $\beta_j = 0$ and $g_j(X) < 0,\ j = 1,\ldots,m$ (the inequality constraints are passive) or $g_j(X) = 0$ (the inequality constraints are active) and $\beta_j > 0,\ j = 1,\ldots,m$. In other terms, if $\beta_j > 0,\ j = 1,\ldots,m$ then the corresponding constraints are equalities. A simple reasoning was used to show that the sign of the multipliers must be positive to ensure (2.18). The sign of the multipliers is added as a part of the FOC for inequality constrained problem.

Formally, to ensure the optimality, the second derivative of the Lagrangian must be considered.

For the purpose of clarification, the illustartion example presented in the previous section is re-used here as follow:

$$\begin{cases} \min f(x_1,x_2) = -x_1 x_2 \\ \text{subject to} \quad g_1(x_1,x_2) : 20x_1 + 15x_2 - 30 \leq 0 \\ \qquad\qquad\ \ g_2(x_1,x_2) : x_1^2/4 + x_2^2 - 1 \leq 0 \\ \qquad\qquad\ \ 0 \leqslant x_1 \leqslant 3;\ 0 \leqslant x_2 \leqslant 3 \end{cases} \tag{2.19}$$



(a) Analytical conditions.

(b) Zoom in the neighbourhood of the optimum.

Figure 2.8: Optimization problem with inequality constraints.

Figure 2.8 shows that at the solution $X^* = [0.8151,\ 0.9132]$ of problem (2.19) both inequality constraints are active and the associated multipliers β_1, β_2 are positive.

General problem

Let consider the optimization problem described in equation (2.1) with l equality constraints $h_i(X)$, $i = 1,\ldots l$, and m inequality constraints $g_j(X)$, $j = 1,\ldots m$.

This general optimization problem is transformed into an equivalent problem in order to minimize the Lagrangian function as follow:

$$\begin{cases} \min L(X,\lambda,\beta) = f(X) + \sum_{i=1}^{l} \lambda_i h_i(X) + \sum_{j=1}^{m} \beta_i [g_j(X) + z_j^2] \\ X^{Low} \leqslant X \leqslant X^{Up}, \qquad X^{Low},\, X,\, X^{Up} \in R^n \end{cases} \tag{2.20}$$

where $[\lambda, \beta]^T$ is the vector of the Lagrangian coefficients associated with this optimization problem.

By considering the Lagrangian of an unconstrained optimization problem, the FOC have $n + l + m$ equations with respect of $n + l + m$ variables as follow:

$$\begin{cases} \dfrac{L(X,\lambda,\beta)}{\partial x_k} = \dfrac{f(X)}{\partial x_k} + \sum_{i=1}^{l} \lambda_i \dfrac{h_i(X)}{\partial x_k} \\ \qquad\qquad + \sum_{j=1}^{m} \beta_i \dfrac{g_j(X)}{\partial x_k} = 0,\ k = 1, \ldots, n \\ \dfrac{L(X,\lambda,\beta)}{\partial \lambda_i} = h_i(X) = 0,\ i = 1, \ldots, l \\ \dfrac{L(X,\lambda,\beta)}{\partial \beta_j \partial z_j} = \beta_j g_j(X) = 0,\ j = 1, \ldots, m \end{cases} \tag{2.21}$$

As explained for a general optimization problem, including both equality and inequality constraints, the necessary and sufficient conditions are as follow:

A candidate solution X^* with corresponding multipliers λ^* and β^* such:

$$\nabla L(X^*, \lambda^*, \beta^*) = 0 \tag{2.22}$$

$$\Delta X^T H(X^*) \Delta X > 0 \tag{2.23}$$

$$\nabla h(X^*) \Delta X = 0 \tag{2.24}$$

$$\nabla g(X^*) \Delta X < 0 \tag{2.25}$$

is a local minimum for problem (2.1).

Note that, if problem (2.1) is a convex optimization problem, then the local optimum is also the global one. Otherwise, there is no proof that the obtained solution is the global optimum.

To ensure that the Hessian matrix be positive definite, three ways [Venkataraman, 2001] are possible:

1. for all $\Delta X,\ \Delta X^T H(X^*) \Delta X > 0$,
2. the eigenvalues of $H(X^*)$ are all positive,
3. the determinants of all lower orders of $H(X^*)$ that include the main diagonal are all positive.

According to [Adeli, 1994], [Fox, 1971] and [Minoux, 1983] only (1) and (2) can be practically applied. Indeed, as it will be shown in the next section, numerical techniques, like SQP, use an approximation calculation of the Hessian.

By using the mathematical concept presented here, all mathematical techniques of optimization provide formal means to look for the optimal solution. However, with most engineering design problems some information like the derivatives are not available, difficult to be extracted, or just inaccessible. Then stochastic optimization methods, which the most do not need any information about the objective function, have an important role in the optimization domain [Spall, 2003]. For all these reasons, stochastic optimization has been studied and applied in this thesis. In the following, some elements of stochastic global optimization methods are listed.

Elements of stochastic global optimization methods

In this part, the complexity of global optimization problems is discussed especially the necessary techniques that should be used by stochastic global optimization methods in order to overcome this complexity. The considered methods here are those containing probabilistic technique and exploring the research space S by evaluating the objective function f for points sampled in S.

When solving global optimization problem the result is dependent on the complexity of the problem. According to [Torn *et al.*, 1999] the complexity is dependent on the following features of the problem:

- the size of the region of attraction of the global minimum,
- the affordable number of function evaluations N_f,
- the type of the global optimum (embedded or isolated),
- the number of local minima.

In the following, how these features affect the complexity for obtaining an optimal solution is discussed briefly.

Size of the region of attraction

When sampling in the search space, if the region of attraction (defined below) of a local optimum is large then it is easy to detect it.

Definition 2.1.6 *(Attraction region [Torn* et al.*, 1999])*
The region of attraction $A(X^*) \in S$ of a local optimum X^* is defined as the largest region containing X^* such that, at each time starting descending local minimization, with an infinitely small step, from any point in $A(X^*)$ then the local optimum X^* will be found. The region of attraction of a minimum f^* is the union of the regions of attraction of all solution X for which $f(X) = f^*$.

By sampling at a random in the search space S, the probability that a point falls in the region of attraction $A(f^*)$ of the global optimum f^* is p^*. Then:

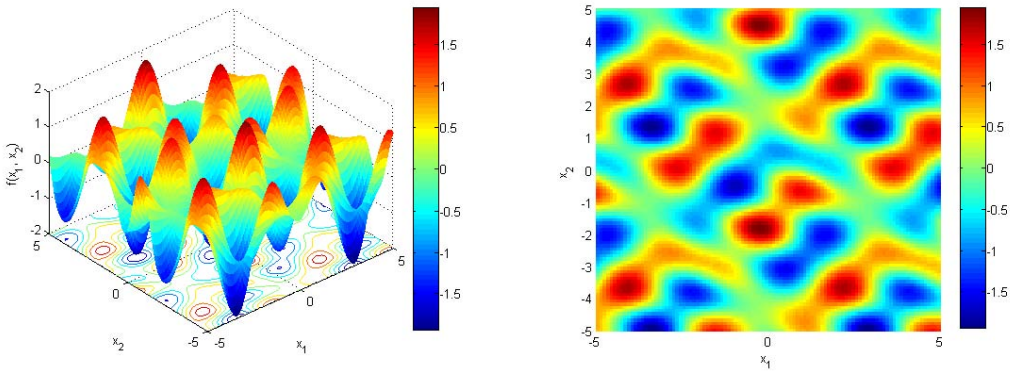
$$(1 - p^*)^N \tag{2.26}$$

is the probability that the region of attraction of f^* is missed when sampling with N points at random in S.

Figure 2.9(b) represents an optimization problem with several local minima, where each one have an attraction region coloured blue. Indeed, considering any point from the dark blue region as an initial point for a local optimization algorithm, the local solution located at the bottom of this region will be found at each time.

Type of global optimum

An other feature which has an influence is if the global optimum is embedded or not. For [Torn *et al.*, 1999] an embedded minimum f^* means that there is several local minima in its neighbourhood. By sampling at a random near these local minima, one point may fall in the region of attraction of f^*. Thus an optimization problem with an embedded minimum is easier to solve than the one having an isolating optimum.



(a) Embedded global optimum. (b) Attraction regions.

Figure 2.9: Complexity of a global optimization problem.

Number of local minima

To find the number of local minima of an optimization problem, the suitable manner is the multi-start technique presented below, i.e. starting a local minimization algorithm from N randomly sampled initial points in S. If several different solutions are obtained then many local optimum will be expected, otherwise few.

A local search will become increasingly ineffective when the number of local minima is increasing because the size of attraction region is a decreasing function of this number.

Furthermore search space should be as small as possible. For example, if a function f have the same global optimum in both search spaces $S_1 = [0,1]^n$ and $S_2 = [0,2]^n$ then choosing and sampling points in S_2 instead of S_1 will increase geometrically the search space with a factor of 2^n and $(1-p^*)$ will reach $(1-\frac{p^*}{2^n})$. Where n is the dimension of the optimization problem (number of design variables).

To overcome this drawback, this study proposes to use the multi-level optimization approach which is discussed in the next chapter of this state of the art. This method consists breakdown a complex problem into smaller ones easier to be solved. Then, for each small problem, the number of local minima and its dimension n is expected to decrease and the size of the attraction region of the sub-global optimum is expected to increase.

Number of function evaluations

One way to establish a stopping criteria is to use the necessary condition presented before. An alternative way is to limit the number of functions evaluations. For this purpose, [Torn *et al.*, 1999] proposed some material to calculate the suitable number of function evaluations in order to ensure with a certain probability the global minimum.

If sampling a point at random in the search space S, the probability that this point falls in $A(X^*)$ is at least p. Sampling k points at random the probability that no point falls in $A(X^*)$ is then at most $(1-p)^k$ and the probability to obtain at least one point in $A(X^*)$ is at least:

$$q = 1 - (1-p)^k \tag{2.27}$$

then $k = log(1-q)/log(1-p)$ is obtained.

For example, $q = 0.95$ and $p = 0.001$ means that 2995 points are needed to be sampled at random in S to ensure with 95% the global optimum.

Strategies in choosing points

All methods evaluate the objective function $f(X)$ in some points $X^1, \ldots, X^N \in S$, and they differ only in their choice of these points. Because there is no information about where in S to find the global minimum, one strategy that must be used is to spread out some of the points to cover S like uniform sampling.

In the second part of this study, chapter 4 is devoted to study the influence of the initial points sampling on the solutions obtained by some optimization algorithms like genetic algorithms (GA).

Given a point X, it is normally possible to find a nearby point with a smaller function value. Many global optimization method will use local optimization search, at least to improve the convergence [Torn *et al.*, 1999]. Indeed, in 5 the genetic algorithms are combined with the local method, sequential quadratic programming (SQP), in the aim to find the global solution. Where GA uses a uniform sampling in order to generate the initial population.

Stopping conditions

Stopping criteria indicate when an algorithm is close enough to the solution that it can be stopped. Then in any computer algorithm there must be some stopping conditions which after some finite number of function evaluations stops the computation. The condition should of course relate to the quality of the solution achieved.

This is a very crucial point in global optimization because they will always remain an unexplored region with any finite number of iterations (or number of function evaluations). Without some additional information or assumptions about the problem there is no way to decide on the quality of the solution after a given number of iterations.

From this we may conclude that the global optimization problem in general is unsolvable and that we must be prepared to accept an approximation of the global optimum as a solution.

In the following, the most important optimization methods are listed. For the purpose of clarification, only the large used ones are presented briefly.

2.1.3 Solving mono-objective optimization problems

Local optimization methods

In general, the classical optimization techniques have difficulties in dealing with global optimization problems. One of the main reasons of their failure is that they can easily be trapped in local minima. Moreover, these techniques cannot generate or even use the global information needed to find the global minimum for a function with multiple local minima.

However, solutions obtained by this kind of algorithms are very accurate. They have a fast convergence comparing to global methods. Furthermore, local search algorithms can be used to guide global optimization methods in the vicinity of local minima and speed up their convergence.

For this reason, "Sequential Quadratic Programming" algorithm is selected to be combined with genetic algorithms, ant colony optimization, and particle swarm optimization which are studied and improved in this research.

Sequential Quadratic Programming

Sequential quadratic programming method (SQP) represents the state of the art in nonlinear programming methods. It is an iterative method, which solves a quadratic problem (QP) at each iteration [Venkataraman, 2001].

In the vicinity of the current point X_n, SQP algorithm replaces the objective function $f(X)$ with its quadratic approximation:

$$QP\left(\Delta X\right) = \frac{1}{2}\left(\Delta X\right)^T L\left(X_n, \lambda_n\right) \Delta X + \nabla f^T\left(X_n\right) \Delta X \tag{2.28}$$

where $L\left(X_n, \lambda_n\right)$ is the Lagrange function and λ_n are the Lagrange multipliers. Then it replaces the l equality and m inequality constraints, respectively $h_k\left(X_n\right)$ and $g_j\left(X_n\right)$, by their linear approximations. The step ΔX is calculated by solving the following quadratic problem:

$$\begin{cases} \min & QP\left(\Delta X\right) \\ \text{subject to} & \nabla h_k^T\left(X_n\right) \triangle X + h_k\left(X_n\right) = 0,\ k = 1, 2, \ldots, l \\ & \nabla g_j^T\left(X_n\right) \triangle X + g_j\left(X_n\right) = 0,\ j = 1, 2, \ldots, m \end{cases} \tag{2.29}$$

According to [Venkataraman, 2001], the optimal solution (X^*, λ^*) must satisfy the second-order condition of optimality. If the starting point X_0 is sufficiently close to X^*, and λ_n stay sufficiently close to λ^*, then $X_{n+1} = X_n + \Delta X$ converges to X^*.

As explained before, SQP must be combined with multi-start technique in order to converge towards the global optimum. An other way to avoid local optima, consist on the combination of SQP with a global optimization algorithm. The new algorithm is called hybrid method. In the following the most known global optimization algorithms are presented.

Global optimization methods

Using deterministic optimization methods like SQP, perfect information about the objective and constraint functions and its derivatives are needed in order to determine the search direction in deterministic manner at each step of the algorithm [Spall, 2003]. However, in many practical optimization problems, such information is not available. Then, deterministic algorithms are inappropriate in the case of noisy functions.

Furthermore, this research aims to solve:

- several objective functions simultaneously;
- and mixed-integer optimization problems dealing with continuous, discrete, integer, and binary design variables.

This is the main drawback of most of the classical deterministic approaches. For all these reasons, this thesis focuses in the stochastic search and optimization, like metaheuristics especially genetic algorithms, ant colony optimization, and particle swarm optimization.

Metaheuristics

The interaction between computer science and optimization has yielded new practical solvers for global optimization problems, called metaheuristics. The structures of metaheuristics are mainly based on simulating nature and artificial intelligence tools.

Metaheuristics mainly invoke exploration and exploitation search procedures in order to diversify the search all over the search space and intensify the search in some promising areas. In particular, these algorithms manage very well combinatorial and mixed problems. In general, metaheuristics are stochastic algorithms.

Genetic Algorithms (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Tabu Search (TS), Simulated Annealing (SA) and Scatter Search are designed to be effective to find good solutions to non-smooth optimization problems [3]. These algorithms can be applied to smooth non-linear problems to seek a globally optimal solution. Furthermore, they are often more effective to find better solutions than a local search algorithm alone.

However, metaheuristics are computationally time consuming due to their slow convergence. One of the main reasons for their slow convergence is that they may fail to detect promising search directions especially in the vicinity of local minima due to their random constructions.

For the purpose of clarification, only the algorithms that arouse our interest are presented. Which are:

- Genetic Algorithms (GA),
- Ant Colony Optimization (ACO),
- Particle Swarm Optimization (PSO).

GA, PSO and ACO are considered as global optimization methods. When GA mimics the natural biological evolution, ACO and PSO draw inspiration from the interactive behaviour of social species [Dorigo *et al.*, 1996, Kennedy & Eberhat, 1995].

GA has a relatively old history since the first work of its author John Holland backs to 1962. ACO was proposed by Dorigo et al. in 1990 [Dorigo *et al.*, 1996]. It is inspired by the collaborative behaviour of the ants [Dorigo *et al.*, 1996]. PSO is the most recent one of both. It was developed by Eberhart and Kennedy in 1995 [Kennedy & Eberhat, 1995], [Clerc & Kennedy, 2002]. PSO is inspired by social behaviour of bird flocking.

The three algorithms are population based stochastic optimization techniques. PSO and ACO have many similarities with GA. Indeed, all are initialized with a randomly generated population and updating solution following the fitness values at each iteration. However, unlike GA, ACO and PSO have no evolution operators like crossover and mutation. But both have memory, which is important to the algorithms , [Kennedy & Eberhat, 1995] [Dorigo *et al.*, 1996], [Dorigo & Gambardelle, 1997], [Dorigo & Blumb, 2005], [Trelea, 2003b], [Trelea, 2003a], [Zhang *et al.*, 2005]. Compared with GA where individuals are in rivalry, the efficiency of PSO and ACO is in the cooperative behaviour of agents (particles or ants).

Below, the three algorithms GA, ACO, and PSO are presented briefly.

Genetic Algorithms

Since at least the 1950s, several papers showed that there has been interest in the application of biological evolutionary concepts to mathematical search and optimization domain [Spall, 2003]. Evolutionary Strategies (ES), Evolutionary Programming (EP) and Genetic Algorithms (GA), cornerstones of the evolutionary algorithms (EA), are developed independently of each other in the 1960s and 1970s [Spall, 2003]. Rechenberg (1965) introduced ES for optimization of continuous variables. EP, described by Fogel et al. (1966), treats the candidate solutions in the population as symbols via finite-state machines. Genetic algorithms are the most popular of the evolutionary algorithms [Michalewicz, 1994], [Venkataraman, 2001], [Spall, 2003], [Dréo *et al.*, 2006].

There are two issues in the genetic algorithms: population diversity and selective pressure. These factors are strongly related, so that an increase in the selective pressure decreases the diversity of the population, and vice versa.

As shown in Figure 2.10, a genetic algorithm implements a loop iteration that incorporates the application of three reproduction operators:

[3] The most difficult type of optimization problem to solve is a non-smooth problem (NSP) which are non-convex.

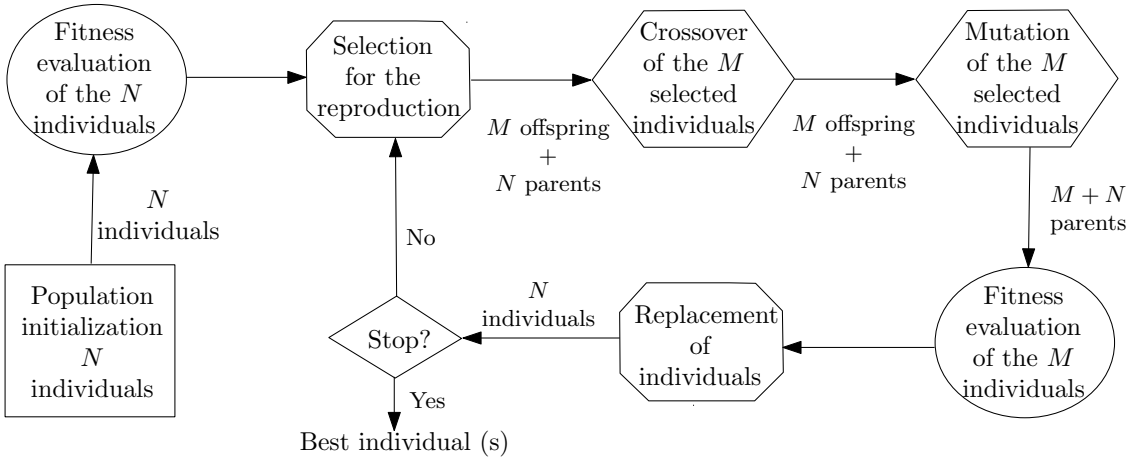


Figure 2.10: Genetic algorithm [Dréo *et al.*, 2006].

1. selection of parents among a population of N individuals for reproduction;
2. crossover and mutation of the M selected individuals (parent) in order to generate M offspring;
3. fitness evaluation of the offspring;
4. selection of the N survival individuals among the M offspring and N parents, or only among the M offspring according to the choice made by the user, in order to build the population for the next generation.

Particle swarm optimzation

The PSO algorithm is based on two rules [Kennedy & Eberhat, 1995], [Clerc, 2003a], [Trelea, 2003b], [Trelea, 2003a], and [Zhang *et al.*, 2005]:

1. each particle has a memory that enables him to memorize its best position found in the past, and it tends to be attracted by this point;
2. each particle is informed about the best position find in its vicinity, and it tends to go towards this point.

In other words, starting from some information, particles must be able to choose their next movement, i.e., to calculate a new velocity which is an updating operator for their positions.

Therefore, a particle i combines, linearly, three components:

- its current velocity: v_i,
- its best position: p_i,
- the best position of its neighbours: G_i,

by using three parameters which balance three tendencies:

- c_1: tendency to follow its own way,
- c_2: tendency to reconsider its steps (preserving),
- c_3: tendency to follow the best neighbor,

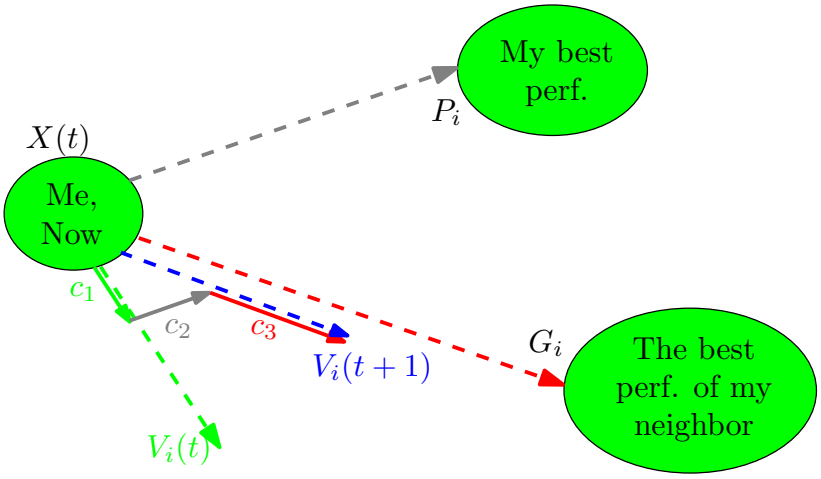


Figure 2.11: Particle swarm optimization [Clerc, 2003b].

To summarize, in PSO each single solution is a particle (an agent) that has fitness value, and a velocity to direct his flying. As shown in Figure 2.11 the PSO algorithm consists on the calculation of the velocity:

$$v_i(k+1) = c_{1_i} v_i(k) + c_{2_i} \cdot r \cdot (p_i - x_i(k)) + c_{3_i} \cdot r \cdot (G_i - x_i(k)) \tag{2.30}$$

and the position:

$$x_i(k+1) = x_i(k) + v_i(k+1) \tag{2.31}$$

where i is the particle index, v, and x are the velocity and the current position of the i^{th} particle respectively. p is the best position found by the i^{th} particle, and G is the best position found by the whole of the swarm. $r \in [0, 1]$ is a random number. c_2, c_3 are the learning factors. c_1 is the inertia of the i^{th} particle.

Ant colony optimization

Contrary to a natural ant, an artificial ant is not completely blind and can see a little further that its direct entourage. Furthermore, an artificial ant has a certain memory enabling it to adopt the current solution.

An artificial ant put down a quantity of pheromone that depends on the quality of the solution [Colorni *et al.*, 1992], [Dréo, 2004], [Dutot, 2005], and
[Dorigo *et al.*, 1996, Dorigo & Gambardelle, 1997, Dorigo *et al.*, 1999, Dorigo & Blumb, 2005]
. To summarize, at each point requiring a choice, a decision table produces a probability distribution on which the movements of the ants are based. Thus, by considering that an ant is on the point i, the probability p_{ij} which it moves towards the point j will be given as:

$$p_{ij} = \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{k \in V(i)}[\tau_{ik}(t)]^{\alpha}[\eta_{ik}(t)]^{\beta}} \tag{2.32}$$

where at iteration t, τ_{ij} is the concentration of pheromone associated with the chosen solution, and η_{ij} is the attraction of this solution. $V(i)$ represents the vicinity of the point i. α and β are two parameters giving a relative importance to the pheromone and the attraction value of each ant.

The attraction η_{ij} represents how much the point j improves the objective function compared to the point i. It can be calculated for example as:

$$\eta_{ij} = (f_i(X) - f_j(X)) + |\max_{k \in V(i)}(\eta_{ik})| \tag{2.33}$$

where $f_i(X)$, $f_j(X)$ are the objective function values of both points i and j respectively. The term $|\max_{k \in V(i)}(\eta_{ik})|$ is added to ensure positive values for the probabilities p_{ij}.

The concentration of pheromone is updating at each iteration t according to the following:

$$\tau_{ij}(k+1) = \rho\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t+1) \tag{2.34}$$

where, $\rho \in [0\,1]$ is the pheromone persistence such that $(1-\rho)$ represents the evaporation of trail between iteration $t+1$ and t. $\Delta\tau_{ij}$ is the pheromone addition when the k^{th} ant moves from the decision point i to the solution j.

Figure 2.12, illustrates the main steps of the ACO algorithm.

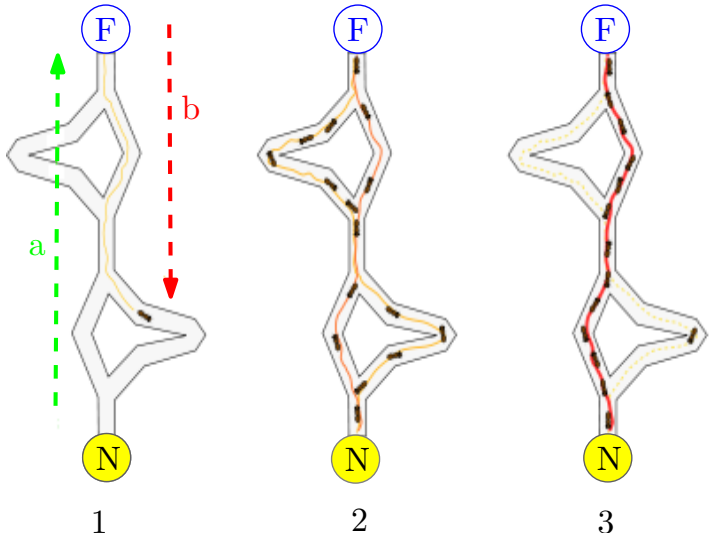


Figure 2.12: Ant colony optimization.

1. The first ant finds the food, puts down a quantity of pheromone and come back to the colony.
2. In the beginning the other ants take randomly one of the four paths.
3. The pheromone of the long ways evaporates and there is less chance to be chosen by the ants. The ants finish by chosen the shortest path which have the greater quantity of pheromone.

General classification

To summarize, the more important local and global optimization methods are listed in tables 2.2 and 2.3.

Furthermore, the strengths and weaknesses of each method are given. Some approaches are not introduced in this study. Then for the purpose of clarification, they are defined brievely in the following:

- Bayesian method is based on the probability theory. Thus, given the probability distribution, the algorithm can leand to the optimal solution [Mockus *et al.*, 1996].
- Interval arithmetic based methods are developed [Hansen & William Walster, 2003] as an approach to putting bounds on rounding errors in mathematical computation and thus developing numerical methods that yield reliable results.

Table 2.2: Local optimization methods classification.

Family type	Methods	Strengths	Weaknesses
Deterministic methods using derivatives	- Sequential Quadratic Programming (SQP) - Conjugate Gradient Method - Projected Gradient Method	- Speed convergence. - Accuracy. - Less parameters to be tuned.	Inappropriate in the case of: - noisy functions, - non-convex functions, - functions with several minima. Initial point dependence.
Direct methods	Nelder-Mead Method	- Speed convergence. - Not need to derivatives. - Easy implementation.	- Not dealing with constraints. - Can be trapped in ocal minima.
Stochastic methods	- Monte Carlo Method	- Not need to derivatives. - Easy implementation.	- No proof of convergence. - Not robust.

Table 2.3: Global optimization methods classification.

Family type	Methods	Strengths	Weaknesses
Adaptive partition enumerative and search strategies	- Branch-and-bound algorithms (BB) - Bayesian approaches - Interval arithmetic based methods	- Easy implementation. - Possibility to find exactly the global optimum	- Inappropriate to handle large scale optimization problems.
Adaptive stochastic search algorithms (Metaheuristics)	- Evolution and Genetic algorithms (GA) - Ant colony optimization (ACO) - Particle Swarm Optimization (PSO) - Simulated annealing (SA) - Tabu Search (TS)	Can be used in the case of: - noisy functions, - non-convex functions, - functions with several minima. Not need to derivatives. Easy implementation. Exploitation and exploration of the search space	- Slow convergence. - Many parameters to be tuned. - Less accurate.
Deterministic global methods	- Generalized descent methods	- Not need to derivatives. - Easy implementation.	- May introduce new local minima due to the use of the penalisation. - No stability.

2.1.4 Outcome

According to the literature, the identification of the characterization of the optimization problem is a part of the solution. Indeed, practical optimization problems, where no information about the objective function and its derivative are available, can not be solved using deterministic methods like SQP. Furthermore, optimization problem with several local optima can not be handled with SQP which is a local optimization method.

All of these show the interest of the use of the stochastic global optimization methods GA, ACO, and PSO in order to solve electrical optimization problems as presented in part II. However, these algorithms are less accurate and time consuming compared to local search methods like SQP. Furthermore, the obtained solutions depend on the tuning of the parameters of the metaheuristic itself. Figure 2.13 shows the trade-off that metaheuristics must take into account in order to ensure a good approximation of the global optimum.

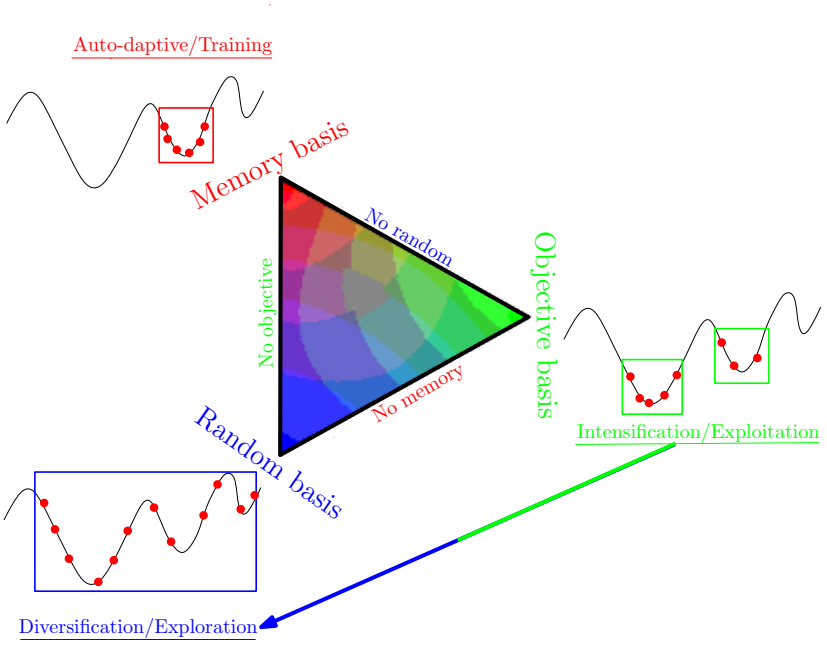


Figure 2.13: Main steps of metaheuristics.

Indeed, metaheuristics must make trade-off between:

- diversification, the process to large exploration of the search space;
- intensification, the process to exploit the information around of a solution;
- training, the process to use a memory in the aim to intensify the search around a specific solution.

A good alternative to overcome these disadvantages is the use of hybrid methods. Indeed, combining metaheuristics with local search methods is a practical remedy to overcome:

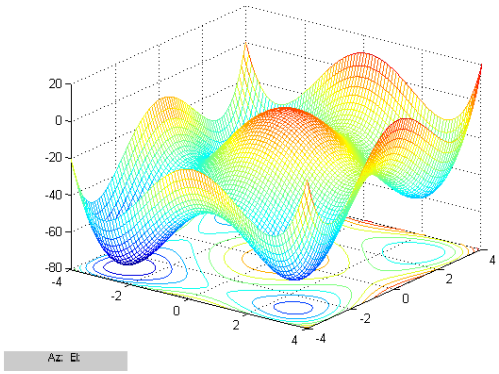
1. the trap of local optima concerning local search methods;
2. and the drawbacks of slow convergence and random constructions of metaheuristics.

As shown in the part II of this thesis, local search strategies can be inlaid inside metaheuristics (like for ACO and SQP), or linked sequentially (like for GA, PSO and SQP) in order to guide the metaheuristics (GA, ACO, and PSO) especially in the vicinity of local minima, and overcome their slow convergence especially in the final stage of the search.

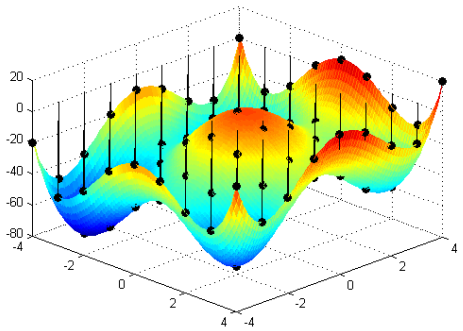
2.2 Mixed-integer optimization

Mixed-integer programming (MIP) problem is one where some of the decision variables are constrained to have only integer values (i.e. whole numbers such as -2, -1, 0, 1, 2,... etc.) at the optimal solution. A special case is a decision variable that are integers with either 0 or 1 values at the solution. This kind of variables, called $0-1$ (zero-one) or binary integer variables, are used to model yes/no decisions.

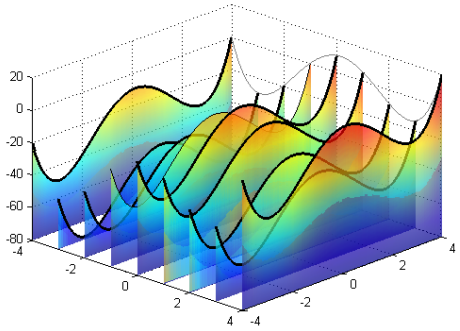
With many integer optimization problems, higher-level constraints can be applied to the integer variables. The most common constraint is the one which assumes that the variables can have only a finite number of possible values (say 1 through 10), and specifies that the variables must be all different at the optimal solution. This kind of optimization is called **Constraint programming** [Van Hoeve, 2006]. A classic example of a constraint programming problem is the travelling salesman problem: a salesman plans to visit N cities and must drive varying distances between them. In what order should he visit the cities to minimize the total distance travelled, while visiting each city exactly once?



(a) Continuous optimization problem.



(b) Discrete optimization problem.



(c) Mixed-integer optimization problem.

Figure 2.14: Optimization problem classification according to the design variables type.

The optimization problem which contains both the discrete and continuous variables is often called mixed-integer non-linear optimization problems (MINLP) when the objective or the constraints functions are non-linear; and called linear optimization problems (MILP) when the functions are all linear. Much research work has been reported on this class of problems [Sandgren, 1990], [HAJELA & C.-J., 1990], [Cao & Wu, 1997], [Kim *et al.*, 1998], [Lin *et al.*, 1999], [Alighanbari *et al.*, 2005], and [De Wit, 2005].

Electrical engineering design optimization problems encounter integer, discrete, and zero-one variables as well as continuous variables. Discrete variables are used in many ways such

as the choice between different design options. For example, the number of primary turn, and the number of the teeth for a gear must be chosen as integers, whether or not selecting a machine is a zero-one variable, and the type of material, type of lamination that will be used must be discrete variables.

Note that, this study focuses on the MIP optimization problems that are non-linear, then on the MINLP optimization problems.

Mixed-integer optimization is considered as a hard optimization and difficult to solve, because the integer variables make the optimization problem non-convex. Indeed to solve such a problem, many combinations of the integer variables values must be tested, and each combination requires the solution of a linear or non-linear optimization problem. The number of combinations can rise exponentially with the size of the problem. Figure 2.14 illustrates three types of optimization problems: continuous, integer, and mixed-integer optimization problems.

2.2.1 Optimization problem formulation

As a continuous optimization problem, the main aim of a mixed-integer optimization problem is to find the design variables values which minimize the objective function, with respect to some design constraints. Furthermore, they have basically the same mathematical formulation except that the design variables may take on any form of continuous, zero-one, integer and discrete variables:

$$\begin{cases} \min f(X) & \\ \textit{subject to} & h_i(X) = 0,\ i = 1, \ldots, l \\ & g_j(X) \leq 0,\ j = 1, \ldots, m \\ & X = \begin{pmatrix} X^c \\ X^z \\ X^i \\ X^d \end{pmatrix} \\ & \quad = [x_1^c, \ldots, x_{n_z}^z, x_1^z, \ldots, x_{n_z}^z, x_1^i, \ldots, x_{n_i}^i, x_1^d, \ldots, x_{n_d}^d]^T \\ & X^{cL} \leq X^c \leq X^{cU} \\ & X^z \in [0,\ 1]^{n_z} \\ & X^i \in Z^{n_i} \\ & X^d \in D^{n_d}, \end{cases} \tag{2.35}$$

Where R^{n_c}, $[0,\ 1]^{n_z}$, Z^{n_i} and D^{n_d} denote the feasible subsets of the continuous, zero-one, integer, and discrete variables, respectively. R is the real domain, Z is the integer space which its elements are ordered, and D is the discrete set in which the points are isolated from each other in a certain sense and not ordered. n_c, n_z,n_i and n_d are the number of continuous, zero-one, integer and discrete variables, respectively. Then, the total number of the design variables is $n = n_c + n_z + n_i + n_d$.

2.2.2 Solving mixed-integer optimization problem

Since MIP problems are non-convex, they must be solved by some kind of potentially exhaustive search. Branch and Bound method [Tran *et al.*, 2007c] is the classic method for solving MIP problems.

This method begins by finding the optimal solution to the **relaxed** problem by using standard linear or non-linear optimization methods. Where the relaxation consists on the transformation of integer variables into continuous ones. If in the obtained solution, the relaxed variables have integer values, then no further work is required. However, if one

or more integer variables have continuous values, the algorithm chooses one such variable and branches creating two new sub-problems where this variable is strongly constrained. Thereafter, these sub-problems are solved and the process is repeated until a solution that satisfies all of the integer constraints is found.

[Sandgren, 1990] has proposed non-linear branch and bound algorithms. In the beginning, these methods obtain a solution by dropping the discrete conditions. Cha and Mayne proposed an algorithm which combined a sequential quadratic programming (SQP) method or an other local search strategy [Leyffer, 2001].

Alternative methods, such as the evolutionary algorithms, can be used to solve the MINLP optimization problems. Indeed, for example genetic algorithms (GA) randomly generate candidate solutions that satisfy the integer constraints. Such initial solutions are usually far from the optimum. But through preserving operators like **integer mutation and crossover**, GA transforms the initial population into new candidate solutions which satisfy the integer bounded constraints. Thereafter, the reproduction operators ensure that the offspring solutions have the better objective values. This process is repeated until a sufficiently good solution is found. Chapter 7 is devoted to show how genetic algorithms can be modified and improved in order to deal with mixed-integer design variables in the case of mono-objective and multi-objective optimization problems.

2.2.3 Outcome

One aim of this work is devoted to develop a mixed-variable genetic algorithm (MIGA) for electrical design optimization problems. Chapter 7 of the second part, Improvement of algorithms, answer to how genetic algorithms can be modified and improved in order to be able to handle mixed-integer non-linear optimization problems.

2.3 Multi-objective optimization

Most realistic optimization problems, particularly those in design engineering, require the simultaneous optimization of more than one objective functions.

Indeed, multiobjective optimization problems (MOP) can be found in various fields: product and process design, finance, aircraft design, train design, the oil and gas industry, and automobile design.

For example train design requires simultaneous optimization of consumption, efficiency, and weight. The portfolio optimization problem attempts to simultaneously minimize the risk and maximize the fiscal return. Maximizing profit and minimizing the cost of a product, maximizing the performance and minimizing the fuel consumption of a vehicle are also examples of multi-objective optimization problems.

2.3.1 Optimization problem formulation

To be more formal, the multi-objective optimization is the process of the simultaneous optimization of two or more conflicting objectives subject to certain constraints. The mathematical description is as follows:

$$\begin{cases} \min F(X) = \begin{pmatrix} f_1(X) \\ f_2(X) \\ \dots \\ f_k(X) \end{pmatrix} & \\ \textit{subject to} \quad h_i(X) = 0,\ i = 1, \dots, l & \\ \qquad\qquad\ g_j(X) \leq 0,\ j = 1, \dots, m & \\ \qquad\qquad\ X^{Low} \leq X \leq X^{Up} & \end{cases} \tag{2.36}$$

where $k \geq 2$ is the number of objective functions $f_i : R^n \longrightarrow R$. The nonempty set

$$S = \left\{ X \in R^n / \ X^{Low} \leq X \leq X^{Up},\ h_i(X) = 0,\ i = 1, \dots, l,\ g_j(X) \leq 0,\ j = 1, \dots, m \right\}$$

denotes the feasible region constrained by equality and inequality constraints and explicit bounds of variables.

The space in which the objective function vectors belong is called the objective space, and the image of the feasible region S under F, $C = F(S)$, is called the feasible objective region. Then, S is a subset of the decision variables space R^n, and C is a subset of the objective space R^k. Instead of objective function vectors, for short, the elements $Z_i = f_i(X),\ i = 1, \dots, k$ of C are called objective vectors in what follows.

Note that different types of multi-objective optimization problems can be defined. Indeed, when all the objective and the constraint functions that form the feasible region are linear, then the problem is linear and it is called multi-objective optimization linear programming (MOLP). If at least one of the functions (objectives or constraints) is non-linear, the problem is called non-linear multi-objective optimization problem (MONLP).

Furthermore, a well formed multi-objective optimization problem is the one where there is not a single solution that simultaneously minimizes each objective to its fullest.

2.3.2 Pareto dominance and optimality

In mono-objective optimization problems the main focus is on the decision variable space. On the other hand, in multi-objective optimization problems the interest is focused on the objective space because it is on lower dimension than the design variable space.

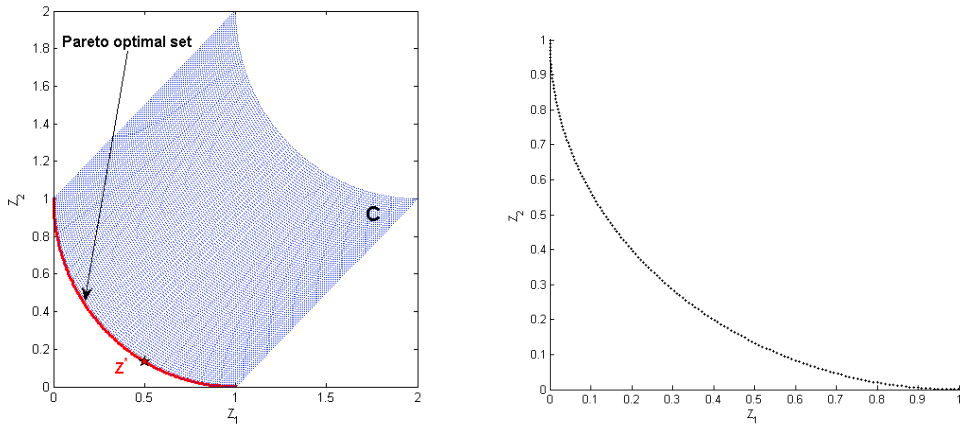
Furthermore, the scalar concept of optimality, discussed before in section 2.1, does not apply directly in the multi-objective setting. A useful replacement is the notion of **Pareto optimality** which needs the use of the objective values $Z_i,\ i = 1, \dots, k$.

Definition 2.3.1 *(Pareto optimality [Kaisa, 2004]) A decision vector $X^* \in S$ is a Pareto optimal if there does not exist another decision vector $X \in S$ such that $f_i(X) \leq f_i(X^*)$ for all $i = 1, \ldots, k$ and $f_j(X) < f_j(X^*)$ for at least one index j.*
An objective vector $Z^* \in C$ is Pareto optimal if there does not exist another objective vector $Z \in C$ such that $Z_i \leq Z_i^*$ for all $i = 1, \ldots, k$ and $Z_j < Z_j^*$ for at least one index j; or equivalently, Z^* is Pareto optimal if the decision vector X^* corresponding to it is Pareto optimal.

In other term, a vector is said to be Pareto optimal for problem ((2.36)) if all other vectors have a higher value for at least one of the objective functions, or else have the same value for all objectives. In fact, this definition introduces the concept of global Pareto optimality.

Figure 2.15, illustrates a feasible objective region $C \subset [0,\ 2]^2$ which is the image of $S \subset [0,\ 1] * [0,\ \frac{\pi}{2}]$ under both functions $f_1(X)$ and $f_2(X)$, as follow:

$$\begin{cases} \min f_1([x_1,\ x_2]) = 1 - cos(x_1) + x_2 \\ \min f_2([x_1,\ x_2]) = 1 - sin(x_1) + x_2 \\ with: \qquad 0 \leq x_1 \leq \dfrac{\pi}{2} \\ \qquad\qquad\quad 0 \leq x_2 \leq 1 \end{cases} \tag{2.37}$$



(a) Objective space. (b) Trade-off space.

Figure 2.15: Objective space and the Pareto optimal vectors Z^* [Collette & Siarry, 2002].

In addition to Pareto optimality, several terms are sometimes used like non-inferiority, efficiency and non-dominance. Pareto optimal points are also known as efficient, non-dominated, or non-inferior points. In this thesis the non-dominance term is widely used.

Before presenting the most-known multi-objective optimization algorithms, local Pareto optimality and the non-dominance relation are introduced first.

According to the definition given before, a point $X^* \in S$ is said to be globally Pareto optimal or a non-dominated point for a MOP if and only if there is no $X \in S$ such that $f_i(X) \leq f_i(X^*)$ for all $i = 1, \ldots, k$, with at least one strict inequality.

In the same way, locally Pareto optimal points, for which the definition is the same as the one given before, except that we restrict an attention to the feasible neighbourhood of X^*: $B(X^*, \delta)$ which is a ball of radius δ around the point X^*. The definition is as follow:

Definition 2.3.2 *(Local Pareto optimality [Kaisa, 2004]) A decision vector $X^* \in S$ is locally Pareto optimal if there exist $\delta > 0$ such that X^* is Pareto optimal in $S \cap B(X^*, \delta)$.*
An objective vector $Z^* \in C$ is locally Pareto optimal if the decision vector corresponding to it is locally Pareto optimal.

Typically, there is an entire curve or surface of Pareto points, whose shape indicates the nature of the trade-off between different objectives. There are usually a lot of (infinite number) Pareto optimal solutions which form a Pareto optimal set or front.

In practical optimization problems, the Pareto set can be non-convex and disconnected, then difficult to be obtained. This thesis focuses on the proposition of a multi-objective optimization algorithm which is able to find the concave parts of the Pareto front, and the disconnected solutions (see chapter 6).

The main difficulty of multi-objective optimization problems is how to choose the suitable optimization method in order to solve them. In the following, we try to give the well-known and important methods for handling such hard problems.

2.3.3 Solving multi-objective optimization problems

As the multi-objective optimization is a branch of multi-criteria decision analysis (MCDA), this thesis follows and presents the classification of multi-objective optimization methods [Hwang & Masud, 1979], [Kaisa, 2004], and [Collette & Siarry, 2002], made according to the participation of the decision maker in the solution process. This classification is presented by [Hwang & Masud, 1979]:

1. A priori methods: where a priori join of a preference information is used. For example the weighting methods with fixed weighting coefficients belong to this kind.

2. A posteriori methods: where a posteriori join of a preference information is used as for the multi-objective evolutionary algorithms.

3. Progressive methods: where a progressive join of a preference is used as for the reference point approach. This kind of methods are not presented in this study due to the hight computational time, as it is difficult to the decision maker to provide consistent answers throughout the decision process. An other drawback is that the obtained solution can be not a Pareto optimal [Hwang & Masud, 1979] and [Collette & Siarry, 2002].

For the purpose of clarification, only some transformation methods (Weighted-sum and ϵ-constraint methods) and the multi-objective evolutionary algorithms are presented.

Weighting methods

The most intuitive approach to solve the multi-objective optimization problem consists on the construction of a single aggregate objective function (AOF). The basic idea is to combine all of the objective functions into a single function. A well-known combination is the weighted linear sum of the objectives (WS). By means of specific scalar weights, the objective functions are combined into a single function that can be solved by any mono-objective optimizer (such as SQP, pattern search,... etc.).

The positively weighted convex sum of the objectives, is formulated as follow:

$$f(X) = \sum_{i=1}^{k} \alpha_i f_i(X),\ \alpha_i > 0,\ i = 1, \ldots, k \tag{2.38}$$

Frequently, these weighting coefficients are positive and respect the following: $\sum_{i=1}^{k} \alpha_i = 1$.

For illustration, Figure 2.16 shows a Pareto front obtained by an optimization of only two objective functions f_1 and f_2. The weighted-sum is as follow:

$$f(X) = \alpha f_1(X) + (1 - \alpha) f_2(X),\ \alpha \geq 0$$

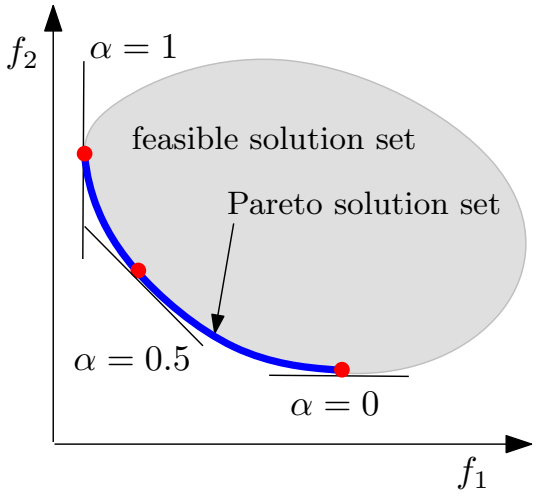


Figure 2.16: The concept of the weighted-sum of objectives.

One weight value and one optimization using for example SQP yield a Pareto optimum solution. The weighted-sum optimization problem is then run with different weight values $\alpha \in [0, 1]$ in order to find an optimal Pareto front. Note that the Pareto set have tangents with slopes that depend on the values of the weighing factor α.

Indeed, with the WS, the solutions depend on the relative values of the weights specified. For example, if we are trying to maximize the efficiency of a motor and minimize its mass, and if a higher weight for the mass objective is specified, then the solution will be one that favours lower mass over higher efficiency.

Discussion: strengths and weaknesses

The solutions obtained using the weighted sum are always Pareto optimal, and this approach gives an idea of the shape of the Pareto surface and provides the user with more information about the trade-off between all objectives.

However, this method has two drawbacks. First, the relationship between the vector of weights and the Pareto curve is such that a uniform spread of weight parameters rarely produces a uniform spread of points on the Pareto set. Often, all the points found are clustered in certain parts of the Pareto set with no point in the bend of the trade-off curve (middle part of the set). The second drawback is that non-convex Pareto sets are seldom found and can not be obtained with such as methods, which minimize convex combinations of the objective functions.

To overcome these disadvantages, more ambitious approaches, that aim to minimize the convex sums of the objectives for various settings of the convex weights, are proposed. The most interesting one is the adaptive weighted sum of objectives (AWS) presented in the chapter 6.

ϵ-constraint method

In the ϵ-constraint method introduced by [Haimes *et al.*, 1971], only one of the objective functions is chosen to be optimized. All the other objective functions are considered as constraints by setting an upper bound to each of them. The ϵ-constraint optimization problem

to be solved is now of the form:

$$\begin{cases} \min\limits_{x} f_1 \\ subject\ to\ f_i(X) \leq \epsilon_i,\ i = 2, \ldots, k \\ \qquad\qquad h_p(X) = 0,\ p = 1, \ldots, l \\ \qquad\qquad g_j(X) \leq 0,\ j = 1, \ldots, m \\ \qquad\qquad X \in S \end{cases} \tag{2.39}$$

where the parameters $\epsilon_i,\ i = 2, \ldots, k$ are additional constraint values. By varying $\epsilon_i,\ i = 2, \ldots, k$ for each optimization, the Pareto front can be found (see Figure 2.17).

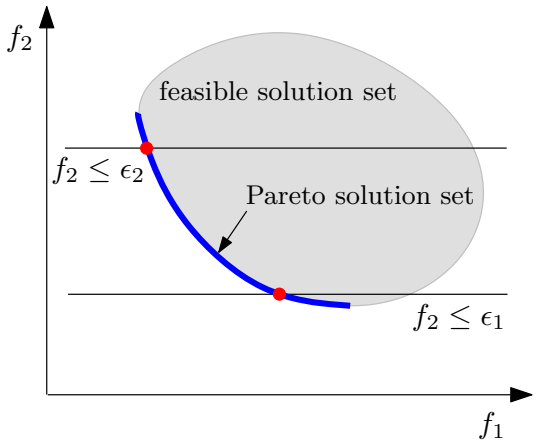


Figure 2.17: The concept of the ϵ-constraint method.

Discussion: strengths and weaknesses

A major strength of ϵ-constraint is its simplicity and the ease of use. Furthermore, this method can locate the whole Pareto front even in the non convex region. However, the necessity to determine the constraint values a priori presents a serious drawback. One way to overcome this disadvantage concists on varying $\epsilon_i,\ i = 2, \ldots, k$ between ϵ_i^{min} and ϵ_i^{max}. Then, two mono-objective optimizations are computed to determine the upper and lower threshold values. ϵ_i^{min} is found by minimizing $f_i,\ i = 2, \ldots, k$ and ϵ_i^{max} is found by minimizing f_1.

Hybrid methods are proposed in order to overcome these drawbacks. They consist on the combination between the weighting methods and the ϵ-constraint one. Then, strengths of both methods are combined in the hybrid approach. Note that, in the case of a bi-objective optimization as shown in Figure 2.17, the weight values $\alpha = 0$ and $\alpha = 1$ correspond to the threshold values ϵ_2^{min} and ϵ_2^{max}.

Computationally, an hybrid method is similar to the ϵ-constraint method with an increased number of constraint functions.

Multiobjective Optimization Evolutionary Algorithms (MOEA)

As shown in many papers [Fonseca & Fleming, 1996], [Kurpati *et al.*, 2002], [Deb *et al.*, 2002], [Erbas *et al.*, 2006], [Paraditwong & Yao, 2006], and [Cai & Wang, 2006] evolutionary algorithms are very adapted to handle multi-objective optimization problems.

By mean of the population-based technique, the evolutionary algorithms deal simultaneously with a set of solutions which allows us to find several points of the Pareto optimal set in a single run of the algorithm. However, with the traditional mathematical programming techniques [Kaisa, 2004, Kim & De Weck, 2006] or some metaheuristic like simulated annealing we have to perform several separate runs to obtain a set of Pareto solutions.

Early work

In 1985, Schaffer proposed the first evolutionary algorithm called VEGA (*Vector Evaluated Genetic Algorithm*). VEGA cannot be used to generate the Pareto set effectively. Indeed, the solution will focus on a medium point. The second drawback is that non-convex Pareto sets can not be obtained with VEGA.

Pareto dominance based algorithm

MOGA (*Multiple Objective Genetic Algorithm*), proposed by [Fonseca & Fleming, 1993], is the first algorithm that uses the notion of dominance directly to assess the performance of individuals.

Then, each individual is assigned a rank which allows it to be classified in relation to the others. The rank of an individual is related to the number of individuals that dominate it. Figure 2.18 illustrates this concept, where each point is labelled with the number of individuals that dominate it. Individuals will have a **rank 1** if they are not dominated by any other individual. Note that, both objectives f_1, f_2 are minimized.

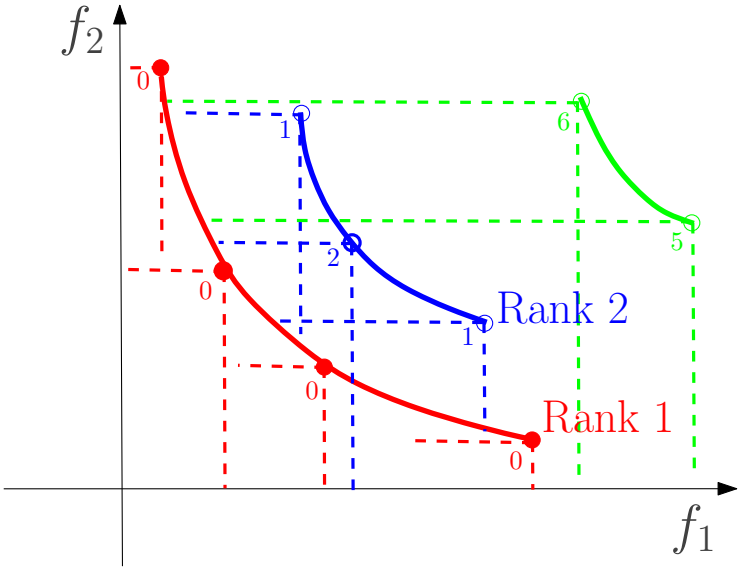


Figure 2.18: Rank of individuals.

The use of the Pareto dominance in order to classify the individuals represents the strength of MOGA. However, this algorithm produces poorly distributed solutions over the Pareto front.

To overcome this drawback, two methods NSGA and NPGA are proposed. NSGA for *Nondominated Sorting Genetic Algorithm* was proposed by [Srinvas & Deb, 1994]. NSGA is similar to MOGA. The main difference occurs when calculating the rank of the individuals. Based on Goldberg's suggestion [Roudenko, 2004] and [Goldberg, 1989], NSGA eliminates the shortcoming of MOGA and thereby distributes the population along the Pareto front using a sharing function. The drawback of NSGA is its dependence of the sharing distance σ_{share} and the niche count m_i of the individuals i.

