MASTER'S DEGREE THESIS

Interuniversity Master in Statistics and Operations Research

Title: Statistical Methods for Parameter Fine-Tuning of Metaheuristics

Author: Laura Calvet Liñán

Advisor: Carles Serrat i Piè

Co-Advisor: Ángel Alejandro Juan Pérez

Department: Applied Mathematics I, UPC - BarcelonaTECH

Academic year: 2013/2014

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

UNIVERSITAT DE BARCELONA

Facultat de Matemàtiques i Estadística

Universitat Politècnica de Catalunya

Master’s Degree Thesis

Statistical Methods for Parameter Fine-Tuning of Metaheuristics

Laura Calvet Liñán

Advisor: Carles Serrat i Piè

Department of Applied Mathematics I, UPC-BarcelonaTECH

Co-Advisor: Ángel Alejandro Juan Pérez

Computer Science Department at the Open University of Catalonia (UOC)

Acknowledgements

I would like to express my deep gratitude to everyone who has helped me to complete this master's degree thesis.

My special thanks to Dr. Ángel Alejandro Juan Pérez for agreeing to be my co-advisor and for his unflagging encouragement, insights and patience. I wish to thank Dr. Carles Serrat, my advisor, for his support, constant guidance and empathy, also for his willingness to spend time correcting this project and sharing his knowledge. I am highly indebted to them both for teaching me, providing me with valuable material, helping me to define my academic interests and introducing me to the challenging and rewarding world of research.

My grateful appreciations are extended to the members of the *Open University of Catalonia* that I have had the pleasure to meet during my internship, the development of this master's degree thesis and other enriching activities. I am particularly grateful to Daniel Córdoba Higueras for assisting me with some computational issues.

Finally, my sincerest thanks go to my family for giving me the opportunity to study and believing in me.

Abstract

Metaheuristics are an approximate method widely used to solve many hard optimization problems in a multitude of fields. They depend on a variable number of parameters. Despite the fact that they are usually capable of finding good solutions within a reasonable time, the difficulty in selecting appropriate values for their parameters causes a loss of efficiency, as it normally requires much time, skills and experience. This master degree's thesis provides a survey of the main approaches developed in the last decade to tackle the problem of choosing a good set of parameter values, called the Parameter Setting Problem, and compares them from a methodological point of view focusing on the statistical procedures used so far by the scientific community. This analysis is accompanied by a proposal of a general methodology. The results of applying it to fine-tuning the parameters of a hybrid algorithm, which combines Biased Randomization with the Iterated Local Search metaheuristic, for solving the Multi-depot Vehicle Routing Problem are also reported. The computational experiment shows promising results and the need / suitability of further investigations based on a wider range of statistical learning techniques. Along these same lines, different suggestions for future work are described. In addition, this work highlights the importance of statistics in operations research giving a real-world example.

Keywords: Parameter Fine-tuning of Metaheuristics, Design of Experiments, Regression Models, Multi-Objective Optimization

Contents

List of Tables

List of Figures

1. Introduction

1.1 Motivation

Mathematical optimization is an extensive collection of methods and algorithms for solving a huge variety of problems. Nowadays, it plays an essential role in different research areas: management of portfolios (finance), development of balanced diets, location of facilities, DNA sequences assembly, among others. Its use has been boosted in the last decades by the increasing availability of computing power.

The most desirable characteristics of an optimization method are **efficiency** (being able to quickly find a satisfactory solution) and **manageability** (being easy to adapt to similar problems).

Unfortunately, some of the most relevant problems are particularly difficult to solve because they require too much computer memory and/or time. In these cases, **heuristics** (from Greek, means "find" or "discover") offer experience-based techniques that implement strategies for obtaining a sufficiently good solution in a short amount of time. For instance, the most fundamental heuristic is trial and error. Although they do not provide any theoretical guarantee, their use has been widely spread among scientists due to its success.