The originality of NPGA is the implementation of a new selection operator called Pareto domination tournament [Horn *et al.*, 1993] and [Horn *et al.*, 1994]. As NSGA, NPGA is a parameter-dependent algorithm. The main parameter is the number of individual t_{dom} that are compared in order to be selected for reproduction.

Elitism in the sense of dominance

The modified Nondominated Sorting Genetic Algorithm (NSGA-II) [Deb *et al.*, 2002] and

the modified Strength Pareto Evolutionary Algorithm (SPEA2) [Zitzler *et al.*, 2001] are the most important multi-objective evolutionary algorithms that use elitism approach. Elitist approach has proved to be the most efficient. The main difference between both MOEA is in their fitness assignment schemes.

According to NSGA-II, the population is sorted based on non-domination into different front. For each individual in each front is assigned a fitness (rank) value based on front to which belongs it. Concerning SPEA2, for each individual is assigned a raw fitness calculated on basis of the strength value of solutions who dominate it. To discriminate between individuals having identical raw fitness values additional density information is incorporated [Zitzler *et al.*, 2001].

Nevertheless, to form the next generation, both algorithms combine offspring and current population. Subsequently, the best individuals in terms of nondominance and diversity are chosen using the binary tournament selection. For diversity preservation, NSGA-II [Deb *et al.*, 2002] [Deb, 2002] [Mohan *et al.*, 2003] uses a crowding approach which leads to a time complexity of $O(N\,log(N))$, where N is the population size. On the other hand, SPEA2 [Zitzler *et al.*, 2001] [Mohan *et al.*, 2003] uses a truncation procedure that requires a computational complexity of $O(M^2 log(M))$ where $M = N + N'$, and N' is the archive population size (see Annex B).

Both algorithms are compared in chapter 6. According to the optimization results, NSGA-II outperforms SPEA2 [Moussouni *et al.*, 2007]. Then NSGA-II is selected to be studied, modified, and improved in order to handle constrained multi-objective optimization problems dealing with mixed variables. For the reason of the large use of NSGA-II in this research, the main steps of this algorithms are highlighted below, and the Figure 2.20 is given as an illustration.

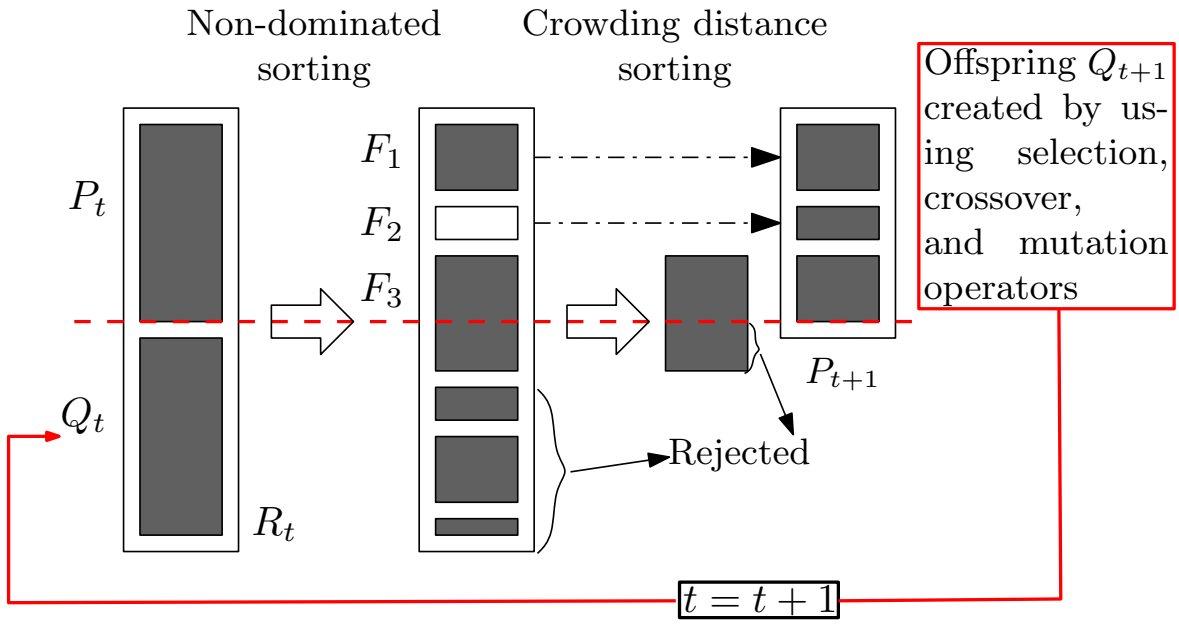


Figure 2.19: NSGA-II.

NSGA-II algorithm can be summarized by the following steps:

1. Generate a random population P_0 of size N, sort them using non-domination order and compute their crowding distance.

2. Initialize current population $P_t = P_0$.

3. Create a mating pool of parents of size $N/2$ from P_t using a standard binary tournament based on Pareto front ranking and crowding distance.

4. Randomly select couples from mating pool and apply bimodal crossover and polynomial mutation operators in order to generate offspring population Q_t of size N.

5. Combine current population P_t and offspring Q_t and sort them using non-domination order relation.

2.3.4 Outcome

Work comparing the evolutionary algorithms multi-objectif (AEMO) developed by Zitzler, Deb and Thiele in 2000, leaves no doubt that the introduction of elitism leads to improved performance. As against, it is difficult to say whether the performance of SPEA2 is better than NSGA-II because comparisons between these two methods elitists do not make a choice without hesitation [Zitzler *et al.*, 2001]. Figure 2.20 compares all the discussed evolutionary algorithms where the arrows point to the best one.

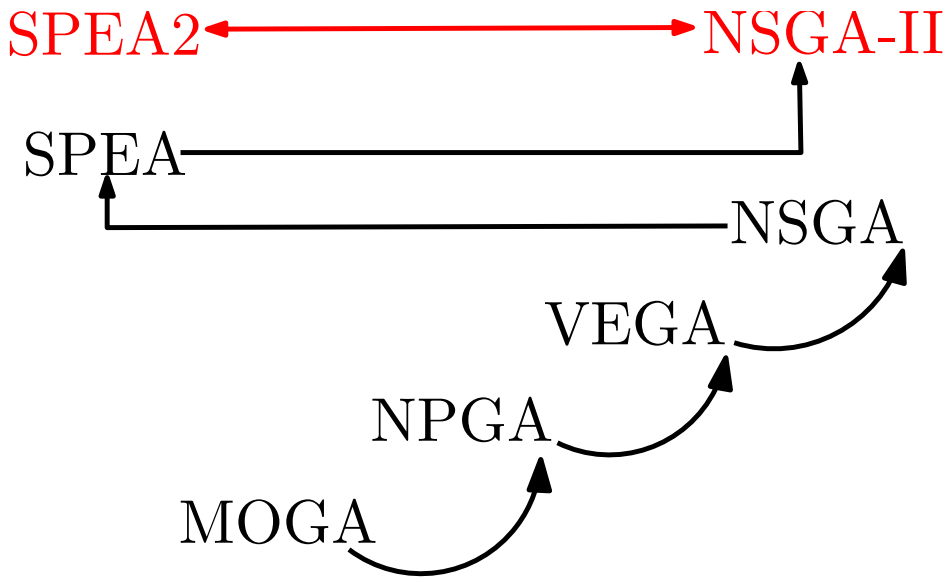


Figure 2.20: Multi-objective evolutionary algorithms comparison.

As discussed before, genetic algorithms can handle problems where the Pareto surface is discontinuous or non-smooth, unlike most other approaches.

2.4 Conclusion

In the optimization domain, the main question is: what method should be applied to solve my optimization problem?

- hard constrained multi-objective optimization problems;
- mixed-integer optimization problems;
- mixed-integer and multi-objective optimization problems.

According to the ***No free lunch theorem****: there is fundamental trade-off between algorithm efficiency and algorithm robustness (reliability and stability in a board range of problems) [Spall, 2003].*

Indeed, when efficiency, robustness and solution quality are all considered, there is no method better than any others. Furthermore, very few optimization methods can find a global optimal solution when the related functions of the optimization problems are not convex and non-differentiable.

Deterministic algorithms like SQP are inappropriate in the case of noisy functions and non-convex functions with several local minima. In order to overcome these disadvantages, the literature recommended to use stochastic global optimization methods like metaheuristics especially genetic algorithms (GA), ant colony optimization (ACO), and particle swarm optimization (PSO).

Evolutionary algorithms have become an increasingly popular design and optimization tool in the last few years, with a constantly growing development of new applications and algorithms especially the genetic algorithms. Despite this, new areas remain to be explored with sufficient depth. One of them is the use of genetic algorithms.

Indeed, GA is favourable for solving multi-objective optimization problems. This because it deals simultaneously with a set of solutions (population) allowing to find several points of the Pareto set in a single run of the algorithm. However, the traditional mathematical programming techniques or some metaheuristics like simulated annealing need to perform a series of separate runs to obtain the Pareto set. Contrarily to the mathematical programming techniques, GA can easily deal with discontinuous and concave Pareto fronts. However, these algorithms are less accurate and time consuming comparing to local search methods like SQP. Furthermore, the obtained solutions depend on the tuning of the parameters of the metaheuristic itself.

An other good alternative to overcome these disadvantages is the use of hybrid methods. Indeed, combining metaheuristics with local search methods is a practical remedy to overcome:

1. the trap of local optima concerning local search methods;
2. and the drawbacks of slow convergence and random constructions of metaheuristics.

Figure 2.21 summarizes the characteristics of the well-known classes of mono-level optimization algorithms.

For all these reasons, this thesis focuses on the improvement of genetic algorithms.

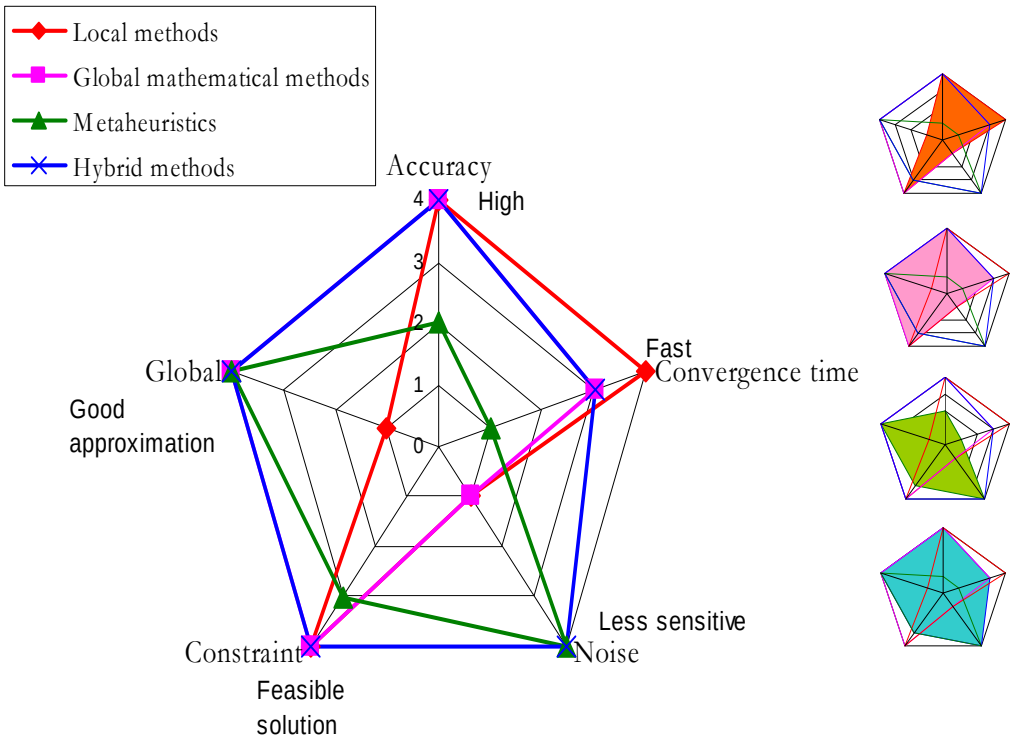


Figure 2.21: Summary of single-objective optimization algorithms

In part II, the stochastic global optimization methods (GA, ACO, PSO, SQP, and hybrid method) are studied, improved, and compared on the optimization of several practical optimization problems given from the electrical engineering domain.

Illustrative examples are employed to establish the more important concepts discussed in this state of the art.

3

Multi-level optimization

3.1 Overview

As introduced in this thesis, an important issue that this research will answer is how to deal with complex system design problems because they are very hard to solve.

When complexity of engineering system increases, the ability of engineers to understand the computed trade-offs declines rapidly. An approach to overcome these challenges is evident: try to break-down the problem into smaller ones that can be solved more easily and then compose the system solution from the solution of its components.

In practical, a complex engineering system is partitioned by object and/or by disciplines (aspects). Object decomposition involves dividing a system by physical component.

For example, according to business unit management organization of ALSTOM-Transport company, train design problem is decomposed into several problems. They are coordinated by system engineers at train level. System engineering team decides on allocation of several targets such as reliability, mass, volume, cost, and noise. These targets are assigned to sub-system engineering teams including e.g. bogie, carriage, and traction system. One department is in charge of traction system design and manufacturing. Inside this department, traction system design problem is also decomposed into sub-system and component design problems (traction box, motor, transformer, etc.). At lower level, a traction box contains several converters (rectifier, inverter, and chopper) and their cooling system. Each sub-system/component design problem is solved by several engineering units and suppliers. Moreover, specialized engineering teams can only focus on the design sub-problem in the domain that they are expert.

Such as decomposition and interaction between components are mainly concerned in **Multi-level Optimization** which is the subject of this chapter. However, aspect partitionning divides the system by disciplines. The same tram traction system might be partitioned by aspect into mechanical, electrical, and thermal disciplines. Interaction between such disciplines is mainly concerned in Multidisciplinary Design Optimization (MDO) which is not the scope of this research. For more detail please refer to [Braun *et al.*, 1996], [Alexandrov & Lewis, 1999], [Alexandrov, 2002], [Allison, 2004], [Allison *et al.*, 2005b].

As illustrated in Figure 3.1, the engineering design problems can be viewed as a hierarchical process, i.e. they have a hierarchical decomposition. In other terms, the whole design system is decomposed into sub-systems, which are decomposed into components and so on.

In this work, such hierarchical design problems are solved using a hierarchical multi-level optimization approach called "Analytical Target Cascading" (ATC) [Kim *et al.*, 2003b], [Kim *et al.*, 2003a], [Kokkolaras *et al.*, 2004], [Choudhary *et al.*, 2005], [Huang *et al.*, 2005], [Michalek *et al.*, 2005], [Cooper *et al.*, 2006], [Ling *et al.*, 2006], [Liu *et al.*, 2006], and

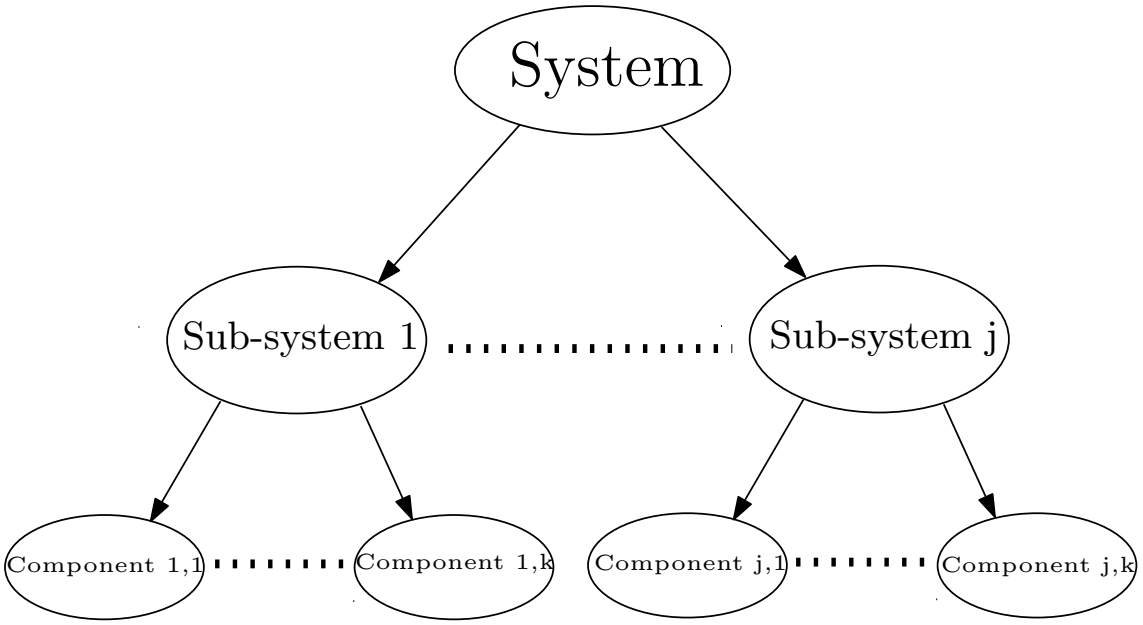


Figure 3.1: Hierarchical decomposition of a complex problem.

[Michalek & Papalambros, 2006].

3.2 Analytical target cascading

Analytical target cascading (ATC) is a hierarchical multi-level design methodology. As explained by [Michelena *et al.*, 1999], under ATC paradigm, product design is a four-step process:

1. specify overall product targets,
2. propagate product targets to system, sub-system and component sub-targets;
3. design system, sub-systems and components to achieve their respective sub-targets;
4. and verify that the obtained product meets overall product targets.

Note that, sub-targets are the responses and linking variables defined below. To summarize, the aim of ATC process is to enssure that systems, sub-systems, and components are compatible and operate together in a consistent way.

This approach is used to deal with large-scale design problems, which are broken down into several small sub-problems. Thus, with ATC we have:

- smaller problems to solve:
  - lower number of local minima,
  - lower dimension (number of the design variables) for each sub-problem,
  - high size of attraction region of the sub-solutions;
- coordination to ensure optimality and optimum design of sub-systems and components.

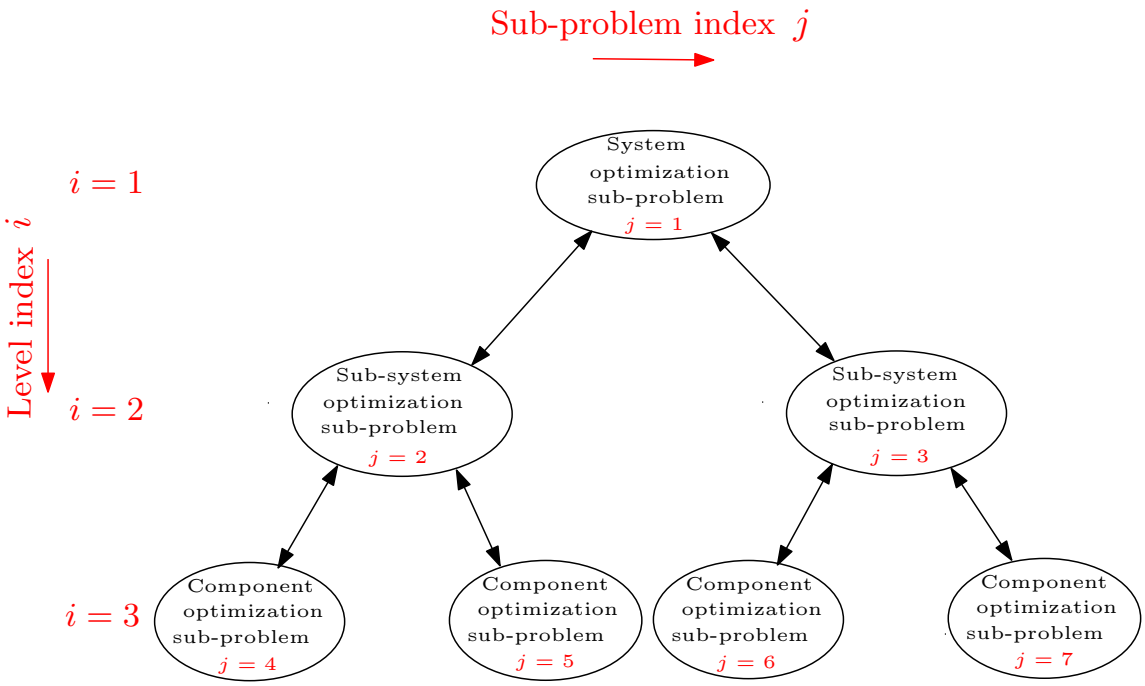


Figure 3.2: ATC: level and sub-problem indexes.

To simplify the presentation of ATC, [Michelena *et al.*, 2003] proposed to index levels and sub-problems with i and j respectively (Figure 3.2).

Figure 3.2 shows an example of an optimization problem decomposition into three levels: system level, sub-system level and component level. Using ATC, for each optimization sub-problem a suitable optimization method (for example, one among those presented in the previous chapter) is applied to solve it. In this multi-level process, each optimization sub-problem has its own local design variables, local constraints as presented in Figure 3.3.

To ensure the consistency of the multi-level process, ATC connects the child sub-problems with its parent sub-problem by using the so-called:

- Targets (T): are the overall product targets which are cascaded from the entreprise level to the system one. ATC process aims to achieve a system design that satisfies the product specifications. This is guaranteed only when the responses system R_{11}^1 mets the product targets. Then the optimization problem at the system level is to minimize the error between the product targets T and system responses R_{11}^1. The deviation error is measured using the square **L-2 norm**, i.e. $\left\|R_{11}^1 - T\right\|_2^2 = \sum_i (R_{11}^1(i) - T(i))^2$.
- Responses (R): are back up from the children sub-problems to their parent sub-problem.
- Linking variables (Y): which characterise the shared variables and interactions between sub-problems.

Note that, T, R, and Y are vectors.

Figure 3.3 summarizes the information flow for each sub-problem (element) of ATC process, where the indexes c and p represent the children and parents of the element $subpb_{ij}$. One can observe that there is no direct link between any two sub-problems in the same level. The interactions are managed by their parent sub-problem via the linking variables. Then, by decoupling the sub-problems of the same level, they can be launched in parallel.

To summarize, at the system level, ATC minimize discrepancy between the targets and the computed responses [Kim *et al.*, 2003b]. Thereafter, these responses and the linking variables

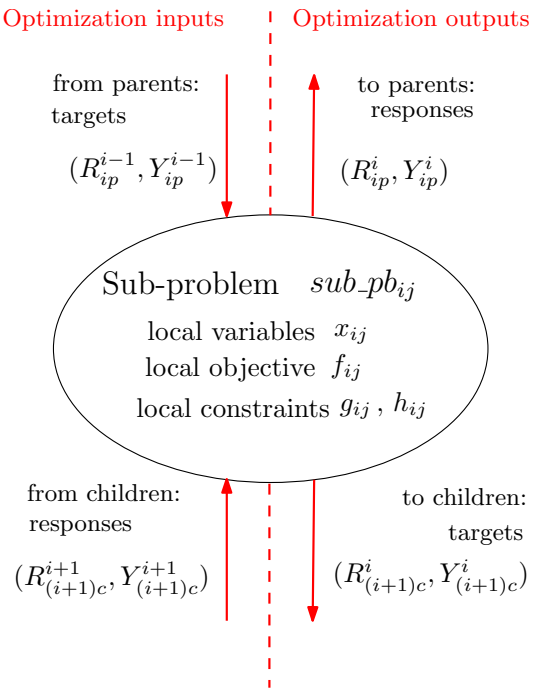


Figure 3.3: Information flow for ATC sub-problem.

are cascaded to the sub-system sub-problems and so on. These cascaded information will be considered as targets for the lower-level sub-problems.

At lower-levels, the sub-problems minimise discrepancy between its responses and linking variables and those of the upper level. The process continues until reaching the bottom level. The information from bottom level are then returned back to upper level. The optimizations at the upper level are re-run until convergence, i.e. consistency is reached.

Note that the term analytical target cascading is used to describe the relevant decision models. In the following the ATC formulation is detailed.

3.3 Optimization problem formulation

According to the literature, there is two main features in analytical target cascading (ATC): breaking-down and coordination [Allison, 2004] - [Tzevelekos *et al.*, 2003].

A large scale problem is broken down into sub-problems, thereafter a coordination strategy is applied to ensure consistency between these sub-problems solutions. For simplicity, only two levels are considered to explain the ATC concept.

3.3.1 Break down

Following the ATC approach, a complex problem is decomposed into levels and sub-problems [Kim *et al.*, 2003b], [Michelena *et al.*, 2003], [Kokkolaras *et al.*, 2004], [Huang *et al.*, 2005], [Ling *et al.*, 2006]. In Figure 3.4, the whole problem can be decomposed into three sub-problems sub_pb_{ij}, where $i \in 1, 2$ and $j \in 1, 2, 3$. There are one upper-level sub-problem (sub_pb_{11}) and two lower-level sub-problems (sub_pb_{22} and sub_pb_{23}). Furthermore, general targets T are assigned at the upper-level.

3.3.2 Coordination

In the ATC assumption, at least one output of a lower sub-problem has to be an input to the upper-level sub-problem with which it is linked. Components of the same system may share one or more design variables. In Figure 3.4 both lower-level sub-problems share a linking variable denoted Y_{22}^1 in sub_pb_{22} and Y_{23}^1 in sub_pb_{23}. This linking variable is coordinated at the upper-level and denoted Y_{11}^1, i.e. at optimality $Y_{22}^1 = Y_{23}^1 = Y_{21}^1$. Similarly, for each lower-level sub-problem responses R_{22}^2 and,R_{23}^2, duplicate variables are created at the top-level sub-problem and,$(R_{22}^2 = R_{22}^1)$, $(R_{23}^2 = R_{23}^1)$ at optimality. In the ATC approach, the adjacent-level sub-problems optimization are solved consecutively, which allows cascading targets between these levels. At each ATC iteration, the overall responses R_{11}^1) are adjusted, based on the feasibility of lower-level design, to match the design targets T with minimum deviation.

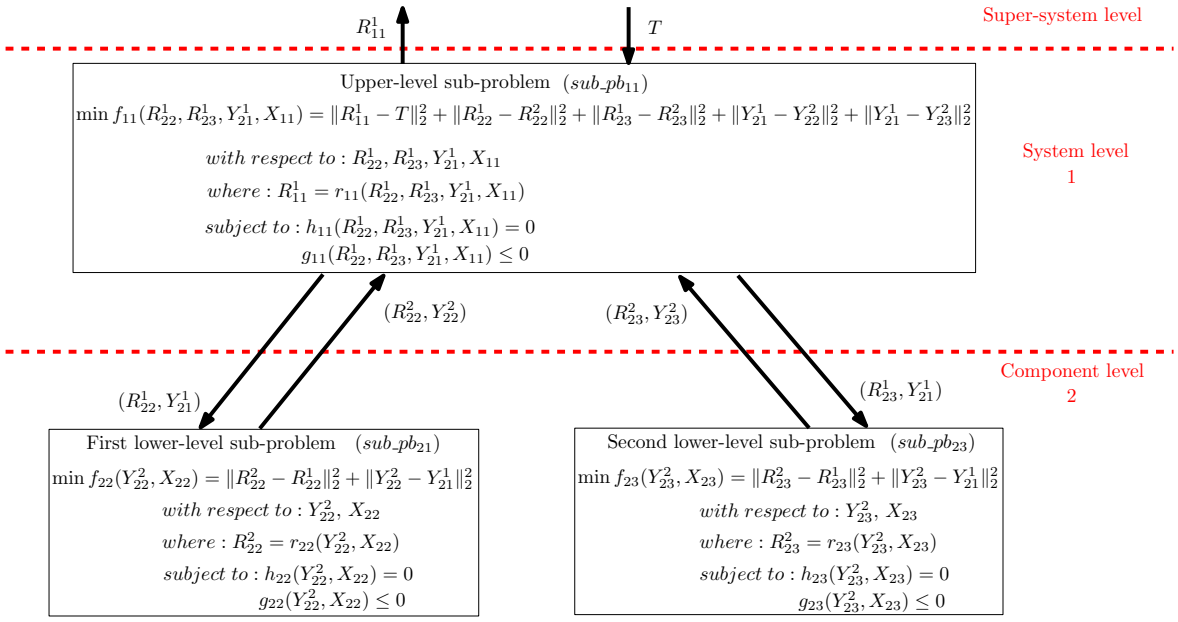


Figure 3.4: ATC: decomposition and coordination.

In addition to X_{11}, the optimization of the upper-level sub-problem sub_pb_{11} searches for responses R_{22}^1, R_{23}^1, and linking variables Y_{21}^1. Note that R_{22}^2, R_{23}^2 and Y_{22}^2, Y_{23}^2 are the optimal results sent back from the lower-level sub-problems.

With ATC, a large scale problem is decomposed into several levels, where each sub-problem sub_pb_{ij} of the level i is decomposed into c_{ij} children sub-problems regrouped in the set C_{ij}.

First, the system level sub-problem formulation is presented. Thereafter, the general notation for a single ATC element sub_pb_{ij} in the hierarchy is given.

3.3.3 System level sub-problem

The optimization problem at the system level is to minimize the error between the targets T, cascaded from the firm level, and the system responses R_{11}^1. The sub_pb_{ij}, $i = 1$, $j = 1$ is

defined as:

$$
\begin{array}{lll}
sub-pb_{11}: & \min f_{11} = & \left\|R_{11}^1 - T\right\|_2^2 + \sum_{k\in C_{11}} \left\|R_{2k}^1 - R_{2k}^2\right\|_2^2 \\
 & & + \sum_{k\in C_{11}} \left\|S_k Y_{21}^1 - Y_{2k}^2\right\| \\
 & \text{with respect to} & X_{11}, Y_{21}^1, R_{2k}^1 \\
 & \text{where} & R_{11}^1 = r_{11}\left(R1_{2k}, Y_{21}^1, X_{11}\right) \\
 & \text{subject to} & g_{11}\left(R_{2k}^1, Y_{21}^1, X_{11}\right) \leq 0 \\
 & & h_{11}\left(R_{2k}^1, Y_{21}^1, X_{11}\right) = 0
\end{array}
\tag{3.1}
$$

Note that R_{11}^1 is a function of the local design variables X_{11}, the responses R_{2k}^1, and the linking variables Y_{21}^1. $k \in C_{11}$, where $c_{11} = |C_{11}|$ is the number of the children sub-problems associated to the system level sub-problem sub_pb_{11}. Addition to X_{11}, the optimization problem must search for the responses R_{2k}^1, and the linking variables Y_{21}^1.

The terms $\sum_{k\in C_{11}} \left\|R_{2k}^1 - R_{2k}^2\right\|_2^2$ and $\sum_{k\in C_{11}} \left\|S_k Y_{21}^1 - Y_{2k}^2\right\|$ are used in the system objective function to ensure consistency between system and sub-system levels for each child sub-problem of system $k \in C_{11}$. Where S_k is the selection matrix indicating which terms of the parent coordinating linking variable vector Y_{21}^1 are relevant to the linking variable vector Y_{2k}^2 at sub-problem sub_pb_{ij}. The deviations are measured using the square **L-2 norm** denoted by $\|\cdot\|_2^2$ (e.g.,$\|A\|_2^2 = \sum_i A_i^2$.

Furtheremore, R_{2k}^2 and Y_{2k}^2 are the optimal results sent back from system level. The superscript 2 indicates that the value comes from the 2^{th} level.

3.3.4 Single ATC element in the hierarchy

The objective of the j^{th} sub-problem in the i^{th} level, called $sub - pb_{ij}$, is to minimise the discrepancy between its responses R_{ij}^i and its parent sub-problem targets (R_{ij}^{i-1}); and between its linking variables Y_{ij}^i and those of its parent sub-problem Y_{ij}^{i-1}.

Note that R_{ij}^{i-1} and Y_{ij}^{i-1} are the optimum responses and linking variables values found by the optimization of the parent sub-problem in the $i-1^{th}$ level.

$$
\begin{array}{lll}
sub\_pb_{i,j}: & \min & f_{ij} = \left\|R_{ij}^i - R_{ij}^{i-1}\right\|_2^2 + \left\|Y_{ij}^i - S_j Y_{ij}^{i-1}\right\|_2^2 \\
 & & + \sum_{k\in C_{ij}} \left\|R_{(i+1)k}^i - R_{(i+1)k}^{i+1}\right\|_2^2 \\
 & & + \sum_{k\in C_{ij}} \left\|S_k Y_{(i+1)j}^i - Y_{(i+1)k}^{i+1}\right\|_2^2 \\
 & \text{with respect to} & X_{ij}, Y_{ij}^i, y_{(i+1)k}^i, R_{(i+1)k}^i \\
 & \text{where} & R_{ij}^i = r_{ij}\left(X_{ij}, Y_{ij}^i, y_{(i+1)k}^i, R_{(i+1)k}^i\right) \\
 & \text{subject to} & g_{ij}\left(X_{ij}, Y_{ij}^i, Y_{(i+1)k}^i, R_{(i+1)k}^i\right) \leq 0 \\
 & & h_{ij}\left(X_{ij}, Y_{ij}^i, Y_{(i+1)k}^i, R_{(i+1)k}^i\right) = 0
\end{array}
\tag{3.2}
$$

It can be observed that R_{ij}^i is a function of the local design variables X_{ij}, the responses $R_{(i+1)k}^i$, and linking variables Y_{ij}^i and $Y_{(i+1)k}^i$. Addition to X_{ij}, the optimization problem must search for responses vector $R_{(i+1)k}^i$, linking variables vectors Y_{ij}^i and $Y_{(i+1)k}^i$.

Similar to S_k, S_j is the selection matrix indicating which terms of the parent coordinating linking variable vector Y_{ij}^{i-1} are relevant to the linking variable vector Y_{ij}^i at sub-problem $sub - pb_{ij}$.

Note that $R_{(i+1)k}^{i+1}$ and $Y_{(i+1)k}^{i+1}$ are the optimal results sent back from the children sub-problems. The superscript $i-1$ indicates that the value comes from the upper-level $(i-1)^{th}$, and the index $i+1$ means that the optimal values will be sent to the lower level $(i+1)^{th}$.

3.4 Conclusion

Analytical target cascading (ATC) is a hierarchical multi-level design methodology. It has been applied in many application such as aircraft design [Allison *et al.*, 2005a], automotive vehicle design [Kim *et al.*, 2003a].

Furthermore, this hierarchical decomposition of complex systems is suitable to railway systems. As the study presented in this thesis is supported by Alstom-Transport, ATC is the powerful method that can be applied in order to yield to the requirements of Alstom-Transport. Indeed, ATC allows applying optimization in large-scale design problem which is broken-down into several small sub-problems. By using ATC formulation, each design sub-problem is formulated as an optimization problem. Thereafter, ATC coordinates interactions and information exchanged between them.

When a complex design system is treated as a whole optimization problem, optimization algorithm may suffer from high number of design variables and constraints. In contrast, ATC works with several small optimization problems, which are easier to solve.

However, as for all methods, ATC have strengths and weaknesses. Indeed, when dealing with unattainable overall targets, a strict design consistency cannot be achieved. Then the convergence of ATC cannot be met.

Chapter 8 is devoted to the application of ATC in electrical engineering domain, when the practical problems have always unattainable tragets. To overcome this drawback, chapter 8 proposed a novel algorithm to improve the convergence of ATC.

Part II

Improvement of algorithms

4

Tuning of optimization algorithm's parameters

As highlighted in the **State of the art**, the metaheuristics: genetic algorithms (GA), ant colony optimization (ACO), and particle swarm optimization (PSO) are sensible to its control parameters itself. Table 4.1 presents the most important parameters of the three algorithms.

Table 4.1: List of optimization algorithm's parameters.

GA	ACO	PSO
- Population size - Initial population - Crossover probability - Mutation probability - Crossover operator - Mutation operator - Selection operator - Reproduction operator - Fitness operator	- Ant number - Concentration of pheromone - Attraction factor (ant visibility) - Pheromone persistence - Relative importance to the pheromone - Relative importance to the visibility	- Particle number - Inertia factor - Preserving factor - Tendency factor to follow the best neighbour

Beside these metaheuristics, sequential quadratic programming (SQP), which is the reference in the non-linear programming, has less parameters to be tuned. However, in addition to the need of derivative calculation, the more critical parameter for SQP is the choice of the starting point. Like SQP, the solutions given by these three metaheuristics are sensible to the initial points employed in the beginning of the optimization process.

To take advantage on these optimization algorithms, this chapter focuses on the tuning of GA, ACO, PSO, and SQP parameters in the aim to improve their accuracy and time of convergence. First, the influence of the initial points on the algorithm's solution is studied. Second,the remainder parameters of GA, ACO, and PSO are tuned.

In the aim to show the influence of the algorithm's parameters, two practical example are used as benchmarks in this study:

1. Optimal design of a tram traction system.
2. Optimal design of a brushless DC wheel motor.

4.1 Initial points sampling

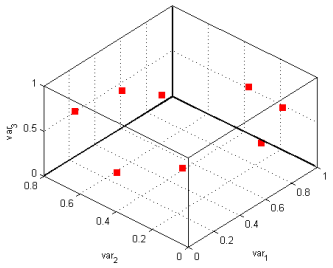
SQP is a non-linear programming method which starts from a single searching point and finds a solution using the gradient information. Therefore, if the starting point is far from the global solution, the SQP algorithm can be trapped easily in local optima. To overcome this disadvantage, SQP should be restarted with several initial points. Like SQP, GA is an initial population dependent algorithm. Indeed, when the initial population is well distributed over the design space, GA explore more interesting space and a good approximation of the global solution can be obtained.

Several sampling techniques like Latin Hypercube, Grid, and Monte-Carlo sampling, can be used to provide good starting points for GA and SQP. Using these sampling techniques, this section intends to highlight the influence of the starting points distribution - employed by GA and SQP - on the behaviour of the algorithms and the quality of the obtained solutions.

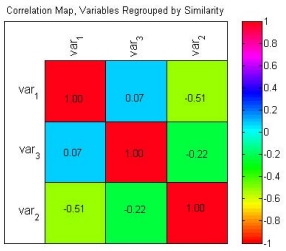
4.1.1 Sampling techniques

Ideally, for an optimization algorithm, a set of candidate points should have a good distribution of design variables configurations. This equates to have a low correlation between these design variables.

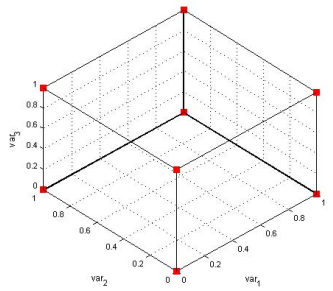
For example as illustrated in Figure 4.1(a), if the combinations of the three variables (var_1, var_2, var_3) are all in the same part of the design space, the correlation between them is high (Figure 4.1(b)). The search space is then poorly explored compared to Figure 4.1(c) and Figure 4.1(d).



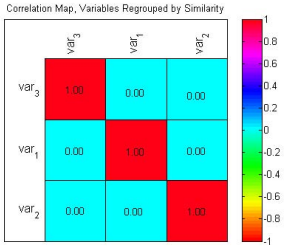
(a) Bad distribution.



(b) Correlated variables.



(c) Good distribution.



(d) Not correlated variables.

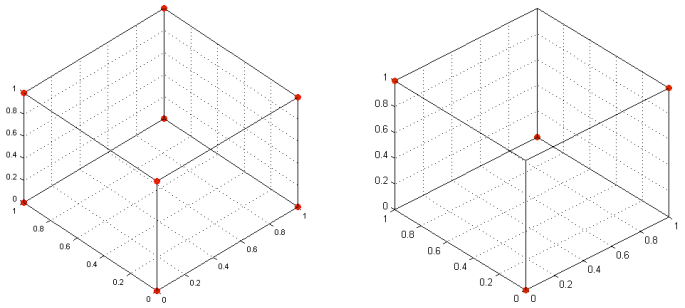
Figure 4.1: Distribution of points over the search space.

To provide a good starting point for the optimization algorithms SQP and GA, three initial point sampling methods are used: Monte Carlo Sampling (MCS), Grid Sampling (Grid) and Latin Hypercube Sampling (LHS). Notes that, grid design is a kind of the classical experimental design.

Classical experimental design

The most basic design is two-level full factorial design [Vivier, 2002b]. It places sample points at all combination of lower and higher level of each design variable. This results in 2^n points, where n is the number of design variables.

By using $2-level$ factorial design, it cannot capture some information and explore the design domain. Various designs are possible for this purpose like $p-level$ factorial design with p^n sample points. However, as the number of design variables increases the number of sample point increases rapidly. Then, for high dimensional design space, fractional design can be used, when the number of sample point is decreased to p^{n-r}, r is a reduce order. Figure 4.2(a) shows how sample points are placed on the **fractional factorial design hypercube** with 2 levels and 3 design variables. With 2^{3-1} fractional design, the number of points is reduced from 8 to 4.



(a) Full factorial design space. (b) Fractional factorial design space.

Figure 4.2: Full and fractional factorial design with 3 variables.

Monte Carlo Sampling

Monte Carlo Sampling (MCS) [Cioffi *et al.*, 2005], also called random design, intends to mimic a random natural process. MCS is very easy to implemented when using a programming language in which a *random* function is available. Position of a sample point in the m^{th} dimension can be located by selecting a random number in an interval of design variable in the m^{th} dimension. Therefore, a MCS point for $m-dimensional$ design space is represented by a vector of n random numbers in the interval $[X^{Low}, X^{Up}]$, where X^{Low}, X^{Up} are the lower and upper bound vectors of the variables. For example, Figure 4.3 represents MCS sampling points in $3-dimensional$ design space.

Latin Hypercube Sampling (LHS)

As MCS, Latin Hypercubes Sampling (LHS) [McKay *et al.*, 2000] is also a modern DOE or space-filling design. Sample points are scattered throughout the whole design space.

By giving a number of sample points, p, the design space is divided into p intervals of equal probability for each dimension. This results in p^n hypercubes or bins. LHS requires that sample sites meet the following criteria:

- A bin can contain only one point, which is placed randomly in the bin.
- When projecting design space in any two dimensions, there are p points and bins. Only one bin is selected in each row and column.

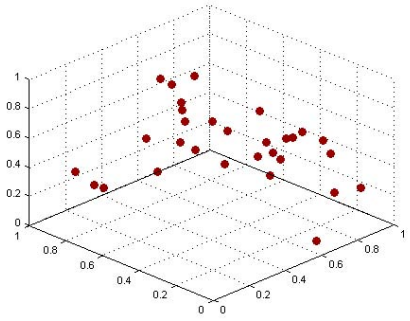


Figure 4.3: Monte-Carlos sampling.

LHS are represented using LHS matrix of p rows and n columns. This matrix is quite easy to generate. It requires a random permutation of p levels for each column. Each column of LHS matrix is defined as:

$$LHS_j = \frac{U_j - R_j}{p} \tag{4.1}$$

where U_j is uniformly distributed random permutation of the integers from 1 to p, R_j is [0, 1] uniformly distributed random number vector of p elements. In Figure 4.4 sampling points in 3-dimensional design space is shown.

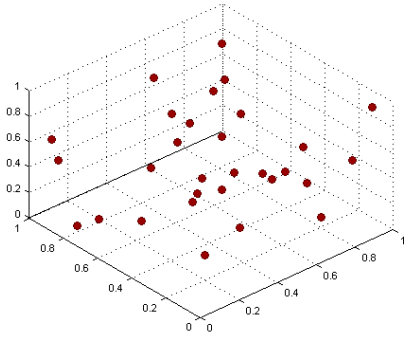


Figure 4.4: Latin hypercube sampling.

In order to highlight the influence of the initial conditions, starting point for SQP and initial population for GA, on the behaviour of the algorithms, the optimization problem of a tram traction system is used as a benchmark.

First, an application of tram traction system is presented. Thereafter, the affect of the initial points, sampled using MCS, grid design, and LHS, on the obtained solution is studied.

4.1.2 Optimal design of a traction system

The main issue of this chapter is to study the affect of the starting points sampling on the obtained solutions by optimization algorithms like SQP and GA. A mono-objective design of a traction system design problem is chosen as a benchmark.

The objective in this section is not to study and discuss the optimization design of this application. Then for the purpose of clarification, this benchmark is described briefly. Only the design variables, constraints and objective functions are described. The traction system model is detailed in chapter 8, where it is used in a more complex test case.

Optimization problem

Alstom Transport has developed a standard traction system for tram application in order to minimize the cost and the delivery time. Each applicative project has different customer specifications. These components are chosen and adapted according to these specifications in the goal of satisfying the global performance of tram operating on a predefined track. In this study, the goal of Alstom is to increase the passenger capacity of the tram. The traction system would be re-designed due to the increase in global weight. The hardware re-design is discarded because the cost is unacceptable. For this reason, only control parameters can be modified. The solution has to meet not only the hardware constraints but also the customers' specification.

Due to a change in passenger capacity, the vehicle weight is increased. In the normal operating mode the simulation results show that the tram can be operated perfectly without any problem. According to customer contract specification, in the faulty operating mode (one traction box is defective) the tram must be able to continue the service until the end of the trip with the remaining 50% motorization (3 motors) with the same performance as in the normal operating mode except that the maximum speed is reduced. The simulation results show that the required performance can be achieved. However, some problems are observed:

1. the temperature of IGBT (Insulate Gate Bipolar Transistor) modules used in VSIs was over the limit of $125°C$ defined by the semi-conductor manufacturer. The normal operation of IGBT cannot be guaranteed beyond this temperature limit;

2. Due to the increased torque, the line current did not respect to the line filter specification.

The objective fixed in this study is to minimize the maximum IGBT junction temperature (T_j) during the round-trip of tram in faulty operating mode on the customers' route profile.

Four control parameters are selected (see Table 4.2). The first two design variables are the parameters for the pulse width modulation (PWM) used in VSI. Two Asynchronous Space Vector (ASV) schemes [Holtz, 1992] with different switching frequency are used in different speed zone.

Table 4.2: List of design variables.

Number	Design variable	Unit	Symbol
1	VSI switching frequency	Hz	f_{pwm_low}
2	Min. ratio $\frac{\text{VSI switching frequency}}{\text{VSI output frequency}}$	(-)	n
3	Maximum torque limit	Nm	T
4	Maximum power limit	kW	P

The last two design variables are the maximum torque limit and the maximum power limit of the motor.

Due to the component specifications and performance requirements, this problem presents 11 inequality constraints given in Table 4.3.

Table 4.3: List of constraints.

	Inequality constraint	Unit
1	Inductor current–traction mode	A
2	Inductor current–regenerative braking mode	A
3	Inductor thermal current–braking resistor mode	A
4	Inductor thermal current–regenerative braking mode	A
5	Capacitor thermal current–braking resistor mode	A
6	Capacitor thermal current–regenerative braking mode	A
7	Peak phase current	A
8	Motor rotor temperature	°C
9	Motor winding temperature	°C
10	Motor iron temperature	°C
11	Round-trip time	s

To summary, the optimization design problem of the traction system is represented graphically in Figure 4.5 using the convention proposed by our research group. The left side shows the 4 design variables. The objective function and constraints are in the right side.

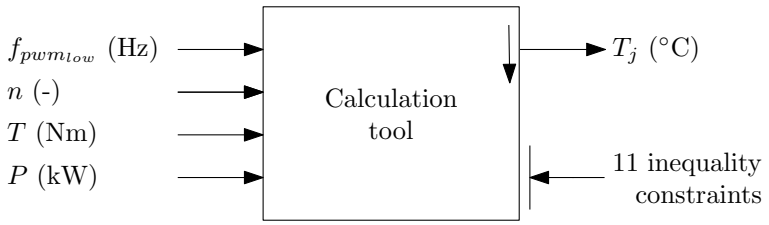


Figure 4.5: Optimization problem.

To show the hardness of this optimization problem, first SQP multi-start with 1000 initial points, chosen randomly, is performed. The found best solution ($T_j = 114.25 \circ C$) is chosen in this study as a reference solution.

As shown in Figure 4.6, the studied problem is a hard optimization problem with several local optima. Using 1000 starting points only 20% allow SQP method to converge to the best solution represented by the first bar with $T_j = 114.2514°C$. It means that approximately 80% of initial points lead to local optima. Therefore, it is obvious that the initial points sampling may affect the quality of the obtained solutions.

To explain this purpose, three algorithms SQP, GA and hybrid method combined both are applied with starting points sampled differently over the design space. By using the three sampling methods presented in the previous section: MCS, Grid Sampling and LHS.

Setting of initial points

Beside the sampling techniques, the number of sample points is also one of the main criteria in order to ensure a good distribution among the search space. The proposed optimization problem has 4 design variables, then, it seems well that, at least 3 levels per dimension is necessary to capture the non-linearity of the problem. Then 4 levels are ensured for the design variable $f_{pwm_{low}}$ and only 3 levels are guaranteed for the three remainder variables.

In other therms, three sets of $108 = 4 * 3 * 3 * 3$ starting points are chosen by using

1. Monte Carlo sampling (MCS),

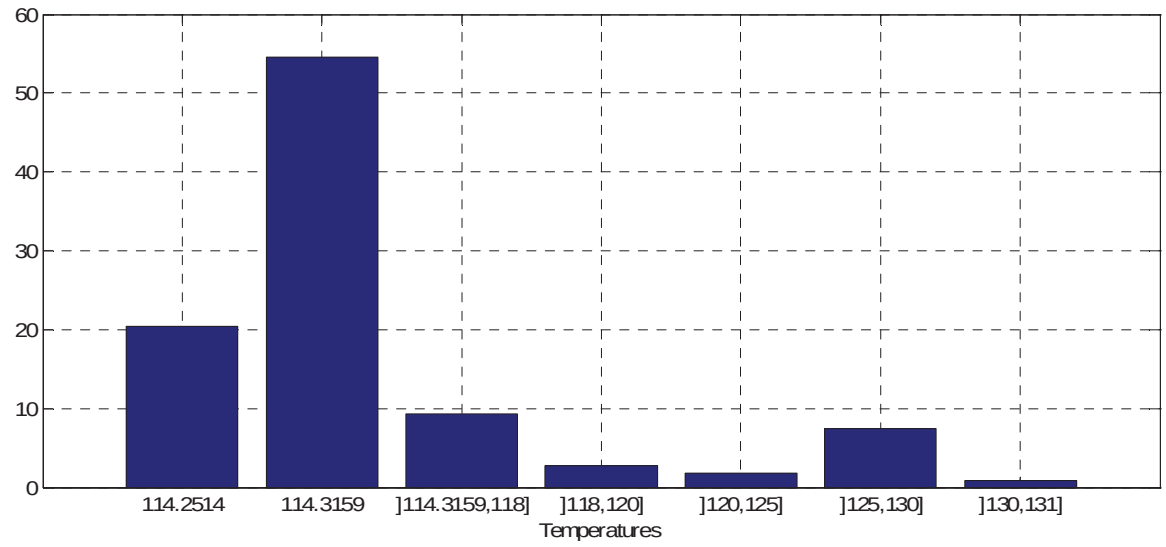


Figure 4.6: Optimization problem with many local optima.

2. Grid sampling (Grid) ,
3. and Latin hypercube sampling (LHS).

These three sampling sets are exploited as follow:

- First, they are used as initial points for SQP multi-start.
- Second, they are applied as GA's initial populations containing 108 individuals each.

In this comparison, using SQP, all design variables are scaled to $[0, 1]$. For GA, the parameters have the following values: population size $N = 108$, maximum number of generations $T = 100$, crossover probability 0.8, rank fitness scaling, scattered crossover, and stochastic uniform selection operators. If the objective value is not improved over 10 generations or the objective function tolerance is less than 10^{-6}, the algorithm stops.

Concerning hybrid method combining GA and SQP, first GA algorithm is initialized like presented above, thereafter it is stopped prematurely after only 5 generations. Then the solutions found by GA are used as initial points for SQP algorithm.

Comparative results

Table 4.4 shows the influence of each sampling technique on the solutions obtained by the three optimization algorithms.

This table compares the three optimization algorithms applied with different starting sampling sets containing 108 points each. They are compared in term of percentage of convergence (denoted PC) to the reference point ($T_j = 114.25°C$) with an Euclidean distance lower than 10^{-3}, and the average number of evaluations (denoted NE mean). Note that the computational time is given in term of the mean number of evaluations. For example, the computation time of SQP with LHS approach is equal to $121 \cdot 0.08s = 9.68s$, where $0.08s$ is the average needed time to evaluate the objective and constraint functions, 121 is the number of SQP's evaluations.

Concerning both algorithms SQP and GA, LHS performs slightly better than other sampling approach. As shown in Table 4.4, SQP converged at average of one hundred evaluations

of the objective function. However, whatever the choice of the initial point set, with any ten points, SQP can converge at least once ($PC = 15.74\%$) or at most twice ($PC = 23.15\%$) to the reference solution. Indeed, whatever the used sampling technique the percentage of convergence of SQP is between 15.74% and 23.15%.

Table 4.4: Optimization results.

	SQP		GA		Hybrid	
	PC	NE mean	PC	NE mean	PC	NE mean
Rand	20.37	116	1	5987	11	3380
Grid	15.74	113	19	5025	70	3380
LHS	23.15	121	16	9906	32	3323

It can be observed from this table, that GA results are less accurate than SQP ones. With an accuracy of 10^{-1}, GA found solutions, which are close as soon as possible to the best solution. However, the percentages of convergences for a precision of 10^{-3} are zero for any used sampling method. This is because, as will be shown in the second part of this chapter (Section 4.2), the solution obtained by GA depends on other parameters of the algorithm itself like the population size, probabilities of crossover and mutation and so on.

However, these results reveal the performance of the hybrid method though the parameters of the used GA are also not well tuned. Indeed, the results of hybrid algorithm combining GA and SQP are more accurate than SQP and GA alone while fewer evaluations are required. This shows the dominance of the hybrid technique as will be confirmed in chapter 5.

Furthermore, performing GA with a grid sampling initial population the obtained solutions are more close to the reference point than when other sampling technique is used. Thereafter, each solution is used by SQP as a starting point in order to overcome local optima. This explains the highest percentage of convergence of the hybrid method with the grid sampling approach.

4.1.3 Outcome

This section aims to stress the fact that, prior to decide what further optimization method to use, it may be useful or even essential to carry out a preliminary exploration of the design space. This help us to provide good initial values of the design variables or to build some understanding of the behaviour of the objectives and constraints. For this purpose, some sampling methods can be used like Monte Carlo sampling (MCS), Grid sampling and Latin hypercube sampling (LHS).

Several optimization algorithms like GA, ACO, PSO and SQP, are starting points dependent. This study revealed that quality of solutions obtained by such algorithms depend on the sampling technique used to distribute the initial point over the design space.

Three sampling approaches - MCS, Grid, and LHS - are used with GA, SQP multi-start, and hybrid method combining both algorithms. The optimization results show that hybrid method with initial population distributed over the search using grid sampling is powerful. Indeed, it gives a greater percentage of convergence and lower computational time compared to SQP and GA performed alone.

The disadvantage of MCS is that the points may not well cover all design space due to the fact that each dimension of a sample point is selected independently. Furthermore, it provides less uniformly distributed sample points. However, Grid and LHS techniques are more structured compared to MCS.

In this study, genetic algorithms give the worst solutions in term of accuracy and time of convergence. This is due to the dependence of GA on other parameters like the population size, crossover probability and so on. In the next section, instead of the initial points sampling,

the influence of other parameters on the optimal solution given by GA, ACO, and PSO is studied.

4.2 Other parameters

This section is devoted to study the influence of other control parameters on the obtained solutions by the optimization algorithms (GA, ACO, and PSO).

With genetic algorithms (GA), ant colony optimization (ACO), and particle swarm (PSO) there are many parameters to be tuned in the aim to provide solutions with a high quality. Unfortunately, to now the tuning of the control parameters for such as algorithms is not acquired [Colorni *et al.*, 1992] [Eiben *et al.*, 1999], [Clerc & Kennedy, 2002], [Trelea, 2003b], and [Dorigo & Blumb, 2005].

Figure 4.7 illustrates how for example control parameters of GA algorithm can be tuned when it is used to solve an optimization problem. In the beginning, GA with an initial values of its control parameters is applied. Then the Euclidean distance between the known optimal solution of the solved problem and the solution obtained by GA is calculated. Furthermore, the convergence time of GA is computed as the necessary times that the objective and constraint functions of this optimization problem are evaluated to give a solution. This optimization process continue until, obtaining an optimal set of parameter configurations which gives a trade-off between accuracy and convergence time of GA.

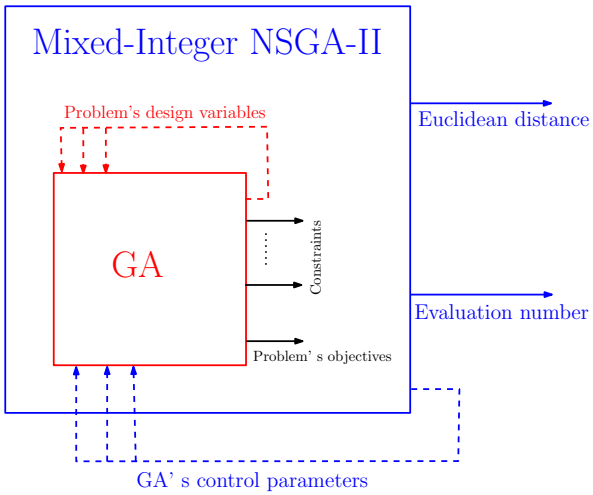


Figure 4.7: GA parameters tuning using Mixed-integer NSGA-II.

It is clear that, to ensure a successful selection of GA parameters, an optimization algorithm must be applied (Figure 4.7). For this purpose, this thesis proposed a mixed-integer and multi-objective optimization method called mixed-integer NSGA-II [Moussouni *et al.*, 2007] which is presented and studied in chapter 7.2.

Indeed, the control parameters can be:

- continuous like crossover and mutation probability for GA, pheromone persistence for ACO, and inertia factor for PSO;
- integer as the population size (GA), number of ants (ACO), and number of particles (PSO);
- and not countable like types of selection, crossover, and mutation operators.

Then optimization method to be applied to tune these algorithms parameters must be a mixed-integer algorithm. Furthermore, the three algorithms (GA, ACO, and PSO) are evaluated in term of

1. the Euclidean distance between the known optimal solution and the returned solution,

2. and also on the number of evaluations.

In this study, the control parameters of the optimization algorithms (GA, ACO, and PSO) are optimally tuned on the optimal design of a brushless DC wheel motor that is known as a non-linear multi-modal benchmark.

For clarity, this benchmark is presented briefly described in section 5.1. One who wants more details refer to chapter 5. Note that all materials can be downloaded at `http://l2ep.univ-lille1.fr/come/benchmark-wheel-motor.htm`.

Compared to ACO and PSO, GA have more parameters to be tuned. Then, first the tuning of GA parameters is studied. Thereafter, the tuning of ACO and PSO parameters is compared with GA's parameters tuning in term of accuracy of the obtained solutions and computation time.

4.2.1 Optimization test case

To ensure the optimal tuning of GA, ACO, and PSO control parameters a test case is used. This benchmark consists on an analytical model for the design of a brushless DC wheel motor [Brisset & Brochet, 2005], where the aim is to design a motor with the best efficiency η. In this study, only five design variables and six constraints are considered. Figure 5.2 describes the optimization problem using our graphical convention.

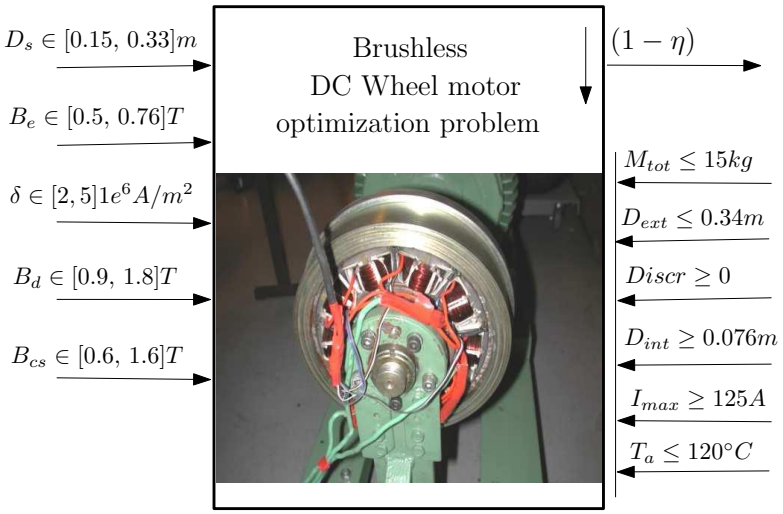


Figure 4.8: Optimization problem.

4.2.2 GA: Optimal tuning of parameters

In this study only six GA parameters are tuned:
two continuous parameters:

- the rate of the elite individuals (elit),
- and the probability of crossover (pc).

and four discrete parameters that are:

- the population size (npop),
- the fitness scaling type (fit),
- the selection scheme (sel),
- and the type of the crossover (cros).

In Matlab Genetic Algorithms and Direct Search Toolbox, the GA parameters are classified such as (Table 4.5):

- reproduction (elit, pc, npop),
- fitness scaling (fit),
- selection (sel),
- and crossover (cros).

Table 4.5: Genetic algorithms parameters and values.

Parameters	Values
- Elite individual rate (elit)	[0.05, 0.4]
- Crossover fraction (pc)	[0.1, 0.95]
- Population size (npop)	20, 50, 100, 160
- Fitness scaling (fit)	Rank (rank), Proportional (prop), Top (top), and Shift linear (shift)
- Selection operator (sel)	Stochastic uniform (stoc), Remainder (rem), Roulette (rou), and Tournament (tour)
- Crossover operator (cros)	Scattered (scat), Intermediate (int), Heuristic (heu), and Arithmetic (ari)

The fact that the tuning of GA parameters is a bi-objective optimization problem, there are a set of optimal parameter configurations distributed over a Pareto front. Figure 4.9 shows the Pareto front found by the Mixed-integer NSGA-II. As GA is a stochastic method, the optimization process is performed 20 times, then the Pareto fronts built with the average and better values of the two objectives are displayed (Figure 4.9).

The star and plus signs respectively represent the best and the average solutions found by the mixed-integer NSGA-II using a real code. Similarly, the triangles and the points respectively represent the best and the average solutions found using a mixed code.

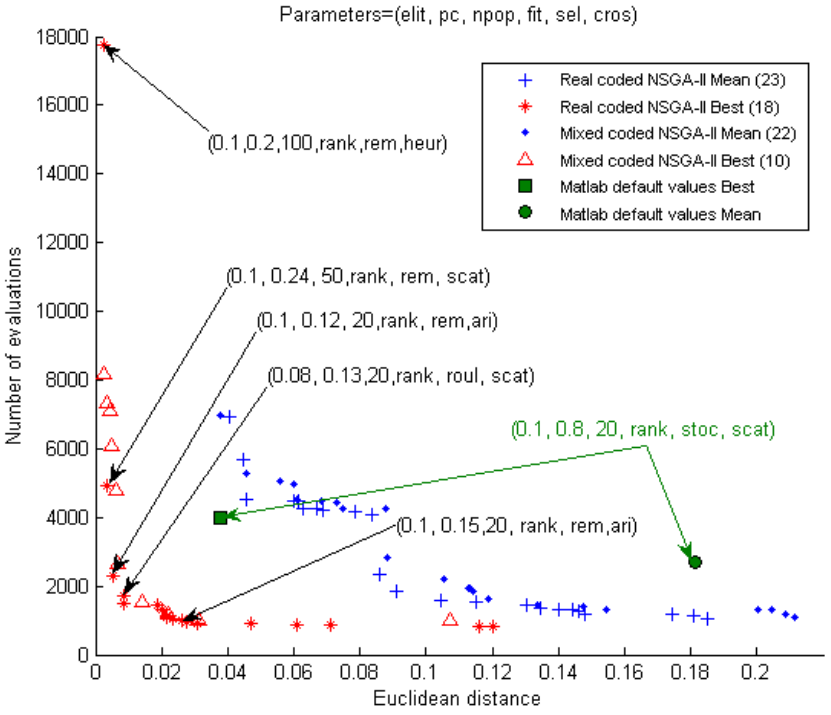


Figure 4.9: GA parameters tuning.

The square and the ring give respectively the best and the average solutions using the Matlab default values of the GA parameters. Some interesting points are highlighted. The optimal values of the GA parameters founded by Mixed-integer NSGA-II are presented in Table 5.5. Note that the parameter values shown in this table correspond to a set of Pareto points with a lower Euclidean distance.

Table 4.6: Optimal parameter's values for genetic algorithms.

Parameters	Optimal values	Matlab default values
- Elite individual rate (elit)	0.1	0.1
- Crossover fraction (pc)	less than 0.5	0.8
- Population size (npop)	20 - 100	20
- Fitness scaling (fit)	Rank	Rank
- Selection operator (sel)	Remainder (rem), Roulette (rou)	Stochastic uniform (stoc)
- Crossover operator (cros)	indifferent	Scattered (scat)

To summarize, a population size equal of 20 to 100, crossover fraction value less than 0.5, rank fitness scaling, and remainder or roulette selection operators seem to be a successful selection of GA parameters for the brushless DC wheel motor optimization problem.

In the following the tuning of GA's parameters is compared with ACO and PSO parameters tuning in term of accuracy of the obtained solutions and computation time.

4.2.3 ACO and PSO: Optimal tuning of parameters

Contrarily to GA, with ACO and PSO there are few parameters to be tuned as explained in the state of the art. Furthermore, PSO is the less sensible to the control parameters.

In order to obtain a maximum of interesting information on the optimal tuning of the parameters of both algorithms, multiple parametric configurations are applied. Thereafter, the mixed-integer NSGA-II [Moussouni *et al.*, 2007], presented in chapter 7.2, is used. Like the previous section, this proposed method evaluates both algorithms in term of the Euclidean distance between the known optimal point and the solution found, and also the number of evaluations. As for GA, the tuning of ACO and PSO parameters is applied on the design of a brushless DC wheel motor.

To draw on the experiences of the other researchers [Dorigo & Blumb, 2005]-[Trelea, 2003b] and the state of art on the swarm intelligence domain, only four parameters for each algorithm are tuned.

Table 4.7 represents the ACO parameters that are the number of ants, the two parameters α, β that allow a user control on the relative importance of trail versus visibility, and the pheromone's persistence factor ρ.

Table 4.7: Ant colony optimization parameters and values.

Parameters	Values
- Number of ants	10, 20, 45, 60
- Pheromone's persistence factor ρ	[0, 1]
- Parameter α	[0, 1]
- Parameter β	[0, 1]

Regarding to PSO the tuned parameters are number of particles, the learning factors c_2, c_3 and the inertia factor c_1 which are shown in Table 4.8.

Table 4.8: Particle swarm optimization parameters and values.

Parameters	Values
- Number of particles	10, 20, 45, 60
- Inertia factor c_1	[0.4, 1]
- Learning factor c_2	[0, 2]
- Learning factor c_3	[0, 2]

To ensure a rigorous study, the tuning of ACO and PSO parameters is compared with GA's parameters tuning in term of accuracy of the obtained solutions and computation time. As presented in the previous section, the same GA parameters are tuned: rate of the elite individuals, probability of crossover, population size, fitness scaling type, selection scheme, and type of crossover.

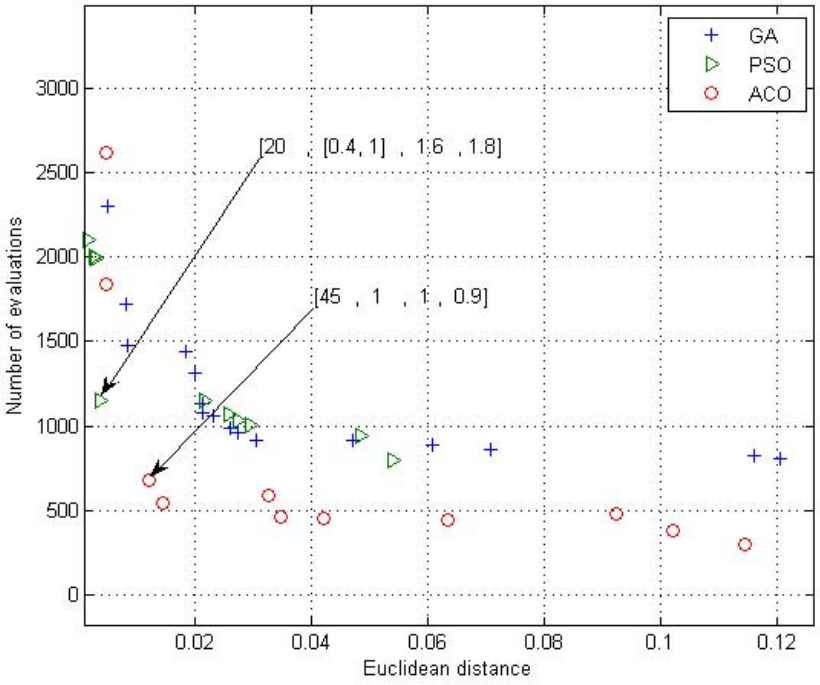


Figure 4.10: GA, ACO and PSO parameters tuning.

Figure 4.10 shows the Pareto's fronts found by the Mixed-integer NSGA-II. As PSO and ACO are stochastic methods, the optimization is performed 20 times, and then the Pareto's fronts are built with the average values of the two objectives (Figure 4.10). Some interesting points are highlighted. The optimal values of the PSO parameters founded by Mixed-integer

Table 4.9: Optimal parameter's values for PSO.

Parameters	Optimal values
- Number of particles	20
- Inertia factor c_1	[0.4, 1] decreasing in time
- Learning factor c_2	1.6
- Learning factor c_3	1.8

NSGA-II are: particle number is $N = 20$, $c_2 = 1.6$, $c_3 = 1.8$ and the inertia factor $c_1 \in [0.4, 1]$ is decreasing in time (Table 5.6).

As shown in Table 5.7 the optimal parameters of ACO are: ant number is 45, $\alpha = \beta = 1$. Pheromone's persistence factor is $\rho = 0.9$. As shown in Figure 4.10, ACO is the most efficient method whereas PSO and GA have partially the same solutions.

Note that the version of ACO algorithm applied in this research used SQP for its local search procedure. This is explain why ACO is less sensible to its control parameters. One more time, the importance of hybrid methods is highlighted.

4.2.4 Outcome

To conclude we can say that there is an important influence of optimization algorithm parameters on the accuracy of returned solutions and computation time.

Table 4.10: Optimal parameter's values for ACO.

Parameters	Optimal values
- Number of ants	45
- Pheromone's persistence factor ρ	0.9
- Parameter α	1
- Parameter β	1

Indeed, GA, ACO, and PSO parameters are tuned. The results show the efficiency of ACO compared to both algorithms. PSO is the less powerful but research about it is still ongoing. However, compared to GA and ACO, the advantages of PSO are its easy implementation and the few parameters to tune.

4.3 Conclusion

This study highlighted that the metaheuristics: genetic algorithms (GA), ant colony optimization (ACO), and particle swarm optimization (PSO) are global optimization methods which are powerful if they are well managed especially when its parameters are well tuned.

As will be shown in chapter 5, one way to overcome the drawback of the parameters tuning and the huge convergence time of the metaheuristics consists in their combination with local search methods like SQP.

Using the optimal tuning of control parameters provided in this study - as initial points distribution over the design space, and the setting of other parameters - in the next chapter, all algorithms (GA, ACO, PSO, SQP, and hybrid method) are compared on the mono-objective optimization design of a brushless DC wheel motor.

5

Mono-objective optimization algorithms

This chapter intends to give practical information on mono-objective optimization algorithms to solve a multi-modal problem. Five popular algorithms: sequential quadratic programming (SQP), genetic algorithms (GA), ant colony optimization (ACO), particle swarm optimization (PSO), and hybrid method combining GA with SQP are studied, tested, and compared on the mono-objective optimization of a brushless DC wheel motor that is known as a non-linear multi-modal benchmark.

First, the mono-objective optimization design problem is described briefly, where some questions about the model are tackled:

1. What is the best way to solve the implicit equations included in the model?

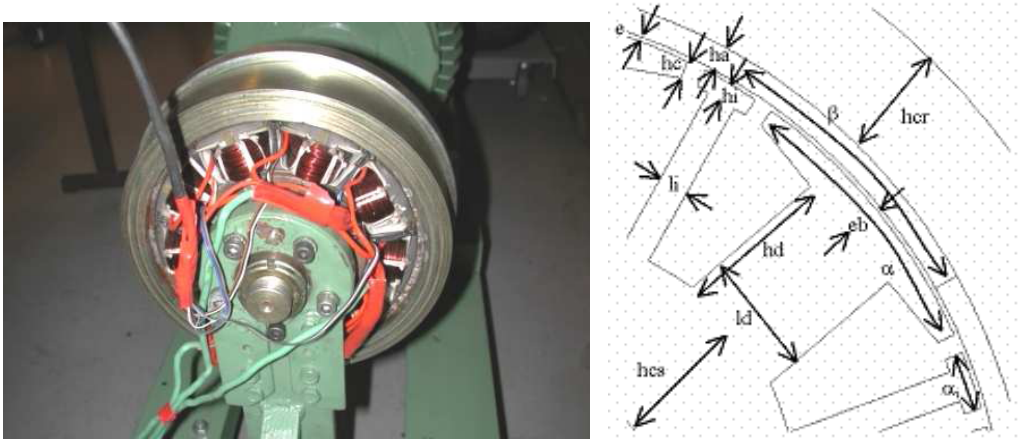
2. What is the best way to calculate the derivative of the objective function?

3. How the constraints are taken into account by each applied algorithm.

5.1 Mono-objective optimization design of a brushless DC wheel motor

In [Brisset & Brochet, 2005] an original motor is proposed as a benchmark to compare optimization methods. As it is sufficiently detailed to be adapted to different design contexts, this benchmark is considered as a reference for the optimal design of electrical machines optimization problems.

It consists on an analytical model for the design of a brushless DC wheel motor, where the aim is to design a motor with the best efficiency η. In this study, only five design variables are considered. This analytical model has been validated via prototype measures (Figure 5.1). As it is shown in [Brisset & Brochet, 2005] and [Moussouni *et al.*, 2007] this constraint optimization problem is multi-modal which makes optimization process very difficult. All useful materials to deal with this benchmark can be found in [Brisset & Brochet, 2005]. Furthermore, all materials can be downloaded at `http://l2ep.univ-lille1.fr/come/benchmark-wheel-motor.htm`.



(a) (b)

Figure 5.1: Prototype of the DC wheel motor.

5.1.1 Optimization problem formulation

The optimization problem was defined in [Brisset & Brochet, 2005] as follow:

$$\begin{cases} \min\,(1-\eta) \\ \quad s.t. \quad M_{tot} \leqslant 15\,kg,\ D_{ext} \leqslant 340\,mm,\ D_{int} \geqslant 76\,mm, \\ \qquad I_{max} \geqslant 125\,A,\ disc(D_s,\,\delta,\,B_d,\,B_e) \geqslant 0,\ T_a \leqslant 120°C \\ \quad with \quad 150\,mm \leqslant D_s \leqslant 330\ mm,\ 0.9\,T \leqslant B_d \leqslant 1.8\,T \\ \qquad 2.0\,A/mm^2 \leqslant \delta \leqslant 5.0\,A/mm^2,\ 0.5\,T \leqslant B_e \leqslant 0.76\,T \\ \qquad 0.6\,T \leqslant B_{cs} \leqslant 1.6\,T \end{cases} \tag{5.1}$$

where η, M_{tot}, D_{ext}, D_{int}, I_{max}, T_a are results of the analytical model and the determinant $disc(D_s,\,\delta,\,B_d,\,B_e)$ is used for the calculation of the slot height [Brisset & Brochet, 2005]. D_s, δ, B_d, B_e, B_{cs} are the five design variables presented in Table 5.1.

Table 5.1: List of design variables.

Number	Design variable	Unit	Symbol
1	Stator diameter	mm	D_s
2	Current density in the conductors	A/mm^2	δ
3	Maximum magnetic induction in the teeth	T	B_d
4	Maximum magnetic induction in the air gap	T	B_e
5	Maximum magnetic induction in the stator back iron	T	B_{cs}

This optimization problem presents 6 inequality constraints given in Table 5.2.

Table 5.2: List of constraints.

	Inequality constraint	Unit	Symbol
1	Total mass of the active parts	Kg	M_{tot}
2	Maximum phase current	A	I_{max}
3	Inner diameters	mm	D_{int}
4	Outer diameters	mm	D_{ext}
5	Temperature of the magnets	A	T_a
6	Determinant used to calculate the slot height	-	$disc(D_s,\,\delta,\,B_d,\,B_e)$

To summary, the optimization design problem of the brushless DC wheel motor is represented graphically in Figure 5.2. The left side shows the 5 design variables. The objective and the 6 constraints are listed in the right side.

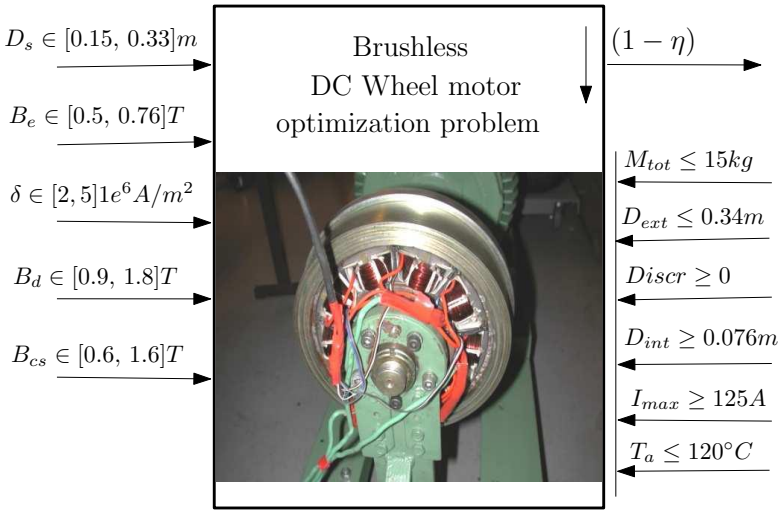
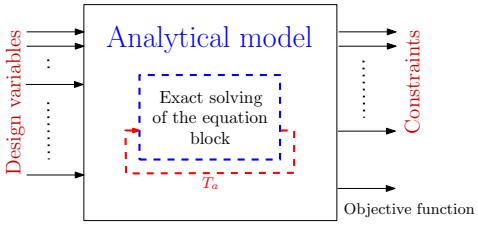


Figure 5.2: Mono-objective optimization of a brushless DC wheel motor.

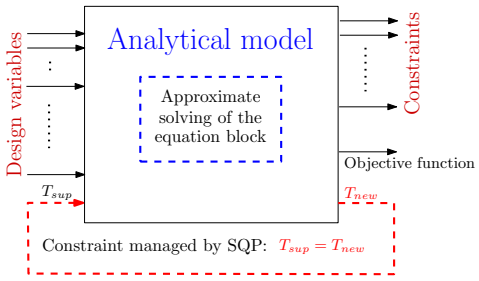
In the following, by mean of this analytical test case, SQP is improved in two points. The first one concerns the resolution of the implicit system of 7 non-linear equations (Section 5.1.2). The second concerns how better the computation of the objective function derivatives can be done (Section 5.1.3).

5.1.2 Implicit equation solving

The analytical model used to design the brushless DC wheel motor [Brisset & Brochet, 2005] includes 78 equations. There is a block of 7 non-linear equations that must be solved simultaneously to calculate the thermal model. Three approaches are proposed to solve this block. In the first one, the temperature is updated using the fixed point technique and the 6 other variables are supposed to converge in the same time. In the second one, Powell's dogleg or Levenberg-Marquet methods [Powell, 1970] is used to solve accurately the block. Both approaches use an inner loop (Figure 5.3(a)).



(a) Exact solving of the equation block.



(b) Approximate solving of the equation block.

Figure 5.3: Solving of the block: inner loop (a), and outer loop (b).

The third method uses an outer loop (Figure 5.3(b)). A supposed temperature T_{sup} is added to the design variables and used to compute approximately the implicit equations. This gives a new value of the temperature T_{new}. A constraint is added to the optimization

problem and imposes both temperatures to be equal.

$$T_{sup} - T_{new} = 0 \tag{5.2}$$

Therefore, the accuracy of block solving increases at each SQP iteration.

Table 5.3: Comparison between internal and external loops.

	internal loop		external loop
	fixed point	dogleg	
convergence (%)	23.7	21.31	49.9
average time (s)	0.106	10.46	0.0823

Table 5.4 compares the percentage of convergence to the known optimum point with a Euclidean distance lower than 10^{-4} and the average time of SQP for the three approaches starting from one thousand initial points. In this comparison all design variables are normalized to $[0, 1]$.

It appears that the external loop approach is better because it gives the greater percentage of convergence and lower average computational time. However, adding new constraints restricts the search space. Furthermore, using the external loop is not always possible. The second approach is the accurate manner to solve the implicit block, but it is time consuming.

In the following the fixed point technique, which gives in an acceptable time a good convergence, is used.

5.1.3 Computation of derivatives

SQP requires the first order derivatives of objective function that can be computed in three ways.

1. The first one is a formal calculation using Pro@Design, Maple or Matlab Symbolic Math Toolbox.

2. The second way is to compute the derivatives with the finite difference technique and a fixed step.

3. The third uses an adaptive step [Fletcher & Powell, 1963].

One hundred initial points are chosen randomly to give the percentage of convergence to the known optimum point. Table 5.4 shows that formal calculation and adaptive step give good results but the adaptive step is faster and easier to implement.

Table 5.4: Percentage of convergence.

precision	formal calculation	fixed step	adaptive step
10^{-4}	37	0	41
10^{-1}	80	5	51

In the following, finite difference technique with an adaptive step is used.

5.2 Constraint handling

As presented in the previous section, the brushless DC wheel motor is a non-linear and non-convex optimization problem with six inequality constraints. Then, the optimization methods

that will be compared on the optimization design of this benchmark must be able to deal with these constraints.

The chosen algorithms to be tested on this optimization problem are:

- Two Matlab's algorithms: SQP, and GA.
- Three other algorithms implemented specially for this research: ACO, PSO, and an hybrid method combining GA and SQP.

In the following, the way each algorithm solves constrained optimization problem is discussed.

5.2.1 Use of SQP

SQP algorithm replaces the objective function $f(X)$ with its quadratic approximation, and replaces the l equality and m inequality constraints, respectively $h_k(X_n)$ and $g_j(X_n)$, by its linear approximations. Then, at each iteration ΔX is calculated by solving the following quadratic problem:

$$\begin{array}{ll} \min & QP(\Delta X) = \frac{1}{2}(\Delta X)^T L(X_n, \lambda_n)\Delta X + \nabla f^T(X_n)\Delta X \\ \text{subject to} & \nabla h_i^T(X_n)\triangle X + h_i(X_n) = 0,\ i = 1, 2, \ldots, l \\ & \nabla g_j^T(X_n)\triangle X + g_j(X_n) = 0,\ j = 1, 2, \ldots, m \end{array} \tag{5.3}$$

where $L(X_n, \lambda_n)$ is the Lagrange function and λ_n are the Lagrange multipliers.

5.2.2 Use of GA

MAtlab' GA is based on the Augmented Lagrangian Genetic Algorithms (ALGA) to solve such as nonlinear constrained problems. By using the Lagrangian and the penalty parameters, the objective function and nonlinear constraints are combined to formulate the following subproblem:

$$\begin{aligned} \min\ \Theta(X, \lambda, s, \rho) = f(X) - \sum_{i=1}^{m} \lambda_i s_i \log(s_i - g_i(X)) + \\ \sum_{i=1}^{l} \lambda_i h_i(X) + \frac{\rho}{2}\sum_{i=1}^{l} h_i(X)^2 \end{aligned} \tag{5.4}$$

where the Lagrangian multiplier estimates λ_i must be zero or positive. ρ is the positive penalty factor and s_i are the non-negative shifts.

5.2.3 Use of ACO and PSO

ACO and PSO algorithms are developed and implemented in this research. Concerning ant colony optimization, an improvement version called GACO (Generalized ant colony optimization) [Hou *et al.*, 2002] is implemented.

As optimization of the Brushless DC wheel motor is a non-linear constrained problem, the external penalty approach is implemented and added in the optimization process of both algorithms.

The objective function and non-linear constraints are combined to form the following sub-problem:

$$\begin{aligned} \min\ \Theta(X,\lambda) = f(X) + \lambda_1 \cdot \sum_{i=1}^{l} (0,\ h_i(X))^2 + \\ \lambda \cdot \sum_{j=1}^{m} \max^2 [0,\ g_i(X)] \end{aligned} \tag{5.5}$$

where $f(X) = (1 - \eta(X))$, is the original objective function to be optimized, $g_i(X)$ are the $m = 6$ inequality constraints.

The parameter λ is a penalty factor updated at each generation k as follow:

$$\begin{cases} \lambda_i(k) = \left(\dfrac{2}{1 + e^{-ak/T}} - 1 \right) \cdot \lambda_0 \\ i = 1, 2 \end{cases} \tag{5.6}$$

where a is a positive parameter set to 10. λ_0 is the upper limit of $\lambda(k)$, given by the users and must be positive and very large. In this study $\lambda_0 = 1000$.

Furthermore, GACO algorithm, with a certain probability of neighbourhood search, carries out the search by the local algorithm SQP. Then, GACO can be considered as an hybrid method.

5.2.4 Use of hybrid algorithm

With hybrid algorithm the Augmented Lagrangian Genetic Algorithms (ALGA) is applied first. Then the constraints are solved as explained before. Thereafter, the best solution obtained by ALGA is used by SQP as an initial point to converge to the solution.

5.3 Optimization results comparison

Following the previous study, the best values of GA parameters that will be used to solve this benchmark are presented in Table 5.5.

Table 5.5: GA parameters: selected values.

Parameters	Values
- Elite individual rate	10%
- Crossover fraction	0.5
- Population size	20
- Fitness scaling	Rank
- Selection operator	Roulette
- Crossover operator	Scattered

Beside the use of a complete neighbourhood, PSO parameters are chosen as illustrated in Table 5.6:

For ACO, the parameters have the following values:

Table 5.6: PSO parameters: selected values.

Parameters	Values
- Particle number	20
- Inertia factor c_1	$[0.4, 1]$ and decreasing in time
- Learning factor c_2	1.6
- Learning factor c_3	1.8

Table 5.7: ACO parameters: selected values.

Parameters	Values
- Ant number	45
- Parameter α	1
- Parameter β	1
- Pheromone's persistence factor ρ	0.9

The three algorithms (GA, ACO, and PSO) stop when the maximum number of iterations or a minimum error requirement is reached.

Concerning SQP:

- the derivative of the objective function is calculated using the finite difference technique with an adaptive step;
- and the fixed point technique is applied to solve the block of the 7 implicit equations of the model.

Furthermore, as SQP is a starting point dependent algorithm the multi-start process is applied. In other term, SQP is restarted with several initial points. The best obtained solution is kept and an average convergence time is calculated.

It can be seen in Table 5.8 that all methods produce a numerical solution which well approximates the best known solution.

As established in the previous study, applying a multi-start approach with SQP is very useful in order to improve its performance in the multi-modal optimization problem. Indeed, the SQP multi-start method obtains an accurate solution in a smaller number of objective function evaluations. Unlike SQP, GA, ACO, PSO and hybrid algorithm always find the global solution at the first run. Furthermore, comparing to GA and PSO, hybrid method and ACO obtain more accurate solutions in a smaller number of objective function evaluations. Indeed, with hybrid method GA, as a global algorithm, explores the search space and a good approximation of the solution is obtained. Thereafter, SQP starts from this point and improves its quality in term of accuracy and time of convergence.

The version of ACO implemented in this study includes, with a certain probability, a local search that is carried out by SQP. Then, ACO can be considered as an hybrid method and this explain its high performance comparing to GA and PSO launched alone.

Table 5.8: Results of the mono-objective optimization.

Symbol (unit)	GA	PSO	ACO	SQP	GA & SQP
M_{tot} (kg)	15	15.0011	15.00	15.00	15.00
I_{max} (A)	125.0128	124.9996	125.00	125.00	125.00
D_{int} (mm)	76.9	79.2	76.0	76.0	76
D_{ext} (mm)	239.2	239.8	238.9	238.9	238.9
T_a (°C)	95.2077	94.9793	95.3464	95.3464	95.3064
D_s (mm)	201.5	202.1	201.2	201.2	201.2
B_e (T)	0.6480	0.6476	0.6481	0.6481	0.6481
δ (A/mm^2)	2.0602	2.0417	2.0437	2.0437	2.0615
B_d (T)	1.7991	1.8	1.8	1.8	1.8
B_{cs} (T)	0.8817	0.9298	0.8959	0.8959	0.8700
η (%)	95.3112	95.32	95.32	95.32	95.31237
Evaluation number	3380	1600	1200	90	1644

5.4 Conclusion

In general, the local optimization techniques like SQP have difficulties in dealing with global optimization problems. One of the main reasons of their failure is that they can easily be trapped in local minima contrarely to the most metaheuristics such as GA, ACO, and PSO.

However, these metaheuristics are computationally time consuming due to their slow convergence. One of the main reasons for their slow convergence is that they may fail to detect promising search directions especially in the vicinity of local minima due to their random constructions.

To overcome the drawbacks of the slow convergence and random constructions, it advice to combine these meta-heuristics with local search methods. In these hybrid methods, local search strategies are inlaid inside metaheuristics in order to guide them especially in the vicinity of local minima, and overcome their slow convergence especially in the final stage of the search.

As it is highlighted in this study, an other alternative to solve multi-modal optimization problems consists in the use of the multi-start technique with SQP. Furthermore, using SQP, it is not necessary to provide a formal calculation of the derivatives because the finite difference technique with an adaptive step gives better optimization results.

6

Multi-objective optimization algorithms

The optimal design of electromagnetic devices is a complex and a complicated task. A way to formulate the problem is to find the trade-off between conflicting goals. Solving this problem requires building the Pareto optimal set with accuracy. With two objectives, the Pareto optimal set can be easily drawn and helps the designer to find a good solution. Many methods are able to find the solutions of a bi-objective optimization: scalar methods such as the well-known weighted sum (WS), ϵ-constraint methods, etc.; and the stochastic methods.

According to the state of the art, the weighted sum (WS) method is still and may always remain the most frequently used approach. The most popular stochastic multi-objective algorithms are the modified Non-dominated Sorting Genetic Algorithm (NSGA-II), and the modified Strength Pareto Evolutionary Algorithm (SPEA2). Both methods use elitism approach because it has proved to be the most efficient.

In this chapter, NSGA-II, SPEA2, and WS technique using SQP are compared on the multi-objective design of the brushless DC wheel motor.

Before comparison, first, the constraint handling and the performance assessment of stochastic optimizers are presented. Thereafter, the multi-objective constrained optimization problem is presented. How do the multi-objective algorithms deal with the constraints is discussed.

After that, a new method called adaptive weighted sum of objectives (AWS) is presented and modified in the aim to overcome to the drawback

1. of WS algorithm when dealing with non-convex and discrete Pareto fronts,

2. of the time consuming of the stochastic algorithm NSGA-II.

In electrical domain, the most optimization design problems use Finite Element models like the 3D FE magneto-thermal models which are known as time consuming.

Unfortunately, the mentioned methods require a very high number of model evaluations, especially the stochastic ones: NSGA-II and SPEA2. The application of these optimization algorithms alone is not compatible with the use of FE models.

Then, a new algorithm is proposed in the aim to handle efficiently multi-objective optimization problems using 3D Finite element models. This algorithm consists on the combination of a multi-objective optimization algorithm with a technique called Space-Mapping (SM). The SM approach, presented in section 6.5.2.0, aims to reduce the computational time when dealing with fine models.

6.1 Constraint handling

Evolutionary algorithms, and thus genetic algorithms (GA) are essentially unconstrained optimization methods. Hence, the way that the constraints are handled in the multi-objective GA (MOGA) is very important.

One way to handle constraints in multi-objective evolutionary algorithms is to add a penalty to the objective functions of the infeasible individuals, as in mono-objective optimization. The other way consists in considering the constraints as additional objectives [Coelho, 2004] and [Michalewicz, 1995]. As we can use the stochastic ranking proposed by [Runarsson & Xin, 200] which aims to balance objective and penalty functions stochastically. This technique is sensible to the use of a probability parameter.

The popular way consist on the use of the Pareto dominance relation. Indeed, the comparison of some multi-objective-based techniques to handle constraints in genetic algorithms did by [Mezura-Montes & Coello, 2002] reveals that the selection criterion of Pareto dominance gives better results.

The constraints are then handled by defining a constrained non-domination relation order. A design X_j of fitness f_j is said to be constraint non-dominated by another design X_i of fitness f_i if:

1. X_j is feasible, and X_i is not.
2. X_j and X_i are both infeasible but X_j has a smaller overall constraint violation.
3. X_j and X_i are both feasible and $X_j \succ X_i$ which means X_j dominates X_i.

Thereafter, [Kurpati *et al.*, 2002] proposed four constraint handling improvements for multi-objective genetic algorithms (MOGA). The main improvement consists on the use of the so called **Constraint-First-Objective-Next model**. The constraints handling process is as follow:

- **Step 1.** Evaluate the constraints for every individual.
- **Step 2.** Identify feasible and infeasible individuals in the current population.
- **Step 3.** Assign a moderate rank to all feasible individuals (i.e. $r = 0.5 \times N$).
- **Step 4.** Evaluate the objective function for all feasible individuals.
- **Step 5.** Identify the non-dominated individuals amongst feasible individuals.
- **Step 6.** Assign a low (i.e. good) rank to feasible non-dominated individuals (i.e. $r = 1$).
- **Step 7.** Obtain fitness values of all infeasible individuals X_i using (6.1)

$$\begin{aligned} fitness(X_i) = \; & [C_{max} - (C_{max} - C_{min})\,(r-1)\,/\,(N-1)] \\ & - [w_1 \times factor_1(X_i) + w_2 \times factor_2(X_i)] \end{aligned} \tag{6.1}$$

where

$$\begin{cases} factor_1(X_i) = CF_1 \times \\ \left[\dfrac{\left[\sum_{j=1}^{m} \max(g_j(X_i), 0) + \sum_{p}^{l} |h_p(X_i)|\right]}{\left(\left[\sum_{i'=1}^{N}\sum_{j=1}^{m} \max(g_j(X_{i'}), 0) + \sum_{i'=1}^{N}\sum_{p=1}^{l} |h_p(X_{i'})|\right] / N\right)} \right] \\ \\ factor_2(X_i) = CF_2 \; \times \left[\left(\displaystyle\sum_{j=1}^{m} \delta_j(X_i) + \sum_{p=1}^{l} \delta_p(X_i) \right) / (m+l) \right] \end{cases} \tag{6.2}$$

where $r = 0.8 \times N$, $\delta_j(X_i)$ (or $\delta_p(X_i)$) $= 1$ if the constraint $g_j(X_i)$ (or $h_p(X_i)$) is violated for the i^{th} individual X_i. Otherwise $\delta_j(X_i)$ (or $\delta_p(X_i)$) $= 0$. The quantities CF_1 and CF_2 are correction factors whose values are defined in the interval [0.0005, 0.015].

Note that, $C_{max} = 1.2$, $C_{min} = 0.8$, N is the population size, and r is the rank of the individual. The quantities w_1 and w_2 are weighting factors determined with the following procedure:

1. Compute $avgfactor_1$, the average of $factor_1$ for all infeasible individuals in the population.
2. Compute $avgfactor_2$, the average of $factor_2$ for all infeasible individuals in the population.
3. For each individual in the population, compare $factor_1$ with $avgfactor_1$ and $factor_2$ with $avgfactor_2$:
   - **Case 1.** If $(factor_1 > avgfactor_1)$ and $(factor_2 < avgfactor_2)$ then $(factor_1 \succ factor_2)$. Set $w_1 = 0.75$ and $w_2 = 0.25$.
   - **Case 2.** If $(factor_1 < avgfactor_1)$ and $(factor_2 < avgfactor_2)$ then $(factor_1 \prec factor_2)$. Set $w_1 = 0.25$ and $w_2 = 0.75$.
   - **Case 3.** For all other cases, both $factor_1$ and $factor_2$ are considered equally important. Set $w_1 = 0.50$ and $w_2 = 0.50$.

where $a \succ b$ means that a dominates b.

6.2 Performance assessment

As is presented before in this thesis, the mono-objective optimization algorithms are compared in term of the quality of the obtained solutions, where the quality means:

1. the convergence accuracy: the smaller the value, the better the solution,
2. and the computational time (can be also expressed as the number of the fitness evaluations) needed to achieve this accuracy.

Concerning the second aspect, there is no difference between mono - and multi-objective optimization [Zitzler *et al.*, 2004]. However, it gets even more complicated when two solutions are compared in the presence of several objective functions. Indeed, in the case of two solutions that are incomparable, i.e. neither dominates the other, their comparison is a complicated task.

The main issue is to answer what quality means in the multi-objective optimization? Is it the convergence to the Pareto optimal front? the coverage of the objective space? or other criteria [Zitzler *et al.*, 2004]?

In principle, to assess the performance of multi-objective optimizers, two basic approaches exist in the literature [Fonseca & Fleming, 1996], [Zitzler *et al.*, 2002], [Deb *et al.*, 2002], [Zitzler *et al.*, 2004], [Joshua *et al.*, 2006], and [Erbas *et al.*, 2006]:

1. the attainment function approach,
2. and the indicator approach.

Most popular are methods that assign each Pareto solutions set a vector of real numbers that reflect different aspects of the quality as shown in Figures 6.1, 6.2, 6.3, and 6.4.

To assess just one criteria such as:

- convergence to the Pareto-optimal front,
- having an uniform distribution of the Pareto front,
- and having a better coverage of the objective space,

many quality indicators (metrics) have been proposed [Deb *et al.*, 2002], [Erbas *et al.*, 2006]. We can list:

- *D-metric* is used to assess the convergence to the Pareto-optimal front (considered as a reference set) (Figure 6.1). It is calculated as:

$$D(\overline{Z}_1, \overline{Z}_2) = \sum_{\overline{z}_1 \in \overline{Z}_1} \frac{\|\overline{z}_1 - \overline{z}_2\|_{max}}{\sqrt{M}|\overline{Z}_1|}$$

  where $\overline{Z}_1$, and $\overline{Z}_2$ are normalized non-dominated sets. $\overline{z}_1 \in \overline{Z}_1$, and $\overline{z}_2 \in \overline{Z}_2$ are two normalized non-dominated objective vectors such as $\overline{z}_2 \prec \overline{z}_1$. $\|\cdot\|_{max}$ is the maximum of the Euclidean distances, $|\cdot|$ represents the cardinality of a finite set, and M is the number of the objective functions (or the dimension of the objective space).

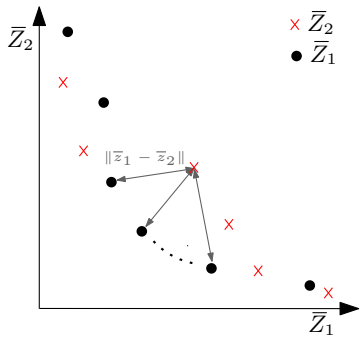


Figure 6.1: Convergence quality: *D-metric*.

- Δ-*metric* is used to assess the performance of multi-objective algorithms in term of the uniformity propriety (Figure 6.2). It is calculated as:

$$\Delta(\overline{Z}) = \sum_{i=1}^{|\overline{Z}|-1} \frac{|d_i - \overline{d}|}{\sqrt{M}\left(|\overline{Z}| - 1\right)}$$

  where $d_i,\ i = 1, \cdots, \left(|\overline{Z}| - 1\right)$ is the Euclidean distance between two consecutive vectors, and $\overline{d} = \dfrac{\sum_i^{|\overline{Z}|-1} d_i}{|\overline{Z}| - 1}$ is the average distance. The quantities $|d_i - \overline{d}|$ represent the absolute deviations from the average distance.

- ∇-*metric* is used to compare multi-objective optimizers in term of extent criteria (Figure 6.3). It is calculated as:

$$\nabla(Z) = \prod_{i=1}^{M} \mid z_k^{min} - z_k^{max} \mid$$

  where $z_k^{min} = \min\{z_k \in Z_k\}$, $z_k^{max} = \max\{z_k \in Z_k\}$, $k = 1, \cdots, M$ and M is the number of the objective functions.

- The contribution metric evaluates the proportion of Pareto solutions given by each front (see Figure 6.4(a)).

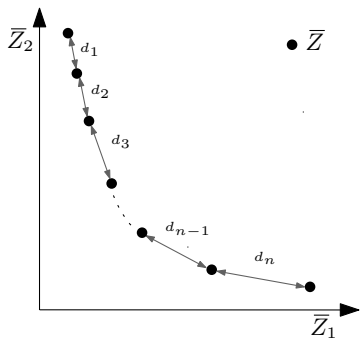


Figure 6.2: Diversity quality: Δ-*metric*.

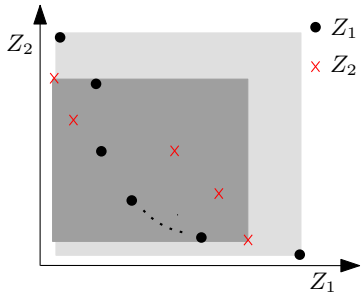


Figure 6.3: Extent quality: ∇-*metric*.

- The coverage metric, as shown in Figure 6.4(b) evaluates the dominated area given by each front.

In the following, the discussed concepts are used and tested on the multi-objective optimization design of the brushless DC wheel motor.

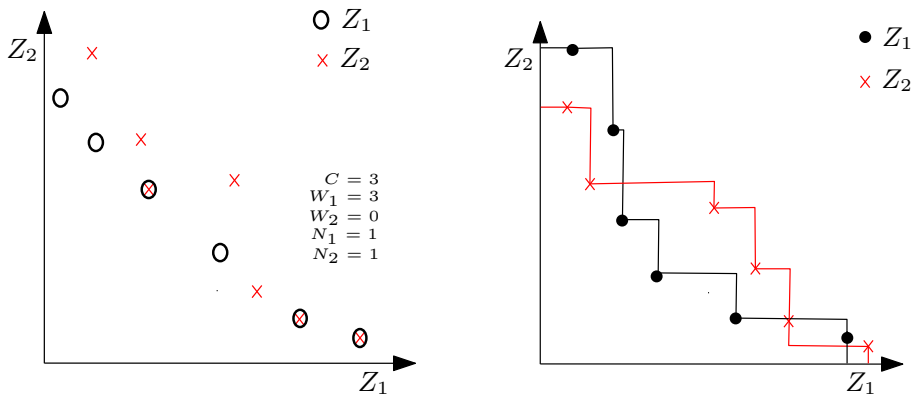
6.3 Multi-objective optimal design of a brushless DC wheel motor

In chapter 5 the brushless DC wheel motor benchmark is used as a mono-objective optimization problem for the comparison of the optimization algorithms: SQP, GA, ACO, and PSO. In this chapter, it is taken as a multi-objective problem by minimizing the losses $(1-\eta)$ and the total mass of the active parts M_{tot} which was a constraint in the mono-objective optimization problem described in section 5.1.1.

6.3.1 Optimization problem formulation

Beside the minimization of the losses $(1-\eta)$, and by considering the total mass M_{tot} as the second objective to be minimized, the multi-objective optimization deign problem of the brushless DC wheel motor is formulated as follow:

$$\begin{cases} \min\ M_{tot} \\ \min\ (1-\eta) \\ \text{with respect to}\ \ 150\,mm \leqslant D_s \leqslant 330\ mm,\ 0.9\,T \leqslant B_d \leqslant 1.8\,T \\ \qquad\qquad\qquad 2.0\,A/mm^2 \leqslant \delta \leqslant 5.0\,A/mm^2,\ 0.5\,T \leqslant B_e \leqslant 0.76\,T \\ \qquad\qquad\qquad 0.6\,T \leqslant B_{cs} \leqslant 1.6\,T \\ \text{subject to}\qquad D_{ext} \leqslant 340\,mm,\ D_{int} \geqslant 76\,mm, \\ \qquad\qquad\qquad I_{max} \geqslant 125\,A,\ disc(D_s,\,\delta,\,B_d,\,B_e) \geqslant 0,\ T_a \leqslant 120°C \end{cases} \tag{6.3}$$



(a) Contribution metric. (b) Coverage metric.

Figure 6.4: Contribution to the Pareto front and the coverage quality.

where D_s, δ, B_d, B_e, B_{cs} are the five design variables, which are the same considered in the case of the mono-objective design optimization presented in chapter 5. Table 6.1 reminds readers of these design variables.

The two objectives η and M_{tot}, and the inequality constraints D_{ext}, D_{int}, I_{max}, T_a are results of an analytical model. The determinant $disc(D_s, \delta, B_d, B_e)$ is used in order to calculate the slot height [Brisset & Brochet, 2005].

Table 6.1: List of design variables.

Number	Design variable	Unit	Symbol
1	Stator diameter	mm	D_s
2	Current density in the conductors	A/mm^2	δ
3	Maximum magnetic induction in the teeth	T	B_d
4	Maximum magnetic induction in the air gap	T	B_e
5	Maximum magnetic induction in the stator back iron	T	B_{cs}

Contrarily to the mono-objective optimization case, the multi-objective optimization problem of the brushless DC wheel motor presents 5 inequality constraints given in Table 6.2.

Table 6.2: List of constraints.

	Inequality constraint	Unit	Symbol
1	Maximum phase current	A	I_{max}
2	Inner diameters	mm	D_{int}
3	Outer diameters	mm	D_{ext}
4	Temperature of the magnets	A	T_a
5	Determinant used to calculate the slot height	-	$disc(D_s, \delta, B_d, B_e)$

To summarize, the multi-objective optimization design problem of the brushless DC wheel motor is represented graphically in Figure 6.5. The left side shows the 5 design variables. Both objectives and the 5 constraints are listed in the right side.

In the aim to test the constraint non-dominance operator and to compare several multi-objective optimization methods, the multi-objective optimization design of the brushless DC wheel motor is used as a benchmark.

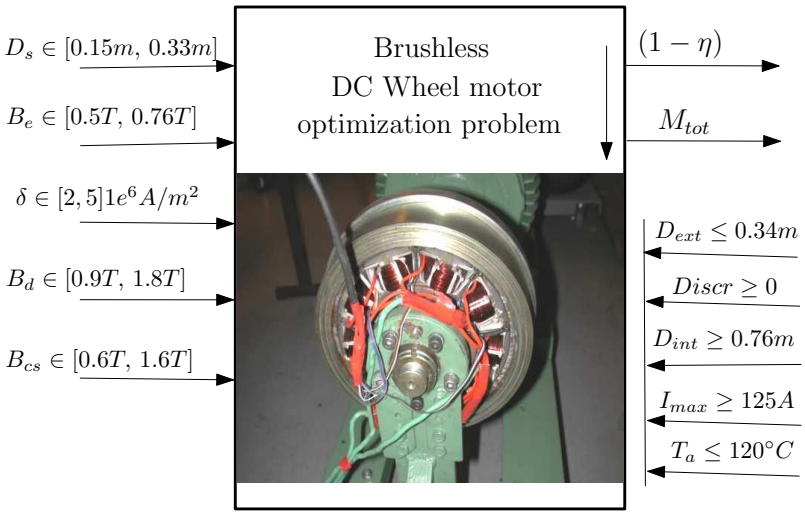


Figure 6.5: Multi-objective optimization of a brushless DC wheel motor.

In the following, this multi-objective optimization benchmark is solved using the Non-dominated Sorting Genetic Algorithm (NSGA-II), the Strength Pareto Evolutionary Algorithm (SPEA2), and the variable weighted sum of objectives with the sequential quadratic programming (WS-SQP). Thereafter, the Pareto fronts are compared in term of accuracy, uniformity, and coverage criteria using the quality indicators presented in section 6.2.

6.3.2 Optimization results comparison

This optimization problem is therefore a multi-objective constrained optimisation problem. To solve it, NSGA-II, SPEA2, and the WS-SQP algorithms have been chosen.

- For NSGA-II and SPEA2 the objectives vector is:

$$\overrightarrow{F}(X) = [M_{tot}(X),\ (1-\eta(X))]^T$$

- For WS-SQP, the two objectives are weighted with the coefficient $\alpha \in [0,1]$:

$$f(X) = \alpha M_{tot}(X) + (1-\alpha)(1-\eta(X))$$

where η and M_{tot} are respectively the efficiency and the total mass of the brushless DC wheel motor. $X = [D_s\, \delta\, B_d\, B_e\, B_{cs}]$ is a vector of the five design variables.

For both MOEA: NSGA-II and SPEA2, standard bimodal crossover (SBX) and polynomial mutation operators [Beyer & Deb, 2001] are used. For the other parameters, the following values are applied: population size $N = 100$, maximum number of generations $T = 100$, mutation probability 0.1, crossover probability 0.9, and the distribution indexes for crossover and mutation operators are $\eta_c = 20$ and $\eta_m = 20$ respectively.

Table 6.3 summarizes the selected parameters for both algorithms.

The constraints are then handled by using the constrained non-domination relation order. A motor design X_i of fitness f_i is said to be constraint non-dominated by another design $X_{i'}$ of fitness $f_{i'}$ if:

1. X_i is feasible (i.e. all the constraints listed in (6.3) are respected), and $X_{i'}$ is not.

Table 6.3: NSGA-II parameters and values.

Parameters	Values
- Mutation probability	0.1
- Crossover probability	0.9
- Population size N	100
- Fitness scaling	Normalization of the objective functions
- Selection operator	Tournament selection
- Crossover operator	Standard bimodal crossover (SBX)
- Mutation operator	Polynomial mutation
- Crossover distribution index η_c	20
- Mutation distribution index η_m	20
- Number of generations	$T = 100$

2. X_i and $X_{i'}$ are both infeasible but X_i has a smaller overall inequality constraint violation which is quantified as
$$\frac{\sum_{j=1}^{5}(\delta_j(X_i))}{\sum_{i'=1}^{N}\sum_{j=1}^{5}(\delta_j(X_{i'}))}.$$
$\delta_j(X_i)) = \max(g_j(X_i), 0)$ is how much the inequality constraint $g_j(X_i)$ is violated for the individual X_i.

3. X_j and X_i are both feasible and $X_j \succ X_i$ (dominates).

Figure 6.6 presents the Pareto front obtained by the three methods. In order to compare the quality of the solutions, the metrics presented in section 6.2 are calculated.

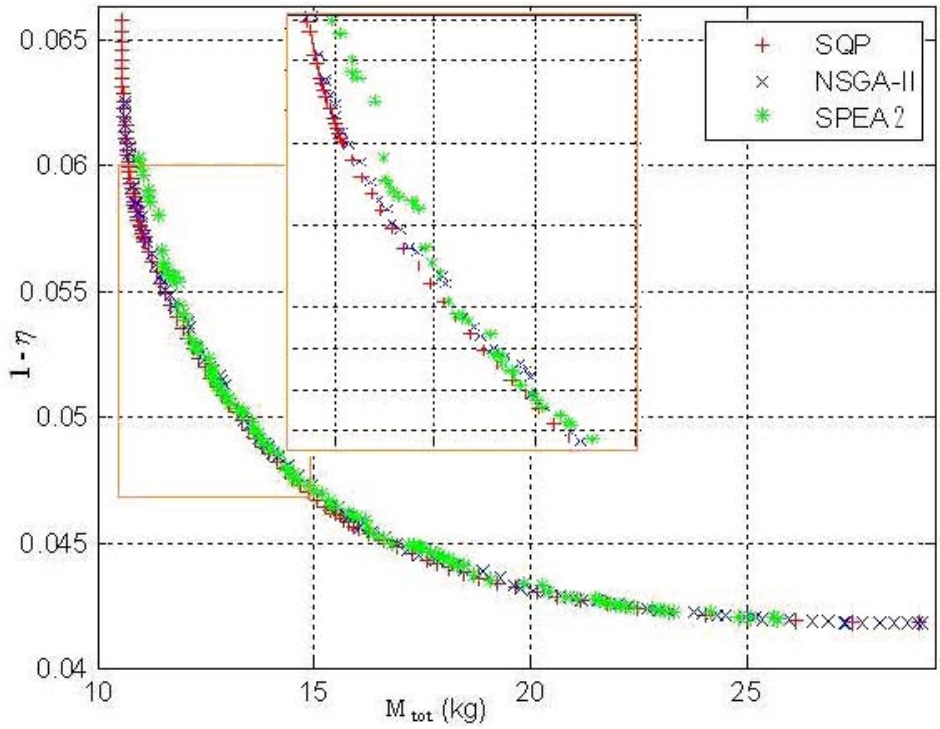


Figure 6.6: Pareto front found with the all optimizers.

To compare the convergence of both stochastic multi-objective optimizers (NSGA-II and SPEA2) and the variable weighted sum of objectives method using SQP (WS-SQP) *D-metric*

is used [Deb *et al.*, 2002] [Erbas *et al.*, 2006]. Concerning the uniformity and the extent criteria Δ-*metric* [Erbas *et al.*, 2006] and ∇-*metric* [Erbas *et al.*, 2006] are used. However, *D-metric* needs a reference set, so, the Pareto front obtained by the variable weighted sum of objectives WS-SQP is taken as the reference set. This Pareto front has been obtained using 100 values of the weight $alpha \in [0, 1]$. Table 6.4 presents the averages and standard deviations of the three performance metrics for 100 runs (i.e. 100 Pareto fronts for each stochastic optimizer NSGA-II and SPEA2).

Note that,as WS-SQP is a deterministic method giving only one Pareto set, there are only constant values for the Δ-*metric* and ∇-*metric*. Thus there are no standard deviation values for both metrics. Furthermore, as a reference set, the Pareto front obtained by WS-SQP can not be evaluated using the *D-metric*. All of these explain the empty boxes in Table 6.4.

On the other hand two other quality indicators are used. The contribution metric [Meunier *et al.*, 2000] evaluate the proportion of Pareto solutions given by each front, and coverage metric [Zitzler, 1999] evaluate the dominated area.

In terms of the convergence metric (*D-metric*) and diversity metric (Δ-*metric*), the scores of NSGA-II and SPEA2 are very close. In terms of the Δ-*metric*, the WS-SQP clearly outperforms both MOEA. NSGA-II and WS-SQP outperform SPEA-2 in terms of the extent metric (∇-*metric*).

Table 6.4: Performance assessment of the multi-objective algorithms.

	D-metric		Δ-*metric*		∇-*metric*	
Optimizers	avr.	std. dev.	avr.	std. dev.	avr.	std. dev.
NSGA-II	$8.51e^{-2}$	$1.70e^{-2}$	$3.51e^{-1}$	$1.20e^{-1}$	$4.22e^{-1}$	$1.99e^{-2}$
SPEA2	$8.78e^{-2}$	4.22^{-2}	$3.46e^{-1}$	$1.12e^{-1}$	$3.34e^{-1}$	$5.01e^{-1}$
WS-SQP	/	/	$1.86e^{-1}$	/	$4.0e^{-1}$	/

The Pareto fronts generated by NSGA-II and SPEA2 are compared with the Pareto set obtained by the WS-SQP in terms of contribution and coverage metrics. Table 6.5 presents the average values of the contributions obtained for 100 runs. The computed values of the contribution metric show that WS-SQP outperforms NSGA-II and SPEA2. The average contribution of WS-SQP is 65% of the non-dominated solutions of the union of NSGA-II and WS-SQP fronts, and 80% of the union of SPEA2 and WS-SQP fronts.

Table 6.5: Contribution metric.

	NSGA-II	SPEA2
WS-SQP	0.653	0.801

Table 6.6 presents the average value of the coverage metric obtained for 100 runs. This result confirms that WS-SQP outperforms NSGA-II and SPEA2. WS-SQP front covers 46% of the NSGA-II front, and 75% of the SPEA2 front. On the other hand, NSGA-II front dominates 44% of the SPEA2 front which dominates in its turn 17% of NSGA-II front. Neither NSGA-II nor SPEA2 front covers WS-SQP front.

Table 6.6: Coverage metric.