However, the application of heuristics presents two major drawbacks: they are problem-dependent and their development is very laborious.

As a consequence, in the last 30 years, a new set of procedures has been created, **metaheuristics** ('meta' means "upper level methodology"). Birattari (2005) defines them as general algorithmic templates that can be easily adapted to solve the most different optimization problems. Boussaïd et al. (2013) list their characteristics: they are nature-inspired (based on some principles like physics or biology), include stochastic components and have several parameters that need to be chosen.

The problem of selecting good settings for the parameters of metaheuristics is called the **Parameter Setting Problem** (PSP). Solving it is an arduous task, Montero et al. (2014) summarize the main difficulties: it is time consuming, the best parameter values set depends on the problem at hand and the parameters can be interrelated. Adenso-Díaz et al. (2006) state that there is anecdotal evidence that about 90% of the time dedicated to designing and testing a new heuristic or metaheuristic is consumed by fine-tuning parameters. According to Eiben et al. (1999), during the first decades of metaheuristics research, many scientists based their choices on tuning the parameters "by hand", experimenting with different values and selecting the ones related with the best outputs. Also there were others practitioners that used "parameter setting by analogy", it consists on taking the settings that have been proved successful for "similar" problems.

At the present time there are two main approaches to tackle this problem. The first one, so-called **Parameter Tuning**, is based on selecting proper values before running the algorithm and keep them constant during the execution of it. The other one, **Parameter Control**, allows the parameters to change or adapt. Each approach has its advantages and disadvantages, so there is no consensus about the best one. Due to their different characteristics, it could be possible that the most adequate depended on the specific problem to solve, the chosen metaheuristic, the available time and the skills of the researcher.

Even studying each approach separately, it can be observed that there are many different methodologies applied to tackle the PSP. Some of these procedures opt for employing **statistical tools**. Design of experiments and regression model have much relevance.

In summary, selecting appropriately values for the parameters of metaheuristics is an indispensable and important step, but it is, for the time being, a hard and tedious activity for the scientific community. On the other hand, solving satisfactorily the problem can result in a better performance of the metaheuristics.

1.2 Objectives of this Master's Degree Thesis

This project has four main purposes:

- Identifying and comparing the most relevant approaches to analyze the PSP

Despite being relatively new, this research is very varied. Therefore, an important aim will be to describe which the principal approaches are, characterize and differentiate them, and determine why there is no one that protrudes above all.

- **Reviewing the major contributions of the scientific community**

The most referenced papers will be indicated and examined. They will be classified according to the statistical methods that employ. The most fundamental tools will be briefly explained.

- **Putting forward a methodology**

The principal objective of this master's degree thesis is to develop an efficient, complete and easily adaptable procedure based on statistical techniques, incorporating and combining the best insights of other authors and of our own.

- **Applying this methodology** to a specific problem and analyse the outputs.

In order to assess the quality of the methodology, some experiments will be carried out. The performance measures and some conclusions will be reported.

1.3 Structure of the document

Beyond this introduction, the master's degree thesis is organized in the following manner:

Chapter 2 presents some basic concepts. It is structured in three sections. The first is dedicated to metaheuristics; it contains a definition, a classification and an explanation of the most important ones. The second exposes the Parameter Setting Problem, gathers the former ideas or ways in which this problem was encompassed in the past, outlines the different approaches and compares them. The third section offers a short redaction of Regression Models, Design of Experiments and Multi-Objective Optimization, techniques that will be the principal components of the proposed methodology.

A wide review of the most fruitful research grouped by approach and statistical tools is provided in *Chapter 3.*

In order to illustrate the differences between the approaches listed in *Chapter 2* and the techniques employed by researchers considered in *Chapter 3*, three papers, one per approach, are selected and summarized in *Chapter 4*.

Chapter 5 proposes a detailed methodology to solve the Parameter Setting Problem. It is a general procedure which requires some flexibility to enable its adaptation to a specific metaheuristic.