	WS-SQP	NSGA-II	SPEA2
WS-SQP	0	0.4641	0.75
NSGA-II	0.004	0	0.44
SPEA2	0	0.17	0

6.3.3 Outcome

In the Engineering design, a lot of problems deal with conflicting objectives. The main difficulty of such hard problems is to choose the suitable optimization method. Indeed, an ideal method gives a Pareto optimal set having a uniform distribution and a better coverage of the objective space, and even finds solutions in non-convex region. Unfortunately such method does not exist. This study focuses on three multi-objective optimization algorithms, in the aim to provide the more suitable one for the multi-objective design of a brushless DC wheel motor in the shortest time.

The Pareto fronts found by NSGA-II, SPEA2, and weighted sum of objectives with SQP (WS-SQP) were presented and compared in terms of convergence, uniformity, extent, contribution and coverage quality assessments. According to the optimization results, it is shown that NSGA-II is not superior to SPEA2 in terms of accuracy (*D-metric*) and diversity (Δ-*metric*). Both MOEA are very close to the Pareto front given by WS-SQP, which is taken as a reference set. Therefore, regarding the remainder of metrics, NSGA-II outperforms SPEA2. Both evolutionary algorithms need high computational effort (function evaluations) to build the Pareto set.

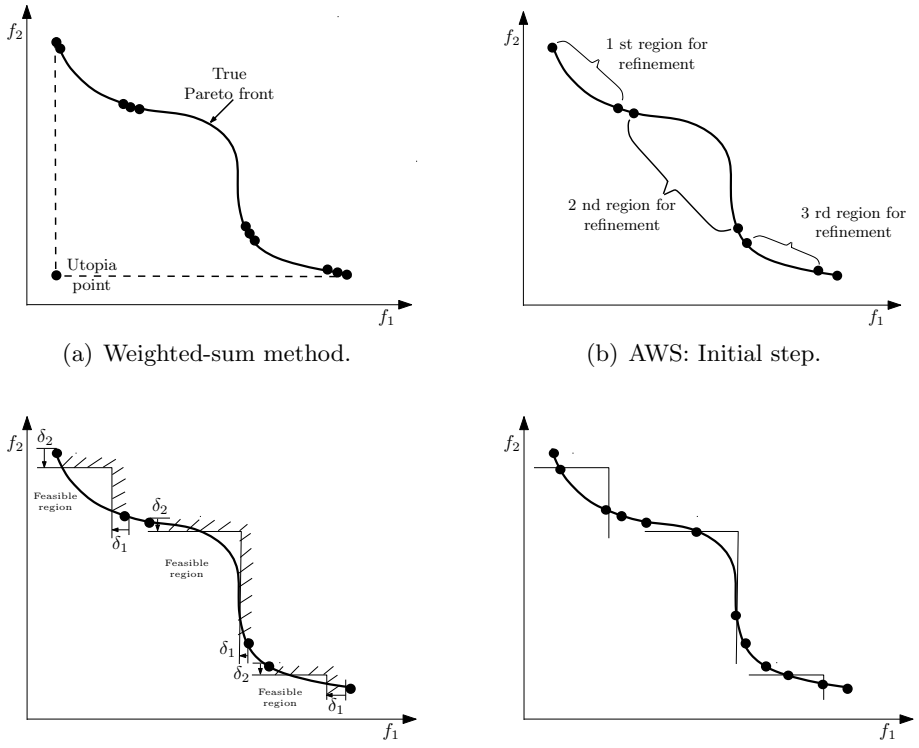
The main disadvantages of the traditional weighted sum of objectives using SQP (WS-SQP) are:

- the difficulty to choose correctly the initial point and the need of multiple starting points, which do not exist in NSGA-II and SPEA2,
- the poorly distributed solutions along the Pareto front,
- the not ability to find Pareto solutions in non-convex regions (contrary to NSGA-II and SPEA2).

The following section presents a new algorithm called Adaptive Weighted-Sum (AWS) that swiftly and efficiently builds a Pareto optimal set. The AWS is improved and compared with the traditional Weighted Sum technique WS and NSGA-II on the multi-objective design of the brushless DC wheel motor. As before, the Pareto fronts are compared in term of accuracy, uniformity, and coverage criteria.

6.4 Adaptive weighted sum of objectives (AWS)

The weighted sum (WS) method consists on the association of each objective function with a weighting coefficient and optimizes the weighted sum of the objectives. In this way, the multi-objective problem is transformed into a set of single objective ones. However, as shown in Figure 6.7(a), this method produces poorly distributed solutions along a Pareto front, and doesn't find Pareto optimal solutions in non-convex region. For all these reasons, an improvement algorithm called adaptive weighted sum (AWS) method is proposed by [Kim & De Weck, 2005].



(a) Weighted-sum method.

(b) AWS: Initial step.

(c) AWS: Constraint imposition.

(d) AWS: Refinement.

Figure 6.7: Concept of the AWS method [Kim & De Weck, 2005].

Contrarily to WS, in AWS [Kim & De Weck, 2005], the weights are changed adaptively and additional inequality constraints are imposed to find the concave parts of the Pareto front as illustrated in Figure 6.7. Below, the main steps of AWS are given.

- **Step 1.** In the beginning, WS method is applied with a small number of weighting factors $\alpha \in [0, 1]$.

- **Step 2.** When a new feasible region is detected for refinement (Figure 6.7(b)), additional inequality constraints are imposed in order to find the concave parts of the Pareto front (Figure 6.7(c)). The constraints are: $\mathrm{f}_1(X) \leq z^1 - \delta_1$
  $f_2(X) \leq z^2 - \delta_2$

  where z^1, z^2 are the objective values, and δ_1, δ_2 are the distances from the solution vector z in the 1^{st} and 2^{nd} dimension of the objective space respectively.

- **Step 3.** Thereafter, WS is performed again for the new feasible regions.

In the next iteration, AWS method proceeds to find a new region for refinement and so on until solutions are sufficiently distributed along the Pareto front or no more refinement is possible.

According to [Kim & De Weck, 2005], when a new feasible region is detected for refinement, the weighting factors $\alpha \in [0, 1]$ are used. However, at each AWS iteration some Pareto solutions are calculated one more time which lead to increase the computation time of the algorithm. Indeed, solutions associated with factors α, such as $\alpha \leq \alpha_1$ and $\alpha \geq \alpha_2$ are already calculated in the previous iteration, where α_1 and α_2 are the associated factors to the points, that delimit the new feasible region, z_1 and z_2 respectively.

To improve the AWS process, this study proposes the use of $\alpha \in \alpha_1, \alpha_2$ at each iteration. Then, only the solutions belonging to the region for refinement are calculated. Note that, WS and AWS are applied using the SQP algorithm.

In the following, the improved AWS algorithm is tested on the optimization of the brushless DC wheel motor. Thereafter, it is compared to the traditional WS and NSGA-II.

6.4.1 Optimization results

In this section AWS-SQP, WS, and NSGA-II are tested and compared on the multi-objective optimization of the brushless DC wheel motor.

- As before, for NSGA-II the objectives vector is:

$$\overrightarrow{F}(X) = [M_{tot}(X),\ (1 - \eta(X))]^T$$

- For WS-SQP and AWS-SQP the two objectives are weighted with the weight $\alpha \in [0, 1]$:

$$f(X) = \alpha M_{tot}(X) + (1 - \alpha)(1 - \eta(X))$$

where η and M_{tot} are the efficiency and the total mass of the brushless DC wheel motor respectively. $X = [D_s\, \delta\, B_d\, B_e\, B_{cs}]$ is the vector of the five design variables.

Concerning NSGA-II, its parameters are selected as before (Table 6.3). However, the Constraint-First-Objective-Next model, presented in section 6.1, is used in order to handle the inequality constraints of the optimization problem. Table 6.7 summarizes the new parameters applied in this section. Note that the parameters C_{min}, C_{max}, CF_1, and CF_2 are used by the constraint operator as presented in equations (6.1) and (6.2).

Table 6.7: NSGA-II parameters and values.

Parameters	Values
- Fitness scaling	Defined in (6.1)
- Parameter C_{min}	0.8
- Parameter C_{max}	1.2
- Parameter CF_1	0.015
- Parameter CF_2	0.015

By mean of AWS-SQP and WS-SQP, the Pareto fronts have been obtained using only 28 values of the weight $\alpha \in [0, 1]$. The obtained Pareto points are plotted in Figure 6.8.

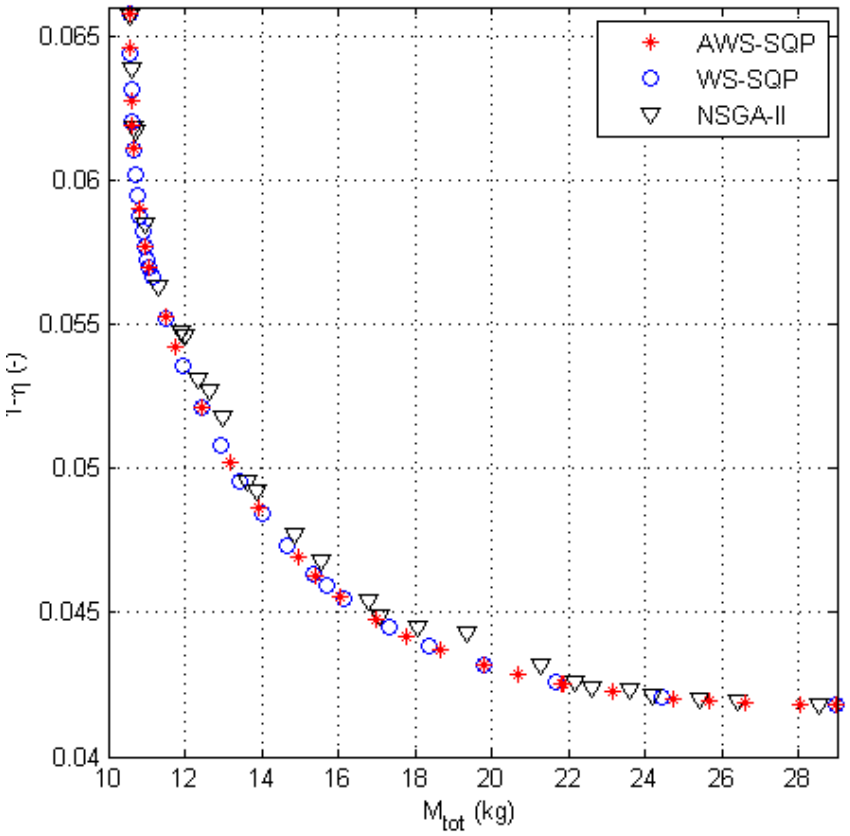


Figure 6.8: Pareto front found with the all optimizers.

The quality of the obtained Pareto solutions are compared in term of uniformity and extent criteria by using the Δ-*metric* and ∇-*metric*. Table 6.8 presents the averages and standard deviations of the three performance metrics for 100 runs.

In terms of the Δ-*metric*, the AWS-SQP clearly outperforms both algorithms. AWS-SQP and WS-SQP outperform NSGA-II in terms of the extent metric (∇-*metric*).

Table 6.8: Performance assessment of the multi-objective algorithms.

	Δ-*metric*		∇-*metric*	
Optimizers	avr.	std. dev.	avr.	std. dev.
NSGA-II	$1.86e^{-1}$	$1.20e^{-1}$	$7.75e^{-1}$	0.2606
WS-SQP	$0.20e^{-1}$	/	$9.89e^{-1}$	/
AWS-SQP	$0.16e^{-1}$	/	$9.89e^{-1}$	/

However, the results of NSGA-II obtained in this section are better than those presented in section 8.5.2. This is explained by the use of different constraint handling operator.

As explained before, with the deterministic methods AWS-SQP and WS-SQP there are no standard deviation values for both metrics. This explains the empty boxes in Table 6.8.

6.4.2 Outcome

The adaptive weighted sum method (AWS) is improved in the aim to provide a uniformly distributed and extent Pareto front in a lower computational time, allowing multi-objective optimization of electrical device. The results demonstrate that AWS produces well-distributed solutions compared to the traditional weighted-sum method, and have a better coverage of the objective space than the multi-objective evolutionary algorithm NSGA-II.

Furthermore AWS, by imposing additional inequality constraints, is able to find Pareto solutions in non-convex regions and neglects non-Pareto optimal ones. Then, to find Pareto optimal solutions, AWS is more robust than traditional weighted-sum method. The drawback of AWS is only used to handle bi-objective optimization problems. Because of its success, the authors [Kim & De Weck, 2006] generalized the AWS method for multi-objective optimization problems.

NSGA-II is a powerful optimization algorithm especially when its parameters are well-tuned. Indeed, by adding novel operators in order to

- handle effectively the constraints,
- ensure an optimal balancing between the information intensification and the exploration the search space,

NSGA-II is able to find more accurate and interesting results. Furthermore, NSGA-II can be used in order to solve mixed-integer optimization problems with several kind if design variables: continuous, discrete, integer and zero-one.

Unfortunately, those methods require a very high number of model evaluations. This is not compatible with the use of fine models, like the finite element models, that require a huge calculation time (more than 1 hour per one objective function evaluation).

Indeed, multi-objective optimizations by means of 3D finite element models result in very high computation burden. To have an affordable computation cost, a new optimization method is proposed, in the next section, in order to ensure a fast building of Pareto optimal

sets when dealing with fine models. The proposed algorithm is used and tested on the bi-objective optimization of a safety isolating transformer and allows finding the complete Pareto optimal set in less than two days on a laptop.

6.5 Fast building of a Pareto set

6.5.1 Designer's dilemma

This section is devoted to answer how it is possible to build Pareto sets by means of fine models like 3D finite element models.

Let's consider the bi-objective optimization of a safety isolating transformer dealing with 3D finite element models (3D FE). Where the multi-physical phenomena are modeled by a lumped-mass (coarse) model (LM) and a 3D finite element (fine) model (3D FE).

What kind of optimization techniques should be used in order to find the complete Pareto optimal set in a short time of convergence?

Indeed, generating 100 optimal solutions of the Pareto optimal set by means of the 3D FE model requires 10000 hours while 5 minutes only are needed with the LM model. In fact, as solving 3D FE model is very expensive only six points are computed and considered as a reference set (Figure 6.9). The LM model is used to build an extended Pareto optimal set in a short time. It is obvious that this last optimal set is far from the reference set.

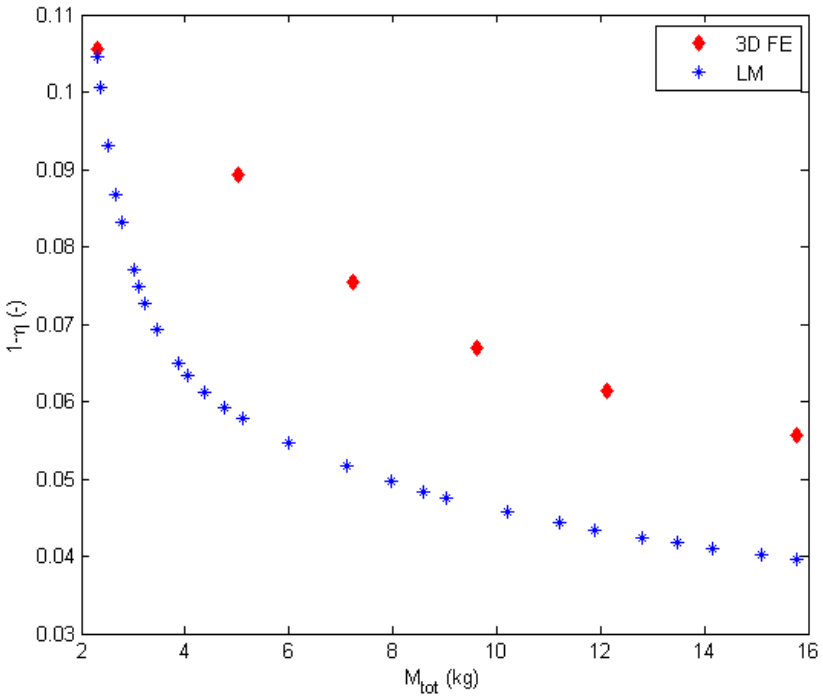


Figure 6.9: Pareto fronts using LM and 3D FE models.

A first solution to this dilemma is to interpolate the reference set. Unfortunately, no information on the design parameters can be obtained because the interpolation can only be made in the objective space.

Therefore, we propose to combine a multi-objective optimization method with a space-mapping (SM) technique in order to generate an extended and accurate Pareto optimal set in less than 100 hours.

6.5.2 Concept of the proposed method

The concept consist on the combination of a multi-objective optimization algorithm with the space-mapping technique.

As a multi-objective optimization method, the ϵ-constraint is selected in order to be combined with a space-mapping technique.

According to the "State of the art", the ϵ-constraint method is a useful approach to build a Pareto optimal set. Moreover, this method may reach Pareto optimal solutions in the non-convex region . The ϵ-constraint method consists to transform the multi-objective problem in a mono-objective problem. Among the objectives, one is kept and the others are transformed in inequality constraints.

$$\begin{cases} x^* = \min\limits_{x \in X} f_j(x) \\ \text{subject to } f_{j \neq i}(x) \leq \epsilon_j \\ \qquad\qquad g(x) \leq 0 \end{cases} \tag{6.4}$$

where $f_i(x)$ is the kept objective, $f_{j \neq i}(x)$ are the other objectives. $g(x)$ are the ordinary constraints vector. ϵ_j are the threshold values.

By varying ϵ_j between ϵ_j^{min} and ϵ_j^{max}, the whole Pareto optimal set may be found. Two mono-objective optimizations are computed to determine the upper and lower threshold values. ϵ_j^{min} is found by minimizing $f_{j \neq i}(x)$ and ϵ_j^{max} is found by minimizing $f_i(x)$. Using the ϵ-constraint method with the coarse model, a Pareto front with 100 points is quickly built and shown in Figure 6.9.

In the following, the concept of the space-mapping technique is given first. Thereafter, the key of these techniques, which consist on the correction factors, is modified and improved in order to combine the SM with the ϵ-constraint method.

Space-mapping techniques

Space mapping techniques aim to use both coarse and fine models in order to reduce the computational time and increase the accuracy of the obtained solution [Choi *et al.*, 2001], [Bandler *et al.*, 2004], [Echeverria *et al.*, 2006], and [Encica *et al.*, 2008].

In general, the coarse computationally cheaper model is denoted by $c(z) \in R^m$ with $z \in Z \subset R^n$, and the fine computationally expensive model is denoted by $f(x) \in R^m$ with $x \in X \subset R^n$. The non-linear constraints computed by the coarse and the fine models are $k_c(z)$ and $k_f(x)$, respectively. The set of variables x^* represents the solution of a given optimization problem:

$$x^* = argmin_{x \in X} \parallel f(x) - y \parallel \ \ s.t. \ \ k_f(x) \leq 0 \tag{6.5}$$

where $y \in R^n$ denotes a vector of design specifications and can be zeros in the case of a minimization. In practice, solving (6.5) is very expensive. Therefore, a faster optimization problem based on the coarse model is preferred:

$$z^* = argmin_{z \in Z} \parallel c(z) - y \parallel \ \ s.t. \ k_c(z) \leq 0 \tag{6.6}$$

One approach of SM techniques, called Output Space-Mapping (OSM) [Encica *et al.*, 2008] consists to modify the coarse model by adding some corrective coefficients $\theta \in \Theta \subset R^p$, p is the number of responses (objectives and constraints) computed by the fine model, in order to align the results of the coarse model with those of the fine model. The coefficients are updated at each iteration to minimize the discrepancy between both models:

$$\theta^{(i+1)} = argmin_{\theta \in \Theta} \left\| \begin{bmatrix} c(x^{(i)}, \theta) \\ k_c(x^{(i)}, \theta) \end{bmatrix} - \begin{bmatrix} f(x^{(i)}) \\ k_f(x^{(i)}) \end{bmatrix} \right\| \tag{6.7}$$

Then, they are introduced in the coarse model to compute an optimal solution x^{i+1} for the next iteration:

$$x^{(i+1)} = argmin_{x \in X} \| c(x, \theta^{(i+1)}) - y \|, \ s.t. \ k_c(x, \theta^{(i+1)}) \leq 0 \tag{6.8}$$

where $c(x, \theta)$ and $k_c(x, \theta)$ denote the corrected coarse models of the objective and constraint functions respectively.

After that, a simulation with x^{i+1} by the fine model is performed to compute the fine responses $f(x^{i+1})$ and $k_f(x^{i+1})$. The algorithm stops when all the corrective coefficients are unchanged from one iteration to the next one.

In order to combine the OSM technique with the multi-objective optimization method ϵ-constraint, how the corrective coefficients are updated at each iteration is improved. Then corrective spline functions are proposed in the next section.

Corrective spline functions

A corrective coefficient is introduced for each objective and constraint therefore the corrected coarse model has the following expression:

$$\begin{bmatrix} c(x, \theta) \\ k_c(x, \theta) \end{bmatrix} = Diag(\theta(\epsilon)) \cdot \begin{bmatrix} c(x) \\ k_c(x) \end{bmatrix} \tag{6.9}$$

In the case of a mono-objective optimization only a set of p scalar coefficients is searched. The parameter p is the number of the objectives and the constraints computed using the fine model.

$$\theta^{(j+1)} = \begin{bmatrix} c(x^{(i)})/f(x^{(i)}) \\ k_c(x^{(i)})/k_f(x^{(i)}) \end{bmatrix} \tag{6.10}$$

where the indices f and c denote the fine and coarse model, respectively.

In the case of a multi-objective optimization, the p corrective coefficient's values must be changed for each solution from the Pareto set, i.e. for each value of the threshold value $\epsilon \in [\epsilon_{min}, \epsilon_{max}]$. So that these p coefficients are replaced by p corrective functions that are function of the parameter ϵ:

$$\theta^{(j+1)}(\epsilon) = \begin{bmatrix} S_i(\epsilon) \\ : \end{bmatrix} \tag{6.11}$$

where S_i is a spline cubic interpolation function. The spline cubic functions avoid the oscillations that appear in the polynomial interpolation approach when the order is high.

Since the mapping functions are defined, they are used by the multi-objective algorithm in order to correct the coarse model and to obtain a new Pareto optimal set.

The following section presents the proposed algorithm that combines the OSM technique with the ϵ-constraint algorithm.

Proposed algorithm

At the beginning of the algorithm ($j = 0$), all the corrective coefficients are initialized to unity:

$$\theta^{(0)} = I \tag{6.12}$$

At each iteration, a multi-objective optimization is performed by using the ϵ-constraint method and the corrected coarse model in order to obtain a new Pareto optimal set in a short

time:

$$\begin{cases} \min\limits_{x \in X} c_1(x, \theta^{(j)}(\epsilon)) \\ \text{subject to } c_2(x, \theta^{(j)}(\epsilon)) \leq \epsilon \\ \qquad\qquad k_c(x, \theta^{(j)}(\epsilon)) \leq 0 \\ \text{where } \epsilon_{min} \leq \epsilon \leq \epsilon_{max} \textbf{ takes 30 values} \end{cases} \tag{6.13}$$

Then $(2^j + 1)$ solutions x^* belonging to the Pareto optimal set are chosen in order to compute the responses of the fine model $f(x^*)$ and $k_f(x^*)$. Note that only 2^{j-1} points are new. To establish the corrective functions at the next iteration, the new points are at the center of the intervals, in the same way as the Dichotomy method.

If the following condition is check, then the algorithm stops:

$$\left\| \begin{bmatrix} c(x^*, \theta^{(j+1)}(\epsilon)) \\ k_c(x^*, \theta^{(j+1)}(\epsilon)) \end{bmatrix} - \begin{bmatrix} c(x^*, \theta^{(j)}(\epsilon)) \\ k_c(x^*, \theta^{(j)}(\epsilon)) \end{bmatrix} \right\| \leq \tau \tag{6.14}$$

where τ is the required accuracy. If the corrected coarse model responses are not close enough to the fine model ones, the algorithm continues by updating the corrective functions defined in (6.11) with the new points.

To summarize, the proposed algorithm carries out the following main steps:

- **Step 0.** Initialize, $j = 0$, $\theta^0 = I$.
- **Step 1.** Build a Pareto optimal set by using the ϵ-constraint method and the corrected coarse model $c(x, \theta^{(j)})$ to solve (6.13).
- **Step 2.** Chose $2^j + 1$ points on the Pareto optimal set.
- **Step 3.** Evaluate the fine model with the chosen points to compute $f(x)$ and $k_f(x)$
- **Step 4.** Update the p corrective functions $\theta^{(j+1)}(\epsilon)$.
- **Step 5.** If the condition (6.14) is reached then stop the algorithm, else $j = j + 1$, and go to 1.

In the following the robustness of the proposed algorithm is tested on the bi-objective optimization design of the safety isolating transformer.

6.5.3 Multi-objective optimal design of a safety isolating transformer

The safety isolating transformer is a one-phase step-down transformer. It uses grain-oriented E-I laminations. The primary and secondary windings are both wound around the frame surrounding the central core (Figure 6.10).

The multi-physical phenomena within the transformer are electric, magnetic and thermal. They are modeled by a lumped-mass (coarse) model (LM) and a 3D finite element (fine) model (3D FE) [Tran *et al.*, 2007a]. The LM model is built with the assumption that the voltage drop due to the magnetizing current is neglected. Therefore the maximal induction depends on the primary voltage. The computational time of the coarse model is very short (50 ms on an Intel Pentium M 2.13 GHz machine).

The 3D FE magneto-thermal model is built with the assumption that all magnetic and electric quantities are sinusoidal. The iron loss is computed with Steinmetz formula and the

leakage inductances are calculated with the magnetic energy. Full-load and no-load simulations are used to compute all characteristics of the safety isolating transformer. The 3D FE model with magneto-thermal coupling requires a very expensive computational time (about 2 hours on an Intel Pentium M 2.13 GHz laptop).

Readers who want more details about this benchmark should refer to [Tran *et al.*, 2007a]. Furthermore, all materials can be downloaded at
`http://l2ep.univ-lille1.fr/come/benchmark-transformer.htm`.

Problem formulation

The multi-objective optimization problem of a safety isolating transformer contains 7 design variables. There are three parameters a, b, c for the shape of the lamination, one for the frame d, two for the section of conductors S_1, S_2, and one for the number of primary turns n_1. The upper and lower bounds of these geometrical parameters are given in Figure 6.10. The seven design variables are summarized in Table 6.9.

Table 6.9: List of design variables.

Number	Design variable	Unit	Symbol
1	Shape of the lamination	mm	a
2	Shape of the lamination	mm	b
3	Shape of the lamination	mm	c
4	Frame of the lamination	mm	d
5	Section of conductors	mm^2	S_1
6	Section of conductors	mm^2	S_2
7	Number of primary turns	-	n_1

There are 6 non-linear inequality constraints in this problem. The copper and iron temperatures T_{co}, T_{ir} should be less than $120^{\circ}C$ and $100^{\circ}C$, respectively. The magnetizing current I_μ/I_1 and drop voltage $\Delta V_2/V_2$ should both be less than 10%. Finally, the filling factors of both coils f_1 and f_2 should both be lower than 0.5. The 6 inequality constraints are summarized in Table 6.10.

Table 6.10: List of constraints.

	Inequality constraint	Unit	Symbol
1	Copper temperature	°C	T_{co}
2	Iron temperature	°C	T_{ir}
3	Magnetizing current	-	I_μ/I_1
4	Drop voltage	-	$\Delta V_2/V_2$
5	Filling factor of primary coil	-	f_1
6	Filling factor of secondary coil	-	f_2

The objective functions are to maximize the efficiency η (or minimize the losses $(1-\eta)$) and to minimize the total mass M_{tot} of iron and copper materials. Thus, the multi-objective

optimization problem is expressed as:

$$\begin{cases} \min\ M_{tot} \\ \min\ (1-\eta) \\ \text{with respect to}\ \ 3.0mm \le a \le 30mm,\ 14mm \le b \le 95mm \\ \qquad\qquad\qquad\quad 6.0mm \le c \le 40mm, 10mm \le d \le 80mm \\ \qquad\qquad\qquad\quad 200 \le n_1 \le 1200,\ 0.15mm^2 \le S_1 \le 19mm^2 \\ \qquad\qquad\qquad\quad 0.15mm^2 \le S_2 \le 19mm^2 \\ \text{subject to}\qquad T_{co} \le 120\,^{\circ}C,\ T_{ir} \le 100\,^{\circ}C \\ \qquad\qquad\qquad\quad I_{\mu}/I_1 \le 0.1,\ \Delta V_2/V_2 \le 0.1 \\ \qquad\qquad\qquad\quad f_1 \le 0.5,\ f_2 \le 0.5 \end{cases} \tag{6.15}$$

To summarize, the multi-objective optimization design problem of the safety isolating transformer is represented graphically in Figure 6.10. The left side shows 7 design variables. The objective function and constraints are in the right side.

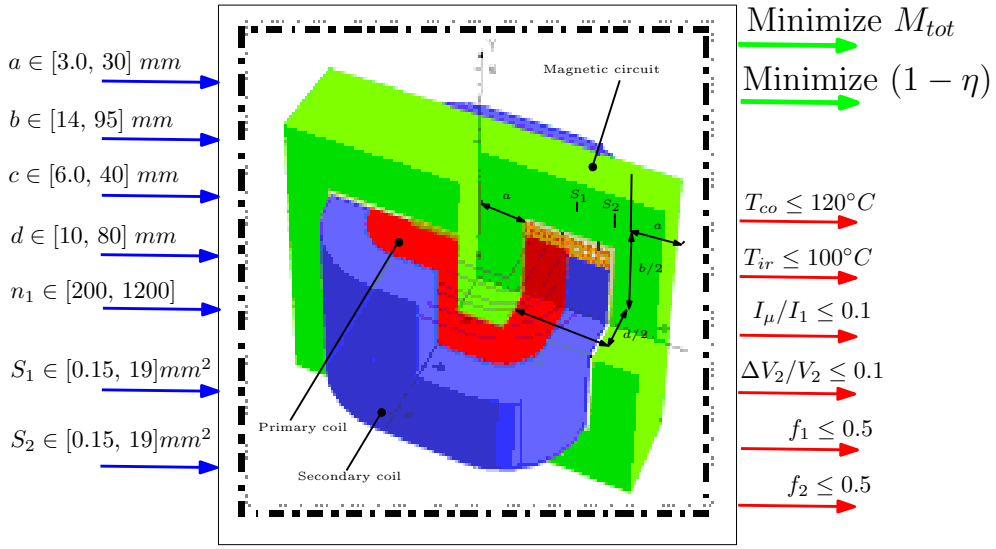


Figure 6.10: Multi-objective optimization problem of a safety isolating transformer.

Application of the proposed algorithm

The bi-objective optimization problem of the safety isolating transformer presented in section 6.5.3 is solved with the proposed algorithm.

For this optimization problem, two constraints (filling factors f_1, f_2) among six and one objective (total mass M_{tot}) among two are not evaluated by the 3D FE model. Indeed, both filling factors and the total mass are analytically computed. So that, five responses η, T_{co}, T_{ir}, I_{μ}/I_1, and $\Delta V_2/V_2$ need 3D FE computation, i.e. only five values of corrective coefficients $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5]$ are computed for each threshold value $\epsilon \in [\epsilon_{min},\ \epsilon_{max}]$.

At initialization ($j = 0$), the coarse model is not corrected yet and the corrective functions are all equal to one. Two points at ϵ_j^{min} and ϵ_j^{max} are computed with the fine model to test the stop criteria and build the corrective functions for the next iteration. At iteration 1, the corrective functions are now linear (Figure 6.5.3.0) and a new optimal set is obtained. In Figure 6.5.3.0, the corrective function is given only for the efficiency of the transformer, i.e. θ_1.

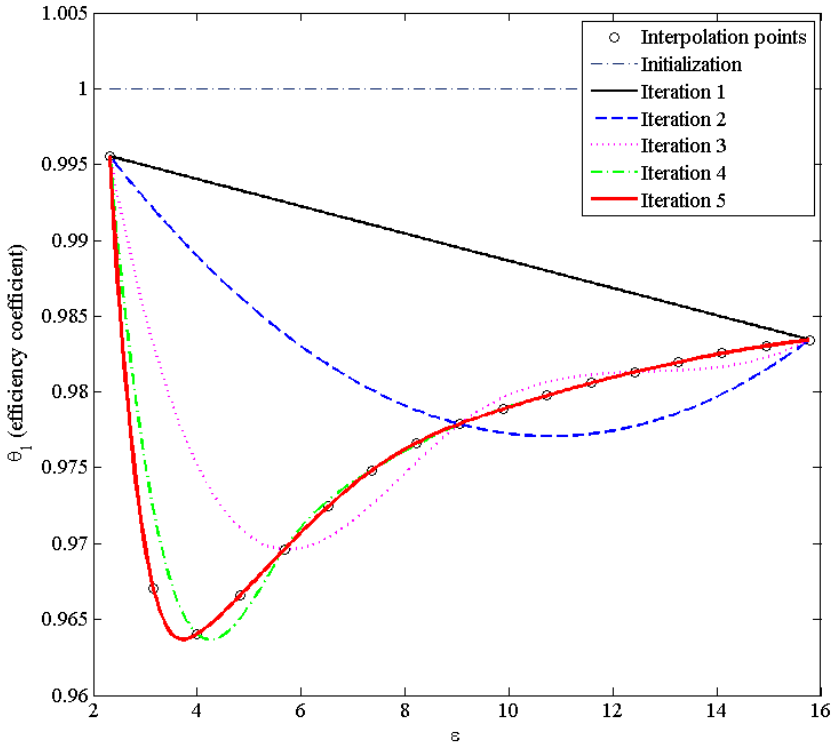


Figure 6.11: Corrective functions for the efficiency of the transformer.

At each iteration, one new point is added at the middle of each couple of points, resulting in 2^{j-1} new points amongst a total of $(2^j + 1)$ points, i.e. 3D FE model evaluations.

Figure 6.12 shows the Pareto optimal set obtained by using the corrective functions. 30 points are given at each iteration. At the end of the optimization, a Pareto optimal set very close to the reference set is found.

Finally, only 17 3D FE simulations are needed to obtain an accurate Pareto optimal set. The computational time of the proposed algorithm is approximately 34 hours (1.4 day) on an Intel Pentium M $2.13\,GHz$ laptop. The reference Pareto optimal set requires 60 hours (2.5 days) on the same machine. To have an accurate Pareto optimal set by using the 3D FE model only, approximately $10,000$ hours (14 months) are required. All the computed points are shown in Figure 6.5.3.0.

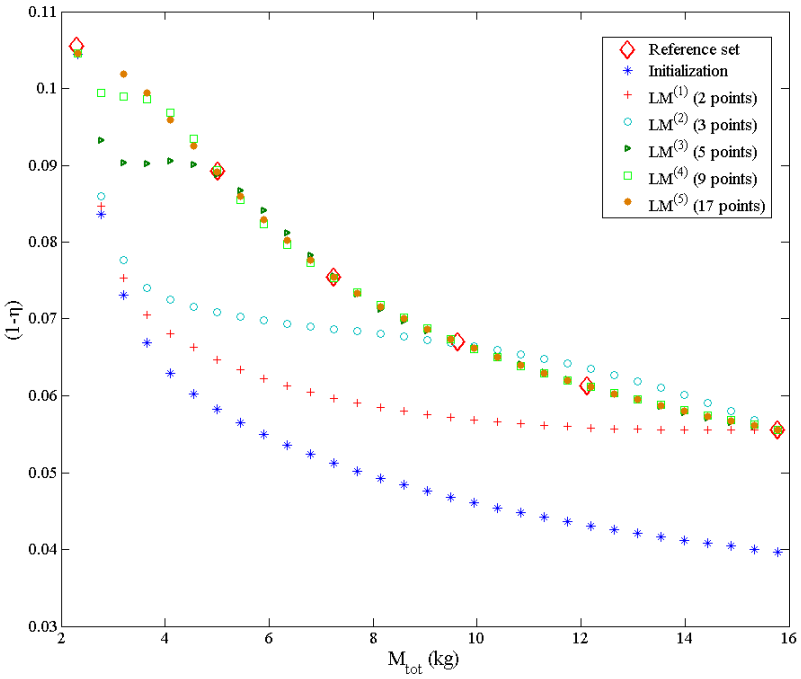


Figure 6.12: Pareto fronts obtained by the proposed algorithm.

6.5.4 Outcome

This section answered the question: how Pareto sets by means of 3D finite element models can be built effectively and in a short time.

A part of this answer is the use of the space-mapping techniques (SM) that combine coarse and fine models in order to reduce the computational time and increase the accuracy of the Pareto solutions.

Then, an original optimization algorithm is proposed which consists on the combination of output space-mapping (OSM) technique by [Tran *et al.*, 2009] with the multi-objective optimization algorithm ϵ-constraint.

The output space-mapping technique is adapted to provide a practical way to build an accurate Pareto optimal set in bi-objective optimization using full 3D coupled finite element model and keeping the computational time in an acceptable limit. Cubic spline interpolation functions are used to predict the values of the corrective coefficients. By using these corrective functions, the ϵ-constraint optimization algorithm gives an extended and accurate Pareto optimal set in less than 2 days.

6.6 Conclusion

In this chapter, several multi-objective optimization algorithms are tested, compared and improved on the multi-objective design of a brushless DC wheel motor and a safety isolating transformer.

The results demonstrate that Adaptive weighted sum AWS and the multi-objective evolutionary algorithm NSGA-II are the preferred algorithms which are able to produce well-distributed solutions among the Pareto front and found solutions in non-convex regions.

NSGA-II is a powerful optimization algorithm that is used, in chapter 7, in order to solve a mixed-integer optimization problem dealing with continuous and discrete design variables.

This study, is devoted too to the fast building of a Pareto front dealing with 3D finite element models. Then, an original optimization algorithm is proposed which combines the output space-mapping (OSM) technique with the multi-objective optimization algorithm ϵ-constraint.

The multi-objective optimization algorithm ϵ-constraint is chosen because it is:

- a useful approach to build a Pareto optimal set,
- able to reach Pareto solutions in a non-convex region,
- a simplified method that requires few objective function evaluations compared to NSGA-II and AWS.

To conclude, we can say, until now, SM was applied to continuous optimization problems. The joint use of SM and AWS for multi-objective optimization problems with discrete variable is a new idea.

7

Mixed-integer optimization algorithms

All the practical test cases presented so far in this thesis have continuous design variables. However, in engineering practice, optimization design problems may include mixed design variables: continuous like geometry parameters, integer as the number of pole pair, and discrete (not countable) like the type of materials that can be used. These kind of problems are called mixed-integer optimization problems (MIP).

Furthermore, several design optimization problems, in addition to include different kind of variables, can deal with many objectives that must be achieved simultaneously. Similarly, this type of problems are called mixed-integer multi-objective optimization problems.

According to the literature, it is difficult to find optimization algorithms that deal with this complexity. Indeed, optimization problems with a non uniformly search space (mixed-integer) have a complexity which is exponential in the number of integer variables. Moreover, multi-objective optimization problems can be hard to solve when the number of objectives increases (for instance above 10) [Deb *et al.*, 2002]. This mismatch is largely due to the inherent complexity of applications and the limited scope of efficient algorithms.

As it is highlighted in chapter 2 **State of the art of the mono-level optimization**, Branch and Bound method (BB) [Tran *et al.*, 2007c] is the classic method for solving MIP problems. Based on the BB concept, other methods are proposed like the non-linear branch and bound algorithm [Sandgren, 1990] and the hybrid method combining BB and SQP [Leyffer, 2001].

Evolutionary algorithms, especially genetic algorithms, are alternative methods that can be used to solve the MIP optimization problems. This chapter focuses on efforts to present and discuss the concept of how standard genetic algorithms can be modified and improved in the aim to solve mixed-integer optimization problems. This concept is tested:

- in the case of a mono-objective optimization of a safety isolating transformer with respect of discrete and integer design variables;
- as with a multi-objective optimization design of a surface-mounted permanent magnet (SMPM) synchronous motor, with respect of mixed-integer (continuous and integer) decision variables.

First, a mixed-integer genetic algorithm is proposed and the key components, allowing this improvement, are discussed. Thereafter, using the same concept, NSGA-II is improved to solve mixed-integer optimization problems.

7.1 Mixed-integer mono-objective optimization algorithms

According to the state of the art, genetic algorithms (GA) manage very well continuous, zero-one, integer and discrete variables.

First, the general concept of a mixed-integer GA is given and the key components, allowing this improvement, are discussed. Thereafter an optimization benchmark is given to show how GA can be adapted to solve it.

The optimization problem consist on the minimization of the mass of a safety isolating transformer where only discrete and integer design variables are considered.

7.1.1 Mixed-integer genetic algorithm

This section intends to show how GA can be modified in the aim to handle in the same time continuous, zero-one, integer and discrete variables. According to the state of the art, crossover and mutation operators are the key of this improvement. In the following, the concept of the proposed algorithm called mixed-integer genetic algorithm is discussed.

Crossover and mutation operators

As it is explained in part **State of the art** where GA is presented briefly, crossover and mutation operators are used to generate the offspring. Furthermore, these components depend on the kind of the design variables. In other term there are real crossover and mutation operators for continuous variables; and binary crossover and mutation operators for the discrete, integer, and zero-one variables.

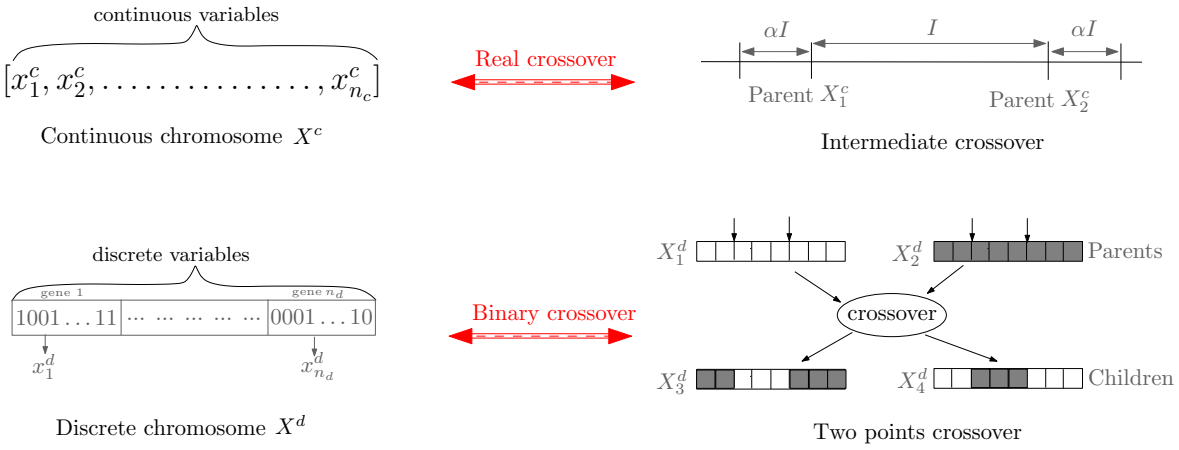


Figure 7.1: Crossover operators for mixed-integer chromosome.

Then to generate mixed chromosomes (mixed-genes) we propose to modify both operators (mutation and crossover) to deal with mixed design variables. Thus, both real and binary operators are combined as shown in Figure 7.1.

For example, with continuous variables, real crossovers like intermediate or arithmetic crossovers can be applied; and binary crossovers like: one, two or multiple points crossovers can be used for the others variables. In the same way, Gaussian and binary mutations can be applied for continuous and zero-one, discrete, and integer genes respectively. In the next section a practical test case is used to illustrate this concept.

The mutation and the crossover operators depend on the type of the chosen representation: binary crossovers for binary representation and real operators for a real representation. Furthermore, with a binary representation for continuous variables, there are less accuracy

than when we use the real one [Michalewicz, 1994], [Spall, 2003], [Dréo *et al.*, 2006]. Then, it is obvious that different representations will be used for different kind of variables. In other term, as shown in Figure 7.1, a real representation is applied for continuous variables and a binary coded is used for the remaining (zero-one, integer, and discrete) variables.

However, the use of binary coded is time consuming. Indeed, to evaluate the individuals the binary-coded variables are converted to their origin, i.e. discrete, integer, or zero-one values. for this reason, only real and discrete coded are used in this study.

Evaluation and Selection

Concerning the evaluation and the selection process, the ones proposed in the literature for classical genetic algorithms can be used. Then, for evaluation procedure the chromosome is considered as a whole and all variables (genes)- continuous, zero-one, integer, discrete - are treated in the same manner. Thereafter, the best individuals (chromosomes) in term of the fitness value are selected to generate offspring by applying mutation and crossover operators as it is explained before.

In the following, the mono-objective optimization of a safety isolating transformer is presented and used to test the proposed mixed-integer genetic algorithm. This benchmark is an optimization problem with only discrete and integer design variables. Then the proposed concept is used without continuous variables, i.e. the number of continuous variables is set to zero ($n_c = 0$), and only binary operators are implemented as shown in section 7.1.2.0. There after, the proposed algorithm is compared to a branch and bound algorithm, and an exhaustive enumeration technique.

7.1.2 Mixed-integer optimization design of a safety isolating transformer

In chapter 6 the safety isolating transformer benchmark is used as a multi-objective optimization problem in order to test the robustness of the proposed algorithm AWS-SM, which consist on the combination of the multi-objective optimization algorithm AWS and the space-mapping technique SM.

In this chapter, it is taken as a mono-objective optimization problem by only minimizing the total mass of the active parts M_{tot} subject to have an efficiency $\eta \geq 0.8$ which was an objective in the multi-objective optimization problem described in section 6.5.3.

Furthermore, the presented benchmark have non-continuous design variables that only have discrete values. Then, the safety isolating transformer, that uses a lumped-mass model and a 3D FE model [Tran *et al.*, 2007c], [Tran *et al.*, 2007b], is treated in this chapter as a mono-objective combinatorial optimization problem. Readers who want more details should refer to [Tran *et al.*, 2007a], furthermore, all materials can be downloaded at `http://l2ep.univ-lille1.fr/come/benchmark-transformer.htm`.

Design problem definition

The optimization problem contains 7 non-continuous design variables. There are three parameters a, b, c for the shape of the lamination, one for the frame d, two for the section of conductors S_1, S_2, and one for the number of primary turns n_1. There are 62 possible combinations for the lamination and the frame, and 63 types of conductors. The number of primary turns n_1 is an integer but only 1000 values are allowed, leading to $246,078,000$ possible combinations. All the feasible values are given in [Tran *et al.*, 2007c]. The seven design variables are presented in Table 7.1.

This optimization problem includes 7 inequality constraints. The copper and iron temperatures T_{co}, T_{ir} respectively are less than $120°C$ and $100°C$. The efficiency η is greater

Table 7.1: List of design variables.

Number	Design variable	Unit	Symbol
1	Shape of the lamination	mm	a
2	Shape of the lamination	mm	b
3	Shape of the lamination	mm	c
4	Frame of the lamination	mm	d
5	Section of conductors	mm^2	S_1
6	Section of conductors	mm^2	S_2
7	Number of primary turns	-	n_1

than 80%. The magnetizing current I_μ/I_1 and drop voltage $\delta V_2/V_2$ are less than 10%. Finally, the filling factor of both coils f_1, f_2 is lower than 0.5. The 7 inequality constraints are summarized in Table 7.2.

Table 7.2: List of constraints.

	Inequality constraint	Unit	Symbol
1	Copper temperature	°C	T_{co}
2	Iron temperature	°C	T_{ir}
3	Efficiency	-	η
4	Magnetizing current	-	I_μ/I_1
5	Drop voltage	-	$\delta V_2/V_2$
6	Filling factor of primary coil	-	f_1
7	Filling factor of secondary coil	-	f_2

The objective function is to minimize the mass M_{tot} of iron and copper materials. Thus, the optimization problem is expressed as:

$$\begin{cases} \min M_{tot}(X) \\ \quad s.t. \quad T_{co} \leq 120\,°C,\ T_{ir} \leq 100\,°C,\ \eta \geq 0.80, \\ \qquad I_\mu/I_1 \leq 0.1,\ \delta V_2/V_2 \leq 0.1,\ f_1 \leq 0.5,\ f_2 \leq 0.5 \\ \quad with\ X = [a,\ b,\ c,\ d,\ n_1,\ S_1,\ S_2] \\ \qquad a, b, c \in EI,\ d \in F,\ S_1,\ S_2 \in C,\ n_1 \in N \end{cases} \tag{7.1}$$

To summarize, the optimization design problem of the safety isolating transformer is represented graphically in Figure 7.2. The left side shows 7 design variables. The objective function and constraints are in the right side.

Beside the proposed mixed-integer algorithm, it exists several manner to solve this optimization problem.

A typical approach for handling discrete variables is to solve the corresponding continuous optimization problem, using for example SQP, and then adjusts the optimal design to neighbouring discrete values. But there is no guarantee that this process is correct or that a good solution can be obtained by this way.

An other way, is to evaluate all the $246,078,000$ possible combinations of the discrete variables. However, using this exhaustive enumeration technique (EE), the needed time is very high. An efficient method to overcome the drawback is branch and bound (BB) algorithm [Venkataraman, 2001] which is based on partial enumeration where only parts of the combinations are explored.

Then, to show the efficiency of the proposed mixed-integer genetic algorithm, it is compared to the three methods SQP, BB, and EE.

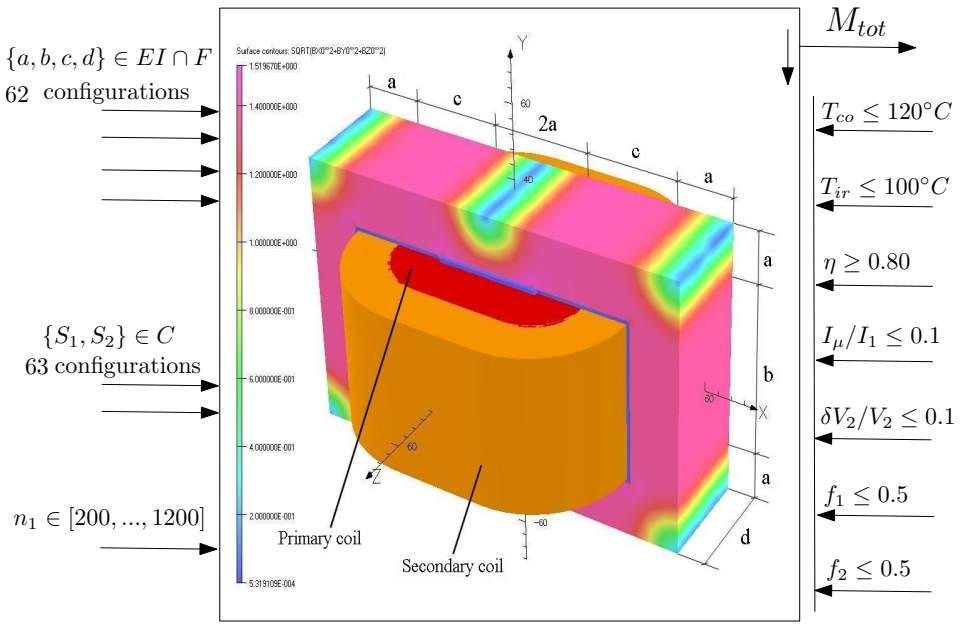


Figure 7.2: Combinatorial optimization of a safety isolating transformer.

Use of genetic algorithm

With the optimization of the safety isolating transformer problem only discrete and integer variables are considered. Then a mixed-integer genetic algorithm, with a number of continuous variable $n_c = 0$, is proposed to solve it.

Each individual in the population is considered as defined with three chromosomes. The genome encoding of the three chromosomes is a discrete value (Figure 7.3).

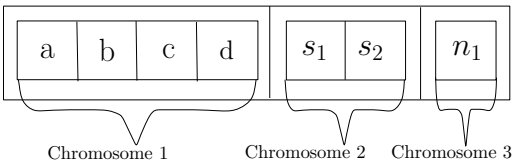


Figure 7.3: Chromosome encoding.

As the reproduction operators [Michalewicz, 1994] depend on the encoding and on the problem, specific crossover and mutation operators are implemented.

At the new generation, the mutation operator makes random changes on some individuals in order to jump outside local optima. In this study, a bit flip mutation which is a two-step process is used. First, the algorithm creates a random binary vector, then, selects genes of an individual for mutation for each bit set to 1. In the second step, the algorithm replaces each selected gene by a new gene selected uniformly from its database (Figure 7.4).

Crossover combines two parents, to form children, for the next generation. A scattered crossover is used. This type of crossover creates a random binary vector. The genes are selected from the first parent for each bit set to 1, and from the second one for each bit set to 0. The genes are combined to form the first child and vice versa to form the second one (Figure 7.5).

Concerning the other GA parameters, the following values are used: population size $N = 60$, maximum number of generations $T = 100$, crossover probability 0.9, binary tournament selection operator is used. If the best objective value is not improved over 10 generations,

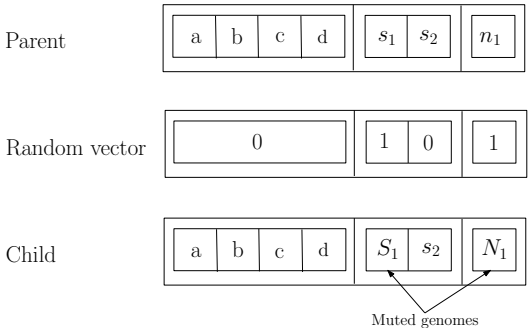


Figure 7.4: Bit flip mutation.

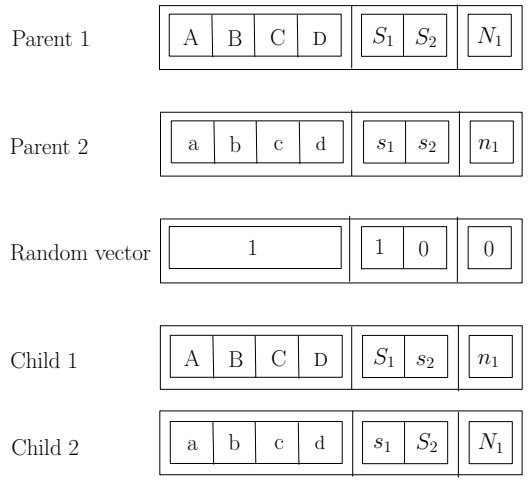


Figure 7.5: Scattered crossover.

the algorithm stops.

As optimization of the safety isolating transformer is a non-linear constrained problem, the external penalty method is used. The objective function and non-linear constraints are combined to form the following sub-problem:

$$\phi(X, \lambda) = f(X) + \lambda \sum_{i=1}^{m} \max^2[0, g_i(X)] \tag{7.2}$$

where $f(X) = M_{tot}(X)$, is the original objective function to be optimized, $g_i(X)$ are the $m = 7$ inequality constraints. The parameter λ is a penalty factor updated at each generation k as follow [Hou *et al.*, 2002]:

$$\lambda(k) = (\frac{2}{1 + e^{ak/T}} - 1)\lambda_0 \tag{7.3}$$

where a is a positive parameter set to 10. $\lambda_0 = 1000$ is the upper limit of $\lambda(k)$, given by the users and must be positive and very large.

Before discussing the optimization results, the exhaustive enumeration (EE) technique is presented. Concerning branch and bound (BB), one version proposed by my colleague TRAN, is used. Author who wants more detail can refer to [Tran *et al.*, 2007c].

Use of exhaustive enumeration

Exhaustive enumeration method (EE) [MathWorks, 1984-2006] is the simplest of the combinatorial optimization techniques. The principle of this method is to evaluate all combinations of the discrete variables. The total number of evaluation n_e is:

$$n_e = \prod_{i}^{n_d} p_i \tag{7.4}$$

where n_d is the number of discrete variables, p_i is the number of values of each discrete variables. The optimal solution is thus the minimum value obtained by scanning the list of all feasible solutions. This method provides the global optimum, but the computational time is huge. To handle the CPU burden of this approach, the Matlab ® distributed computing toolbox and the Matlab ® distributed computing engine [MathWorks, 1984-2006] are used to reduce the computational time.

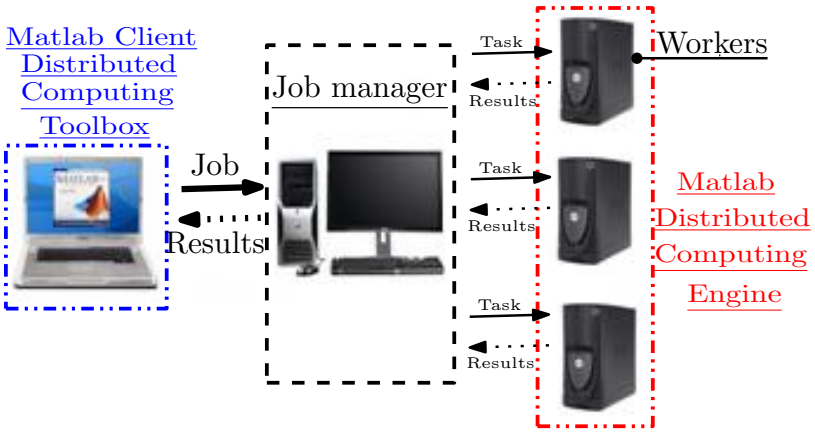


Figure 7.6: Distributed computing description.

Using Matlab ® distributed computing product, the 246.078.000 independent evaluations are executed simultaneously on a cluster of 8 computers. 62 tasks are created for each combination of lamination and frame a, b, c, d. Then, the exhaustive enumeration with the remainder combinations S_1, S_2, n_1 is launched with every combinations (or tasks) a, b, c, d. These tasks form a job solved with a long computational time of 23 days (Figure 7.6).

Optimization results comparison

Mixed-integer genetic algorithm, exhaustive enumeration, branch-and-bound method, and continuous optimization using SQP are examined. For comparison one commercial software called Pro@Design [4] is applied too. The results are presented in Table 7.3.

The convergence of continuous optimization is fast and four constraints are active: temperature of iron T_{ir}, magnetizing current I_μ and the filling factors of the primary and secondary coils. It is obvious that the best solution found by EE method is the global optimum. However, the computational time required is very expensive. The optimization with the commercial software is poor with a total mass 9.8% higher than the global one.

As GA is a stochastic method, some statistics are given. The percentage of convergence is 81% to the best solution found with a Euclidean distance lower than 10^{-4}. In this study all design variables are normalized to $[0, 1]$. The average number of evaluations is 6000 that is 100 generations of 60 individuals.

[4] http://www.designprocessing.com

Table 7.3: Optimization results.

Parameters		Continuous Opt.	Combinatorial optimization			
			ProDesign	EE	GA	BB
a	mm	12.92	18	18	18	18
b	mm	50.12	54	54	54	54
c	mm	16.61	18	18	18	18
d	mm	43.26	33.5	33.5	33.5	33.5
S_1	mm^2	0.325	0.3318	0.2827	0.2376	0.2642
S_2	mm^2	2.912	2.835	2.27	2.835	2.545
n_1	-	640.77	722	610	614	611
M_{tot}	kg	2.311	2.840	2.594	2.633	2.614
$Time$, t	s	4	38	$1.8 * 10^6$	$3 * 10^3$	816
T_{co}	$°C$	108.80	103.60	109.20	106.25	106.48
T_{ir}	$°C$	100.00	94.19	99.58	97.13	97.34

GA ($2.633kg$) gives an acceptable solution very close to the global optimum obtained by EE method ($2.594kg$). Furthermore, all the inequality constraints are satisfied.

Note that, in order to assure the global solution, the BB algorithm is doped up with a strategy verifying the difference between local minima to give the best one.

7.1.3 Outcome

How a standard genetic algorithm can be modified and improved in the aim to deal with mixed-integer design variables is presented. To show the efficiency of the proposed concept, it is tested on the optimal design of a safety isolating transformer.

Then a mixed-integer genetic algorithm, dealing only with discrete and integer design variables, is developed and compared to branch and bound method, and exhaustive enumeration technique.

Note that the mixed-integer GA can be applied on the continuous optimization problems by setting the number of discrete variables to zero ($n_d = 0$).

In the following section, the proposed concept is tested on the case of a multi-objective optimization. It is incorporated in the multi-objective optimization algorithm NSGA-II in the aim to deal with continuous and discrete design variables, in electrical engineering.

7.2 Mixed-integer multi-objective optimization algorithms

In the chapter presented earlier, several multi-objective optimization algorithms are tested, compared and improved on the multi-objective design of a safety isolating transformer, and a brushless DC wheel motor. The optimization results show that the multi-objective algorithm NSGA-II is one of the most efficient multi-objective evolutionary algorithms using elitist approach.

For this reason, in this study, NSGA-II is selected to be modified and improved in order to solve mixed-integer and multi-objective optimization problems.

Indeed, NSGA-II is a powerful algorithm that can be adapted easily to handle mixed-integer multi-objective optimization problems. To test the proposed mixed-integer NSGA-II algorithm, a multi-objective design of Surface-Mounted Permanent Magnet (SMPM) synchronous motor is used as a benchmark.

First, the improved mixed-integer NSGA-II algorithm is discussed. Thereafter, the multi-criteria optimal design of the traction motor is presented.

7.2.1 Mixed-integer NSGA-II

As discussed before, genetic algorithms then NSGA-II manage very well continuous, zero-one, integer and discrete variables.

This section intends to show how NSGA-II can be modified and adapted to handle Mixed-integer multi-objective optimization problems.

Section 7.1 explained that the crossover and mutation operators are the key of the improvement of a standard GA to solve mixed-integer optimization problems. As GA, NSGA-II applies crossover and mutation operators in order to generate offspring population. Then, it is clear that NSGA-II will be modified as explained in section 7.1.1. In other terms, to generate mixed-chromosomes (mixed-genes), both operators will be modified to deal with all types of variables as will be shown in the following.

7.2.2 Mixed-integer optimization design of a traction motor

Design of traction motor [Kreuawan *et al.*, 2007] usually involves trade-off between several objectives. These trade-off solutions, which can be used as support in the decision making, are obtained using a multi-objective optimization (MO) approach. Then, the design problem of a permanent magnet synchronous traction motor, presented here, is formulated as a multi-objective problem.

Indeed, in high competitive market such as the railway market, products must be developed with the lowest cost, the lowest lead time, and the highest quality while satisfying all of customer requirements. Recent research and development in permanent magnet (PM) synchronous motor allows the railway industries to develop and industrialize this kind of motor in their traction application. Two main reasons are its high efficiency and its compactness. Many engineering domains are involved in the design of traction motor for rolling stock and the interaction between traction motor, propulsion system and rolling stock has to be investigated.

Design problem definition

Several objectives are considered important for traction motor design. Three objectives are defined:

1. minimization of mass;
2. minimization of cost;
3. minimization of energy consumption.

Design variables are related to the motor geometry. They include 8 continuous variables (C) such as armature radius, yoke height and 4 integer variables (I) e.g. number of pole pair, number of conductor per slot constrained to a finite number of possible values. The twelve design variables are presented in Table 7.4.

Table 7.4: List of design varaibles.

	Symbol	Description	Unit	Type	Limit
1	Y	Yoke height	mm	C.	$[10, 50]$
2	l_m	PM height	mm	C.	$[1, 20]$
3	g	Airgap	mm	C.	$[1, 10]$
4	d_s	Slot height	mm	C.	$[20, 40]$
5	r_a	Armature radius	mm	C.	$[100, 300]$
6	l_{stk}	Stack length	mm	C.	$[100, 300]$
7	r_{wt}	Tooth width ratio	-	C.	$[0.3, 0.7]$
8	k_p	PM span coeff.	-	C.	$[0.5, 1]$
9	p	Pole pair	-	I.	$\{2, \ldots, 8\}$
10	N_{slot}	Slot/pole/phase	-	I.	$\{1, \ldots, 3\}$
11	N_c	Conductor/slot	-	I.	$\{1, \ldots, 50\}$
12	a	Parallel path	-	I.	$\{1, \ldots, 4\}$

C. = continuous variable; I. = integer variable

The solutions must also satisfy several constraints e.g. maximal winding and PM temperature, maximal flux density in yokes and tooth, maximal rms current, etc. These constraints are imposed by the model's hypothesis, material properties, and the interaction to other component. The maximal rms voltage, material properties, and load profile i.e. required torque and speed as function of time (time dependent inputs) are considered as design parameters. The six inequality constraints are summarized in Table 7.5.

Table 7.5: List of constraints.

	Symbol	Quantity	Unit	Limit
1	T_w	Winding temp.	°C	≤ 200
2	T_{PM}	PM temp.	°C	≤ 150
3	B_Y	Yoke flux density	T	≤ 1.6
4	B_T	Tooth flux density	T	≤ 1.6
5	I_{rms}	rms current	A	≤ 300
6	$Pt_{dem.}$	PM demag. point	-	≤ 0

The multi-criteria design problem can be graphically represented in Figure 7.7. Design variables are defined in the left side, objective functions and constraints are shown in the right side. Parameters are on the top.

This optimisation test problem is therefore a multi-objective, mixed-variable, constrained optimization problem. To solve it, the proposed mixed-integer non-dominated sorting genetic algorithm (Mixed-integer NSGA-II) is used.

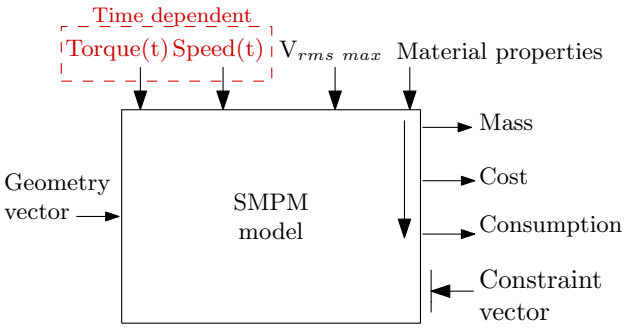


Figure 7.7: Mixed-integer multi-objective optimization of a traction motor.

Use of mixed-integer NSGA-II

As the mixed-integer genetic algorithm proposed in section 7.1.2.0, the NSGA-II is modified and improved in this section.

Each individual in the population is composed of two chromosomes: the first one have real-encoding values, and the second one is a discrete-encoding as shown in Figure 7.8.

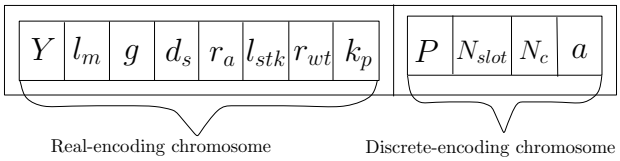


Figure 7.8: Chromosome encoding.

As the reproduction operators [Michalewicz, 1994] depend on the encoding and on the problem, specific crossover and mutation operators are implemented.

Then hybrid crossover and mutation operators are proposed:

- mutation operator combines a bit flip mutation for the discrete-encoding chromosomes, with an auto-adaptive anisotropic mutation applied for the real-encoding chromosome;
- crossover operator combines a scattered and an arithmetic crossovers which are applied for discrete and real chromosomes respectively.

As previous, bit flip mutation is applied for the discrete design variables as illustrated in Figure 7.9.

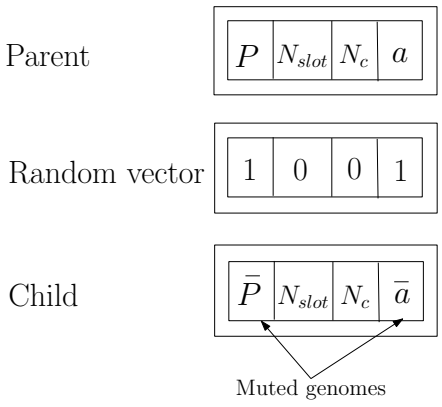


Figure 7.9: Bit flip mutation.

However, for the 8 continuous variables, the auto-adaptive anisotropic Gaussian mutation [Beyer & Deb, 2001] is applied as follow:

$$x_c(i) = x_p(i) + (\sigma_c(i) \cdot N_i(0,1)), \ i = 1, \ldots, 8 \tag{7.5}$$

where, $[x_c(1), \ldots, x_c(8)]$ is the child vector created from the parent vector $[x_p(1), \ldots, x_p(8)]$, and $N_i(0,1)$ is the standard normal distribution of the i^{th} variable. Furthermore, $\sigma_c(i)$ is the standard deviation of the i^{th} variable which is calculated as follow:

$$\sigma_c(i) = \sigma_p(i) \cdot \exp(\tau' \cdot r + \tau \cdot N_i(0,1)), \ i = 1, \ldots, 8 \tag{7.6}$$

where r is a random number. τ and τ' are two parameters fixed to $(2(n_c)^{\frac{1}{2}})^{-\frac{1}{2}}$, and $(2n_c)^{-\frac{1}{2}}$ respectively. $n_c = 8$ is the number of the continuous varaibles.

As in the case of mono-objective optimization, for NSGA-II the scattered crossover is used for the discrete design variables (Figure 7.10).

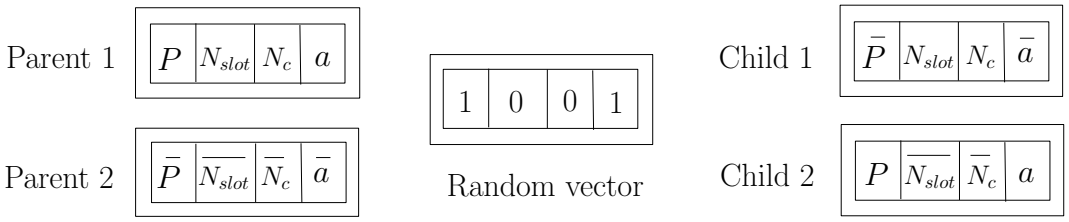


Figure 7.10: Scattered crossover.

However, the continuous part of individuals are crossed using the simulated binary crossover (SBX) [Beyer & Deb, 2001] as follow:

$$\begin{cases} x_{c1}(i) = 0.5[(1+\beta)x_{p1}(i) + (1-\beta)x_{p2}(i)], \ i = 1, \ldots, 8 \\ x_{c2}(i) = 0.5[(1-\beta)x_{p1}(i) + (1+\beta)x_{p2}(i)], \ i = 1, \ldots, 8 \end{cases} \tag{7.7}$$

where β is a dispersion factor calculated as:

$$\beta = \begin{cases} (2r)^{\frac{1}{\eta+1}} & if \ r < 0.5 \\ (\frac{1}{2(1-r)})^{\frac{1}{\eta+1}} & else \end{cases} \tag{7.8}$$

with r is a random variable belonging to the interval $[0,1]$, and η is a non-negative real parameter. A high value of η gives a high probability to create children in the local vicinity of the parents.

Furthermore, the design of the presented traction motor is a hard constrained optimization problem. As highlighted in the previous chapter, a constrained non-domination relation order is included to the proposed mixed-integer NSGA-II algorithm.

The other algorithm parameters are tuned as follow: population size $N = 500$, maximum number of generations $T = 100$, mutation probability 0.1, crossover probability 0.9, and the distribution index for the crossover operator is $\eta = 20$.

Optimization results comparison

The multi-objective optimization takes 8 hours with NSGA-II algorithm. 3D Pareto front and projections of this front in each direction are shown in Figure 7.11. There are mainly four solution groups. Solutions in each group share a common trend in design variable values. The solutions in group 1 and 2 have almost the same design with low conductor number, low armature radius but long stack length. Unlike the first two groups, group 3 and 4 have high armature radius but short stack length. Group 1 has higher PM height value but lower PM span coefficient and thinner airgap compared with other groups.

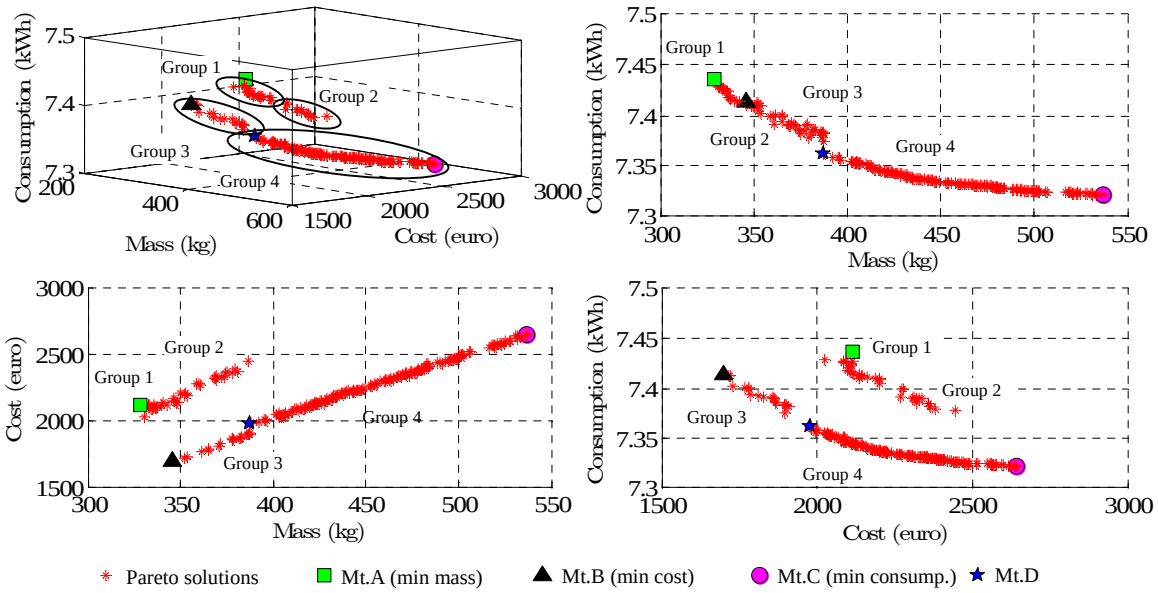


Figure 7.11: Optimization results - Pareto front.

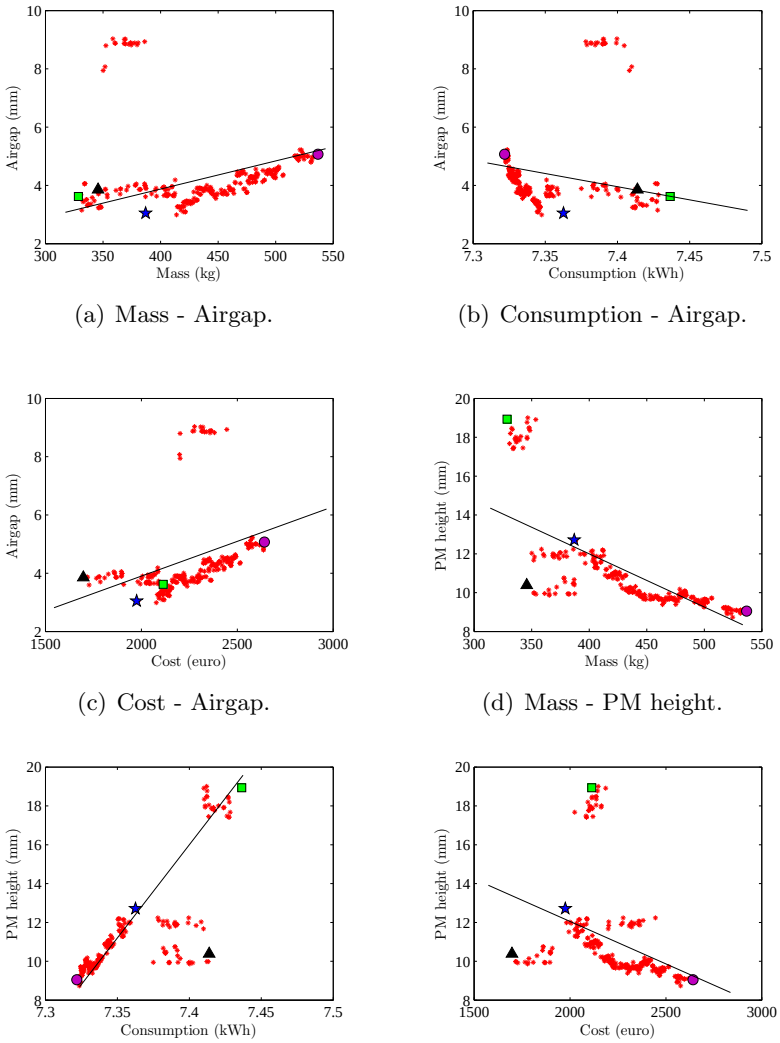
Mass and consumption criteria are clearly conflicting goals. Higher mass motor consumes energy less than motor with low mass. For mass and cost criteria, increase in mass and cost are in the same direction. However, for motors whose mass is lower than $400\,kg$, trade-off exists between the group 2 and 3. Having the same mass, a group 2 motor is more expensive but energy consumption is slightly better than that motors of group 3.

Evolutions of design variables on Pareto front are depicted in Figures 7.12. It is not obvious to clearly identify design criterion trends. They depend on interaction between design variables as well as between other design criteria. Nevertheless, linear global trends are presented in these figures. Without optimal design, designer may have difficulty to find the optimal solutions.

In order to make a decision, four design points are selected among the optimal Pareto solutions: Motor A - minimal mass, Motor B - minimal cost, Motor C - minimal consumption, and Motor D - average mass, cost and consumption.

The comparison of the design variables, constraints, and objectives values of these four solutions are given in Table 7.6 and Figure 7.13. All solutions have 7 pole pairs and 1 slot per pole and per phase. Motor C and D are almost the same design. As motor C is longer, this results in more heat exchange surface than motor D. The temperatures can be kept lower. Combining this effect with a low phase resistance at the low temperature, the high efficiency, hence low energy consumption can be achieved.

Motor A and B are also interesting for traction application due to their low mass regardless their high energy consumption. Rather thick PM (high value of PM height) is used in



(a) Mass - Airgap.

(b) Consumption - Airgap.

(c) Cost - Airgap.

(d) Mass - PM height.

(e) Consumption - PM height.

(f) Cost - PM height.

Figure 7.12: Evolution of design variables on Pareto front.

motor A. It allows lower current compared with other solutions. However, the cost would be higher. One can observe that the flux density in yoke has not reach the constraint limit. In conventional motor design approach, in order to minimise mass, the yoke height has to be minimised. This can be obtained by computing it directly from the allowable flux density. In our case, the temperature constraints are active instead of magnetic constraints. This shows the advantage of using of the thermal model. If the conventional approach is performed, motor could suffer from extremely high temperatures.

The last decision was made. Motor A was chosen using mass and dimension criteria. The energy consumption is a bit higher but this is not crucial because it can be compensated by lower current, hence lower inverter losses. However the cost is much higher than the motor B. It has to be investigated in detail by including manufacturing cost.

Table 7.6: Optimization results.

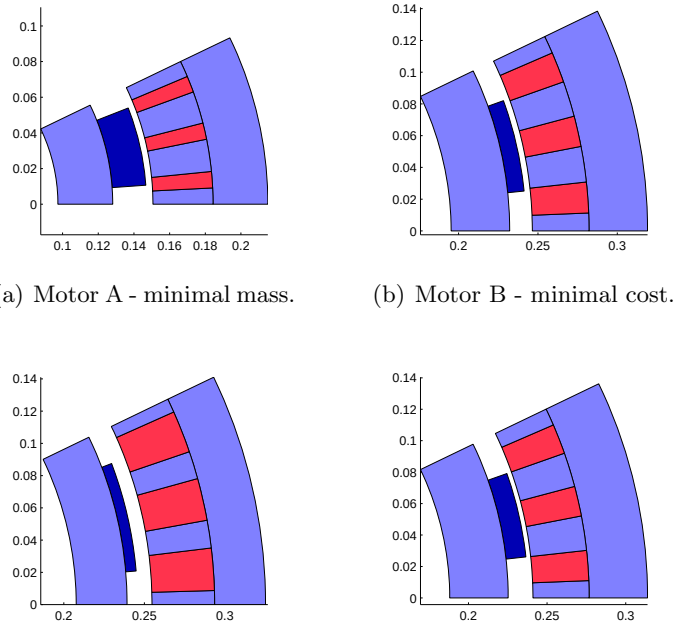
Symbol	Quantity	Unit	Mt.A	Mt.B	Mt.C	Mt.D	Remark	Limit
Y	Yoke height	mm	30.74	36.89	40.19	37.17	D.V.c.	[10, 50]
l_m	PM height	mm	18.93	10.39	9.05	12.71	D.V.c.	[1, 20]
g	Airgap	mm	3.62	3.86	5.07	3.05	D.V.c.	[1, 10]
d_s	Slot height	mm	33.71	35.73	36.77	35.73	D.V.c.	[20, 40]
r_a	Armature radius	mm	150.6	246.4	237.3	241.0	D.V.c.	[100, 300]
l_{stk}	Stack length	mm	265.9	117.4	202.1	139.7	D.V.c.	[100, 300]
r_{wt}	Tooth width ratio	-	0.663	0.536	0.487	0.535	D.V.c.	[0.3, 0.7]
k_p	PM span coeff.	-	0.676	0.538	0.500	0.510	D.V.c.	[0.5, 1]
p	Pole pair	-	7	7	7	7	D.V.d.	[2, 8]
N_{slot}	Slot/pole/phase	-	1	1	1	1	D.V.d.	[1, 3]
N_c	Conductor/slot	-	8	28	10	11	D.V.d.	[1, 50]
a	Parallel path	-	2	4	2	2	D.V.d.	[1, 4]
T_w	Winding temp.	°C	191.4	200	117.7	142.9	Con.	≤ 200
T_{PM}	PM temp.	°C	150	144.8	103	118.1	Con.	≤ 150
B_Y	Yoke flux density	T	1.17	1.19	0.84	1.23	Con.	≤ 1.6
B_T	Tooth flux density	T	1.60	1.48	1.30	1.59	Con.	≤ 1.6
I_{rms}	rms current	A	266.5	286.7	286.7	282.0	Con.	≤ 300
$Pt_{dem.}$	PM demag. point	-	0	0	0	0	Con.	≤ 0
M	Mass	kg	329	346	537	387	Obj.	min
C	Cost	euro	2114	1698	2642	1976	Obj.	min
E	Consumption	kWh	7.44	7.41	7.32	7.36	Obj.	min

D.V. = Design variable; c. = continuous variable; d. = discrete variable
Obj. = Objective function; Con. = Constraint

7.2.3 Outcome

The multi-objective optimization of the traction motor shows the powerful of the proposed mixed-integer NSGA-II algorithm. Indeed, this benchmark is a hard optimization problem dealing with three conflicting objectives: mass, cost and energy consumption minimizations; and mixed-design variables are taken into account: continuous and discrete variables.

Furthermore the proposed algorithm, by mean of the dominance constraint component discussed in the previous chapter, managed very well all the constraints of the optimization problem.



(a) Motor A - minimal mass.

(b) Motor B - minimal cost.

(c) Motor C - minimal consumption.

(d) Motor D - average mass, cost and consumption.

Figure 7.13: Shape of optimal motors in two dimensions.

7.3 Conclusion

This study shows the powerful of the metaheuristics, specially genetic algorithms which are easy to modify with the aim to handle large types of optimization problems.

Indeed, two metaheuristics have been proposed and tested:

- a mixed-integer genetic algorithm which is able to handle combinatorial, continuous and mixed-integer mono-objective optimization problems;
- a mixed-integer NSGA-II (Non-dominated Sorting genetic algorithm) able to handle multi-objective optimization problems with mixed-integer design variables.

Note that beside mixed-integer optimization problems, both proposed algorithms can be used to solve discrete and continuous problems. For example, to deal only with discrete variables, the number of continuous variables is set to zero ($n_c = 0$).

8

Multi-level optimization

As presented in chapter3, analytical target cascading (ATC) is a hierarchical multi-level and multidisciplinary design methodology. It is used to solve optimization problems that have a hierarchical structure.

According to [Tzevelekos *et al.*, 2003] and [Michelena *et al.*, 2003], the theoretical convergence proof for ATC algorithms demonstrated that ATC process is highly sensitive to the numerical inaccuracies in the data communicated among the sub-problems. Indeed, as will be explained in the next section, when dealing with unattainable targets, a strict design consistency can not be achieved. Then the convergence of ATC can not be met. The focus of this chapter is to improve the convergence of the multi-level optimization algorithm ATC by a better management of the consistency constraints.

First the consistency of ATC for unattainable targets is presented. Thereafter, the convergence of ATC and its improvement are discussed. A new formulation is proposed in order to improve the convergence of ATC. As a test case, a non-convex geometric programming problem is solved with the new formulation.

Thereafter an industrial application, from **Alstom-Transport** company, is carried out to design optimally a tram traction system in order to demonstrate the efficiency of the proposed algorithm.

8.1 Consistency for unattainable targets

For the purpose of clarification the ATC sub-problem formulations, described in sections 3.3.3 and 3.3.4 of chapter 3, are rewritten in this section. The system level sub-problem is as follow:

$$\begin{array}{lll} sub-pb_{11}: & \min f_{11} = & \left\|R_{11}^1 - T\right\|_2^2 + \sum_{k\in C_{11}} \left\|R_{2k}^1 - R_{2k}^2\right\|_2^2 \\ & & + \sum_{k\in C_{11}} \left\|S_k Y_{21}^1 - Y_{2k}^2\right\| \\ & \text{with respect to} & X_{11}, Y_{21}^1, R_{2k}^1 \\ & \text{where} & R_{11}^1 = r_{11}\left(R1_{2k}, Y_{21}^1, X_{11}\right) \\ & \text{subject to} & g_{11}\left(R_{2k}^1, Y_{21}^1, X_{11}\right) \leq 0 \\ & & h_{11}\left(R_{2k}^1, Y_{21}^1, X_{11}\right) = 0 \end{array} \tag{8.1}$$

where the objective is to minimize the error between the targets vector T, cascaded from the firm level, and the system responses vector R_{11}^1.

The k^{th} child sub-problems in the lower-level i is set as follow:

$$
\begin{array}{lll}
sub\_pb_{i,j}: & \min & f_{ij} = \left\| R_{ij}^{i} - R_{ij}^{i-1} \right\|_2^2 + \left\| Y_{ij}^{i} - S_j Y_{ij}^{i-1} \right\|_2^2 \\
 & & + \sum_{k \in C_{ij}} \left\| R_{(i+1)k}^{i} - R_{(i+1)k}^{i+1} \right\|_2^2 \\
 & & + \sum_{k \in C_{ij}} \left\| S_k Y_{(i+1)j}^{i} - Y_{(i+1)k}^{i+1} \right\|_2^2 \\
 & \text{with respect to} & X_{ij}, Y_{ij}^{i}, y_{(i+1)k}^{i}, R_{(i+1)k}^{i} \\
 & \text{where} & R_{ij}^{i} = r_{ij}\left(X_{ij}, Y_{ij}^{i}, y_{(i+1)k}^{i}, R_{(i+1)k}^{i}\right) \\
 & \text{subject to} & g_{ij}\left(X_{ij}, Y_{ij}^{i}, Y_{(i+1)k}^{i}, R_{(i+1)k}^{i}\right) \leq 0 \\
 & & h_{ij}\left(X_{ij}, Y_{ij}^{i}, Y_{(i+1)k}^{i}, R_{(i+1)k}^{i}\right) = 0
\end{array}
\tag{8.2}
$$

Note that, all the objective functions are formulated as a weighted sum of deviation metric terms; where the weights are set to 1 and the deviations are measured using the square L-2 norm denoted by $\|.\|_2^2$ (e.g., $\|Y\|_2^2 = \sum_i Y_i^2$). Author who want more details should refer to chapter 3.

According to the state of the art, it is known that for problems with unattainable targets T in the top-level, strict design consistency cannot be achieved with finite weighting factors.

According to [Michelena *et al.*, 2003] the solutions X_t of the system level sub-problem can be visualized as a Pareto front between terms in the objective function f_{11} (Figure 8.1). Furthermore, it has demonstrated that there exist error-zeroing weights that yield solution with lowest consistency deviation as illustrated in Figure 8.1 (proposed by [Michelena *et al.*, 2003]). In other terms, [Michelena *et al.*, 2003] demonstrated that it is possible to find weights W_{2k}^{R} and W_{21}^{Y} leading to zero deviations called ε^{R} and ε^{Y} for responses and linking variables respectively.

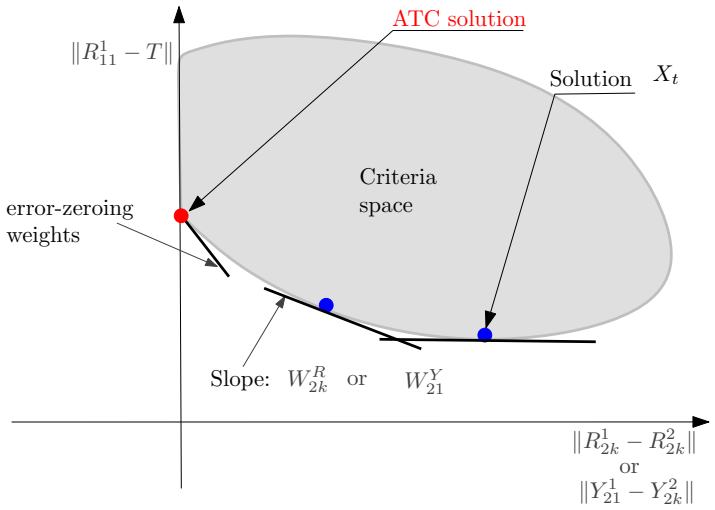


Figure 8.1: Pareto front between consistency deviation and target deviation.

In order to ensure consistency of ATC when the targets are unattainable, derivatives of ATC are proposed. Then [Michalek & Papalambros, 2005a] proposed a weighting update method to identify the appropriate weighting factors to achieve designs with an acceptable deviation tolerance. After that, [Kim *et al.*, 2006a] developed a modified Lagrangian dual formulation and coordination for ATC, where the Lagrange multipliers can be viewed as the weights for deviations.

8.1.1 Weighting update method

[Michalek & Papalambros, 2005a] proposed a weighting update method (denoted in this study WUM) to identify appropriate weighting factors to achieve designs with an acceptable deviation tolerance. For the purpose of clarification, as in chapter 3, only two levels: with one parent and two children sub-problems are considered in order to explain the WUM formulation. The example of the Figure 8.2 is used for illustration.

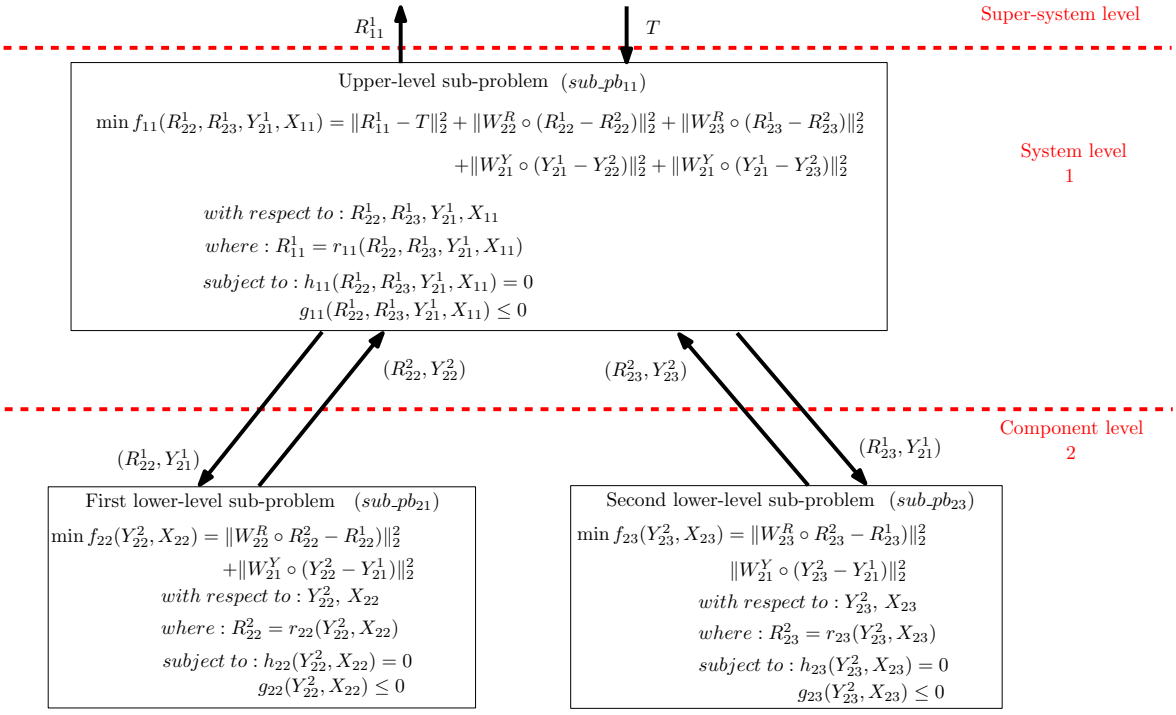


Figure 8.2: Weighting update method formulation.

Where R^1_{22}, R^1_{23}, and R^2_{22}, R^2_{23} are the parent and children responses vectors respectively. Y^1_{21}, and Y^2_{22}, Y^2_{23} are the parent and children linking variables vectors respectively. The symbol $\circ$ is used as a term-by-term multiplication of vectors.

With the WUM, a large scale problem is decomposed into several levels, where each sub-problem sub_pb_{ij} at a level i is decomposed into c_{ij} children sub-problems.

First, the system level sub-problem formulation is presented. Thereafter, the general WUM notation for a single ATC element sub_pb_{ij} in the hierarchy is given. For more details, readers should refer to chapter 3.

System level sub-problem

The optimization problem at the system level is to minimize the error between the targets T, cascaded from the entreprise level, and system responses R_{11}^1. It is defined as:

$$\begin{array}{lllll} sub-pb_{11}: & \min & f_{11} = & \left\|R_{11}^1 - T\right\|_2^2 + \sum_{k\in C_{11}} \left\|W_{2k}^R \circ (R_{2k}^1 - R_{2k}^2)\right\|_2^2 \\ & & & + \sum_{k\in C_{11}} \left\|S_k W_{2j}^Y \circ (S_k Y_{21}^1 - Y_{2k}^2)\right\| \\ & \text{with respect to} & & X_{11}, Y_2^1, R_{2k}^1 \\ & \text{where} & & R_{11}^1 = r_{11}\left(R1_{2k}, Y_{21}^1, X_{11}\right) \\ & \text{subject to} & & g_{11}\left(R_{2k}^1, Y_{21}^1, X_{11}\right) \leq 0 \\ & & & h_{11}\left(R_{2k}^1, Y_{21}^1, X_{11}\right) = 0 \\ & & & k \in C_{11} \end{array} \tag{8.3}$$

Note that $k \in C_{11}$, where C_{11} is the set containing the children sub-problems associated to the system level sub-problem sub_pb_{11}.

The terms $\sum_{k\in C_{11}} \left\|W_{2k}^R \circ (R_{2k}^1 - R_{2k}^2)\right\|_2^2$ and $\sum_{k\in C_{11}} \left\|S_k W_{2j}^Y \circ (S_k Y_{21}^1 - Y_{2k}^2)\right\|$ are used in the system objective function to ensure consistency between system and sub-system levels for each child sub-problem of system $k \in C_{11}$. Where S_k is the selection matrix indicating which terms of the parent coordinating the linking variable vector Y_{2k}^1 are relevant to the linking variable vector Y_{2k}^2 at the sub-problem sub_pb_{ij}. The deviations are measured using the square **L-2 norm** denoted by $\|\cdot\|_2^2$ (e.g.,$\|A\|_2^2 = \sum_i A_i^2$).

Single ATC element in the hierarchy

The objective of the j^{th} sub-problem in the i^{th} level, called $sub-pb_{ij}$, is to minimise the discrepancy between its responses R_{ij}^i and its parent sub-problem targets (R_{ij}^{i-1}); and between its linking variables Y_{ij}^i and those of its parent sub-problem Y_{ip}^{i-1}. The index p means that the variable is sent from the parent sub-problem.

Note that R_{ij}^{i-1} and Y_{ip}^{i-1} are the optimum responses and linking variables values found by the optimization of the parent sub-problem in the $(i-1)^{th}$ level.

$$\begin{array}{lllll} sub\_pb_{i,j}: & \min f_{ij} = & & \left\|W_{ij}^R \circ (R_{ij}^i - R_{ij}^{i-1})\right\|_2^2 + \left\|S_j W_{ip}^Y \circ (Y_{ij}^i - S_j Y_{ip}^{i-1})\right\|_2^2 \\ & & & + \sum_{k\in C_{ij}} \left\|W_{(i+1)k}^R \circ (R_{(i+1)k}^i - R_{(i+1)k}^{i+1})\right\|_2^2 \\ & & & + \sum_{k\in C_{ij}} \left\|S_j W_{(i+1)j}^Y \circ (S_k Y_{(i+1)j}^i - Y_{(i+1)k}^{i+1})\right\|_2^2 \\ & \text{with respect to} & & X_{ij}, Y_{ij}^i, y_{(i+1)k}^i, R_{(i+1)k}^i \\ & \text{where} & & R_{ij}^i = r_{ij}\left(X_{ij}, Y_{ij}^i, y_{(i+1)k}^i, R_{(i+1)k}^i\right) \\ & \text{subject to} & & g_{ij}\left(X_{ij}, Y_{ij}^i, Y_{(i+1)k}^i, R_{(i+1)k}^i\right) \leq 0 \\ & & & h_{ij}\left(X_{ij}, Y_{ij}^i, Y_{(i+1)k}^i, R_{(i+1)k}^i\right) = 0 \end{array} \tag{8.4}$$

Note that the superscript $(i-1)$ indicates that the value comes from the system level $(i-1)$, and the index $(i+1)$ means that the optimal values will be sent to the lower-level $(i+1)$.

In order to find the weights $W_{(i+1)k}^R$ and W_{ip}^Y, [Michalek & Papalambros, 2005a] proposed to use the first-order necessary conditions for optimality (discussed in section 2.1.2.0 of the "State of the art").

Then the Lagrangian of the objective function of the sub-problem $sub-pb_{ij}$ described in

(8.4) is written as follow:

$$\begin{aligned} L_{ij} = \; & \left\| W_{ij}^{R} \circ (r_{ij} - R_{ij}^{i-1}) \right\|_2^2 + \left\| S_j W_{ip}^{Y} \circ (Y_{ij}^{i} - S_j Y_{ip}^{i-1}) \right\|_2^2 \\ & + \sum_{k \in C_{ij}} \left\| W_{(i+1)k}^{R} \circ (R_{(i+1)k}^{i} - R_{(i+1)k}^{i+1}) \right\|_2^2 \\ & + \sum_{k \in C_{ij}} \left\| S_j W_{(i+1)j}^{Y} \circ (S_k Y_{(i+1)j}^{i} - Y_{(i+1)k}^{i+1}) \right\|_2^2 \\ & + \mu_{ij}^{T} g_{ij} + \lambda_{ij}^{T} h_{ij} \end{aligned} \tag{8.5}$$

where μ and λ are the vectors of the Lagrange multipliers for the inequality and equality constraints of the sub-problem $sub - pb_{ij}$. Expressing the square L-2 norm with vector elements indexed with the symbol α, equation (8.5) is written as follow:

$$\begin{aligned} L_{ij} = \; & \sum_{\alpha_1} (\left\langle W_{ij}^{R} \right\rangle_{\alpha_1} \left\langle r_{ij} - R_{ij}^{i-1} \right\rangle_{\alpha_1})^2 + \sum_{\alpha_2} (\left\langle S_j W_{ip}^{Y} \right\rangle_{\alpha_2} \left\langle Y_{ij}^{i} - S_j Y_{ip}^{i-1} \right\rangle_{\alpha_2})^2 \\ & + \sum_{k \in C_{ij}} \sum_{\alpha_3} (\left\langle W_{(i+1)k}^{R} \right\rangle_{\alpha_3} \left\langle R_{(i+1)k}^{i} - R_{(i+1)k}^{i+1} \right\rangle_{\alpha_3})^2 \\ & + \sum_{k \in C_{ij}} \sum_{\alpha_4} (\left\langle S_k W_{(i+1)j}^{Y} \right\rangle_{\alpha_4} \left\langle S_k Y_{(i+1)j}^{i} - Y_{(i+1)k}^{i+1} \right\rangle_{\alpha_4})^2 \\ & + \mu_{ij}^{T} \, g_{ij} + \lambda_{ij}^{T} \, h_{ij} \end{aligned} \tag{8.6}$$

The first-order necessary conditions state that at the local solution, the gradient of the Lagrangian with respect to each term α_3 of the responses vector $R_{(i+1)k}^{i}$ that will be sent to the element k is zero:

$$\frac{\partial L_{ij}}{\partial \left\langle R_{(i+1)k}^{i} \right\rangle_{\alpha_3}} = 0 \tag{8.7}$$

Then with WUM, the weights $W_{(i+1)k}^{R}$ are calculated as follow:

$$\begin{cases} \left\langle W_{(i+1)k}^{R} \right\rangle_{\alpha_3} = \left| \dfrac{\psi_{k\alpha_3}}{\left\langle \varepsilon_{(i+1)k}^{R} \right\rangle_{\alpha_3}} \right|^{1/2} \\ \text{where} \\ \psi_{k\alpha_3} = \displaystyle\sum_{\alpha_1} \left((\left\langle W_{ij}^{R} \right\rangle_{\alpha_1})^2 \left\langle R_{ij}^{i-1} - r_{ij} \right\rangle_{\alpha_1} \frac{\partial \left\langle r_{ij} \right\rangle_{\alpha_1}}{\partial \left\langle R_{(i+1)k}^{i} \right\rangle_{\alpha_3}} \right) \\ \qquad -\dfrac{1}{2} \left(\mu_{ij}^{T} \dfrac{\partial g_{ij}}{\partial \left\langle R_{(i+1)k}^{i} \right\rangle_{\alpha_3}} + \lambda_{ij}^{T} \dfrac{\partial h_{ij}}{\partial \left\langle R_{(i+1)k}^{i} \right\rangle_{\alpha_3}} \right) \end{cases} \tag{8.8}$$

where, $\left\langle \varepsilon_{(i+1)k}^{R} \right\rangle_{\alpha_3} = \left\langle R_{(i+1)k}^{i} - R_{(i+1)k}^{i+1} \right\rangle_{\alpha_3}$ are the acceptable deviation tolerance values for the responses. Furthermore, the gradient of the Lagrangian with respect to each term α_2 of the linking variables vector Y_{ij}^{i} for element j is zero:

$$\frac{\partial L_{ij}}{\partial \left\langle Y_{ij}^{i} \right\rangle_{\alpha_2}} = 0 \tag{8.9}$$

Then with WUM, the weights W_{ip}^{Y} are calculated as follow:

$$\begin{cases} \left\langle W_{ip}^{Y} \right\rangle_{\alpha_2} = \max\limits_{\forall j,j' \in C_{(i-1)p}^{\alpha_2}} \left| \dfrac{\psi_{j\alpha_2} - \psi_{j'\alpha_2}}{\left\langle \varepsilon_{ip}^{Y} \right\rangle_{\alpha_2}} \right|^{1/2} \\ \text{where} \\ \psi_{j\alpha_2} = \sum\limits_{\alpha_1} \left(\left(\left\langle W_{ij}^{R} \right\rangle_{\alpha_1} \right)^2 \left\langle R_{ij}^{i-1} - r_{ij} \right\rangle_{\alpha_1} \dfrac{\partial \left\langle r_{ij} \right\rangle_{\alpha_1}}{\partial \left\langle S_i^T Y_{ij}^i \right\rangle_{\alpha_2}} \right) \\ \qquad -\dfrac{1}{2} \left(\mu_{ij}^T \dfrac{\partial g_{ij}}{\partial \left\langle S_i^T Y_{ij}^i \right\rangle_{\alpha_2}} + \lambda_{ij}^T \dfrac{\partial h_{ij}}{\partial \left\langle S_i^T Y_{ij}^i \right\rangle_{\alpha_2}} \right) \end{cases} \tag{8.10}$$

where, $\left\langle \varepsilon_{ip}^{Y} \right\rangle_{\alpha_2} = \left\langle Y_{ij}^i - S_j Y_{ip}^{i-1} \right\rangle_{\alpha_3}$ are the acceptable deviation tolerance values for the responses.

Then the aim of the weighting update method (WUM) is to identify the appropriate weighting factors in order to achieve the ATC designs with acceptable deviation tolerances $\left\langle \varepsilon_{(i+1)k}^{R} \right\rangle_{\alpha_2}$ and $\left\langle \varepsilon_{ip}^{Y} \right\rangle_{\alpha_3}$ for each term of the responses vectors and the linking variables vector respectively. Note that, these deviation tolerance values are set by the user.

The drawback of this method is that the derivatives of the objective functions must be calculated to update the values of the response and the linking variable deviations weighting factors W^R and W^Y respectively. This fact discard this algorithm when derivatives can not be obtained.

8.1.2 Lagrangian dual coordination

In order to enhance the formulation and the coordination proposed earlier in the literature, [Kim *et al.*, 2006a] developed a modified Lagrangian dual formulation and coordination called in this study LDC. As established before, using the first-order necessary conditions (in the convex case), the proposed LDC converges to a global optimal solution with optimal Lagrange multipliers in the dual space [Kim *et al.*, 2006a]. Here these Lagrange multipliers can be viewed as the weights for the deviations.

Furthermore, LDC uses the augmented Lagrangian in order to improve the convergence of the algorithm. For more details, readers should be refer to [Kim *et al.*, 2006a].

For the purpose of clarification, as before, only two levels: with one parent and two children sub-problems are considered in order to explain the LDC formulation. The example of the Figure 8.3 is used for illustration.

The symbol $\circ$ is used as a term-by-term multiplication of vectors.

However, suitable initial values for the Lagrange multipliers λ_k^R and λ_k^Y, $k = \{2, 3\}$ are needed to achieve the optimal designs. This is not obvious when dealing with practical engineering problems.

In the following, an improvement of analytical target cascading convergence is proposed without need to specify neither the derivatives, nor the Lagrangian factors.

8.2 Improvement of analytical target cascading convergence

As explained before, the WUM and LDC methods present the drawback of the need either to calculate the derivatives or to specify the suitable initial values of the Lagrangian multipliers.

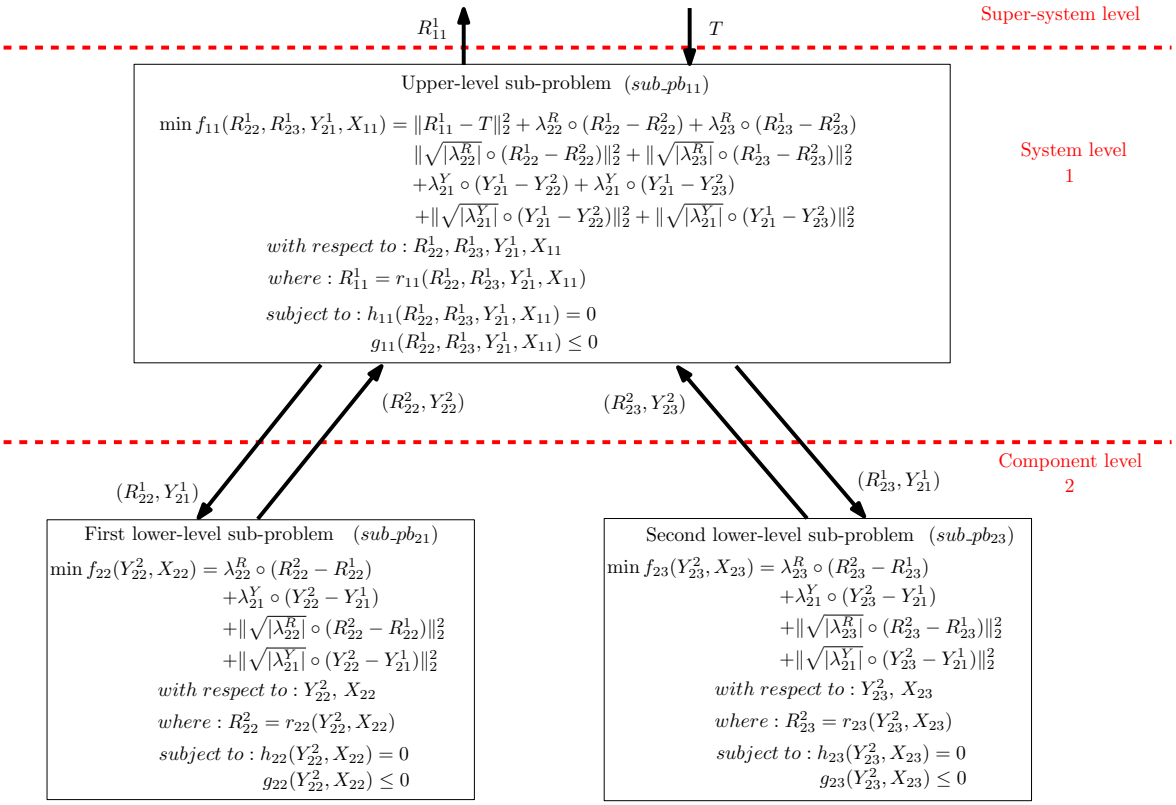


Figure 8.3: Lagrangian dual coordination method formulation.

To overcome these disadvantages, based on the Pareto set illustrated in Figure 8.1, we propose to transform the weighted sum of deviation metric formulation of the system level sub-problem function f_{11} into a multi-objective formulation, called MO, as detailed below. Furthermore, Non-dominated Sorting Genetic Algorithm (NSGA-II) [Deb *et al.*, 2002] is applied to solve the system level sub-problem. The lower-level sub-problems remain mono-objective optimization problems and can be solved with the sequential quadratic programming method (SQP).

8.2.1 Multi-objective formulation

Under the target cascading paradigm, the original problem is decomposed into multiple levels (two or more levels). For simplicity, only the general bi-level case is considered. The original problem is only decomposed into two levels: one system level sub-problem and two lower-level sub-problems as shown in Figure 3.4 with $c_{11} = 2$ is the cardinal of the set C_{11}.

As explained before, to overcome the disadvantages of the previous formulations, the weighted sum of deviation metric of the system level sub-problem is transformed into a

three-objective formulation as follow:

$$\begin{cases} \min \|R_1^{11} - T\|_2^2 \\ \min \sum\limits_{k \in C_{11}} \|R_{2k}^1 - R_{2k}^2\|_2^2 \\ \min \sum\limits_{k \in C_{11}} \|Y_{21}^1 - Y_{2k}^2\|_2^2 \\ \text{with respect to} \quad X_{11},\, R_{2k}^1,\, Y_{21}^1 \\ \text{subject to} \quad g_{11}(R_{2k}^1,\, Y_{21}^1,\, X_{11}) \leq 0 \\ \qquad\qquad h_{11}(R_{2k}^1,\, Y_{21}^1,\, X_{11}) = 0 \\ \text{where} \quad R_1^{11} = r_{11}(R_{2k}^1,\, Y_{21}^1,\, X_{11}) \\ \qquad k \in C_{11},\; k = \{2, 3\} \end{cases} \tag{8.11}$$

where the objective is to minimize simultaneously:

- $\|R_{11}^1 - T\|_2^2$: the deviation between the system targets T and the system responses R_{11}^1 that are functions of the system design variables X_{11} and the c_{11} sub-system responses vectors $R_{2k}^1,\; k \in C_{11}$,
- $\|R_{2k}^1 - R_{2k}^2\|_2^2$: the response's deviation,
- $\|Y_{21}^1 - Y_{2k}^2\|_2^2$: the deviation between the linking variables.

X_{11} are the local variables. The lower-level is composed of c_{11} sub-problems. The optimization problems are stated as follow:

$$\begin{cases} \min \|R_{2j}^2 - R_{2j}^1\|_2^2 + \|Y_{2j}^2 - Y_{21}^1\|_2^2 \\ \text{with respect to} \quad X_{2j},\, Y_{2j}^2 \\ \text{subject to} \quad g_{2j}(X_{2j},\, Y_{2j}^2) \leq 0 \\ \qquad\qquad h_{2j}(X_{2j},\, Y_{2j}^2) = 0 \\ \text{where} \quad R_{2j}^2 = r_{2j}(X_{2j},\, Y_{2j}^2) \\ \qquad j = \{2, 3\} \end{cases} \tag{8.12}$$

where the responses and linking variables targets $(R_{2j}^1,\, Y_{21}^1)$ from the system level system become parameters, and X_{2j} are the local variables of the j^{th} lower-level sub-problem.

Note that the lower-level sub-problems, defined in (8.12), can be decomposed into c_{2j} sub-problems having the formulation given in section 3.3.4 of the chapter 3.

At each ATC iteration, the non-dominated sorting genetic algorithm (NSGA-II) proposed by [Deb *et al.*, 2002] is applied to solve the system level sub-problem; and the sequential quadratic programming (SQP) method is used to solve the problems at the lower-levels.

The new proposed algorithm follows. As shown in Figure 8.4, at each ATC iteration t, a Pareto set is obtained by using NSGA-II to solve the multi-objective optimization sub-problem described in (8.11). Thereafter with respect of a tolerance ε_t, which is decreased at each iteration t, one solution S_t is extracted. The process is repeated until convergence, i.e. an optimal solution is obtained with respect of a minimal tolerance ε.

8.2.2 Proposed algorithm

The ATC algorithm given in [Michalek & Papalambros, 2005a] and [Kim *et al.*, 2006a] is modified with a multi-objective formulation of the system level sub-problem as follow:

1. Set t=0. Choose initial NSGA-II control parameters and initialize the k^{th} lower-level responses vectors R_{2k}^1 and the linking variables vector Y_{2k}^1, where $k \in C_{11}$.

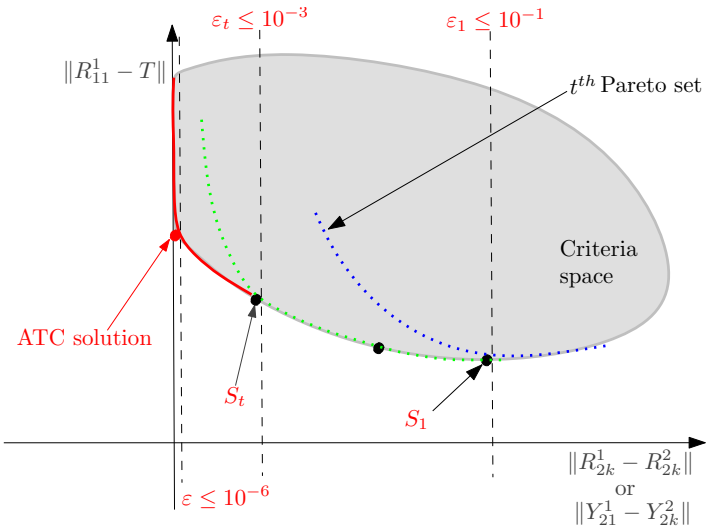


Figure 8.4: Multi-objective formulation: extraction of the system sub-problem solution.

2. Use of NSGA-II to solve the system level sub-problem with respect of X_{11}, R^1_{2k}, Y^1_{2k}, $k \in C_{11}$. A Pareto set is given.

3. With respect of a tolerance ε, where $\|R^1_{2k} - R^2_{2k}\|^2_2 \leq \varepsilon$ and $\|Y^1_{21} - Y^2_{2k}\|^2_2 \leq \varepsilon$, optimal values of $(R^1_{2k}, Y^1_{21}) = argmin(\|R^1_{11} - T\|^2_2)$, $k \in C_{11}$ are extracted from the Pareto front found at step 2 (as shown in Figure 8.4).

4. Pass optimal values of R^1_{2k}, Y^1_{21}, $k \in C_{11}$ to the children lower-level sub-problems.

5. Use of SQP to solve the lower-level sub-problems with respect of X_{2k}, Y^2_{2k}, $k \in C_{11}$ for each one. Optimal values of Y^2_{2k}, $k \in C_{11}$ are given, and the responses R^2_{2k}, $k \in C_{11}$ are calculated.

6. Calculate the deviation for the optimal responses and linking variables: $\begin{bmatrix} \varepsilon^R_{k,t} \\ \varepsilon^Y_{k,t} \end{bmatrix} = \begin{bmatrix} R^1_{ik,t} - R^2_{ik,t} \\ Y^1_{i1,t} - Y^i_{ik,t} \end{bmatrix}$, $k \in C_{ik}$, and$i = 2, \cdots$. if $\|(\varepsilon^R_{k,t}, \varepsilon^Y_{k,t})\| \leq tolerance$, then stop. Where $\|.\|$ denotes the Euclidean distance.

7. Pass the optimal values R^2_{2k}, Y^2_{2k}, $k \in C_{11}$ to system level parent sub-problem.

8. Set $t = t + 1$ and go to step 2.

Note that at the step 5, if the k^{th} sub-problem of the 2^{nd} level is composed of c_{2k} sub-problems, then the optimal values of the linking variables vector Y^2_{3p} and the responses vectors $R^2_{3k'}$, $k' \in C_{2k}$ are given and cascaded to the 3^{rd} level and so one.

As for all other algorithms, our proposed algorithm has advantages and disadvantages. Indeed, with the multi-objective formulation (MO), in one hand, we do not need to specify neither the derivatives like for WUM, nor the initial Lagrangian multipliers like for LDC. In other hand, by way of NSGA-II, it doesn't need to specify a suitable starting point for the ATC optimization process. Furthermore, with the MO formulation the consistency is ensured at each time even if the targets are unattainable.

However, the number of objectives increases when the number of the responses and linking variables, shared between upper and lower-level, increases. To overcome this disadvantage we propose to gather the deviations, then the problem (8.11) can be written as a bi-objective

optimization problem:

$$\begin{cases} \min \|R_{11}^1 - T\|_2^2 \\ \min \sum\limits_{k \in C_{11}} \|R_{2k}^1 - R_{2k}^2\|_2^2 + \sum\limits_{k \in C_{11}} \|Y_{21}^1 - Y_{2k}^2\|_2^2 \\ \text{with respect to} \quad X_{11},\, R_{2k}^1,\, Y_{21}^1 \\ \text{subject to} \quad g_{11}(R_{2k}^1,\, Y_{21}^1,\, X_{11}) \leq 0 \\ \qquad\qquad\; h_{11}(R_{2k}^1,\, Y_{21}^1,\, X_{11}) = 0 \\ \text{where} \quad\; R_{11}^1 = r_{11}(R_{2k}^1,\, Y_{21}^1,\, X_{11}) \\ \qquad\qquad\; k \in C_{11} \end{cases} \tag{8.13}$$

Note that, all the deviation terms must be normalized in order to ensure the balance between them.

8.2.3 Demonstration example

In this section, a non-convex geometric programming problem taken from [Kim *et al.*, 2006a] and [Michalek & Papalambros, 2005a] is solved by using the proposed multi-objective formulation. The original geometric programming problem is formulated as follow:

$$\begin{cases} \min \quad x_1^2 + x_2^2 \\ \text{subject to} \quad x_3^{-2} + x_4^2 - x_5^2 \leq 0 \\ \qquad\qquad x_5^2 + x_6^{-2} - x_7^2 \leq 0 \\ \qquad\qquad x_8^2 + x_9^2 - x_{11}^2 \leq 0 \\ \qquad\qquad x_8^{-2} + x_{10}^2 - x_{11}^2 \leq 0 \\ \qquad\qquad x_{11}^2 + x_{12}^{-2} - x_{13}^2 \leq 0 \\ \qquad\qquad x_{11}^2 + x_{12}^2 - x_{14}^2 \leq 0 \\ \qquad\qquad x_1^2 - x_3^2 - x_4^{-2} - x_5^2 = 0 \\ \qquad\qquad x_2^2 - x_5^2 - x_6^2 - x_7^2 = 0 \\ \qquad\qquad x_3^2 - x_8^2 - x_9^{-2} - x_{10}^{-2} - x_{11}^2 = 0 \\ \qquad\qquad x_6^2 - x_{11}^2 - x_{12}^2 - x_{13}^2 - x_{14}^2 = 0 \\ \qquad\qquad x_i \geq 0,\; i = 1, \ldots,\, 14 \end{cases} \tag{8.14}$$

This problem is broken-down into two levels; thus we have a bi-level hierarchy with two lower sub-problems. Both lower-level sub-problems share one linking variable x_{11} denoted Y_{22}^2, and Y_{23}^2 at the sub-problems $subpb_{22}$ and $subpb_{23}$ respectively. This linking variable is coordinated at the upper-level sub-problem $subpb_{11}$ and denoted as Y_{21}^1.

Furthermore, $x_3 = R_{22}^2$ and $x_6 = R_{23}^2$ are responses calculated in the lower-level sub-problems $subpb_{22}$ and $subpb_{23}$ respectively. Similarly, these responses are denoted at the upper-level sub-problem $subpb_{11}$ as R_{22}^1 and R_{23}^1 for x_3 and x_6 respectively. In other terms, for the same lower-level response, a duplicate variable is created at the upper-level. Thus, with respect of acceptable deviation tolerance values for the optimal responses and linking variables we have: $R_{22}^1 = R_{22}^2$, $R_{23}^1 = R_{23}^2$, $Y_{21}^1 = Y_{22}^2$, and $Y_{21}^1 = Y_{23}^2$.

The system responses x_1 and x_2 denoted as $R_{11}^1(1)$ and $R_{11}^1(2)$ must met the overall target $T = [0,\, 0]^T$.

Note that, the responses x_1, x_2, x_3, x_6 are calculated by mean of the equality constraints of problem (8.14). Figure 8.5 represents the bi-level decomposition of this optimization problem.

To validate the proposed algorithm, in term of accuracy of the solution, the performance of the multi-objective formulation (MO) was compared to the two other approaches: the weighting update method (WUM) of [Michalek & Papalambros, 2005a] and the Lagrangian dual coordination (LDC) of [Kim *et al.*, 2006a]. For a rigorous comparison, WUM and LDC are implemented in this study, and the obtained results are not much different from those obtainned by [Michalek & Papalambros, 2005a] and [Kim *et al.*, 2006a]. Table 8.1 shows the

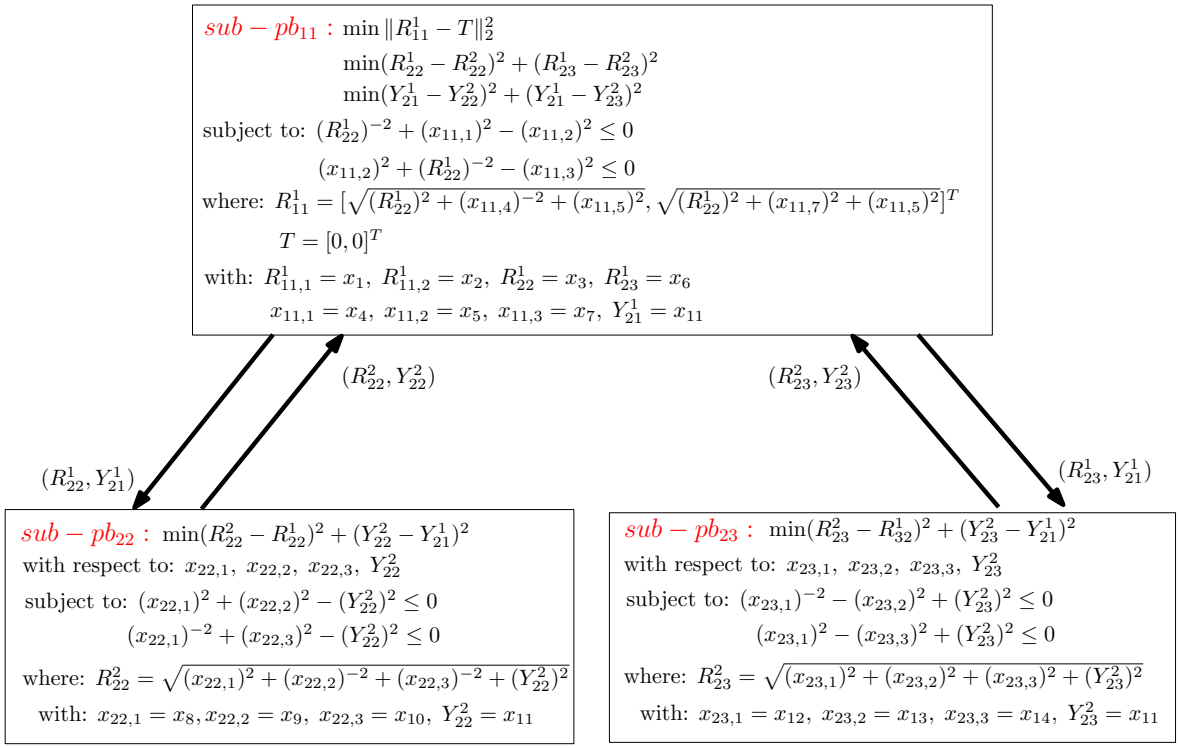


Figure 8.5: ATC decomposition and coordination of the example (8.14).

results obtained by using the WUM, LDC and MO formulations.

The termination *tolerance* of the coordination process was set to 0.01. The optimal design shown in the second column of Table 8.1 is obtained by solving directly the whole problem without its decomposition. For this purpose, SQP is run with

$$X_0 = [5, 5, 2.76, 0.25, 1.26, 4.64, 1.39, 0.67, 0.76, 1.70, 2.26, 1, 41, 2.71, 2.66]^T$$

as the suitable starting point given by [Michalek & Papalambros, 2005a]. For each method, the final design (F. digest) and the error deviation from the optimal design are given.

Because of use of NSGA-II, the optimization results of MO method are less accurate that those of WUM and LDC which use SQP algorithm. The MO solution is less accurate than the WUM solution, and LDC found the most accurate result. Even if the error deviation of MO method is sometimes about 10%, the MO solution converges to the optimal design. It is reminded that the purpose of MO is to find a good solution of an ATC problem with unattainable targets without need of neither Lagrange multipliers nor weight coefficients (or computation of derivatives).

For an efficiency comparison, the computational effort of the three algorithms is taken into account. For the same accuracy ($tolerance = 0.01$), the numbers of function evaluations are similar, when using LDC (about 1300 evaluations) method as well as for WUM with 1251 evaluations. However, with MO formulation the effort is about 5000 evaluations. This is explained by the population size $N = 20$ and the number of generations $gen = 50$ needed by NSGA-II, at each ATC iteration, to converge. Indeed, with MO, 5 Pareto sets are obtained when NSGA-II solves the system multi-objective optimization sub-problem (see Figure 8.4) at each ATC iteration. Figure 8.6 shows the 5 fronts, where each **circle** represents the system

Table 8.1: Optimal design with the three ATC formulations.

Var.	Opt. design	WUM		LDC		MO	
		F. digest	Err. (%)	F. digest	Err. (%)	F. digest	Err. (%)
x1	2.84	2.79	-1.63	2.84	0.12	2.79	-1.59
x2	3.09	3.04	-1.67	3.07	-0.65	3.10	0.34
x3	2.36	2.30	-2.46	2.36	0.19	2.34	-0.61
x4	0.76	0.76	0.00	0.76	0.04	0.84	10.97
x5	0.87	0.88	0.60	0.87	-0.02	0.94	8.74
x6	2.81	2.75	-2.17	2.8	-0.36	2.77	-1.63
x7	0.94	0.95	0.84	0.94	0.07	1.03	9.89
x8	0.97	0.97	-0.47	0.97	0.05	0.97	-0.06
x9	0.87	0.89	3.48	0.86	-1.15	0.87	0.45
x10	0.80	0.82	2.64	0.79	-0.39	0.80	0.33
x11	1.30	1.32	1.29	1.29	-0.77	1.30	0.16
x12	0.84	0.84	0.00	0.84	0.00	0.84	0.00
x13	1.76	1.75	-0.99	1.75	-0.57	1.74	-1.14
x14	1.55	1.53	-1.28	1.58	1.94	1.53	-1.74

level sub-problem solution extracted from each Pareto set according to step 3 of the proposed algorithm given in section 8.2.2. Thereafter, this solution is cascaded to the lower-level to be used as parameters for both sub-problems, and so on until convergence.

In the next section ATC with the multi-objective formulation is tested on the multi-level optimization design of a railway traction system of Alstom-Transport. All the models are presented in [Kreuawan *et al.*, 2009]. Among the optimization method that can be used to solve ATC problems when the targets are unattainable, only MO is able to solve the railway traction problem because the computation of the derivatives for such a system is inaccessible.

8.3 Application case: Object-based decomposition of a railway traction system

In order to demonstrate the efficiency of the new method (MO formulation), it is applied to design optimally a tram traction system of Alstom-Transport. As shown in Figure 8.8, a train can be decomposed into bogie, carriage and traction system. The traction system can be again decomposed into sub-systems such as motor and inverter. By this way, the system forms a multi-level hierarchical structure. The system and each sub-system have their own requirements, targets and constraints, which depend on each other and form the global design problem. Nowadays, system engineer ensures the consistency of the whole system.

The simplified power scheme of tram traction system is shown in Figure 8.7. The pantograph collects electricity from the direct current (DC) overhead supply. The current passes through the line filter and then through the voltage source inverter (VSI) and finally the traction motor [Kreuawan *et al.*, 2008b].

In practical, the being studied tram is composed of two traction boxes. Each traction box contains a line filter, two braking chopper, three VSI, which can drive one motor each, and other electrical equipments e.g. breaker, sensors etc.

In this example, the system level describes the whole traction system. Only two subsystems are considered (dashed line in Figure 8.8):

1. the heat sink of the traction inverter,

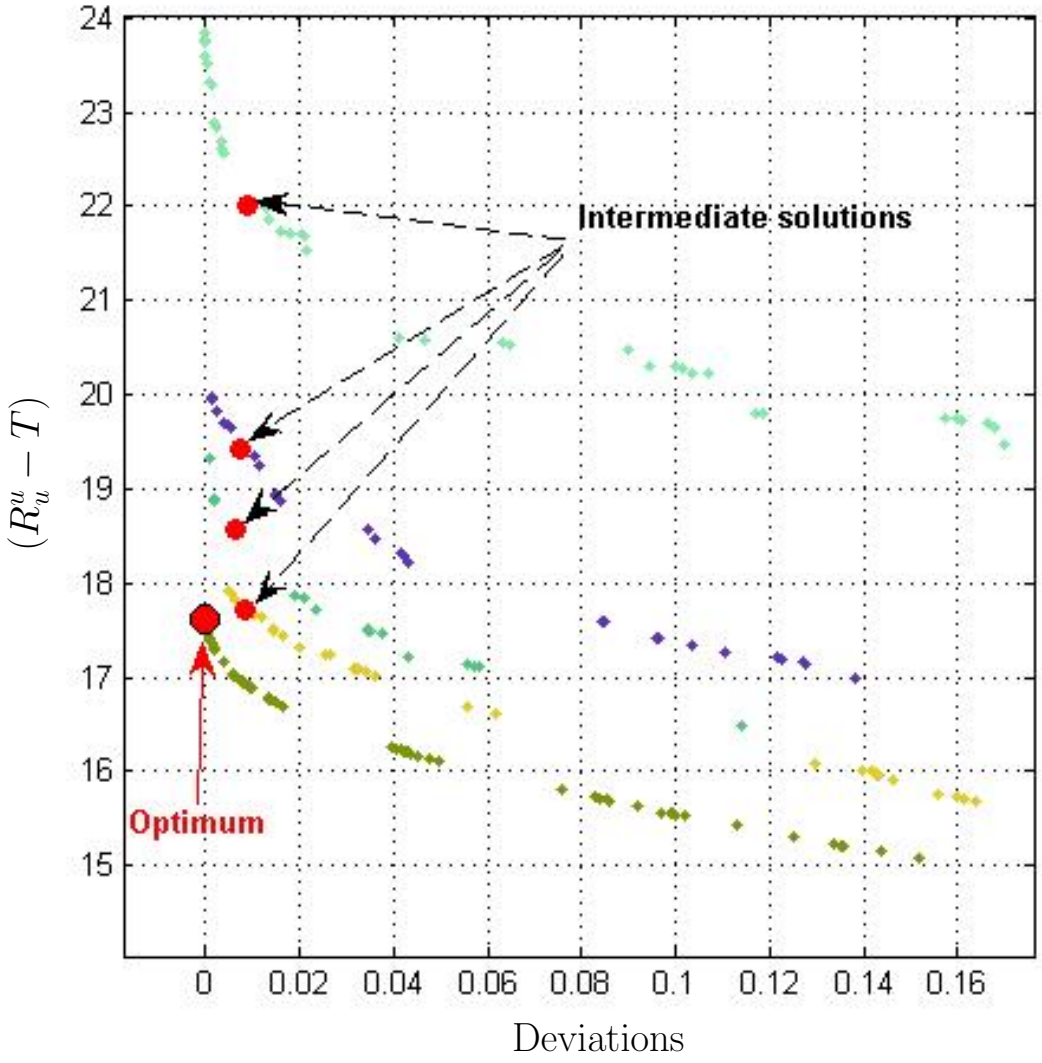


Figure 8.6: Pareto fronts obtained by NSGA-II.

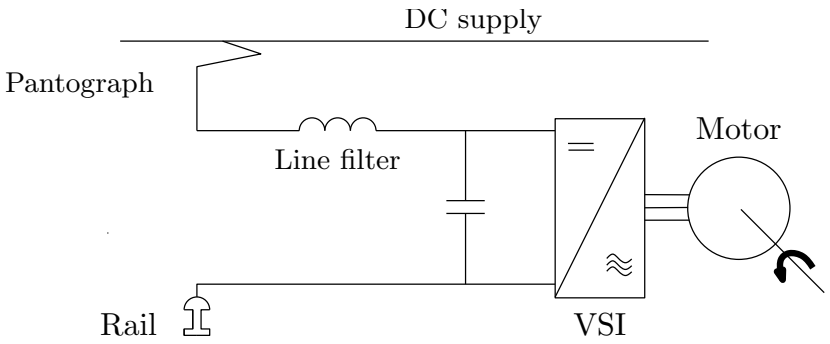


Figure 8.7: Simplified power scheme of tramway traction system.

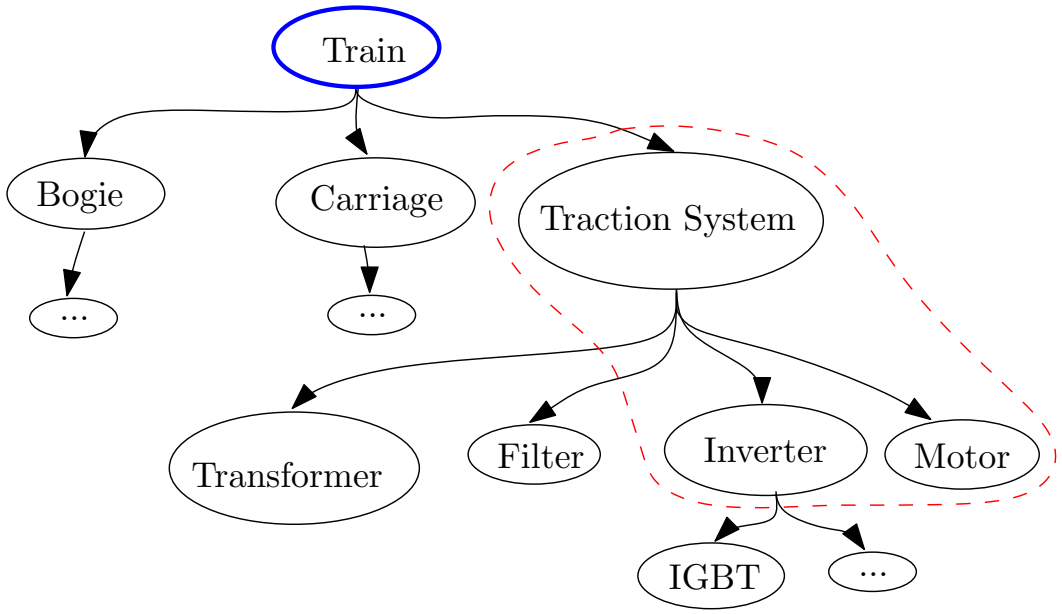


Figure 8.8: Rolling stock system decomposition.

2. and the permanent magnet machine (PMM).

In rolling stock design, the optimization technique has been used to optimize only a sub-system or a component and leads to a sub-optimal system. To optimize the whole system, the system already decomposed needs to be brought back together. The system and sub-system optimization problems can be straightforward coordinated in the same way the system engineer does. This becomes a multi-level hierarchical optimization problem.

As a result of the previous demonstrated example, WUM and LDC give a more accurate solution. But, in electrical engineering domain, giving suitable initial Lagrange multipliers is a complicated task and computing derivatives is very time consuming and often inaccurate. Then for the tram traction system design optimization problem ATC with a MO formulation, which overcomes these disadvantages, is the only one available.

Before the representation of the multi-level optimization of the tram traction system, its modelling is given first.

8.4 Tram traction system modelling

For a hierarchical design problem, several models are needed. The main difficulty is to choose the models that represent the system and its components with suitable details. The system model can be less detailed i.e. presented in a more global point of view and the component models at the sub-system level should typically represent the components in detail and, therefore, provide a higher degree of freedom to the designer. These models are used in a complementary way in order to design the traction system.

8.4.1 Modeling of the traction system

The traction system model is a proprietary code of Alstom Transport–CITHEL. This model integrates the train system, the traction system and its components. Such a model is described with different levels of complexity and formes a system model. It allows simulating the behaviour of a rolling stock operating on a track e.g. a tram on a round-trip (mission of more than 1 hour). It requires a lot of data:

1. the rolling stock characteristics e.g. the aerodynamic properties, and the component parameters,
2. the track characteristics i.e. the track curve, the speed limit, the stop station, the stop time, and the altitude level,
3. the driving performance requirements e.g. the maximal acceleration, and the deceleration.

The simulation, using the CITHEL tool, takes into account the transient thermal behaviour of components such as the motor and the inverter. The electrical computation takes only the steady state into consideration. At each time step, it computes the global dynamic values of the train (speed, acceleration, etc.) and the local electrical, thermal, and mechanical values of the components (line current, IGBT temperature, output torque, etc.).

The main role of CITHEL is to validate the architecture of the traction systems (number of motors, inverters, etc.) and to verify the operation of a train on a track based on the already developed components. The component properties are defined by rather global parameters. For example, the motor model is a single-phase equivalent circuit model and does not depend on the geometry of the motor. The simulation must be performed under many operating scenarios in order to satisfy the specification requirements. Usually, several missions (different tracks) and several operation modes including normal operation and various faulty situations, have to be taken into account. In this work, the traction system must be optimised for these scenarii:

1. the reference tracks,
2. the operation modes: normal operation, 75% motorization and 50% motorization faulty modes,
3. the driving strategies: with and without regenerative braking.

This yields 12 different simulations in total for each design evaluation. These computations take more than one hour. The surrogate model approach [Kreuawan *et al.*, 2008c] is used to decrease the computation time. Figure 8.9 depicts the main idea of a surrogate model approach. The surrogate model uses information from the high fidelity model and provides a cheap-to-evaluate model estimating the high fidelity model. It replaces the high fidelity model in the optimal design process. Readers who want more details should refer to appendix D.

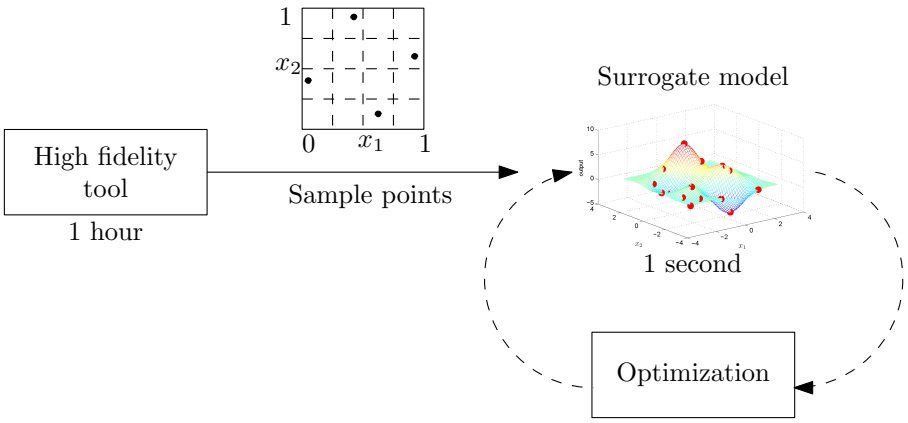


Figure 8.9: Surrogate modelling approach.

In this work, the Kriging surrogate model [Lebensztajn *et al.*, 2004], described in section D.3 of the appendix D, is constructed independently for each output. According to the design problem, there are 5 surrogate models of 7 inputs. The traction system design tool is then replaced by the surrogate model during the optimization process. The computation time decreases from more than one hour to less than one second per design evaluation.

Figure 8.10 depicts the system level model. Beside the 5 Kriging surrogate model, it contains analytical equations that are used in order to compute the total mass.

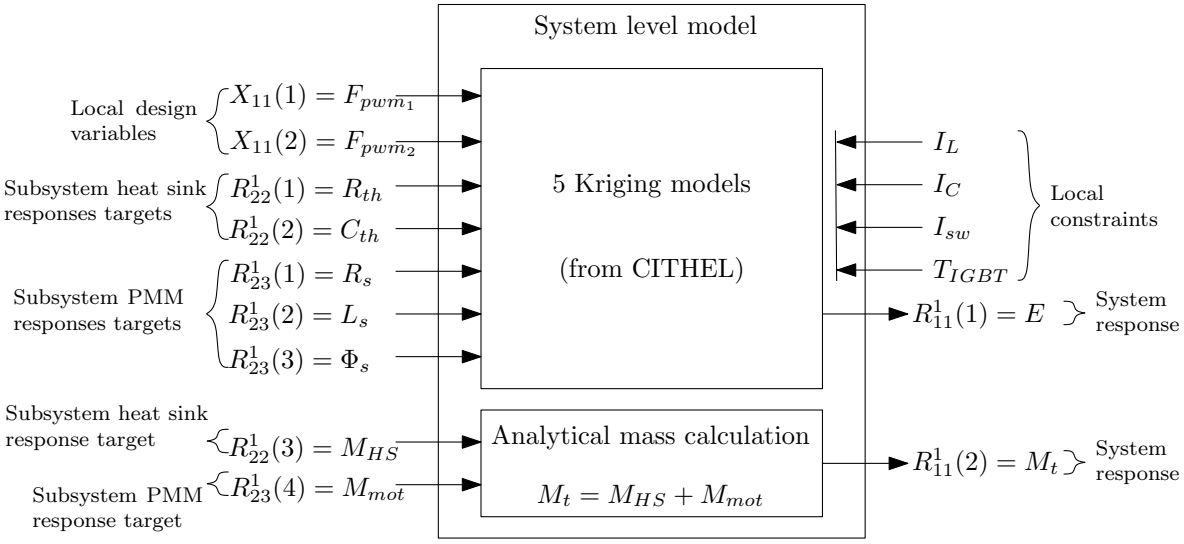


Figure 8.10: System level model.

8.4.2 Modeling of the heat sink

The inverter is equipped with a cooling system to dissipate the heat losses generated by IGBT. In a tram application, the inverter cooling system is typically using air. An air cooling system is more robust, cheaper and simpler than a water cooling system. Even with a fan forced-air convection cooling, its performance is not as good as a water cooling system but it is usually sufficient for this application.

The heat sink model aims at computing the thermal resistance, the thermal capacitance and the mass from the geometry of the heat sink shown in Figure 8.11. These parameters

are used in the traction system design tool in order to simulate the thermal behaviour of an inverter cooling system. The model is based on the theory of a thin plate heat conduction and forced-air convection through fins due to the fan with which the inverter cooling system is equipped [Visser *et al.*, 2000] and [Kim & Nho, 1998]. This model includes 72 equations in total with several if-else conditions depending on experimental coefficients.

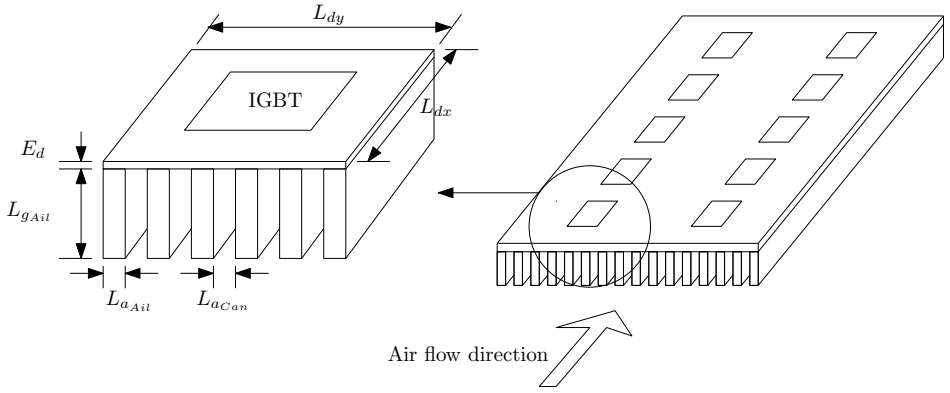


Figure 8.11: Heat sink geometries.

The thermal resistance, thermal capacitance and mass are used in the sub-problem objective function, as described in Table 8.2. Three constraints are defined: pressure loss, air speed between two fins and minimum number of fins.

Inputs and outputs of the heat sink model are shown in Figure 8.12.

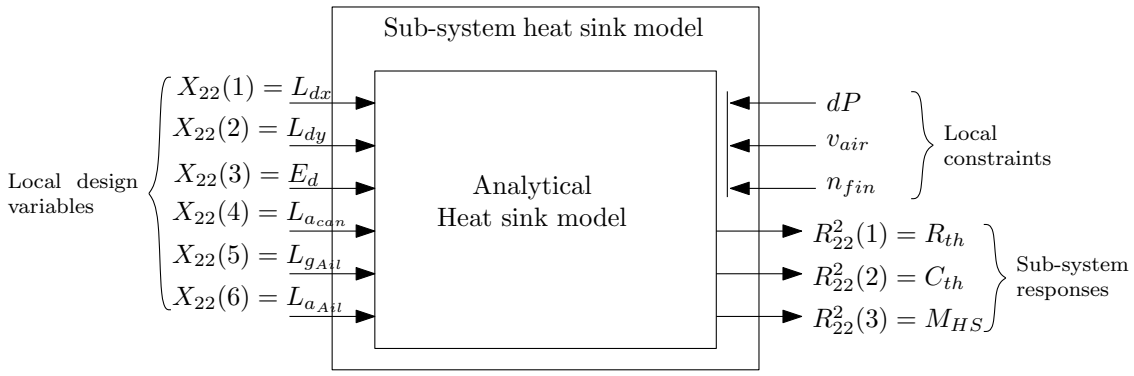


Figure 8.12: Sub-system heat sink model.

8.4.3 Modeling of the traction motor

A design of a PMM implies a multidisciplinary design: magnetic, electric, heat transfer, mechanic, cost, etc. The modelling of PMM is based on a modular approach. Input and output data of each module are defined in the first place. These input and output describe the interactions between modules. Once the input and the output are identified, the model used in each module can be selected on the basis of the need of fidelity. For example, the analytical model or Finite Element Analysis (FEA) can be used in the magnetic or thermal module. In the same manner, the flux-weakening control or the maximal torque per ampere strategy can be used in the control module. The structure of PMM model is shown in Figure 8.13. It consists of eight modules. Each module represents a discipline: magnetic, electric, control, heat transfer etc. Regarding the interactions, there are two temperature feedbacks; (i) the magneto-thermal loop allows taking into account the permanent magnet properties

changes due to the PM temperature changes; (ii) the electro-thermal loop modifies the phase resistance value according to the winding temperature.

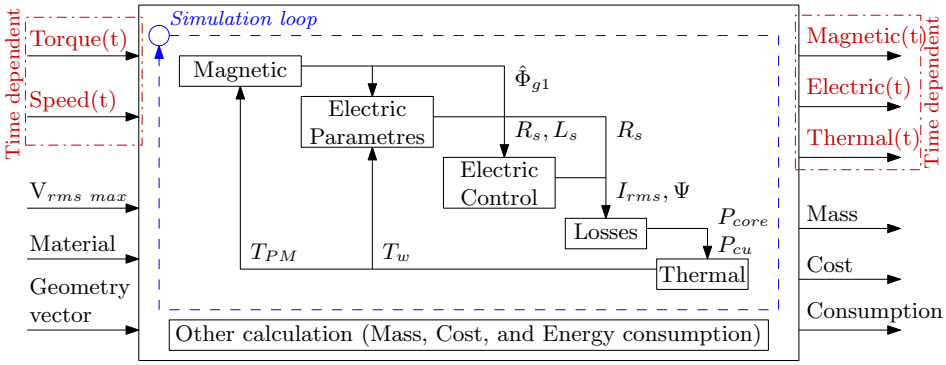


Figure 8.13: Structure of PMM motor model.

In this work, an approach using a semi-numerical model is used. Physical phenomena are described by analytical equations. Thermal interdisciplinary interactions and temperature-material property interactions are solved by an iterative method. This allows fast computation and acceptable accuracy since it is used during the preliminary design phase. The model accuracy and complexity are sufficient to capture interactions between disciplines. The appendix C, extracted from [Kreuawan, 2008] thesis, describes each module in detail. PMM design is based on the nominal operating point approach [Kreuawan *et al.*, 2008a]. This non-linear analytical model comprises more than 150 equations with an iterative thermal loop. In this section, only important points are given, readers should refer to [Kreuawan *et al.*, 2008c] and [Kreuawan *et al.*, 2007] for more details.

Figure 8.14 shows inputs and outputs of the traction motor model. Local design variables are the dimensions of the motor. Constraints are defined in order to satisfy the following requirements:

- the model hypothesis: maximum flux density (linear B-H characteristic),
- the material properties: maximum winding and PM temperatures,
- the failure protection: demagnetization of PM,
- the mechanical interface: external diameter and length,
- the inverter interface: maximum current and voltage.

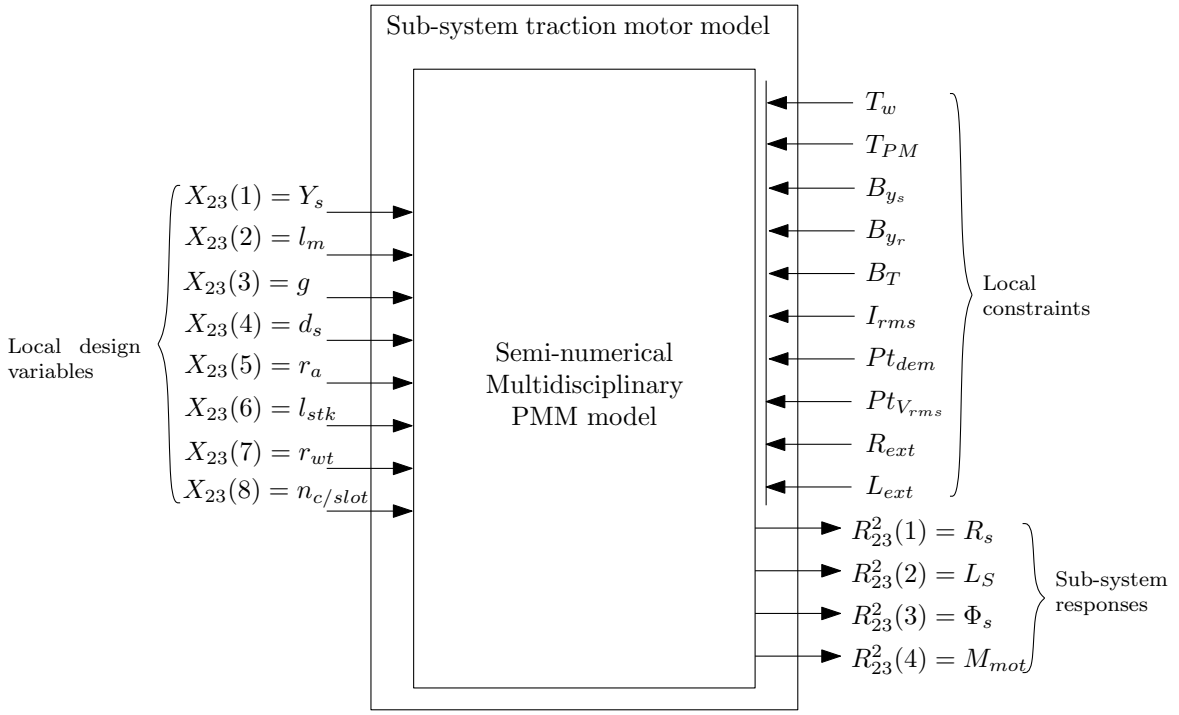


Figure 8.14: Sub-system traction motor model.

8.5 Multi-level optimization design of a tram traction system

8.5.1 Optimization problem formulation

As mentioned before, the overall optimization problem includes three sub-problems: one system level problem (a tram traction system) and two sub-system problems (a heat sink of traction inverter and a permanent magnet machine (PMM)). Two targets are set at the system level: the total energy consumption and the total mass. Note that with this optimization problem the targets are set to unattainable values, i.e. set to zero. Figure 8.15 shows the traction system optimization problem hierarchy.

System level design variables consist of:

1. the system level local variables vector X_{11}: the pulse width modulation (PWM) frequencies (F_{pwm_1} and F_{pwm_2}),

2. the lower-level responses targets vector R^1_{22} that will be sent to the sub-system heat sink: including the thermal parameters (thermal resistor R_{th}, thermal capacitor C_{th}) and heat sink mass M_{HS},

3. the lower-level responses targets vector R^1_{23} that will be sent to the sub-system PMM: electric circuit parameters (phase resistor R_s, inductance L_s, flux Φ_s) and motor mass M_{mot}.

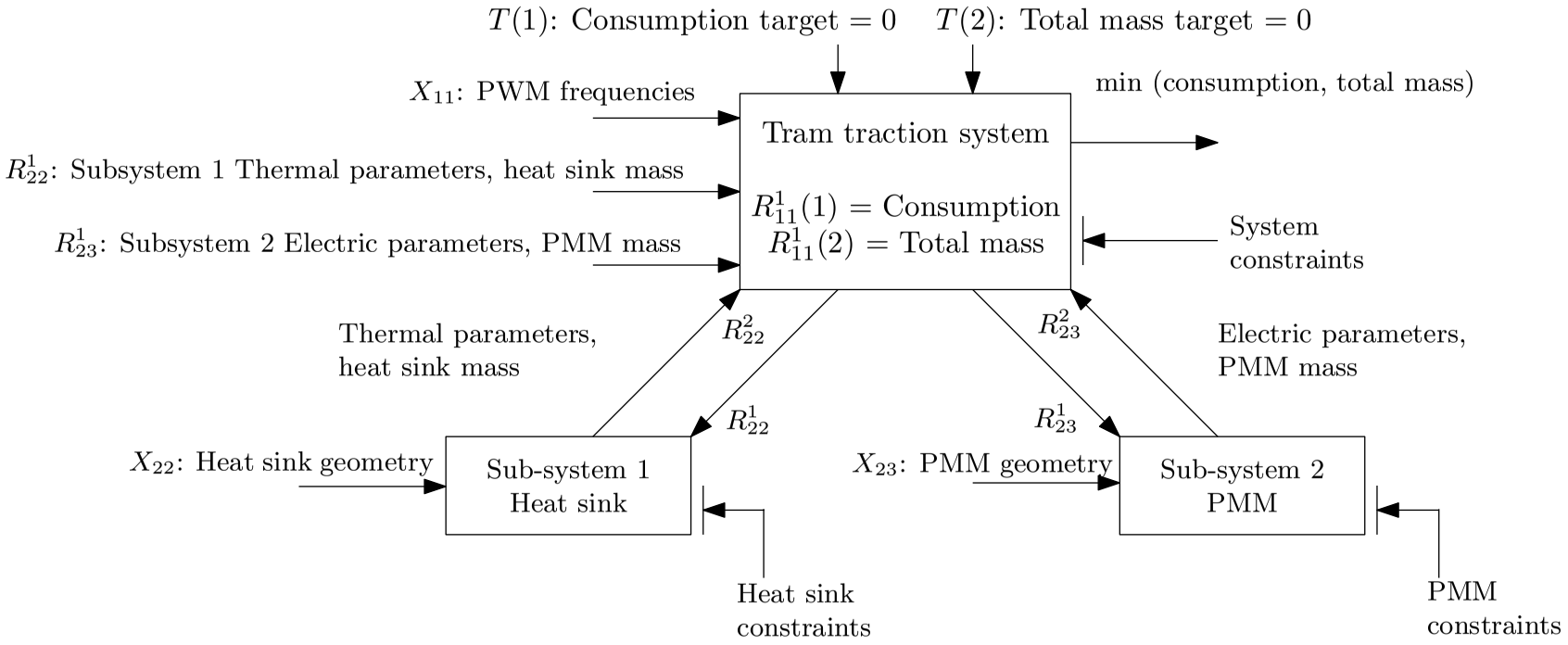


Figure 8.15: Tram traction system optimization problem.

After the first system level optimization, the sub-targets of both sub-systems are then cascaded to sub-system optimization problems, which try to attain these sub-targets by changing their local variables: the motor geometry (8 design variables) and the heat sink geometry (6 design variables).

The sub-systems may not be successful to achieve these targets as the system and each sub-system also have their own constraints (4 in the system level sub-problem, 3 in the Sub-system 1 sub-problem, and 10 for the Sub-system 2 sub-problem).

In this case they return actual calculated lower-level responses to the upper level and the ATC algorithm alternates between upper and lower level optimizations until convergence i.e. sub-targets at system level are equal to lower-level responses computed at the sub-system sub-problems. Table 8.2 summarize the three optimization sub-problems: traction system, heat sink and PMM sub-problems.

Table 8.2: Summary of the 3 optimization sub-problems.

Problem	Find the optimal values of	Responses	Targets	Minimise
System sub_pb_{11}	**- Local design variables:** F_{pwm_1} and F_{pwm_2} **- Heat sink sub-problem responses targets:** R^1_{22} **- PMM sub-problem responses targets:** R^1_{23}	E, M_t	$T(1)$, $T(2)$	$f_{11}(1) = (E - T(1))^2 +$ $(M_t - T(2))^2$ $f_{11}(2) =$ $\|R^1_{22} - R^2_{22}\|_2^2 +$ $\|R^1_{23} - R^2_{23}\|_2^2$
Heat sink sub_pb_{22}	**- Local design variables:** 6 for Geometry	$R^2_{22}(1) = R_{th}$, $R^2_{22}(2) = C_{th}$, $R^2_{22}(3) = M_{HS}$	$R^1_{22}(1)$, $R^1_{22}(2)$, $R^1_{22}(3)$	$\left(R_{th} - R^1_{22}(1)\right)^2 +$ $\left(C_{th} - R^1_{22}(2)\right)^2 +$ $\left(M_{HS} - R^1_{22}(3)\right)^2$
PMM sub_pb_{23}	**- Local design variables:** 8 for Geometry	$R^2_{23}(1) = R_s$, $R^2_{23}(2) = L_s$, $R^2_{23}(3) = \Phi_s$, $R^2_{23}(4) = M_{mot}$	$R^1_{23}(1)$, $R^1_{23}(2)$, $R^1_{23}(3)$, $R^1_{23}(4)$	$\left(R_s - R^1_{23}(1)\right)^2 +$ $\left(L_s - R^1_{23}(2)\right)^2 +$ $\left(\Phi_s - R^1_{23}(3)\right)^2 +$ $\left(M_{mot} - R^1_{23}(4)\right)^2$

8.5.2 Optimization results

ATC alternates between three sub-problem optimizations, the overall targets and system responses progressively get close to each other depending on the sub-targets achievement at the lower level sub-problems.

The ATC process took only 10 iterations. Note that the stopping criterion is set to 1% on the Euclidean distance of the deviations between sub-system targets (given by NSGA-II) and sub-system responses (given by SQP).

ATC with MO formulation gives thus the best compromise between two unattainable targets (set to zero) at the system level according to its formulation. The results are shown in Table 8.3, where the mass and consumption targets are set to unattainable values: $0\,kg$ and $0\,kWh$.

Table 8.3: Traction system level targets and responses.

Description	Symbol	unit	Target	Final results
Total mass	M_t	kg	0	400.40
Consumption	E	kWh	0	194.60

Table 8.4 shows final optimal value of local design variables. At this optimal point, all constraints for each problem are satisfied. The simulation of the whole system shows that the traction system performs well as defined in specification requirements.

Table 8.4: Optimization results - Local optimal design variables.

Sub-problem	Description	Symbol	Unit	Optimal results
System	PWM frequency 1	F_{pwm_1}	Hz	797.04
	PWM frequency 2	F_{pwm_2}	Hz	1711.9
Heat sink	width of heat sink	L_{dx}	mm	493.7
	length of heat sink	L_{dy}	mm	140.9
	Diffuser thickness	E_d	mm	10.7
	Heat sink fin gap	$L_{a_{Can}}$	mm	16
	Fin height	$L_{g_{Ail}}$	mm	32.2
	Fin thickness	$L_{a_{Ail}}$	mm	11.6
PMM	Stator yoke height	Y_s	mm	38.40
	PM height	l_m	mm	15
	Airgap	g	mm	3.86
	Slot height	d_s	mm	33.16
	Armature radius	r_a	mm	118.57
	Stack length	l_{stk}	mm	421.6
	Tooth width ratio	r_{wt}	-	0.6
	Conductor per slot	$n_{c/slot}$	-	4

Note that this optimization problem is not very large. It has 16 design variables and 17 inequality constraints. The complexity of this constrained optimization problem consists on its non-linearity and non-convexity. Furthermore, authors suspect that the feasible search space is very small then it is difficult to find feasible solution either an optimal one. However, solving of large-scale problems with hundreds variables and constraints like in [Wismer, 1971] and [Wagner & Papalambros, 1993] is not in the scope of this study.

8.6 Conclusion

To improve the convergence of analytical target cascading (ATC) in the case of unattainable targets, the weighted sum of deviation metric formulation is transformed into a multi-objective formulation. Then Non-dominated Sorting Genetic Algorithm (NSGA-II) is applied to solve the system level sub-problem. A mathematical benchmark is used to compare the three formulations: weighting update method, Lagrangian dual coordination and the multi-objective formulation. It is shown that target cascading algorithm with the multi-objective formulation finds the optimal solution as well as other both formulations without use of neither the Lagrangian coefficients nor the weighting factors.

ATC with the multi-objective formulation is used for the optimization of a tram traction system of Alstom-Transport. This MO formulation doesn't use derivatives and that's a key-point in electrical engineering where derivatives are very time consuming and often inaccurate.

The great advantage of ATC is that it allows doing the system optimization as a whole i.e. the tram system PWM parameters, heat sink, and PMM geometries are determined concurrently. Regarding the optimization problem formulation, it is simpler to construct many decomposed problems than a global and a complex problem. The full complete system model has not been to be set-up. In fact, it seems a very difficult modelling task. Thus, applying ATC is better when mapping with the organization structure of Alstom-Transport company. In this example, the object-based decomposition is already defined and practical in engineering departments while a system optimization is a novel approach. The problem can be formulated in almost the same way engineers usually do with slightly adaptation. To note also that the resolution of many small optimization problems are easier than solving a large problem. Furthermore, the optimization algorithm is less sensitive to the initial point.

Conclusion

9

Conclusion

9.1 Summary of contribution

This thesis presented the basic concepts of complex system design and how the engineers can lead to an optimal design. As presented throughout this study all the used practical test cases have at least one of the following characteristics:

- non-linear optimization problem with several local optima (multi-modal),
- complex and very hard constrained optimization problem. For example with numerous design variables and constraints, where the research space is very small,
- time-consuming due to the use of fine models like finite element models,
- having many objectives to be considered simultaneously.

Furthermore, there is no optimization method, used alone, which can overcomes these difficulties and deals with such complexity. Then, based on the literature, a set of algorithms are selected to be studied, modified, improved, and combined in order to improve the accuracy of the obtained solutions and the time of convergence.

Furthermore, the selected algorithms are compared on the optimization of several practical test cases. In this thesis, these algorithms are classified into two categories:

- Mono-level optimization algorithms that consider an optimization problem as a whole. This kind of algorithms is used to solve components of a railway traction system.
- Multi-level optimization methods for handling complex optimization problems by their decomposition into several small sub-problems, including the system level sun-problem, that are easier to be solved. The solution is then composed of the sub-problems' solutions. These algorithms are applied to solve the railway traction system itself.

For the mono-level optimization, the following algorithms are selected:

1. in order to solve mono-objective optimization problems:
   - Genetic Algorithms (GA),
   - Ant Colony Optimization (ACO),
   - Particle Swarm Optimization (PSO),
   - Sequential Quadratic Programming (SQP)
2. in order to handle multi-objective optimization problems:
   - Non-dominated Sorting Genetic Algorithms (NSGA-II),

- Strength Pareto Evolutionary Algorithm (SPEA2),
- Traditional weighted sum of objectives (WS),
- Adaptive weighted sum of objectives (AWS)

Concerning the multi-level optimization several derivatives of the Analytical Target Cascading (ATC) method are tested:

- Traditional fixed weights,
- Weighting updated method (WUM),
- Lagrangian dual coordination method (LDC).

The metaheuristics GA, ACO, and PSO, that are considered as global methods, are chosen for their stochastic search component which is appropriate in the case of noisy functions. The deterministic local search algorithm SQP is selected because it represents the state of the art in non-linear programming methods.

In chapter 4, the parameters of the algorithms are tuned, especially the genetic algorithms that have a large number of parameters. The study shown that GA, ACO, and PSO are powerful global optimization methods especially when they are well managed and their parameters are optimally tuned.

This chapter took in hand the weight of the initial conditions and setting of the optimization algorithms. The SQP and GA algorithms are performed several times with different starting points. Three approaches are used to sample these points:

1. Monte Carlo Sampling (MCS),
2. Latin Hypercube Sampling (LHS),
3. Grid Sampling (Grid).

The LHS and grid sampling are the preferred techniques.

In order to overcome the dependence of SQP on the start point, this study advises to use the multi-start technique. This consists on the restart of SQP with several initial points, and the best solution among the others is selected.

One efficient way to overcome the drawback of the parameters tuning and the huge convergence time of the metaheuristics consists in their combination with local search methods like SQP. In this study:

- GA is combined with SQP in a consequence way to form an hybrid method called GA & SQP,
- ACO uses SQP in its local search procedure to form the so called generalized ACO (GACO).

Using the optimal tuning of control parameters, in chapter 5, all algorithms GA, GACO, PSO, SQP, and hybrid method (GA & SQP) are compared in term of the solution accuracy and the computation time. Furthermore, it is shown that using SQP, it is not necessary to provide a formal calculation of the derivatives because the finite difference technique with an adaptive step gives better optimization results. The optimization comparison is done on the mono-objective optimization design of a brushless DC wheel motor.

Beside the parameters tuning of the mentioned algorithms, and their comparison to each other, this thesis answered the following new issues:

- give of suitable multi-objective optimization algorithms in order to fast build a Pareto front with accuracy;
- the proposition of a suitable approach in order to handle various kinds of variables and constraints for the electrical optimization problems;
- the improvement of the convergence of ATC which is a powerful strategy for the complex system optimization.

In chapter 6, two evolutionary multi-objective optimization algorithms NSGA-II and SPEA2 are tested and compared to the traditional weighted sum of objectives WS and the adaptive one AWS. The comparison is did using some quality metrics to assess just one criteria like:

- the convergence to the Pareto-optimal front,
- having a uniform distribution of the Pareto front,
- and having a better coverage of the objective space.

The four algorithms are compared on the multi-objective design of a brushless DC wheel motor. The results shown that AWS and NSGA-II are the preferred algorithms because both are able to produce well-distributed solutions among the Pareto front and found solutions in non-convex regions.

The second part of this chapter is devoted to answer the question: how it is possible to build a Pareto front dealing with 3D finite element models in a short time? To do, the use of technique that is able to reduce the computational time seems a good idea. Then a part of this answer is the use of the Output Space-Mapping technique (OSM) that combines coarse and fine models in order to reduce the computational time and increase the accuracy of the solutions. However, this technique is only applied in the case of a mono-objective optimization.

Thus, an original optimization algorithm is proposed which combines the OSM technique with the multi-objective optimization algorithm ϵ-constraint. The multi-objective optimization algorithm ϵ-constraint is chosen because it is a simplified method that requires few objective function evaluations compared to NSGA-II and AWS.

How to deal with the constraints is also one of the scopes of this thesis. Indeed, GA, ACO, PSO, SPEA2 and NSGA-II are essentially unconstrained optimization methods. In order to solve constrained optimization problems, an adaptive and dynamic penalty approach is implemented and added in the optimization algorithms ACO and PSO. Concerning GA, the augmented Lagrangian is applied.

For the multi-objective algorithms SPEA2 and NSGA-II the following approach are used:

- constrained non-domination relation,
- and the Constraint-First-Objective-Next model.

The study reveals that the selection criterion using both techniques gives better results than the penalty approaches.

In chapter 7, how genetic algorithms can be used to solve optimization problems with continuous, integer, discrete, and zero-one design variables is discussed. A new algorithm, which combines two kind of crossover and mutation operators, is proposed. Indeed, for GA, both operators represents the mean to generate the offspring. Then with the proposed algorithm:

1. continuous operators are applied for the continuous design variables,
2. binary operators are used in order to deal with the discrete, integer, and zero-one variables.

The proposed algorithm is tested on the combinatorial optimization design of a safety isolating transformer dealing with discrete and integer design variables.

Thereafter, by applying the same concept, the multi-objective optimization algorithm NSGA-II is modified and improved in order to solve mixed-integer optimization problems. The proposed algorithm, called "Mixed-integer NSGA-II", is tested on the multi-objective optimization design of a traction motor that deals with continuous and integer design variables.

Note that all the results, presented so far, are published in three international journals [Moussouni *et al.*, 2007], [Moussouni *et al.*, 2008] and [Kreuawan *et al.*, 2008b].

As introduced in this thesis, experience and knowledge of only one expert are not enough to understand and deal with complex engineering systems. An approach to overcome these challenges is to decompose the whole problem into smaller ones that are easier to be solved. Then the system solution is composed of the component's solutions.

Analytical target cascading (ATC) is a hierarchical multi-level and multidisciplinary design methodology. It is used to solve optimization problems that have a hierarchical structure by coordinating their decoupled sub-problems.

The contribution of this thesis is to improve the convergence of ATC by mean of the best management of the consistency constraints. Indeed, chapter 8 shown that ATC process is highly sensitive to the numerical inaccuracies in the data communicated among the sub-problems. As it is known that for problems with unattainable targets in the top-level, strict design consistency cannot be achieved with finite weighting factors.

Then, an original algorithm is proposed (MO) to overcome these difficulties. The improvement consists in the proposition of a multi-objective formulation of the top-level optimization sub-problem and use of the multi-objective optimization algorithm NSGA-II to solve it.

The proposed algorithm is compared with the Weighting updated method (WUM) and the Lagrangian dual coordination method (LDC) on the multi-level optimization design of a tram traction system.

The results shown that with the proposed multi-objective formulation (MO):

- it is not necessary to specify neither the derivatives like for WUM, nor the Lagrangian factors like for LDC. In other hand,
- by mean of NSGA-II, it does not need to specify a suitable starting point for the ATC optimization process,
- the consistency is ensured at each time even if the targets are unattainable.

Note that these optimization results, are published in the international journal *COMPEL* [Moussouni *et al.*, 2009].

Throughout part II, "Improvement of algorithms", practical benchmarks from electrical engineering domain are employed in order to establish the important concepts introduced before. The proposed algorithms are compared and test on the design optimization of:

1. Railway traction system components:
    - a mono-objective and a multi-objective optimization design of a brushless DC wheel motor;

- a combinatorial and a multi-objective optimization design of a Safety isolating transformer dealing with 3D finite element models;
- a mixed-integer multi-objective optimization design of a traction motor.

2. a multi-level optimization design of a railway traction system.

Table 9.1 summarizes all the studied, modified and improved optimization algorithms in this thesis. This results on an optimization toolbox. Table 9.2 summarizes all the practical test cases that are applied in order to show the robustness of the proposed algorithms.

In the following, we tray to give the points that need to be improved, i.e. the future search of further work.

Table 9.1: Optimization toolbox.

Mono-level optimization		
Design variables	**Mono-objective optimization**	**Multi-objective optimization**
- Continuous	- Genetic algorithms (GA) - Generalized ant colony optimization (GACO) - Particle swarm optimization (PSO) - SQP combined with the multi-start technique - Hybrid method GA & SQP	- Non-dominated Sorting Genetic Algorithm (NSGA-II) - Strength Pareto Evolutionary Algorithm (SPEA2) - Adaptive weighted sum of objectives (AWS) - ϵ-constraint
- Discrete	- Mixed-integer GA	- Mixed-integer NSGA-II
- Mixed	- Mixed-integer GA	- Mixed-integer NSGA-II
Constraints handling	- Adaptive and dynamic penalty function - Augmented Lagrangian approach	- Constrained non-domination relation - Constraint-First-Objective-Next model
Multi-level optimization		
- ATC: - Traditional fixed weights - Weighting updated method (WUM) - Lagrangian dual coordination method (LDC) - Multi-objective formulation method (MO)		

Table 9.2: Constrained optimization benchmarks.

Mono-level optimization		
Design variables	**Mono-objective optimization**	**Multi-objective optimization**
- Continuous	- Brushless DC wheel motor - Simplified model of a railway traction system	- Brushless DC wheel motor - Safety isolating transformer
- Discrete	- Safety isolating transformer	- Safety isolating transformer
- Mixed	- Safety isolating transformer	- Traction motor: a permanent magnet machine (PMM)
Multi-level optimization		
- Railway traction system decomposed into two components: - the heat sink of the traction inverter, - and the permanent magnet machine (PMM).		

9.2 Future research of further work

This thesis has shown the interest of the use of hybrid methods. Indeed, the global algorithms GA and ACO are combined with the local search SQP in order to improve the accuracy of the obtained solutions and the time of convergence. The OSM technique is combined with the ϵ-constraint in order to build an accurate Pareto front in a short time. In this way, a further work is to investigate in the combination of different optimization methods in order to exploit the advantages of each one.

In the case of the multi-objective optimization, some metrics are used in this study in order to assess the quality of the obtained Pareto solutions. A further work consist to use these metrics as stopping criteria for the multi-objective optimization algorithms. Then the algorithms stop when solutions with a good quality, according to the used metric, are obtained.

Furthermore, the way that the constraints are handled in the multi-objective GA is very important. In this thesis two powerful techniques are applied: the Pareto dominance model and the Constraint-First-Objective-Next model (CFON). However, using the Pareto dominance as the selection criterion, it is hard to maintain a balance between infeasible and feasible solutions in a population. The drawback of the CFON model is the number of parameters to be tuned in order to obtain good approximation of the Pareto set. Thus, additional mechanisms have to be used and added in order to improve the effectiveness of the constraint handling approaches.

As shown in this thesis, the convergence of the multi-level optimization ATC is improved by using a multi-objective formulation. An other way to improve this convergence consists in the use of the ACO algorithm. Indeed, ACO is a multi-agent algorithm which is able to deal with different sub-problems in the same time. Then a further work is to modify and improve ACO in order to solve hierarchical optimization problems.

To conclude we can say that the analytical target cascading (ATC)is a powerful method which is able to handle complex systems decomposed by objects or disciplines. Furthermore, with ATC, the optimization problem formulation is more likely to the existing management structure of a company. It is mapped directly to each engineering team. This makes it easier for the company to accept and use optimal design in their everyday design task.

A further work consists in investigating the research in this way, i.e. the decomposition of a complex system into several sub-problems and their coordination in order to ensure the consistency of the solution.

Appendices

A

Mathematical concepts

This appendix intends to give some mathematical concepts needed to understand the Kuhn-Tucker conditions which are used to prove the convergence of local search methods.

A.1 Introduction

Some mathematical definitions are necessary to understand the necessary and sufficient conditions for the non-linear optimization problems (NLP). Definitions are needed for both the analytical discussion as well as numerical techniques. First, basic mathematical concepts are given. Thereafter Taylor's theorem is presented.

A.2 Basic mathematical concepts

The basic mathematical elements in the discussion of NLP are derivatives, partial derivatives, vectors, matrices, Jacobian, and Hessian.

A.2.1 Function of one variable

Let's $f(x)$ be a function of one variable. The derivative of the function at the location x is written as

$$\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x)}{\Delta x} = \lim_{\Delta x \to 0} \frac{\Delta f}{\Delta x} \tag{A.1}$$

x is the point about which the derivative is computed. Δx is the distance to a neighbouring point $x + \Delta x$. When the separation Δx is reduced to zero, then the ratio of the difference in the value of the function at the two points (Δf) to the distance separating the two points (Δx) represents the derivative of the function $f(x)$ at the point x.

Numerical derivative computation

Many numerical techniques in NLP require the computation of derivatives. Most software implementing these techniques do not use formal computation. Automation and ease of use require that these derivatives be computed numerically. Indeed, using a very small perturbation Δx, the derivative at a point is numerically calculated as the ratio of the change in the function value, Δf, to the change in the displacement,Δx.

The derivative for the single-variable function at any value x is also called the slope or the gradient of the function at that point. If a line is drawn tangent to the function at the

value x, the tangent of the angle that this line makes with the x-axis will have the same value as the derivative. If this angle is θ, then:

$$\frac{df}{dx} = tan\theta$$

Higher derivatives

The derivative of the derivative is called the second derivative.

$$\frac{d^2f}{dx^2} = \frac{d}{dx}\frac{df}{dx} = \lim_{\Delta x \to 0}\frac{df/dx}{\Delta x} = \lim_{\Delta x \to 0}\frac{df/dx\,|_{x+\Delta x} - df/dx\,|_x}{\Delta x} \tag{A.2}$$

Similarly the third derivative is defined as

$$\frac{d^3f}{dx^3} = \frac{d}{dx}\frac{d^2f}{dx^2} \tag{A.3}$$

A.2.2 Function of n variables

The first important feature of more variables is that the derivatives defined for a single variable do not apply. The equivalent concept here is the partial derivative denoted by the symbol ∂. These derivatives are obtained in the same way as the ordinary derivative except the other variables are held at a constant value.

$$df = \frac{\partial f}{\partial x_1}dx_1 + \frac{\partial f}{\partial x_2}dx_2 + \cdots + \frac{\partial f}{\partial x_n}dx_n$$

Jacobian

The Jacobian J defines a useful way to organize the gradients of several functions. Using n variables and m functions $f_i([x_1, \ldots, x_n])$, $i = 1, \ldots, m$ the Jacobian is defined as follow:

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \tag{A.4}$$

Hessian

The Hessian matrix H is the same as the matrix of second derivatives of a function of several variables.

$$H = \begin{bmatrix} \frac{\partial^2 f_1}{\partial x_1^2} & \frac{\partial^2 f_1}{\partial x_2^2} & \cdots & \frac{\partial^2 f_1}{\partial x_n^2} \\ \frac{\partial^2 f_2}{\partial x_1^2} & \frac{\partial^2 f_2}{\partial x_2^2} & \cdots & \frac{\partial^2 f_2}{\partial x_n^2} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 f_m}{\partial x_1^2} & \frac{\partial^2 f_m}{\partial x_2^2} & \cdots & \frac{\partial^2 f_m}{\partial x_n^2} \end{bmatrix} \tag{A.5}$$

A.3 Taylor's Theorem / Series

Single variables: The Taylor series is useful mechanism to approximate the value of the function $f(X)$ at the point $(X_p + \Delta X)$ if the function is completely known at point X_p. The expansion is for finite n:

$$f(X_p + \Delta X) \cong f(X_p) + \frac{df}{dx}\Big|_{X_p}(\Delta X) + \frac{1}{2!}\frac{d^2f}{dx^2}\Big|_{X_p}(\Delta X)^2 + \cdots + \frac{1}{n!}\frac{d^nf}{dx^n}\,|_{X_p}(\Delta X)^n \tag{A.6}$$

The series is widely used in most disciplines to establish continuous models. It is the mainstay of many numerical techniques, including those in optimization. Equation approximation (A.6) is usually truncated to the first two or three terms with the understanding the approximation will suffer some error whose order depends on the term that is being truncated:

$$f(X_p + \Delta X) \cong f(X_p) + \frac{df}{dx}\Big|_{X_p}(\Delta X) + \frac{1}{2!}\frac{d^2f}{dx^2}\Big|_{X_p}(\Delta X)^2 + O(\Delta X)^3 \quad \text{(A.7)}$$

If the first term is brought to the left, the equation, discarding the error term, can be written as:

$$\Delta f = f(X_p + \Delta X) - f(X_p) = \frac{df}{dx}\Big|_{X_p}(\Delta X) + \frac{1}{2!}\frac{d^2f}{dx^2}\Big|_{X_p}(\Delta X)^2 \quad \text{(A.8)}$$

In (A.8), the first term on the right is called the first-order/linear variation while the second term is the second-order/quadratic variation.

Single variables: The series are only expanded to the quadratic terms. The truncation error is ignored. The two-variable function is shown in detail and also organized in terms of vectors and matrices. The first-order expansion is expressed in terms of the gradient. The second-order expansion will be expressed in terms of the Hessian matrix.

$$f(x_p + \Delta x,\, y_p + \Delta y) = f(x_p, y_p) + \left[\frac{\partial f}{\partial x}\Big|_{(x_p,y_p)}\Delta x + \frac{\partial f}{\partial y}\Big|_{(x_p,y_p)}\Delta y\right] + \frac{1}{2}\left[\frac{\partial^2 f}{\partial^2 x}\Big|_{(x_p,y_p)}(\Delta x)^2 + 2\,\frac{\partial^2 f}{\partial x \partial y}\Big|_{(x_p,y_p)}\Delta x \Delta y + \frac{\partial^2 f}{\partial^2 y}\Big|_{(x_p,y_p)}(\Delta y)^2\right] \quad \text{(A.9)}$$

If the displacement are organized as a column vector $[\Delta x\, \Delta y]^T$, the expansion in (A.9) can be expressed in a condensed manner as:

$$f(x_p + \Delta x,\, y_p + \Delta y) = f(x_p, y_p) + \nabla f^T_{(x_p,y_p)} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} + \frac{1}{2}\,[\Delta x\ \Delta y]\ H(x_p, y_p) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad \text{(A.10)}$$

A.4 Conclusion

In this appendix, how do the Hessian and the Jacobian are calculated is given. Furthermore Taylor series, which are essential and useful mechanism to understand the Kuhn-Tucker conditions presented in chapter 2 are discussed.

B

Complexity of algorithms

This appendix intends to give some definitions needed to understand the concept of complexity of algorithms. Not that all materials presented here are summarized from the electronic lecture notes [Narahari, 1998] at `http://lcm.csa.iisc.ernet.in/dsa`.

B.1 Introduction

As the focus of this thesis is to study, compare, modify and improve some optimization algorithms, it seems well to highlight the concept of complexity of algorithms. Indeed, data structures and algorithms are the cornerstones for developing high quality computer programs.

B.2 Concept of complexity of algorithms

It is very convenient to classify algorithms based on the relative amount of time or relative amount of space they require and specify the growth of time /space requirements as a function of the input size. Thus, we have the notions of:

- Time complexity: running time of the program as a function of the size of input ,
- Space complexity: amount of computer memory required during the program execution, as a function of the input size

A convenient way of describing the growth rate of a function and hence the time complexity of an algorithm is the use of big Oh notation. Let n be the size of the input and $f(n)$, $g(n)$ be positive functions of n.

Definition B.2.1 ***(Big Oh notation)*** *$f(n)$ is $O(g(n))$ if and only if there exists a real, positive constant C and a positive integer n_0 such that*

$$f(n) \leq Cg(n) \; \forall n \geq n_0$$

Note that the "Oh" notation specifies asymptotic upper bounds, and $O(g(n))$ is a class of functions. where:

- $O(1)$ refers to constant time,
- $O(n)$ indicates linear time,
- $O(n^k)$ (k fixed) refers to polynomial time,
- $O(logn)$ is called logarithmic time,
- $O(2^n)$ refers to exponential time, etc.

Example:
Let $f(n) = 2n^2 + 4n + 10$, then as defined before $f(n)$ is $O(n^2)$. Indeed, for $C = 3$ and $n_0 = 6$ we have $f(n) \leq 3n^2 \ \forall n \geq 6$. In the same way, we have also $f(n) \leq 4n^2 \ \forall n \geq 4$ ($C = 4$ and $n_0 = 4$).

Definition B.2.2 *(**Characteristic**) If $f(n)$ is $O(n^k)$ for some k, it is $O(n^h)$ for $h > k$*

In fact, as $f(n) = n^2 + n + 5$ is $O(n^2)$ thus it is also $O(n^3)$, but it cannot $O(n)$.

Definition B.2.3 *(**Characteristic**) If $f_1(n)$ is $O(g_1(n))$ and $f_2(n)$ is $O(g_2(n))$, then*

$f_1(n) + f_2(n)$ is $O(max(g_1(n), g_2(n)))$

Note that, the constant C that appears in the definition of the asymptotic upper bounds is very important. It depends on the algorithm, machine, compiler, etc. Indeed, the big "Oh" notation gives only asymptotic complexity. As such, a polynomial time algorithm with a large value of the constant may turn out to be much less efficient than an exponential time algorithm (with a small constant) for the range of interest of the input values.

B.3 Worst case and average case

Worst case running time is the behaviour of the algorithm with respect to the worst possible case of the input instance. The worst-case running time of an algorithm is an upper bound on the running time for any input. With this information we have a guarantee that the algorithm will never take any longer and there is no need to make an educated guess about the running time.

Average case running time is the expected behaviour when the input is randomly drawn from a given distribution, i.e. it is an estimate of the running time for an average input n. thus, its computation entails knowing all possible input sequences, the probability distribution of occurrence of these sequences, and the running times for the individual sequences. for simplicity, it is often assumed that all inputs of a given size are equally likely.

In the following, the worst case and average case running time are given for some known algorithms in Table B.1:

Table B.1: Worst case and average case running time of algorithms.

Algorithm	Worst case running time	Average case running time
- Problem of finding the minimum element in a list of elements	- $O(n)$	- $O(n)$
- Quick sort	- $O(n^2)$	- $O(n \log n)$
- Merge sort, Heap sort	- $O(n \log n)$	- $O(n \log n)$
- Bubble sort	- $O(n^2)$	- $O(n^2)$
- Binary search tree (Search for an element)	- $O(n)$	- $O(\log n)$

B.4 Conclusion

In this appendix, some definitions needed to understand the concept of complexity of algorithms is given. Note that this annex aims to explain briefly how do the complexity of some procedures of algorithms (like the truncation procedure for SPEA2 and the crowding one for NSGA-II) are calculated in chapter 2.

C

Surface-mounted permanent magnet synchronous motor modelling

This appendix intends to give more detail description concerning the traction motor design. Note that, all materials are extracted from [Kreuawan, 2008] thesis.

C.1 Introduction

A design of a surface-mounted permanent magnet (SMPM) motor implies a multidisciplinary design: magnetic, electric, heat transfer, mechanic, cost, etc. The modelling of SMPM is based on a modular approach. Input and output data of each module are defined in the first place. These input and output describe interactions between modules. Once the input and the output are identified, the model used in each module can be selected based on the need of fidelity. For example, the analytical model or Finite Element Analysis (FEA) can be used in the magnetic or thermal module. In the same manner, the flux-weakening control or the maximal torque per ampere strategy can be used in the control module. The structure of SMPM motor model is shown in Figure C.1. It consists of eight modules. Each module represents a discipline; magnetic, electric, control, thermal etc. Regarding the interaction, there are two temperature feedbacks:

1. The magneto-thermal loop allows taking into account permanent magnet properties change due to the PM temperature change;

2. The electro-thermal loop modifies phase resistance value according to the winding temperature.

There are two types of input and output: time dependent and time independent variable. Time dependent inputs consist of torque and speed (as a function of time). These can be obtained using vehicle dynamic equations. Time independent inputs are voltage limit, material property and motor geometry. Time dependent outputs e.g. flux density, phase current, temperature (as a function of time) and time independent outputs; mass, cost, and consumption are computed by the SMPM model. When using the model in optimization, each input variable can be fixed as a design parameter or assigned as a design variable depending on how an optimal design problem is formulated. Outputs have to be conditioned before used in the optimization e.g. taking the maximal value of time dependent output. Table C.1 - Table C.11 provide an exhaustive list of model's inputs and outputs.

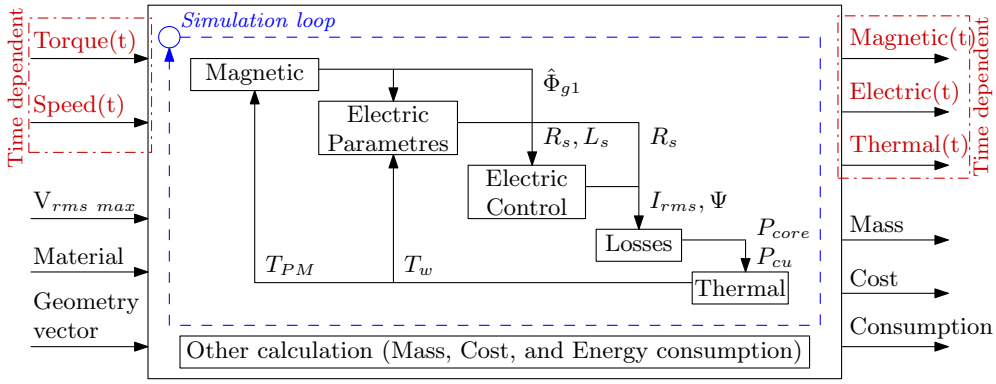


Figure C.1: Structure of SMPM motor model.

In this research, an approach using semi-numerical model is used. Physical phenomena are described by analytical equations. Thermal interdisciplinary interactions and temperature-material property interactions are solved by an iterative method. This allows fast computation and acceptable accuracy since it will be used in preliminary design phase. Model accuracy and complexity are sufficient to capture interactions between disciplines. The following sections describe each module in detail and a simulation example will be presented.

C.2 Magnetic module

A simplified model was built. Several assumptions were made:

1. Squarewave PM flux density distribution is considered.
2. Flux crossing PM and air gap is in radial direction.
3. First harmonic of flux density is extracted from square-wave and used for further calculation.
4. Steel permeability is assumed infinity.

The main flux path of the SMPM motor is shown in Figure C.2(a). Magnet flux is the actual flux that passes the magnet. It is composed of two main flux paths: air gap flux and leakage flux. The air gap flux is effective flux crossing the air gap. The leakage flux can be taking into account by leakage coefficient k_{leak}, which is a ratio of air gap flux and magnet flux.

Figure C.2(b) shows magnetic equivalent circuit of one pole. The air gap reluctance R_g and the magnet reluctance R_m are expressed as:

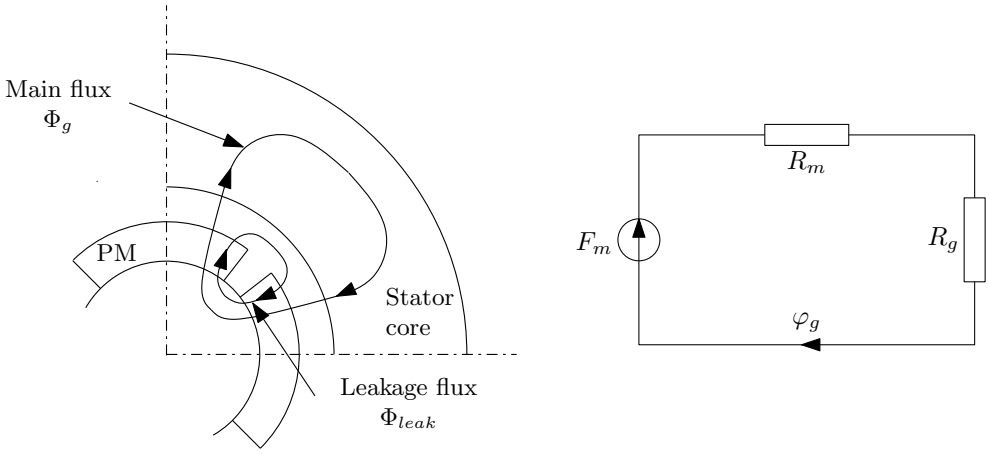
$$R_g = \frac{1}{\frac{\pi}{p} \cdot k_p \cdot \mu_0 \cdot \mu_r \cdot l_{stk}} \cdot \ln\left(\frac{r_a}{r_a - g'}\right) \tag{C.1}$$

$$R_m = \frac{1}{\frac{\pi}{p} \cdot k_p \cdot \mu_0 \cdot l_{stk}} \cdot \ln\left(\frac{r_a - g' - l_m}{r_a - g'}\right) \tag{C.2}$$

where $g' = k_c \cdot g$, k_c Carter coefficient given in [Gillon, 1997], l_{stk} stack length, k_p magnet span coefficient. Other dimensions can be found in Figure C.3.

Magnetomotrice force (MMF) of magnet F_m and air gap flux φ_g are related by equation:

$$F_m = \frac{B_r}{\mu_R \mu_0} l_m = (R_g + R_m)\varphi_g = (R_g + R_m)\hat{B}_g A_m \tag{C.3}$$



(a) Main flux path. (b) Magnetic equivalent circuit.

Figure C.2: Magnetic circuit.

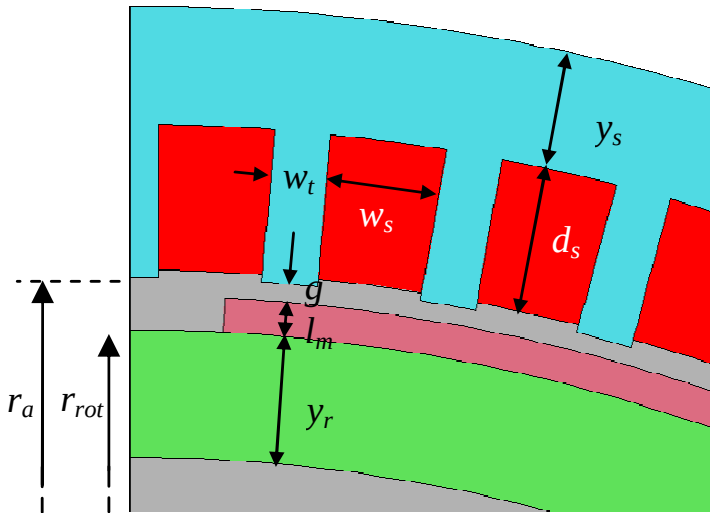


Figure C.3: Motor geometry

with A_m magnet pole area. Therefore, flux density in air gap due to PM can be expressed as:

$$\hat{B}_g = \frac{k_{leak} \cdot B_r \cdot l_m}{r_g} \cdot \left(\ln \left(\frac{r_{rot} + l_m + g'}{r_{rot} + l_m} \right) + \frac{1}{\mu_R} \ln \left(\frac{r_{rot} + l_m}{r_{rot}} \right) \right)^{-1} \tag{C.4}$$

where B_r remanence flux density, μ_R relative permeability of PM, r_g mid-air gap radius ($r_g = r_{rot} + l_m + g/2$). The Carter coefficient has been verified with FEA and the relative error between flux for the case of a slotted stator and a smooth stator with a corrected air gap has been studied. In order to keep relative error below 5%, tooth width ratio $\left(r_{wt} = \frac{w_t}{w_s + w_t} \right)$ higher than 0.4 is preferred.

In sine-wave back-EMF motor, fundamental air gap flux density ($\hat{B}_{g1}$) given in (C.5)is extracted from square-wave flux density distribution as shown in Figure C.4.

$$\hat{B}_{g1} = \frac{4}{\pi} \cdot \hat{B}_g \cdot \sin \left(\frac{\pi}{2} k_p \right) \tag{C.5}$$

Peak value of fundamental flux $\hat{\phi}_{g1}$ can be computed by integrating flux density over a pole area as in (C.6). Total flux per phase $\hat{\Phi}_{g1}$ is obtained by multiplying effective coil number

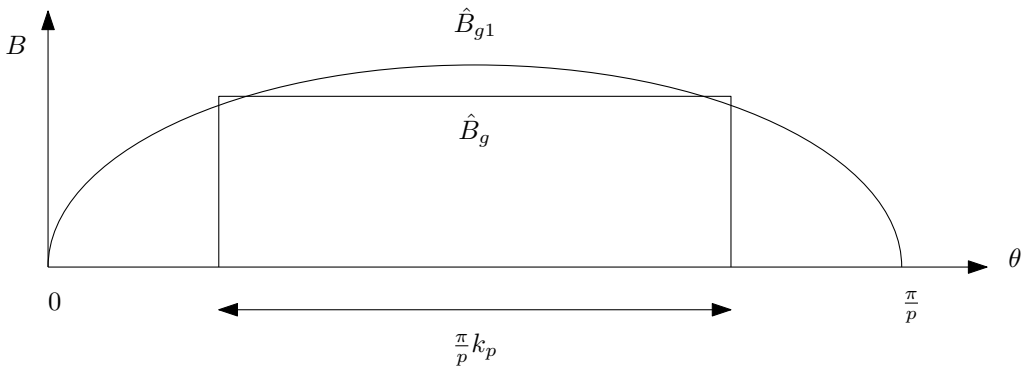


Figure C.4: Squarewave flux density distribution and fundamental flux density

per phase.

$$\hat{\phi}_{g1} = \int_0^{\frac{\pi}{p}} \hat{B}_{g1} l_{stk} r_g \cdot d\theta \tag{C.6}$$

$$\hat{\Phi}_{g1} = \hat{\phi}_{g1} \cdot N_{s,eff} = \frac{2}{p} \cdot \hat{B}_{g1} \cdot \left(r_{rot} + l_m + \frac{g}{2} \right) \cdot l_{stk} \cdot N_{s,eff} \tag{C.7}$$

where p poles pair, k_p PM span ratio, l_{stk} stack length, and $N_{s,eff}$ effective coil number per phase described as:

$$N_{s,eff} = N_s \cdot k_{pitch} \cdot k_w = \left(\frac{N_c \cdot N_{slot} \cdot p}{a} \right) \cdot k_{pitch} \cdot k_w \tag{C.8}$$

where N_s coil number per phase, N_c conductor number per slot, N_{slot} slot number per pole and per phase, a parallel path number, k_{pitch} winding pitch factor, and k_w winding distribution factor.

Flux equation (C.7) is validated against FEA. Figure C.5 shows flux lines computed by FEA. The relative error between (C.7) and FEA is less than 10%, which is acceptable for preliminary design. The main error is due to Carter coefficient, which is inaccurate for the structure with narrow teeth (r_{wt} is equal to 0.26 for this motor.). However, correction factors can be used to calibrate the model in case that the model error is known a priori.

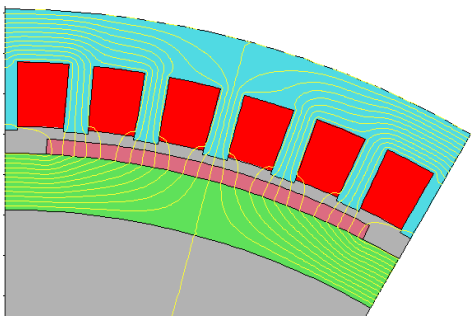


Figure C.5: Flux line from FEA

The properties of the PM are critical in the design of a SMPM. The major factors are the protection against the demagnetization, the property change due to temperature changes and the working temperature limit. The characteristic of permanent magnet such as B_r and the intrinsic coercive field strength (H_{ci}) change as the temperature changes. It can be described by a reversible temperature coefficient of B_r and H_{ci}, α_{B_r} and $\beta_{H_{ci}}$, respectively [Trout, 2001]. For NdFeB, B_r decreases as the temperature increases. However, reversible losses can be recovered when the temperature returns to the initial point. In order to take

into account the PM temperature effects, at each simulation step, B_r will be updated with the actual permanent magnet temperature T_{PM} using the remanence flux density at 20°C (B_{r20}) and remanence temperature coefficient α_{B_r}.

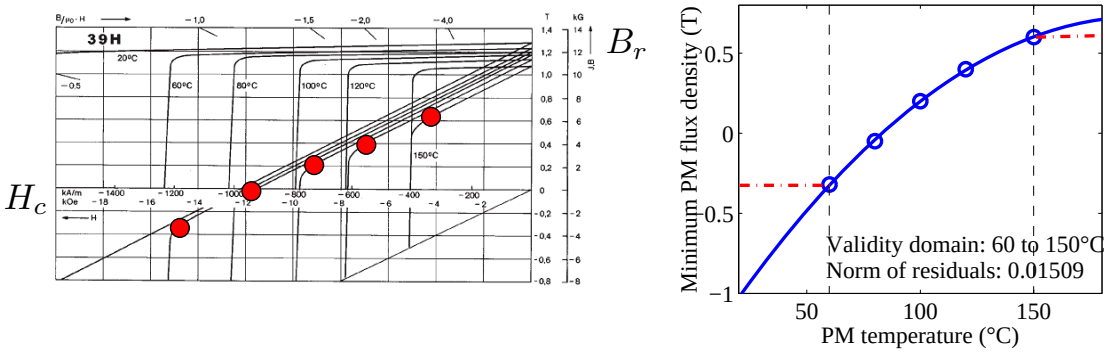
$$B_r = B_{r20}\left(1 + \frac{\alpha_{B_r}}{100} \cdot (T_{PM} - 20)\right) \tag{C.9}$$

Note that T_{PM} is in °C.

The change in "knee point" of normal demagnetization curve is very important (red points in Figure C.6(a)). In order to avoid irreversible loss, the operating point cannot be lower than this point [Hendershot & Miller, 1994] i.e. the allowable motor current is limited by the knee point. Unfortunately, the relationship between the knee point and the temperature is generally not provided by the manufacturer. However, one can always approximate this relationship from B-H curve at various temperatures given in the data sheet. It will be used as a constraint in the optimization problem. For NdFeB 39H, Br_{min} can be given as:

$$B_{r_{min}} = -5.58 \cdot 10^{-5} \cdot T_{PM}^2 + 0.022 \cdot T_{PM} - 1.445 \tag{C.10}$$

with T_{PM} in °C. This relationship is valid from 60°C to 150°C. For T_{PM} lower than 60°C or higher than 150°C, constant values are given as shown in Figure C.6(b).



(a) B-H characteristic from the PM data sheet.

(b) Approximate model.

Figure C.6: Minimal value of the remanence flux density as a function of the temperature can be estimated from the PM B-H curve

C.3 Electric module

This module computes circuit parameters (resistance R_s and inductance L_s) from motor geometry. The phase resistance at 20°C is defined as:

$$R_{s20} = \frac{\rho_{cu} \cdot l}{k_r \cdot w_s \cdot d_s} \cdot 2N_s \tag{C.11}$$

where l one conductor length (including end winding), k_r slot-fill factor and ρ_{cu} copper conductivity at 20°C.

During the simulation phase, the resistance is updated with the actual winding temperature (T_w, in °C) from the heat transfer module.

$$R_s = R_{s20}\left(1 + \alpha_{cu} \cdot (T_w - 20)\right) \tag{C.12}$$

where α_{cu} is the conductivity temperature coefficient of copper.

The phase inductance is composed of 4 components, (i) air gap self inductance L_g, (ii) air gap mutual inductance M_g, (iii) slot leakage inductance L_{slot}, and (iv) end winding inductance L_{end}.

$$L_s = L_g - M_g + (L_{slot} - L_{end}) \tag{C.13}$$

L_g is computed by simply dividing the peak value of fundamental air gap flux due to armature current ($\hat{\Phi}_{ga1}$) by current I. For sinusoidal distributing winding, M_g is obtained using (C.17).

$$L_g = \frac{\hat{\Phi}_{ga1}}{I} \tag{C.14}$$

$$\hat{\Phi}_{ga1} = \frac{2 \cdot l_{stk} \cdot r_a}{p} \cdot \hat{B}_{ga1} \cdot N_{s,eff} \tag{C.15}$$

$$\hat{B}_{ga1} = \frac{4}{\pi} \cdot \frac{\mu_0}{2 \cdot p \cdot (g' + l_m/\mu_R)} \cdot N_{s,eff} \cdot \tag{C.16}$$

$$M_g = -\frac{1}{2} L_g \tag{C.17}$$

End winding leakage inductance L_{end} formula are given in [Bouchard & Olivier, 1997]:

$$L_{end} = 3.15 \cdot 10^{-6} \left(2N_s\right)^2 \frac{3k_w^2}{4\pi p} \left(b_1 + \frac{(b_4 + b_3/2)}{2}\right) \tag{C.18}$$

where b_1, b_3, b_4 are geometries of end winding as shown in Figure C.7[5].

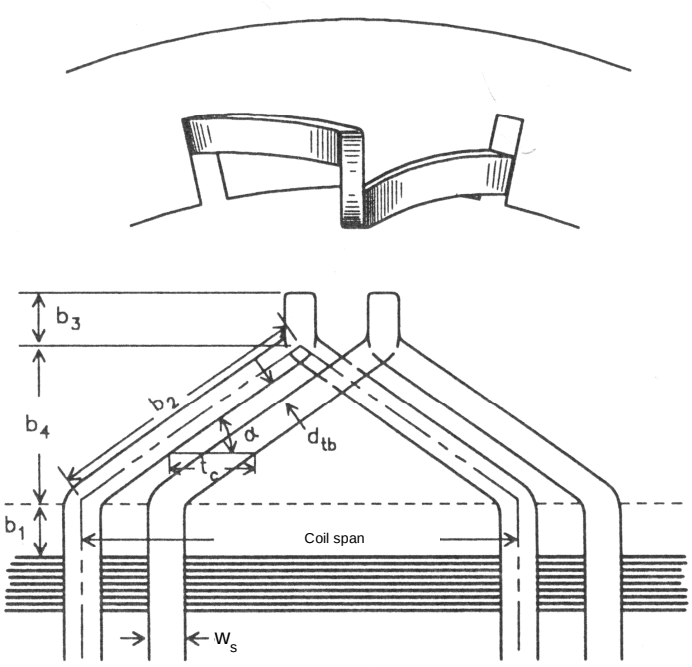


Figure C.7: Dimension of end winding

For open rectangular slot, L_{slot} can be computed as [Hendershot & Miller, 1994]:

$$L_{slot} = \mu_0 N_s^2 l_{stk} \left(\frac{h2}{3w_s} + \frac{h1}{w_s} + \frac{h2}{4w_s}\right) p \left(b_1 + \frac{(b_4 + b_3/2)}{2}\right) \tag{C.19}$$

h_1 and h_2 are slot and winding dimensions (see Figure C.8).

[5]Figure from [Bouchard & Olivier, 1997]

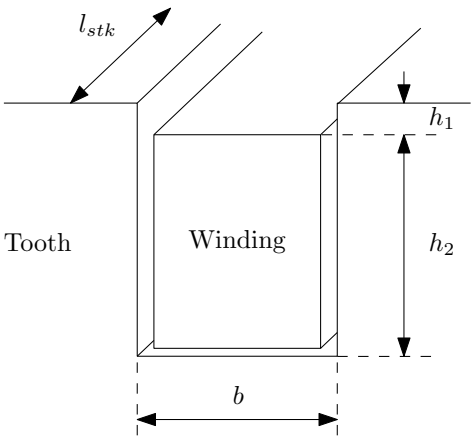


Figure C.8: Dimension of slot

C.4 Control strategy module

Since only electrical steady-state operation is considered, it is assumed that there is no error between the controllers' reference signal and real output. The model is an inverse steady-state single-phase equivalent model as shown in Figure C.9. It is then transformed in d-q reference frame in order to perform the flux-weakening strategy [Morimoto *et al.*, 1990]. It computes necessary d-axis, q-axis, phase current and voltage to produce the required torque and speed at each time step.

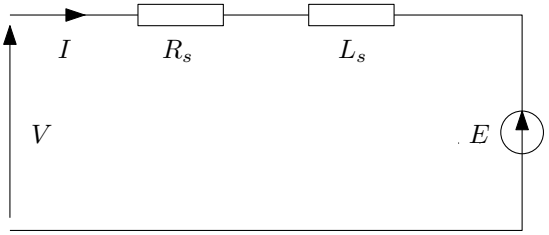


Figure C.9: Single-phase equivalent electric model

Relationships between RMS current (I_{rms}) and d-q axis current i_d and i_q are given as:

$$i_d = \sqrt{3} I_{rms} \sin\psi \tag{C.20}$$

$$i_q = \sqrt{3} I_{rms} \cos\psi \tag{C.21}$$

Therefore voltage equations in d-q coordinates (v_d and v_q) and RMS voltage V_{rms} can be described as follows:

$$v_d = R_s i_d - \omega_s L_q i_q \tag{C.22}$$

$$v_q = R_s i_q + \omega_s \left(L_d i_d + \phi_d\right) \tag{C.23}$$

$$V_{rms} = \sqrt{\frac{v_d^2 + v_q^2}{3}} \tag{C.24}$$

where ω_s electrical speed, L_d and L_q d- and q-axis inductance.

The electromagnetic torque T_{em} can be computed from electromagnetic power balance:

$$T_{em} = \frac{P_{em}}{\Omega} = p\left(L_d - L_q\right) i_d i_q + p\phi_d i_q \tag{C.25}$$

For SMPM motor, both L_d and L_q are equal to L_s (C.13). Equation C.25 becomes

$$T_{em} = p\phi_d i_q \tag{C.26}$$

where ϕ_d d-axis flux computed from fundamental air gap flux (C.7):

$$\phi_d = \sqrt{\frac{3}{2}} \cdot \hat{\Phi}_{g1} \tag{C.27}$$

Two operating modes are shown in Figure C.10. Zone 1: constant torque, in this zone, the torque is proportional to q-axis current i_q. d-axis current i_d is controlled to zero in order to maximise the torque per ampere. As the speed increases, the voltage increases. Once the voltage reaches the maximum inverter voltage, the controller switches to flux-weakening mode (zone 2). The speed at this transition point is called base speed (Ω_b). In this zone, Ψ phase angle between back-EMF and phase current is changed by decreasing i_d. This results in negative value of i_d. It creates a flux in the opposite direction of the air gap flux. The result flux, hence back-EMF, then decreases. Therefore, the speed can be increased. As mentioned above, if the operating flux density is lower than the knee point at the corresponding temperature in the PM characteristic curve, it would cause an irreversible demagnetization of the magnet, which is not preferred.

Figure C.10: Traction motor operating zone

A flux weakening control flowchart is shown in Figure C.11. By giving the required electromagnetic torque (T_{em}), i_q can be defined:

$$i_q = \frac{T_{em}}{p \cdot \phi_d} \tag{C.28}$$

V_{rms} (C.24) is then computed. If V_{rms} is higher than the maximal inverter output RMS voltage ($V_{rms\ max}$), this means that the inverter does not provide sufficient voltage and flux-weakening mode may be performed in order to achieve the required torque and speed.

In the flux weakening zone, i_d can be computed by solving the quadratic equation obtained with (C.22)-(C.24) substituting V_{rms} for $V_{rms\ max}$ and i_q from (C.28):

$$\begin{aligned} V_{rms\ max}^2 = &\ \left(R_s^2 + \omega_s^2 L_d^2\right) i_d^2 + \left(2\omega_s^2 L_d \Phi_d\right) i_d + \\ &\ \left(\omega_s^2 L_q^2 i_q^2 + \omega_s^2 \Phi_d^2 + 2\omega_s \Phi_d R_s i_q + R_s^2 i_q^2\right) \\ 0 = &\ A \cdot i_d^2 + B \cdot i_d + C \end{aligned} \tag{C.29}$$

The solution should be a real negative value. Otherwise, it can be concluded that the voltage is not sufficient to produce the required torque at the corresponding speed. In this case, the necessary voltage (which is higher than the maximum voltage) is given as an output and used as a constraint in optimal design process[6].

The RMS phase current (I_{rms}) can be computed:

$$I_{rms} = \sqrt{\frac{i_d^2 + i_q^2}{3}} \tag{C.30}$$

[6]This voltage output must be lower or equal to $V_{rms\ max}$ in order to ensure that motor can produce required torque at the corresponding speed.

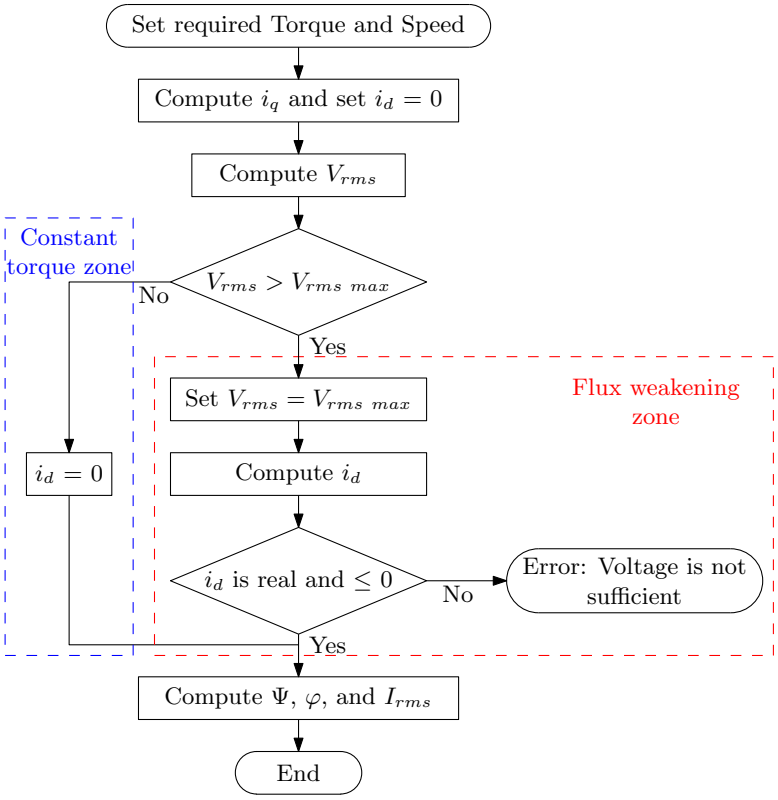


Figure C.11: Flux weakening control flowchart

Ψ and φ, angle between current and voltage, can be written as:

$$\Psi = \arctan\left(\frac{i_d}{i_q}\right) \tag{C.31}$$

$$\varphi = \arctan\left(\frac{v_d}{v_q}\right) - \Psi \tag{C.32}$$

C.5 Losses module

Copper losses can be easily obtained using conventional formula:

$$P_{cu} = 3I_{rms}^2 R_s \tag{C.33}$$

Core losses are computed using an approximate model obtained from a specific losses curve at 50Hz (q_t) given in the lamination data-sheet. This model uses the peak value of the resulting flux density in tooth ($\hat{B}_t$) or stator yoke ($\hat{B}_{Y_s}$) as input. For M235-35A, q_t is given as in (C.34)

$$q_t = -1.2073\hat{B}^4 + 4.5706\hat{B}^3 - 4.6764\hat{B}^2 + 2.4067\hat{B} - 0.2025 \tag{C.34}$$

These specific losses are recalibrated to the corresponding electrical frequency (f_s). They are then multiplied by the mass of teeth and the stator yoke to achieve core losses in the teeth and stator yoke.

$$P_{core\ (tooth,yoke)} = mass_{(teeth,yoke)} \cdot q_t \cdot \left(\frac{f_s}{50}\right)^{1.5} \tag{C.35}$$

Figure C.12 shows the vector diagram of the resulting flux density $\hat{B}_{g\,res1}$, which is vector sum of the air gap flux density due to the PM, mutual, and armature current $\hat{B}_{g1}$, $\hat{B}_{Mg1}$, and $\hat{B}_{ga1}$, respectively. The resulting flux density in the teeth, stator and rotor yoke can be expressed as:

$$\hat{B}_t = \hat{B}_{g\,res1} \cdot \left(\frac{\pi \cdot r_g}{3 \cdot N_{slot} \cdot w_t} \right) \tag{C.36}$$

$$\hat{B}_{Y_s} = \frac{1}{2} \hat{B}_{g\,res1} \cdot \left(\frac{\pi \cdot r_g}{p \cdot Y_s} \right) \tag{C.37}$$

$$\hat{B}_{Y_r} = \frac{1}{2} \hat{B}_{g\,res1} \cdot \left(\frac{\pi \cdot r_g}{p \cdot Y_r} \right) \tag{C.38}$$

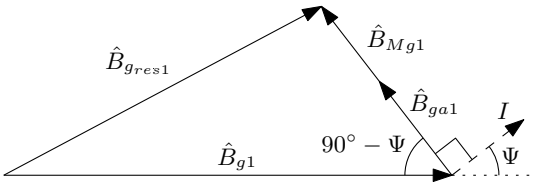


Figure C.12: Vector diagram of the resulting flux density

C.6 Heat transfer module

The transient thermal behaviour of motor is necessary to design a motor in variable speed applications, especially when the load profile is known such as in the traction applications [Mester *et al.*, 2006]. Motor sizing is based on the steady-state temperature on the whole route profile, not on a rated point (base point).

A lumped parameter thermal network allows estimating the temperature in various parts of motor. The model is based on models developed for asynchronous motors [Millor *et al.*, 1991, Boglietti *et al.*, 2003]. They are modified for Totally Enclosed, Fan-Cooled (TEFC) SMPM motors. Several hypotheses are made:

1. Radial heat fluxes are considered, except shaft, heat transfer is axially.
2. Heat sources are concentrated at node.
3. Air flow at end winding is neglected.
4. Only tangential air flow is considered at air gap.

Model is composed of 8 nodes (see Figure C.13). The node temperature represents the average temperature of each element. For each node, the differential equation is:

$$C_{th_i} \frac{dT_i}{dt} = \sum_{i,j=1}^{n} \frac{(T_j - T_i)}{R_{th_{ij}}} + P_i \tag{C.39}$$

Thermal resistances R_{th} characterise conduction and convection heat transfers. Thermal capacitances C_{th} represent thermal inertia in case of a transient simulation. Thermal resistances and capacitors are determined from dimensions and motor materials. Losses generated by a motor act as heat sources P.

The $R_{th_{ij}}$ resistance between two nodes i and j can be defined as:

$$R_{th_{ij}} = \frac{R_{th_i}}{2} + \frac{R_{th_j}}{2} \tag{C.40}$$

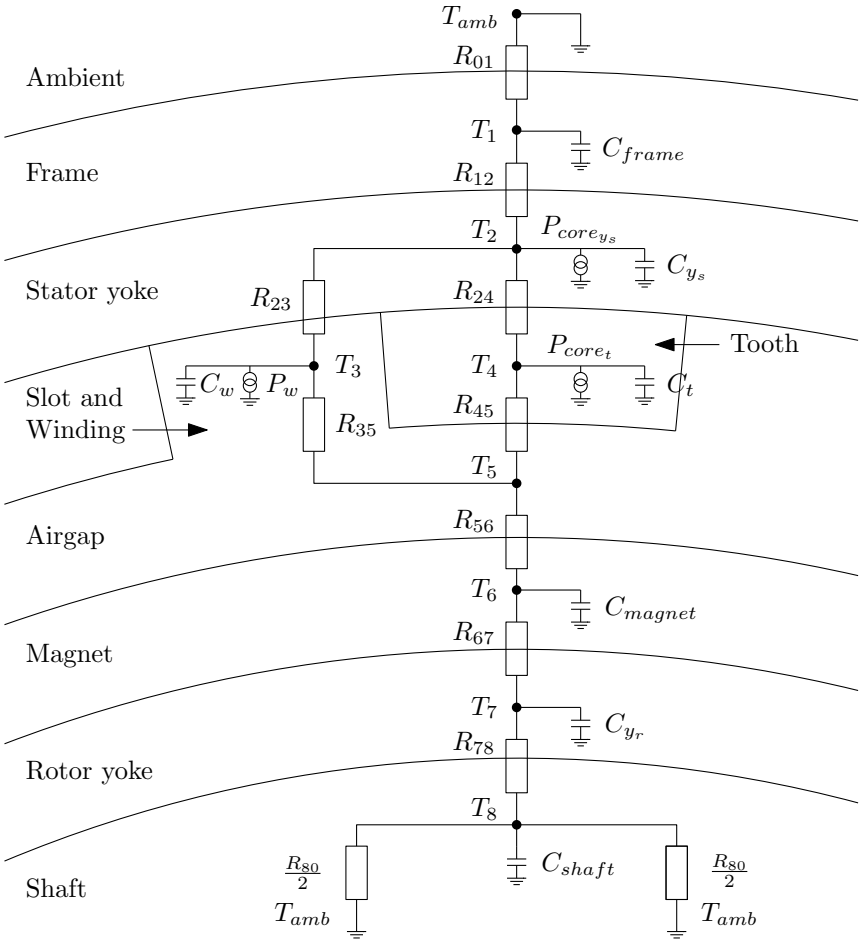


Figure C.13: Thermal model of SMPM motor

Note that the temperature at node represents the average temperature of the element and the node position does not necessary have to be in the middle of elements.

For radial heat conduction, R_{th} of an element can be expressed as:

$$R_{th_i} = R_{cond_i} = \frac{1}{\theta k_i L} \ln\left(\frac{r_{ext_i}}{r_{int_i}}\right) \tag{C.41}$$

where k thermal conductivity coefficient (as given in [Boglietti *et al.*, 2003] k is function of the temperature), r_{int} internal radius, r_{ext} external radius and L longueur and θ open angle comprising between 0–2π radian. (see Figure C.14).

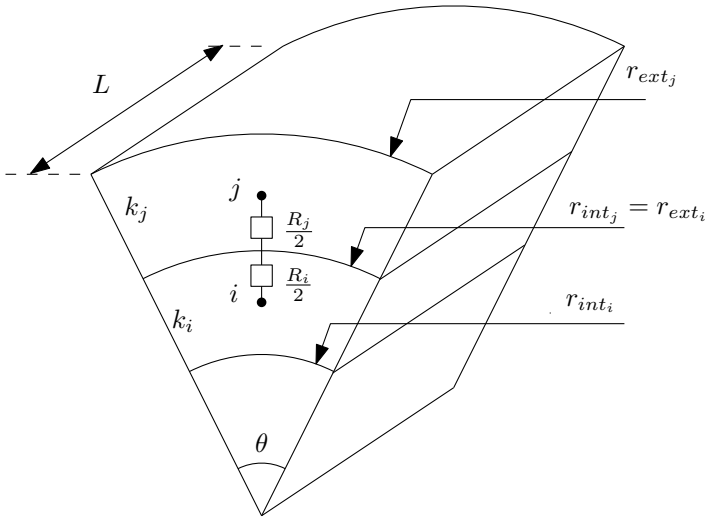


Figure C.14: Thermal resistance of radial convection

Heat convection is considered at the air gap and external frame. Thermal resistance due to heat convection is described as:

$$R_{th_i} = R_{conv_i} = \frac{1}{A \cdot h} \tag{C.42}$$

where A heat exchange surface and h heat convection coefficient.

Air gap heat convection coefficient: At air gap, tangential air speed due to rotation of rotor is:

$$v_{air_g} = (r_a - g)\,\Omega \tag{C.43}$$

Reynolds number and Prandtl number are expressed as:

$$Re_g = \frac{l_{stk} \cdot v_{air_g} \cdot \rho_{air_g}}{\mu_{air_g}} \tag{C.44}$$

$$Pr_g = \frac{C_{p_{air_g}} \cdot \mu_{air_g}}{k_{air_g}} \tag{C.45}$$

where ρ_{air_g} air mass density, μ_{air_g} air dynamic viscosity and $C_{p_{air_g}}$ specific heat capacity of air in the air gap.

h is expressed as:

$$h_g = \frac{Nu_g \cdot k_{air_g}}{g} \tag{C.46}$$

where Nu_g Nusselt number at air gap. Nu_g is function of Taylor number as shown in (C.47)[Millor *et al.*, 1991].

$$\begin{array}{ll} Ta \leq 41, & Nu_g = 2.2 \\ 41 \leq Ta \leq 100, & Nu_g = 0.23 Ta^{0.63} \cdot Pr_g^{0.27} \\ Ta > 100, & Nu_g = 0.3855 Ta^{0.5} \cdot Pr_g^{0.27} \end{array} \tag{C.47}$$

where

$$Ta = Re_g \sqrt{\frac{g}{r_a - g}} \tag{C.48}$$

External frame heat convection coefficient: It depends on external air properties and external air speed in axial direction as depicted in Figure C.15

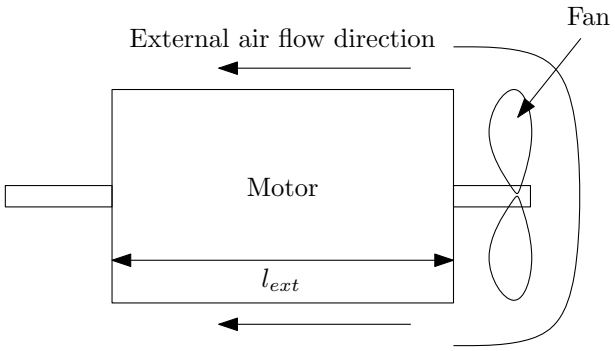


Figure C.15: External air flow

Reynolds number and Nusselt number are expressed as:

$$Re_{ext} = \frac{l_{ext} \cdot v_{air_{amb}} \cdot \rho_{air_{amb}}}{\mu_{air_{amb}}} \tag{C.49}$$

$$\begin{aligned} Re_{air_{ext}} < 10000, \qquad & Nu_{ext} = 0.66 Re_{ext}^{0.5} \cdot Pr_{ext}^{0.33} \\ Re_{air_{ext}} \geq 10000, \qquad & Nu_{ext} = 0.66 Re_{ext}^{0.75} \cdot Pr_{ext}^{0.33} \end{aligned} \tag{C.50}$$

Therefore, external air heat convection coefficient is defined as:

$$h_{ext} = \frac{Nu_{ext} \cdot k_{air_{amb}}}{l_{ext}} \tag{C.51}$$

The heat transfer module includes the numerical algorithm for solving the Ordinary Differential Equation (ODE). At the end of the load profile, steady state convergence over load cycle is checked. The temperatures at the end of the cycle are set as the new initial temperature and the simulation is rerun until the temperatures reach their steady state. This depends on the thermal time constant.

It also allows computing the steady-state temperature at any time step (do not take into account thermal capacitor). This can be useful when only the simple calculation is needed e.g. a design based on the torque as a function of speed characteristics or a base point design, in such cases the simulation is not needed since the time domain is not concerned. Temperatures are implicit variables because material properties depend on temperatures as mentioned above. At any torque-speed point, the input temperatures are forced to be equal to the output temperatures through the fixed-point iteration method.

C.7 Simulation example

The simulation can be done by giving the motor's geometry and material, inverter voltage limit. Motor load profile (torque and speed as a function of time) is given as shown in Figure C.16. The simulation takes 40 load cycles to achieve the steady-state temperatures.

Figure C.17(a) shows the temperatures as a function of time. The temperature evolution starts from the ambient temperature at 40°C. The winding and the PM temperatures reach their steady-state after 40 load cycles. At the steady-state, the winding and the PM temperatures can be observed at 175°C and 145°C, respectively.

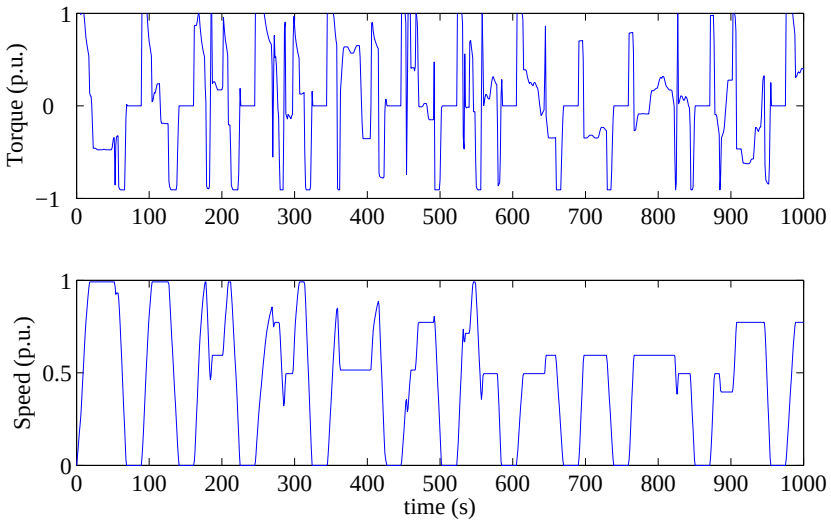


Figure C.16: Torque and speed requirement

The resulting flux density and minimal PM flux density are shown in Figure C.17(b). As the PM temperature increases, the minimal PM flux density increases. The PM demagnetization can be occurred if the resulting flux density is lower than the minimal PM flux density. In this example, it cannot be observed.

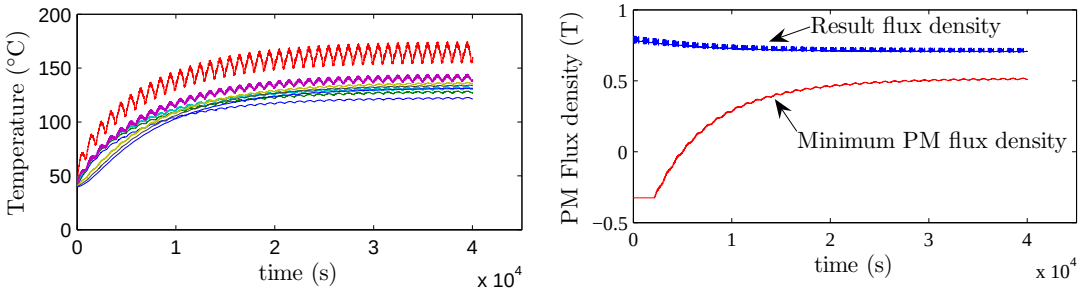
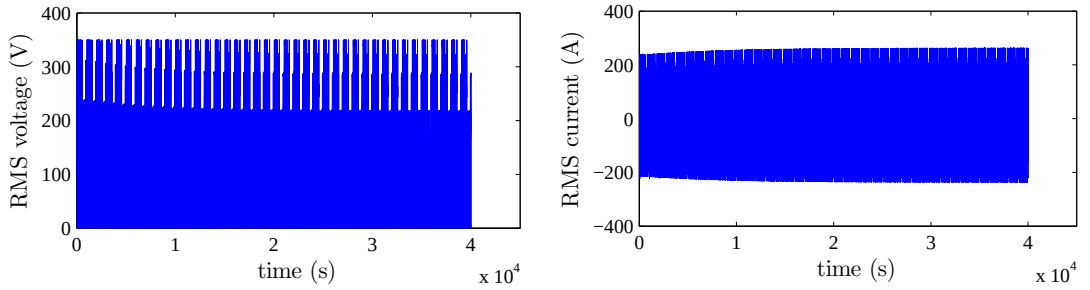


Figure C.17: (b) PM flux density, (a) Temperatures

Figure C.18 show the RMS voltage and current. The maximum RMS voltage is given 290V in traction mode and 350V in braking mode. At the beginning of the simulation, the required RMS current is lower due to the fact that when the PM temperature is low, the PM can provide high remanence flux density. Therefore, motor requires less current to produce the same amount of torque.

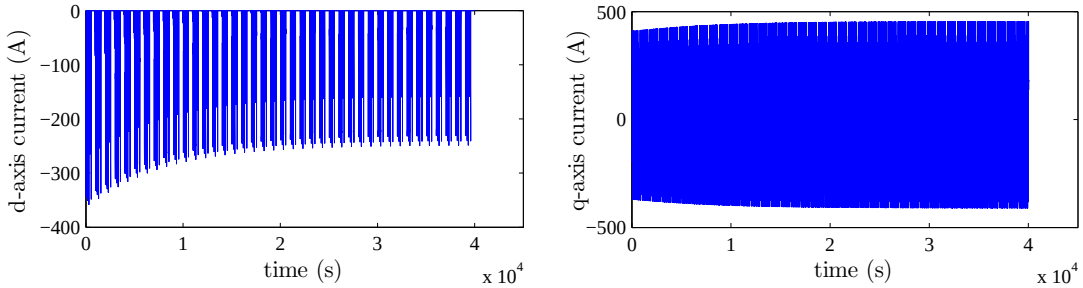
In d-q coordinate, currents are shown in Figure C.19. It can be observed that i_d is negative when the motor is operated in a flux weakening mode.

When using these simulation results in an optimal design process, the results must be reconditioned. The maximum value of the PM temperature is taken. Different value between the resulting flux density and the minimal PM flux density ($B_{res} - B_{r_{min}}$) is used to indicate the demagnetization of the PM.



(a) RMS voltage. (b) RMS current.

Figure C.18: RMS voltage and current



(a) d-axis current. (b) q-axis current.

Figure C.19: Current in d-q coordinate

C.8 Conclusion

In this appendix, a model of a surface-mounted permanent magnet motor is presented. This model is built using a modular approach. Each module represents a discipline. They are described by analytical relationship. Implicit variables and interaction loops are solved by using the iterative method. This model is suitable for preliminary design and for using in an optimal design process. Applications of this model are presented in chapter 7 and 8.

Table C.1: Input–Geometries

Symbol	Quantity	Unit
Y_s	Stator yoke height	m
Y_r	Rotor yoke height	m
l_m	PM height	m
g	Air gap	m
d_s	Slot height	m
r_a	Armature radius	m
l_{stk}	Stack length	m
d_{open}	Slot opening depth	m
t_{insu}	Insulation thickness	m
t_{frame}	Frame thickness	m
r_{wt}	Tooth width ratio	-
k_p	PM span coefficient	-
p	Pole pair	-
N_{slot}	Slot/pole/phase	-
k_{leak}	Leakage flux coefficient	-

Table C.2: Input–Winding

Symbol	Quantity	Unit
n_{ph}	Number of phases	-
N_c	Conductor/slot	-
A	Parallel path	-
$pitch$	Winding pitch	-
k_r	Slot filling coefficient	-

Table C.3: Input–Inverter

Symbol	Quantity	Unit
$V_{max_{Tr}}$	Maximal inverter voltage (traction)	V
$V_{max_{Br}}$	Maximal inverter voltage (braking)	V
$flag_{control}$	Flag control model: 1=with R_s, 2=without	-
$flag_{fw}$	Flag flux-weakening: true=with R_s, false=without	-

Table C.4: Input–Performances

Symbol	Quantity	Unit
T	Required torque vector	Nm
Ω	Required speed vector	rad/s
t	Time vector	s

Table C.5: Input–Simulation options

Symbol	Quantity	Unit
h	Time step	s
ode	1=transient, 0=steady-state	-
$cyclemax$	Maximal cycle to simulate	-

Table C.6: Input–Thermal

Symbol	Quantity	Unit
T_{amb}	Ambient temperature	K
T_{init}	Initial temperature vector	K

Table C.7: Input–Material properties

Symbol	Quantity	Unit
General		
ρ	Mass density	kg/m^3
cost	Material cost	Euro/kg
k	Thermal conductivity	W/(m·K)
c_p	Specific heat	J/kg/K
Copper		
ρ_{elec20}	Resistivity at 20°C	Ωm
α_{cu}	Conductivity temp. coefficient	°C^{-1}
PM		
B_{r20}	Remanence flux density at 20°C	T
μ_r	Relative permeability	-
α_{Br}	Temperature coefficient	% °C^{-1}
-	Demagnetization model	-
Steel sheet		
-	Core loss model	-

Table C.8: Output–Electric parameter module

Symbol	Quantity	Unit
R_{s20}	Phase resistance at °C	Ω
R_s	Phase resistance at actual temperature	Ω
L_s	Inductance	H

Table C.9: Output–Magnetic module

Symbol	Quantity	Unit
$\hat{B_{ga1}}$	Peak air gap flux density due to armature current	T
$\hat{B_{g1_{Br1}}}$	Peak air gap flux density due to PM for B_r=1	T
$\hat{\Phi_{g1_{Br1}}}$	Total flux per phase for B_r=1	Wb
B_r	PM remanence flux density	T
$\hat{B_{g1}}$	Peak air gap flux density due to PM	T
$\hat{\Phi_{g1_{Br1}}}$	Total flux per phase	Wb
$\hat{B_{gres1}}$	Result flux density in air gap	T
B_{Y_s}	Resulting flux density in stator yoke	T
B_{Y_r}	Resulting flux density in rotor yoke	T
B_t	Resulting flux density in tooth	T
B_{rmin}	Minimum allowable PM flux density	T

Table C.10: Output–Electric control module

Symbol	Quantity	Unit
I_{rms}	RMS current	A
i_d	d-axis current	A
i_q	q-axis current	A
V_{rms}	RMS voltage	V
V_d	d-axis voltage	V
V_q	q-axis voltage	V
Ψ	Angle between back-EMF and phase current	°
φ	Angle between current and voltage	°
T_{em}	Electromagnetic torque	Nm

Table C.11: Output–Other modules

Symbol	Quantity	Unit
Losses		
P_{cu}	Copper losses	W
P_{core_t}	Core losses in teeth	W
$P_{core_{Ys}}$	Core losses in stator yoke	W
η	Efficiency	%
E	Energy consumption	kWh
Thermal		
T_i	Temperature at node i	K
T_w	Winding temperature	K
T_{PM}	PM temperature	K
Mass		
M_j	Mass of element j	kg
M	Total mass	kg
Cost		
$Cost_j$	Cost of element j	euro
$Cost$	Total cost	euro

D

Optimal design of a traction system

As appendix C, all materials of this appendix are extracted from [Kreuawan, 2008] thesis and [Kreuawan *et al.*, 2008b].

D.1 Introduction

As presented in [Kreuawan, 2008] thesis, Alstom Transport has developed the standard traction system for tram applications in order to minimise cost and delivery time. Each applicative project has different customer specifications. These components are chosen and adapted according to these specifications in order to satisfy the global performance of trams operating on a predefined track.

In this study, the goal of Alstom is to increase the passenger capacity of the tram. The traction system would be re-designed due to the increase in global weight. The hardware redesign is discarded because the cost is unacceptable. For this reason, only control parameters can be modified. The solution has to meet not only the hardware constraints but also the customers' specifications.

D.2 Tram traction system design problem

D.2.1 Problem description

Due to a change in passenger capacity, the vehicle weight is increased. In the normal operating mode, the simulation results show that the tram can be operated perfectly without any problem. According to the customer contract specifications, in the faulty operating mode (one traction box is defective) the tram must be able to continue the service until the end of the trip with the remaining 50% motorization (3 motors) with the same performance as in the normal operating mode, except that the maximum speed had been reduced. The simulation results show that the required performance can be achieved [Kreuawan *et al.*, 2008b]. However, some problems are observed:

1. The temperature of IGBT (Insulate Gate Bipolar Transistor) modules used in VSIs was over the limit of 125°C defined by the semi-conductor manufacturer. The normal operation of IGBT cannot be guaranteed beyond this temperature limit. Figure D.1 shows the simulation results obtained by using the simulation tools i.e. CITHEL. The maximum temperature reaches 170°C;

2. Due to the increased torque, the line current did not respect the line filter specifications.

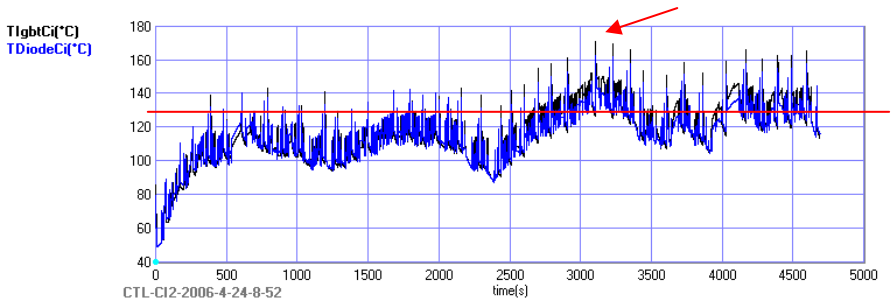


Figure D.1: IGBT temperature – initial design, 50% motorization mode

D.2.2 Modelling of traction systems

An overview of railway traction systems has been given in [Kreuawan, 2008] thesis. A tram is usually equipped with a DC traction system. For low power application, the DC power supply allows decreasing the cost of the traction system, as a transformer and a rectifier are not needed.

A simplified schema of a tram traction system is shown in Figure D.2. A pantograph collects electricity from the direct current (DC) overhead supply with a nominal voltage of 750V. Electricity passes through a line filter and then through a voltage source inverter (VSI) and finally traction motors. Normally, a regenerative braking is used. However, for security reasons, a braking chopper with a braking resistance is connected to a DC bus, for example in the case of voltage saturation of the DC line.

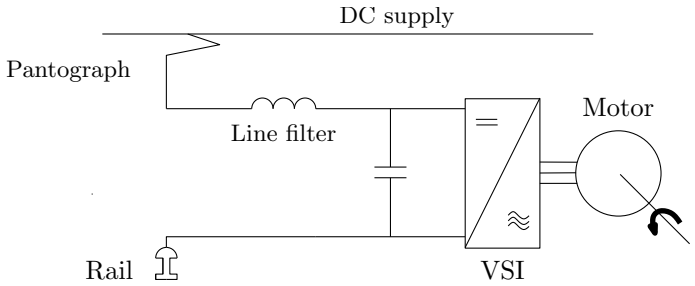


Figure D.2: Simplified power scheme of tram traction system.

In reality, the studied tram is composed of 2 traction boxes. Each traction box contains a line filter, 2 braking choppers and 3 VSIs, which can drive one motor each, and other electrical equipments e.g. breakers, sensors etc.

The traction system model is an in-house code of Alstom Transport–CITHEL. The model is the integration of the train system, the traction system and its components. These models are described with different levels of complexity and form a system model. It allows simulating the behaviour of rolling stocks operating on tracks e.g. a round-trip of tram (mission of more than 1 hour). It requires a lot of knowledge:

1. the rolling stock characteristics e.g. the aerodynamic properties, the component parameters.

2. the track characteristics i.e. the track curve, the speed limit, the stop station, the stop time, the altitude level.

3. the driving performance requirements e.g. the maximal acceleration, the deceleration.

The simulation takes into account the transient thermal behaviour of components such as the motor and the inverter. The electrical computation only takes the steady state into

consideration. At each time step, it computes the global dynamic values of the train (speed, acceleration, etc.) and the local electrical, thermal, and mechanical values of the components (line current, IGBT temperature, output torque, etc.).

The main role of CITHEL is to define the architecture of traction systems (number of motors, inverters, etc.) and to verify the operation of a train on a track based on the components already developed. The component properties are defined by rather global parameters, for example, the motor model is a single-phase equivalent circuit model and is not linked to the geometries of the motor. The simulation must be performed in many operating scenarios in order to satisfy the specification requirements.

Usually, several missions (different tracks) and several operation modes including normal operation and various faulty situations, have to be taken into account. However in this problem, the track profile of only one customer and a 50% motorization faulty mode are taken into consideration. Two driving strategies are tested: with and without regenerative braking. This allows verifying the maximum current passing through the line filter and the braking resistance.

The surrogate model approach [Kreuawan *et al.*, 2008b] is used to decrease the computation time. Figure D.3 depicts the main idea of a surrogate model approach. The surrogate model uses information from the high fidelity model and provides a cheap-to-evaluate model estimating the high fidelity model. It replaces the high fidelity model in the optimal design process. In this work, the Kriging surrogate model [Lebensztajn *et al.*, 2004] is constructed independently for each output.

The traction system design tool is then replaced by the surrogate models during the optimization process. The computation time is then decreased from more than one hour to less than one second per design evaluation.

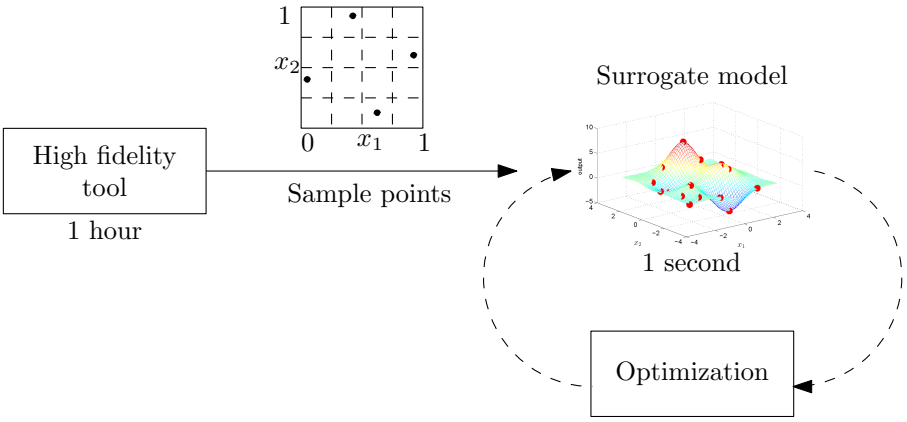


Figure D.3: Surrogate modelling approach.

In this example, 108 sample points have been computed with the high fidelity tool i.e. CITHEL. These sample points are used to build 12 surrogate models of CITHEL outputs.

D.3 Design optimization using surrogate model

D.3.1 Overview

In spite of the increasing in computer speed, engineers always meet the difficulty in integrating the expensive computational cost and high fidelity simulation software in the design optimization process. Many researchers are working on the optimization algorithm to decrease model evaluation number requirement [Moussouni *et al.*, 2007]. Others focus on the development of approximation method called surrogate model such as Response Surface Methodology, Ra-

dial Basis Function, Neural Network, Kriging, etc. [Jin *et al.*, 2001]. The surrogate model (or meta-model) is constructed using the sample data computed by high fidelity software (or fine model). It replaces the fine model during the optimization process. A great advantage of the surrogate model is the reduction in the computation time. However the main drawback is the trade-off between the accuracy and the computational time. Once the surrogate model is built, it should be validated. The assessment strategies of surrogate model can be found in [Meckesheimer *et al.*, 2002].

Figure D.4 shows the typical flowchart of an optimization process with a surrogate model. This optimization approach begins with the selection of sample points. The expensive models are then evaluated at these points. The responses are therefore used to construct surrogate models. These surrogate models replace the expensive models in the optimization process. Finally the optimization results are analysed and validated by the engineers.

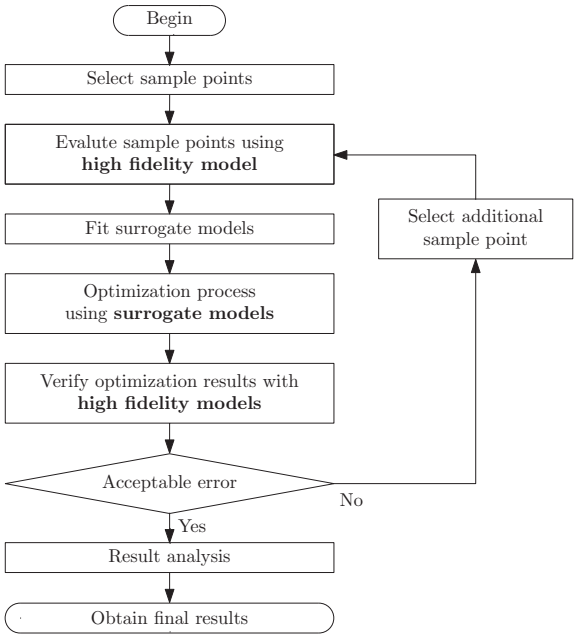


Figure D.4: Optimization using surrogate model.

The sample points used to build the surrogate model can be selected based on design of experiment (DOE) (e.g. full factorial design, central composite design) [Vivier, 2002a] or based on Space filling design (e.g. Latin Hypercube Sampling, Orthogonal array) [Giunta *et al.*, 2003].

D.3.2 Kriging surrogate model

Kriging was first developed by D. Krige, mining engineer. It was used with success in the field of geological statistics to estimate mineral concentrations over area of interest given a set of sampled sites [Chiles & Delfiner, 1999]. In the field of computer science and engineering, it was introduced by Sacks et al. as Design and Analysis of Computer Experiments (DACE) [Sacks *et al.*, 1989]. Kriging method was also used with success in many engineering application (e.g. [Lebensztajn *et al.*, 2004]). Therefore, in this paper, Kriging is used to build surrogate models.

In Kriging approach, a modelling function can be written in this form:

$$y(x) = B(x) + Z(x) \tag{D.1}$$

The first term $B(x)$ is an unknown constant or a regression model, which represents the global trend of the function. The second term $Z(x)$ gives the localized deviations from the

global trend. $Z(x)$ is a model of a stochastic process with zero mean and variance of σ^2 and covariance defined by:

$$Cov\left[Z\left(x^{(i)}\right),\left(x^{(j)}\right)\right]=\sigma^2\mathbf{R}\left[R\left(x^{(i)},x^{(j)}\right)\right] \tag{D.2}$$

where $\mathbf{R}$ is the correlation matrix, R is the correlation function, which is selected by user, i and j are the sample points (from 1 to sample point number, n_s). The correlation function controls the smoothness of the model. Various correlation functions are given in [Sacks *et al.*, 1989]. The Gaussian function is the most often used one.

$$R\left(x^{(i)},x^{(j)}\right)=\exp\left[-\sum_{k=1}^{n_v}\theta_k\left|x_k^{(i)}-x_k^{(j)}\right|^2\right] \tag{D.3}$$

where n_v is the number of design variables and θ_k is the unknown correlation function parameter vector.

While (D.1) defines the true response value y, the Kriging model predicts the estimated response value $\hat{y}$. The Mean Square Error (MSE) is the expected value of difference between the true response and the estimated one.

$$MSE=E\left(\left(y\left(x\right)-\hat{y}\left(x\right)\right)^2\right) \tag{D.4}$$

Since Kriging interpolates the data, MSE is zero at the sample points. At the other unknown points, MSE is supposed to be minimized in order to obtain a good approximation. If MSE is minimized, Kriging model becomes:

$$\hat{y}=\hat{B}+\mathbf{r}^T(x)\mathbf{R}^{-1}\left(y-\mathbf{f}\hat{B}\right) \tag{D.5}$$

and MSE (D.4) can be rewritten as:

$$MSE=\sigma^2\left[1-\mathbf{r}^T(x)\mathbf{R}^{-1}\mathbf{r}+\frac{\left(1-\mathbf{f}^T(R)^{-1}\mathbf{r}\right)^2}{\mathbf{f}^T(R)^{-1}\mathbf{f}}\right] \tag{D.6}$$

where $\hat{B}$ is unknown, $r(x)$ is the correlation vector between a new location x to be estimated and the sampled locations, $\mathbf{f}$ is a unit vector with length of n_s.

$$\mathbf{r}(x)=\left[R\left(x,x^{(1)}\right),R\left(x,x^{(2)}\right),\ldots,R\left(x,x^{(n_s)}\right)\right] \tag{D.7}$$

$$\mathbf{R}=\begin{bmatrix} R\left(x^{(1)},x^{(1)}\right) & R\left(x^{(1)},x^{(2)}\right) & \cdots & R\left(x^{(1)},x^{(n_s)}\right)\\ R\left(x^{(2)},x^{(1)}\right) & R\left(x^{(2)},x^{(2)}\right) & \cdots & R\left(x^{(2)},x^{(n_s)}\right)\\ \vdots & \vdots & \ddots & \vdots\\ R\left(x^{(n_s)},x^{(1)}\right) & R\left(x^{(n_s)},x^{(2)}\right) & \cdots & R\left(x^{(n_s)},x^{(n_s)}\right)\end{bmatrix} \tag{D.8}$$

Note that $\mathbf{R}$ is symmetric $R\left(x^{(i)},x^{(j)}\right)=R\left(x^{(j)},x^{(i)}\right)$ with ones along diagonal $R\left(x^{(i)},x^{(i)}\right)=1$.

$\mathbf{R}$ and $\mathbf{r}$ depend on θ_k, which can be found using Maximum Likelihood Estimation (MLE).

The likelihood function is defined as:

$$\frac{1}{\sqrt{(2\pi\hat{\sigma}^2)^{n_s}\left|\mathbf{R}\right|}}\cdot\exp\left(\frac{-\left(y-\mathbf{f}\hat{B}\right)^T\mathbf{R}^{-1}\left(y-\mathbf{f}\hat{B}\right)}{2\hat{\sigma}^2}\right) \tag{D.9}$$

The log-likelihood function is usually used in order to simplify (D.9).

$$\frac{-\left(y-\mathbf{f}\hat{B}\right)^{T}\mathbf{R}^{-1}\left(y-\mathbf{f}\hat{B}\right)}{2\hat{\sigma}^{2}}-\frac{n_s\ln\left(2\pi\hat{\sigma}^{2}\right)+\ln\left(|\mathbf{R}|\right)}{2} \tag{D.10}$$

$\hat{\beta}$ and $\hat{\sigma}^2$ can be estimated as:

$$\hat{\beta}=\left(\mathbf{f}^{T}\mathbf{R}^{-1}\mathbf{f}\right)^{-1}\mathbf{f}^{T}\mathbf{R}^{-1}y \tag{D.11}$$

$$\hat{\sigma}^{2}=\frac{1}{n_s}\left(\left(y-\mathbf{f}\hat{\beta}\right)^{T}\cdot\mathbf{R}^{-1}\cdot\left(y-\mathbf{f}\hat{\beta}\right)\right) \tag{D.12}$$

Substituting $\hat{\beta}$ and $\hat{\sigma}^2$, (D.10) becomes:

$$-\frac{n_s\ln\left(\hat{\sigma}^{2}\right)+\ln\left(|\mathbf{R}|\right)}{2} \tag{D.13}$$

By solving the MLE optimization problem (D.14), θ_k can be determined. The response at any design point x can be estimated using (D.5).

$$\begin{array}{ll}\max\limits_{\theta_k} & -\dfrac{n_s\ln\left(\hat{\sigma}^{2}\right)+\ln\left(|\mathbf{R}|\right)}{2}\\ \text{subject to} & 0<\theta_k<\infty\end{array} \tag{D.14}$$

Reference [Jin *et al.*, 2001] shows that the Kriging method works slightly better that other surrogate models. However, Kriging needs more computational time to solve the MLE optimization problem in order to construct the model. This time could be very high in large-scale problem with large sample size. However, in the real implementation, it does not cause any trouble due to the fact that the surrogate model is built only one time, moreover the using of this model is very fast compared to the fine model.

D.4 Conclusion

In this appendix, a design of traction system is presented. The models are built using the surrogate approach. This traction system is used as a benchmark in chapter 8 in order to show the efficiency of the proposed multi-level design optimization.

Bibliography

[Adeli, 1994] Adeli, H. 1994. *Advances in design optimization.* Chapman and Hall.

[Alexandrov & Lewis, 1999] Alexandrov, N.M., & Lewis, R.M. 1999. *Comparative Properties of Collaborative Optimization and Other Approaches to MDO.* Tech. rept. Institute for Computer Applications in Science and Engineering, NASA Langley Research Center.

[Alexandrov, 2002] Alexandrov, Natalia M.; Lewis, Robert Michael. 2002. Analytical and Computational Aspects of Collaborative Optimization for Multidisciplinary Design. *American Institute of Aeronautics and Astraunotics (AIAA) Journal*, **40**(2), 301–309.

[Alighanbari *et al.*, 2005] Alighanbari, M., Homaifar, A., & Sayarrodsari, B. 2005 (Oct.). Robust adaptive control and parameter estimation using multi objective evolutionary algorithm. *Pages 1326–1333 of: IEEE International Conference on Systems, Man and Cybernetics*, vol. 2.

[Allison *et al.*, 2005a] Allison, J., Walsh, D., Kokkolaras, M., Papalambros, P. Y., & Cartmell, M. 2005a. *Analytical target cascading in aircraft design.* Tech. rept. American Institute of Aeronautics and Astronautics.

[Allison *et al.*, 2005b] Allison, J., Kokkolaras, M., Zawislak, M., & Papalambros, P. Y. 2005b (May). On the use of analytical target cascading and collaborative optimization for complex system design. *In: 6th World Congress on Structural and Multidisciplinary Optimization.*

[Allison *et al.*, 2007] Allison, J. T., Kokkolaras, M., & Papalambros, P. Y. 2007. On selecting single-level formulation for complex system design optimization. *Journal of Mechanical Design*, **129**(Sept.), 898–906.

[Allison, 2004] Allison, T. J. 2004. *Complex System Optimization: A Review of Analytical Target Cascading, Collaborative Optimization, and Other Formulation.* M.Phil. thesis, University of Michigan.

[Bandler *et al.*, 2004] Bandler, J.W., Cheng, Q.S., Dakroury, S.A., Mohamed, A.S., Bakr, M.H., Madsen, K., & Sondergaard, J. 2004. Space mapping: the state of the art. *IEEE Trans. Microwave Theory Tech.*, **52**(1), 337–361.

[Besharati *et al.*, 2006] Besharati, B., Luo, L., Azarm, S., & Kannan, P. K. 2006. Multi-objective single product robust optimization: an integrated design and marketing approach. *Journal of Mechanical Design*, **128**(4), 884–892.

[Beyer & Deb, 2001] Beyer, H. G., & Deb, K. 2001. On self-adaptive features in real-parameter evolutionary algorithm. *IEEE Trans. on Evol. Comp.*, **5**(3), 250–270.

[Boglietti *et al.*, 2003] Boglietti, A., Cavagnino, A., Lazzari, M., & Pastorelli, A. 2003. A simplified thermal model for variable speed self cooled industrial induction motor. *IEEE Transactions on Industry Applications*, **39**(4), 945–952.

[Bouchard & Olivier, 1997] Bouchard, R., & Olivier, G. 1997. *Conception de moteurs asynchrones triphasés*. Editions de l'école polytechnique de montréal. In French.

[Braun *et al.*, 1996] Braun, R. D., Moore, A. A., & Kroo, I. M. 1996. *Use of the Collaborative Optimization Architecture for Launch Vehicle Design*. Tech. rept.

[Brisset & Brochet, 2005] Brisset, S., & Brochet, P. 2005. Analytical model for the optimal design of a brushless DC wheel motor. *COMPEL*, **24**(3).

[Cai & Wang, 2006] Cai, Z., & Wang, Y. 2006. A multiobjective optimization-based evolutionary algorithm for constrained optimization. *IEEE Trans. on Evol. Comp.*, **10**(Dec.), 658–675.

[Cao & Wu, 1997] Cao, Y.J., & Wu, Q.H. 1997. Mechanical design optimization by mixed-variable evolutionary programming. *IEEE Trans. on Evol. Comp.*, Apr., 443–446.

[Chiles & Delfiner, 1999] Chiles, J.-P., & Delfiner, P. 1999. *Geostatistics: Modeling Spatial Uncertainty*. New York: Wiley.

[Choi *et al.*, 2001] Choi, H., Kim, D., Park, I., & Hahn, S. 2001. A new design technique of magnetic systems using space mapping algorithm. *IEEE Trans. on Magn.*, **37**(5), 3627–3630.

[Choudhary *et al.*, 2005] Choudhary, R., Malkawi, A., & Papalambros, P. Y. 2005. Analytic target cascading in simulation-based building design. *Automation in Construction*, **14**, 551 568.

[Chung *et al.*, 1996] Chung, V., Zhang, B.X., & Lee, E.T. 1996. A multi-objective optimization of radiative fin array systems in a fuzzy environment. *Journal of Heat Transfer*, **118**(3), 642–649.

[Cioffi *et al.*, 2005] Cioffi, M., Formisano, A., & Martone, R. 2005. Statistical analysis in robust design of superconducting magnets. *COMPEL*, **24**(2), 427–435.

[Clerc, 2003a] Clerc, M. 2003a (Oct.). Exemple de logiciel dOEP sans paramètres de réglage : Tribes, ou la coopération de tribus. *In: Séminaire : L'optimisation par essaim de particules (OEP)*.

[Clerc, 2003b] Clerc, M. 2003b (Oct.). *Tutoriel 1 : L'optimisation par essaim particulaire*. Tech. rept. France Télécom R&D.

[Clerc & Kennedy, 2002] Clerc, M., & Kennedy, J. 2002. The particle swarm : explosion, stability, and convergence in a multi-dimensional complex space. *IEEE Trans. on Evol. Comp.*, **4**, 58–73.

[Coelho, 2004] Coelho, R. F. 2004. *Multicriteria optimization with expert rules for mechanical Design*. Ph.D. thesis, Université Libre de Bruxelles (ULB) Faculté des Sciences Appliquées.

[Collette & Siarry, 2002] Collette, Y., & Siarry, P. 2002. *Optimisation multiobjectif*. EYROLLES.

[Colorni *et al.*, 1992] Colorni, A., Dorigo, M., & Maniezzo, M. 1992. An investigation of some properties of an ant algorithm. *Pages 509–520 of: 2th conference on the parallel solving from nature*. Bruxelles: ELSEVIER Publishing.

[Cooper *et al.*, 2006] Cooper, A. B., Georgiopoulos, P., Kim, H. M., & Papalambros, P. Y. 2006. Analytical target setting: an enterprise context in optimal product design. *Journal of Mechanical Design*, **128**(Jan.), 4–13.

[De Wit, 2005] De Wit, J. 2005 (Sept.). *Evaluation of Mixed Integer Nonlinear Programming routines.* Ph.D. thesis, Technische Universiteit Eindhoven.

[Deb, 2002] Deb, K. 2002. *Multi-objective optimization using evolutionary algorithms.*

[Deb *et al.*, 2002] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Trans. on Evol. Comp.*, **6**(2), 181–197.

[Dorigo & Blumb, 2005] Dorigo, M., & Blumb, C. 2005. Ant colony optimization theory: A survey. *Theoretical Computer Science*, **344**, 243278.

[Dorigo & Gambardelle, 1997] Dorigo, M., & Gambardelle, L. M. 1997. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans. on Evol. Comp.*, **1**(1), 1–24.

[Dorigo *et al.*, 1996] Dorigo, M., Maniezzo, V., & Colorni, A. 1996. The ant system: optimization by a colony of cooperating agents. *IEEE Trans. Systems, Man, Cybernet-Part B*, **26**(1), 1–13.

[Dorigo *et al.*, 1999] Dorigo, M., Di Caro, G., & M., Gambardella L. 1999. Ant algorithms for discrete optimization. *Artificial Life*, **5**(3), 137–172.

[Dréo, 2004] Dréo, J. 2004. *Adaptation de la méthode des colonies de fourmis pour l'optimisation en variables continues : application en génie biomédical.* Ph.D. thesis, Paris 12.

[Dréo *et al.*, 2006] Dréo, J., Pétrowski, A., Siarry, P., & Taillard, E. 2006. *Metaheurstics for hard optimization.* Springer-Verlag Berlin Heidelberg.

[Dutot, 2005] Dutot, A. 2005 (Dec.). *Distribution dynamique adaptative à l'aide des méchanismes d'intelligence collective.* Ph.D. thesis, Université du Havre, LIH Laboratoire Informatique du Havre.

[Echeverria *et al.*, 2006] Echeverria, D., Lahaye, D., Encica, L., Lomonova, E.A., Hemker, P.W., & Vandenput, A.J.A. 2006. Manifold-mapping optimization applied to linear actuator design. *IEEE Trans. on Magn.*, **42**(4), 1183–1186.

[Eiben *et al.*, 1999] Eiben, A.E., Hinterding, R., & Michalewicz, Z. 1999. Parameter control in evolutionary algorithms. *IEEE Trans. on Evol. Comp.*, **3**(2), 124–141.

[Encica *et al.*, 2008] Encica, L., Paulides, J.J.H., Lomonova, E.A., & Vandenput, A.J.A. 2008. Aggressive output space-mapping optimization for electromagnetic actuators. *IEEE Trans. on Magn.*, **44**(6), 1106–1109.

[Erbas *et al.*, 2006] Erbas, C., Cerav-Erbas, S., & Pimentel, A.D. 2006. Multiobjective optimization and evolutionary algorithms for the application mapping problem in multiprocessor system-on-chip design. *IEEE Trans. on Evol. Comp.*, **10**(3), 358–374.

[Fadel *et al.*, 2003] Fadel, G. M., Blouin, V. Y., & Haque, I. 2003. *Continuously variable transmission design for optimum vehicle performance by analytical target cascading.* Business Briefing : GLOBAL AUTOMOTIVE MANUFACTURING & TECHNOLOGY. Department of Mechanical Engineering, Clemson University.

[Fletcher & Powell, 1963] Fletcher, R., & Powell, M.J.D. 1963. A Rapidly Convergent Descent Method for Minimization. *Computer Journal*, **6**, 163–168.

[Fonseca & Fleming, 1996] Fonseca, C. M., & Fleming, P. J. 1996. On the performance assessment and comparison of stochastic multiobjective optimizers. *Lecture Notes in Computer Science, Springer-Verlag*, **1141**, 584593.

[Fonseca & Fleming, 1993] Fonseca, Carlos M., & Fleming, Peter J. 1993. Genetic Algorithms for multiobjective optimization: Formulation, discussion and generalization. *Pages 416–423 of: In Genetic Algorithms: Proceeding of the Fifth International Conference*. Morgan Kaufmann.

[Fox, 1971] Fox, R.L. 1971. *Optimization methods for engineering design*. Mechanics and Thermodynamics. Reading, Massachusetts - Menlo Park, California - London - Don Mills, Ontario: Addition-Wesley Publishing Company.

[Gillon, 1997] Gillon, F. 1997. *Modélisation et optimisation par plans d'expérience déun moteur à commutations électroniques*. Ph.D. thesis, Université des sciences et technologies de lille. In French.

[Giunta *et al.*, 2003] Giunta, A., Wojtkiewicz, S., & Eldred, M. 2003 (Jan.). Overview of Modern Design of Experiments Methods for Computational Simulations. *In: Proceeding of the 41st Aerospace Sciences Meeting and Exhibit*. AIAA–2003–649. 6-9.

[Goldberg, 1989] Goldberg, D.E. 1989. *Genetic algorithm in search, optimization and machine learning*. Addison Wesley.

[Haimes *et al.*, 1971] Haimes, Y.Y., Lasdon, L.S., & Wismer, D. A. 1971. On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization. *IEEE Transactions on Systems Man, and Cybernetics*, **1**(3), 296–297.

[HAJELA & C.-J., 1990] HAJELA, P., & C.-J., SHIH. 1990. Multiobjective optimum design in mixed integer and discrete design variable problems. *American Institute of Aeronautics and Astraunotics (AIAA) journal*, **28**(4), 670–675.

[Hansen & William Walster, 2003] Hansen, E. R., & William Walster, G. 2003. *Global Optimization Using Interval Analysis: Revised And Expanded*. CRC, 2 edition.

[Hendershot & Miller, 1994] Hendershot, J. R., & Miller, T. J. E. 1994. *Design of brushless permanent-magnet motors*. Oxford: Magna physics publishing and Clarendon press.

[Holtz, 1992] Holtz, J. 1992. Pulsewidth modulation - a survey. *IEEE Transactions on Industrial Electronics*, **39**(5), 410–420.

[Horn *et al.*, 1993] Horn, J., Nafpliotis, N., & Golcberg, D.E. 1993. *Multiobjective optimization using the niched Pareto genetic algorithm*. Report 93005. Université de lIllinois, Urbana, Illinois.

[Horn *et al.*, 1994] Horn, J., Nafpliotis, N., & Golcberg, D.E. 1994. A Niched Pareto Genetic Algorithm for Multiobjective Optimization. *IEEE World Congress on Computational Intelligence*, **1**, 82–87.

[Horst & Tuy, 1995] Horst, R., & Tuy, H. 1995. *Global optimization deterministic approaches*. Springer-Verlag Berlin Heidelberg New York.

[Hou *et al.*, 2002] Hou, Y.H., Wu, Y.W., Lu, L.J., & Xiong, X.Y. 2002 (Oct.). Generalized ant colony optimization for economic dispatch of power systems. *Pages 225–229 vol.1 of: International Conference on Power System Technology (PowerCon 2002)*, vol. 1. 13-17 Oct.

[Huang *et al.*, 2005] Huang, G.Q. T.Qu ., Cheung, D. W. L., & Liang, L. 2005. Extensible multi-agent system for optimal design of complex systems using analytical target cascading. *Int. J. Adv0 Manuf. Technol.*, Oct.

[Hwang & Masud, 1979] Hwang, C.-L., & Masud, A.S.M. 1979. *Multiple Objective Decision Making, Methods and Applications: A State-of-the-Art Survey.* Lecture notes in Economics and Mathematical Systems, vol. 186. Springer-Verlag.

[Jin *et al.*, 2001] Jin, R., Chen, W., & Simpson, T.W. 2001. Comparative studies of metamodelling techniques under multiple modelling criteria. *Structural and Multidisciplinary Optimization*, **23**(1), 1–13.

[Joshua *et al.*, 2006] Joshua, D., Knowles, D., Thiele, L., & Zitzler, E. 2006 (Feb.). *A tutorial on the performance assessment of stochastic multi-objective optimizers.* TIK-Report 214. Computer Engineering and Networks Laboratory, ETH Zurich.

[Kaisa, 2004] Kaisa, M. 2004. *Non-linear multiobjective optimization.* Kluwer Academic Publishers Group.

[Kennedy & Eberhat, 1995] Kennedy, J., & Eberhat, R. C. 1995. Particle swarm optimization. *In: IEEE International conference on Neural Networks, Perth, Australia.* IEEE Service Center, Piscatway, NJ.

[Kim & Nho, 1998] Kim, C.-K., & Nho, K.-M. 1998. Heat sink design of high power converter. *Pages 2122–2130 of: proceeding of 29th Annual IEEE Power Electronics Specialists Conference (PESC)*, vol. 2.

[Kim *et al.*, 1998] Kim, D.H., Lee, J.H., Hong, S.H., & Kim, S.R. 1998. A mixed-integer programming approach for the linearized reactive power and voltage control-comparison with gradient projection approach. *Energy Management and Power Delivery*, **1**(Mar.), 67–72 vol.1.

[Kim *et al.*, 2003a] Kim, H. M., Rideout, D. G., Papalambros, P. Y., & Stein, J. 2003a. Analytical target cascading in automotive vehicle design. *Journal of Mechanical Design*, **125**(Sept.), 1–9.

[Kim *et al.*, 2003b] Kim, H.M., Michelena, N.F., Papalambros, P. Y., & Jiang, T. 2003b. Target cascading in optimal system design. *Journal of Mechanical Design*, **125**(3), 474–480.

[Kim *et al.*, 2006a] Kim, H.M., Chen, W., & Wiecek, M. 2006a. Lagrangian coordination for enhancing the convergence of analytical target cascading. *American Institute of Aeronautics and Astraunotics (AIAA) Journal*, **44**(10), 2197–2207.

[Kim *et al.*, 2006b] Kim, H.M., Kumar, D.K.D., & Chen, W. 2006b. Target exploration for disconnected feasible region in entreprise-driven multilevel product design. *American Institute of Aeronautics and Astraunotics (AIAA) Journal*, **44**(Jan.), 67–77.

[Kim & De Weck, 2005] Kim, I.Y., & De Weck, O.L. 2005. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and Multidisciplinary Optimization*, **29**, 149158.

[Kim & De Weck, 2006] Kim, I.Y., & De Weck, O.L. 2006. Adaptive weighted sum method for multiobjective optimization: a new method for Pareto front generation. *Structural and Multidisciplinary Optimization*, **31**, 105116.

[Kokkolaras *et al.*, 2002] Kokkolaras, M., Fellini, R., Kim, H. M., Michelena, N. F., & Papalambros, P. Y. 2002. Extension of the target cascading formulation to the design of product families. *Structural and Multidisciplinary Optimization*, **24**, 293301.

[Kokkolaras *et al.*, 2004] Kokkolaras, M., Louca, L.S., Delagrammatikas, G.J., Michelena, N.F., Filipi, Z.S., Papalambros, P.Y., J.L., Stein, & Assanis, D.N. 2004. Simulation-based optimal design of heavy trucks by model-based decomposition: An extensive analytical target cascading case study. *Int. J. of Heavy Vehicle Systems*, **11**(3/4), 403–433. Department of Mechanical Engineering, University of Michigan, Ann Arbor, Michigan 48109, USA Email: mk@umich.edu.

[Kreuawan *et al.*, 2009] Kreuawan, S. Moussouni, F., Gillon, F., Brisset, S., Brochet, P., & Nicod, L. 2009. Optimal design of traction system using Target Cascading. *International journal of Applied Electromagnetics and Mechanics - IJAEM.*

[Kreuawan, 2008] Kreuawan, S. 2008 (Nov.). *Modelling and optimal design in railway applications.* Ph.D. thesis, Ecole Centrale de Lille, France.

[Kreuawan *et al.*, 2007] Kreuawan, S., Gillon, F., Moussouni, F., Brisset, S., & Brochet, P. 2007 (Sep.). Optimal design of traction motor in railway propulsion system. *Pages 343–348 of: Proceeding of International Aegean Conference on Electric Machines, Power Electronics and Electromotion Joint Conference.*

[Kreuawan *et al.*, 2008a] Kreuawan, S., Gillon, F., & Brochet, P. 2008a. Comparative study of design approach for electric machine in traction application. *International Review of Electrical Engineering (IREE)*, **3**(May), 455–465.

[Kreuawan *et al.*, 2008b] Kreuawan, S., Moussouni, F., Gillon, F., Brisset, S., Brochet, P., & Porcher, F. 2008b. Optimal design process applied to electric railway traction system. *Transactions on Systems, Signals & Devices*, Sep.

[Kreuawan *et al.*, 2008c] Kreuawan, S., Gillon, F., & Brochet, P. 2008c (May). Surrogate-assisted multiobjective optimization for optimal design using finite element analysis. *In: Proceeding of 13th Biennial IEEE Conference on Electromagnetic Field Computation (CEFC).*

[Krishnamachari & Papalambros, 1995] Krishnamachari, R. S., & Papalambros, P. Y. 1995. Hierarchical decomposition synthesis in optimal systems design. *Engineering, College of - Technical Reports*, Aug.

[Kurpati *et al.*, 2002] Kurpati, A., Azarm, S., & Wu, J. 2002. Constraint handling improvements for multiobjective genetic algorithms. *Structural and Multidisciplinary Optimization*, **23**(3), 204–213.

[Lassiter *et al.*, 2005] Lassiter, J. B., Wiecek, M. M., & Andrighetti, K. 2005. Lagrangian coordination and analytical target cascading: solving ATC-Decomposed problems with lagrangian duality. *Optimization and Engineering*, **6**(March), 361381.

[Lebensztajn *et al.*, 2004] Lebensztajn, L., Marretto, C.A.R., Costa, M.C., & Coulomb, J.-L. 2004. Kriging: a useful tool for electromagnetic device optimization. *IEEE Transactions on Magnetics*, **40**(2), 1196–1199.

[Leyffer, 2001] Leyffer, S. 2001. Integrating SQP and Branch-and-Bound for Mixed Integer Nonlinear Programming. *Comput. Optim. Appl.*, **18**(3), 295–309.

[Lin *et al.*, 1999] Lin, Y-C., Wang, F-S., & Hwang, K-S. 1999. A hybrid method of evolutionary algorithms for mixed-integer nonlinear optimization problems. *Proceedings on Evolutionary Computation, CEC 99.*, **3**, 2166.

[Ling *et al.*, 2006] Ling, L-Y., Liang, L., & Pu, X-J. 2006. ATC based coordination of distributed production planning and supplier selection. *Applied Mathematics and Computation.*

[Liu *et al.*, 2006] Liu, H., Chen, W., Kokkolaras, M., Papalambros, P.Y., & Kim, H.M. 2006. Probabilistic analytical target cascading: a moment matching formulation for multilevel optimization under uncertainty. *Journal of Mechanical Design*, **128**(4), 991–1000.

[MathWorks, 1984-2006] MathWorks, The. 1984-2006. *Distributed computing Toolbox for use with Mathlab.* Users Guide version 3. The MathWorks Inc.

[McKay *et al.*, 2000] McKay, M. D., Beckman, R. J., & Conover, W. J. 2000. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, **42**(1), 55–61.

[Meckesheimer *et al.*, 2002] Meckesheimer, M., Booker, A.J., Barton, R.R., & Simpson, T.W. 2002. Computationally inexpensive metamodel assessment strategies. *American Institute of Aeronautics and Astraunotics (AIAA) Journal*, **40**, 2053–2060.

[Mester *et al.*, 2006] Mester, V., Gillon, F., Brisset, S., & Brochet, P. 2006 (Sep.). Global optimal design of a wheel traction motor by a systemic approach of the electric drive train. *In: proceeding of IEEE Vehicle Power and Propulsion Conference (VPPC).*

[Meunier *et al.*, 2000] Meunier, H., Talbi, E.-G., & Reininger, P. 2000. A multiobjective genetic algorithm for radio network optimization. *Proceedings of the 2000 Congress on Evolutionary Computation, CEC*, **1**(Jul.), 317–324.

[Mezura-Montes & Coello, 2002] Mezura-Montes, E., & Coello, C.A.C. 2002 (Sep.). *A numerical comparison of some multiobjective-based techniques to handle constraints in genetic algorithms.* Tech. rept. EVOCINV-03-2002. Dept. de Ingeniería Eléctrica, Evolutionary Computation Group at CINVESTAV-IPN, Mexico, D.F. 07300.

[Michalek *et al.*, 2004] Michalek, J., Papalambros, P.Y., & Skerlos, S. 2004. A Study of Fuel Efficiency and Emission Policy Impact on Optimal Vehicle Design Decisions. *Journal of Mechanical Design*, **126**(6), 1062–1070.

[Michalek, 2005] Michalek, J. J. 2005. *Preference coordination in engineering design decision-making.* Ph.D. thesis, (Mechanical Engineering) in The University of Michigan. Doctoral Committee: Professor Panos Y. Papalambros, Chair Professor Richard D. Gonzalez Associate Professor Fred M. Feinberg Assistant Professor Steven J. Skerlos.

[Michalek *et al.*, 2005] Michalek, J. J., Feinberg, F. M., & Papalambros, P. Y. 2005. Linking marketing and engineering product design decisions via analytical target cascading. *J. Prod. Innov. Manag.*, **62**, 22–42.

[Michalek & Papalambros, 2005a] Michalek, J.J., & Papalambros, P.Y. 2005a. An efficient weighting update method to achieve acceptable consistency deviation in analytical target cascading. *Journal of Mechanical Design*, **127**(Mar.), 206–214.

[Michalek & Papalambros, 2005b] Michalek, J.J., & Papalambros, P.Y. 2005b. Weights, norms, and notation in analytical target cascading. *Journal of Mechanical Design*, **127**(3), 499–501.

[Michalek & Papalambros, 2006] Michalek, J.J., & Papalambros, P.Y. 2006 (Sep.). BB-ATC: Analytical target cascading using branch and bound for mixed-integer nonlinear programming.

[Michalewicz, 1994] Michalewicz, Z. 1994. *Genetic algorithms + Data structures = Evolution Programs.* Springer-Verlag.

[Michalewicz, 1995] Michalewicz, Z. 1995. A Survey of Constraint Handling Techniques in Evolutionary Computation Methods. *In: Proceedings of the4th annual Conference on Evolutionary Programming.*

[Michelena *et al.*, 1999] Michelena, N., Kim, H. M., & Papalambros, P. Y. 1999 (Aug.). A System Partitioning and Optimization Approach to Target Cascading. *In: International Conference on Engineering Design (ICED 99).*

[Michelena *et al.*, 2003] Michelena, N., Park, H., & Papalambros, P. 2003. Convergence properties of analytical target cascading. *American Institute of Aeronautics and Astraunotics (AIAA) Journal*, **41**(05), 897–905.

[Millor *et al.*, 1991] Millor, P. H., Roberts, D., & Turner, D. R. 1991. Lumped parameter thermal model for electrical machine of TEFC design. *Electric Power Application, IEE Proceedings B*, **138**(5), 205–218.

[Minoux, 1983] Minoux, M. 1983. *Programmation mathématique : théorie et algorithmes.* Technique et scientifiques des télécommunications, no. 1. Dunod.

[Mockus *et al.*, 1996] Mockus, J., Eddy, W., & Reklaitis, G. 1996. *Bayesian Heuristic Approach to Discrete and Global Optimization: Algorithms, Visualization, Software, and Applications.* Springer.

[Mohan *et al.*, 2003] Mohan, M., Deb, K., & Mishra, S. 2003. *A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions.* KanGAL Report 2.

[Morimoto *et al.*, 1990] Morimoto, S., Takeda, Y., Hirasa, T., & Taniguchi, K. 1990. Expansion of operating limits for permanent magnet motor by current vector control considering inverter capacity. *IEEE Transactions on Industry Applications*, **26**(5), 866–871.

[Moussouni *et al.*, 2007] Moussouni, F., Brisset, S., & Brochet, P. 2007. Some results on the design of the brushless DC wheel motor using SQP and GA. *International journal of Applied Electromagnetics and Mechanics - IJAEM*, **26**(3,4), 233–241.

[Moussouni *et al.*, 2008] Moussouni, F., Brisset, S., & Brochet, P. 2008. Comparison of two multi-agent algorithms ACO and PSO for the optimization of a brushless DC wheel motor. *SPRINGER, Studies in Computational Intelligence*, **119**(4), 3–10. series 7092.

[Moussouni *et al.*, 2009] Moussouni, F., Kreuawan, S., Brisset, S., Gillon, F., Brochet, P., & Nicod, L. 2009. Multi-level design optimization using target cascading. *COMPEL.*

[Murano *et al.*, 2008] Murano, A., Passaro, A., Abe, N.M., Preto, A.J., & Stephany, S. 2008 (May). Multiobjective Optimization of Electro-optic Modulators by Using the ϵ-constraint Method. *In: proceedings of CEFC 2008.*

[Narahari, 1998] Narahari, Y. 1998 (Feb.). *Data structures and algorithms.* Electronic Lecture Notes. Computer Science and Automation Indian Institute of Science, Bangalore-560012. hari@csa.iisc.ernet.in.

[Oliver *et al.*, 1997] Oliver, D. W., Kelliher, T. P., & G., Keegan Jr. J. 1997. *Engineering complex systems with models and objects.*

[Papalambros & Michelena, 2000] Papalambros, P.Y., & Michelena, N.F. 2000 (Aug.). Trends and challenges in system design optimization. *In: Proceedings of the International Workshop on Multidisciplinary Design Optimization.* Proceedings of the International Workshop on Multidisciplinary Design Optimization.

[Paraditwong & Yao, 2006] Paraditwong, K., & Yao, X. 2006. A new multi-objective evolutionary optimisation algorithm: the two-archive algorithm. *IEEE Comp. Intelligence and Security*, **1**(Nov.), 286–291.

[Powell, 1970] Powell, M. J. D. 1970. *A Fortran Subroutine for Solving Systems of Nonlinear Algebraic Equations.* Numerical Methods for Nonlinear Algebraic Equations. Ch.7.

[Qin & Zhao, 2006] Qin, Y. F., & Zhao, M. Y. 2006. Fuzzy multi-objective evaluating method for hybrid assembly line design. *Control, Automation, Robotics and Vision,*, Dec., 1–5. 5-8 december, ICARCV '06. 9th International Conference on.

[Roudenko, 2004] Roudenko, O. 2004 (Mar.). *Application des Algorithmes Evolutionnaires aux Problèmes dOptimisation Multi-Objectif avec Contraintes.* Ph.D. thesis, Ecole Polytechnique. Thèse soutenue le 5 mars.

[Runarsson & Xin, 200] Runarsson, T.P., & Xin, Y. 200. Stochastic ranking for constrained evolutionary optimization. *IEEE Trans. on Evol. Comp.*, **4**(3), 284 – 294.

[Sacks *et al.*, 1989] Sacks, J., Welch, W.J., Mitchell, T.J., & Wynn, H.P. 1989. Design and analysis of computer experiments. *Statistical Science*, **4**, 409–435.

[Sandgren, 1990] Sandgren, E. 1990. Nonlinear integer and discrete programming in mechanical design optimization. *Journal of Mechanical Design*, **112**(2), 223–229.

[Spall, 2003] Spall, J. C. 2003. *Introduction to stochastic search and optimization: Estimation, Simulation, and control.* Discrete mathematics and optimization. John Wiley & Sons, INC.

[Srinvas & Deb, 1994] Srinvas, N., & Deb, K. 1994. Multi-objective function optimization using non-dominated sorting genetic algorithms. *IEEE Trans. on Evol. Comp.*, **2**(3), 221–248.

[Tappeta & Renaud, 2001] Tappeta, R. V., & Renaud, J. E. 2001. Interactive multiobjective optimization design strategy for decision based design. *Journal of Mechanical Design*, **123**(2), 205–215.

[Torn *et al.*, 1999] Torn, A., Ali, M.M., & Viitanen. 1999. Stochastic global optimization: problem classes and solution techniques. *Global Optimization*, 437–447.

[Tran *et al.*, 2007a] Tran, T. V., Brisset, S., & Brochet, P. 2007a (Jun.). A Benchmark for Multi-Objective, Multi-Level and Combinatorial Optimizations of a Safety Isolating Transformer. *In: 16th International Conference on Computation of Electromagnetic Fields, Aachen, Germany.* June 24-28.

[Tran *et al.*, 2007b] Tran, T. V., Brisset, S., & Brochet, P. 2007b. Combinatorial and Multilevel Optimization of a Safety Isolating Transformer. *International journal of Applied Electromagnetics and Mechanics - IJAEM*, **26**(3, 4), 201–208.

[Tran *et al.*, 2007c] Tran, T.-V., Moussouni, F., Brisset, S., & Brochet, P. 2007c (Jun.). Combinatorial Optimization of a Safety Isolating Transformer using Branch-and-Bound method and Genetic Algorithm. *In: COMPUMAG 2007.*

[Tran *et al.*, 2009] Tran, T. V., Brisset, S., & Brochet, P. 2009. A new efficient method for global discrete multi-level optimization combining Branch-and-Bound and Space-Mapping. *IEEE Transactions on Magnetics*, **45**(3).

[Trelea, 2003a] Trelea, I. C. 2003a (Oct.). Lessaim de particules vu comme un système dynamique : convergence et choix des paramètres. *In: Séminaire : L'optimisation par essaim de particules (OEP).*

[Trelea, 2003b] Trelea, I. C. 2003b. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Proceeding Letters*, **85**(6), 317–325.

[Trout, 2001] Trout, S. R. 2001 (Oct.). Material selection of permanent magnets considering thermal properties correctly. *Pages 365–370 of: proceeding of Electrical Insulation Conference and Electrical Manufacturing & Coil Winding Conference.* 16-18.

[Tzevelekos *et al.*, 2003] Tzevelekos, N., Kokkolaras, M., Papalambros, P.Y., Hulshof, M.F., Etman, P., & Rooda, J. E. K. 2003 (May). An empirical local convergence study of alternative coordination schemes in analytical target cascading. *In: 5th World Congress on Structural and Multidisciplinary Optimization.* Venice, May 19-23.

[Van Hoeve, 2006] Van Hoeve, W.-J. 2006. Exploiting Semidefinite Relaxations in Constraint Programming. *Computers and Operations Research*, **33**(10), 2787–2804.

[Venkataraman, 2001] Venkataraman, P. 2001. *Applied optimization with Matlab Programming.* John Wiley & Sons, INC.

[Visser *et al.*, 2000] Visser, J.A., de Kock, D.J., & Conradie, F.D. 2000. Minimisation of heat sink mass using mathematical optimisation. *Pages 252–259 of: Proceeding of 16th Annual IEEE Semiconductor Thermal Measurement and Management Symposium.*

[Vivier, 2002a] Vivier, S. 2002a (Jul.). *Stratégie d'optimisation par plans d'expériences et Application aux dispositifs électrotechniques modélisés par éléments finis.* Ph.D. thesis, Université des Sciences et Techniques de Lille, France. In French.

[Vivier, 2002b] Vivier, S. 2002b (July). *Stratégie d'optimisation par plans d'expériences et Application aux dispositifs électrotechniques modélisés par éléments finis.* Ph.D. thesis, Université des Sciences et Techniques de Lille, France.

[Wagner & Papalambros, 1993] Wagner, T. C., & Papalambros, P. Y. 1993. A general Framework for Decomposition Analysis in Optimal Design. *Advances in Design Automation, Albuquerque, NM, ASME*, **65**(2), pp. 315–325. New York, NY, September 19-22.

[Wismer, 1971] Wismer, D.A. 1971. *Optimzation methods for large scale systems.* New York: McGraw-Hill Book Company.

[Zeljkovic & Maksimovic, 2006] Zeljkovic, V., & Maksimovic, S. 2006. Multilevel optimization approach applied to aircraft nose landing gear. *Scientific-Technical Review*, **LVI**(2).

[Zhang *et al.*, 2005] Zhang, L-P., Yu, H-J., & Hu, S-X. 2005. Optimal choice of parameters for particle swarm optimization. *Journal of Zhejiang University SCIENCE*, 528–534. Project (No. 20276063) supported by the National Natural Science Foundation of China.

[Zitzler, 1999] Zitzler, E. 1999 (Nov.). *Evolutionary algorithms for multiobjective optimization: methods and applications.* M.Phil. thesis, Swiss federal Institute of technology (ETH), Zurich, Switzerland.

[Zitzler *et al.*, 2001] Zitzler, E., Laumanns, M., & Thiele, L. 2001 (May). *Spea2: Improving the strength Pareto evolutionary algorithm.* Tech. rept. 103. Gloriastrasse 35, CH-8092 Zurich, Switzerland.

[Zitzler *et al.*, 2002] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & Grunert da Fonseca, V. 2002 (Jun.). *Performance Assessment of Multiobjective Optimizers: An Analysis and Review.* Tech. rept. 139. Computer Engineering and Networks Laboratory, ETH Zurich.

[Zitzler *et al.*, 2004] Zitzler, E., Laumanns, M., & Bleuler, S. 2004. A tutorial on evolutionary multiobjective optimization. *Pages 3–38 of: In Metaheuristics for Multiobjective Optimisation.* Gloriastrasse 35, CH-8092 Zurich, Switzerland zitzler,laumanns,bleuler@tik.ee.ethz.ch: Springer-Verlag.