A computational experiment performed to test the methodology is described in *Chapter 6*. Concretely, it includes a simplified explanation of the Multi-Depot Vehicle Routing Problem. It is the problem that is solved by the selected algorithm, which is based on the Iterated Local Search metaheuristic. The project associated to this algorithm is also introduced. The experiment is commented step by step and the results are discussed.

Finally, *Chapter 7* collects some conclusions regarding the review, the existing methodology, and the experiment. Moreover, some suggestions for further research are made, all focusing on expanding our experiment and introducing other statistical techniques in the solving of the Parameter Setting Problem.

2. Basic concepts

2.1 Metaheuristics

Metaheuristics represent a family of algorithms designed to solve a wide range of hard and complex optimization problems[1] without having to deeply adapt them to each problem. Although they do not guarantee optimal solutions, they may provide **sufficiently good solutions in a reasonable time**.

In the last decades, there have been multiple applications of metaheuristics in a large number of areas. Talbi (2009) highlights some of them:

- Engineering design, topology optimization and structural optimization in electronics and VLSI, aerodynamics, fluid dynamics, telecommunications, automotive, and robotics.
- Machine learning and data mining in bioinformatics and computational biology, and finance.
- System modeling, simulation and identification in chemistry, physics, and biology; control, signal, and image processing.
- Planning in routing problems, robot planning, scheduling and production problems, logistics and transportation, supply chain management, environment, and so on.

2.1.1 Classification

Many classification criteria have been used to differentiate metaheuristics. The most important are:

- Memory usage versus memoryless methods.
- Iterative versus greedy. An iterative metaheuristic is built from one (or more) complete solution and it is transformed at each iteration. Nonetheless, a greedy algorithm starts from an empty solution and as the execution proceeds, it is being built progressively.
- Deterministic versus stochastic. A deterministic metaheuristic makes deterministic decisions, consequently, using the same initial solution will lead to the same final solution. Whereas in stochastic metaheuristics, one could obtain different solutions from the same initial one.
- Single-solution based search versus population-based search. Single-solution based metaheuristics transform a single solution during their execution. While in population-based metaheuristics, a group of solutions is considered. The first group is exploitation oriented, they intensify the search in local regions. In contrast, population-based metaheuristics are exploration oriented, they allow a better diversification.

[1] They are called NP-hard (Non-deterministic Polynomial-time hard) in computational complexity theory. Sequencing, scheduling, assignment, location, grouping, routing and covering problems are some examples of NP-hard problems.

Nowadays, a growing trend is to combine different algorithms to solve a specific problem using the best characteristics of each one in different moments and/or for different subtasks of the problem. These procedures, so-called **hybrid metaheuristics**, include different combinations of a metaheuristic and: other metaheuristics, exact methods from mathematical programming, constraint programming and machine learning and data mining techniques.

2.1.2 Description of the most basic and implemented metaheuristics

A brief explanation of the characteristics of the main metaheuristics is presented. It will be useful to better explain the problem that this project addresses and the different proposals offered by the scientific community.

- Single-solution based metaheuristics

1. **Simulated Annealing**

It is inspired by annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals.

In a similar way, the objective function of a problem is minimized by introducing a **temperature measure** *T*, which is a controllable parameter. The algorithm starts generating a base solution and initializing *T.* In each iteration, a solution is randomly selected in the neighbourhood of the current solution. If the chosen one improves the objective result, it replaces the current solution. Otherwise, it can also be accepted with a probability that depends on *T*. The temperature *T* is decreased during the search process.

The most critical parameters are: the starting temperature, the reduction function of this temperature (known as cooling schedule) and the number of iterations.

2. **Tabu Search**

It tries to mimic the human memory processes.

There are several **memory structures**:
- Short-term. A **tabu list** records the last solutions (or some attributes of them) and forbids these solutions (or solutions containing one of these attributes) from being selected again. The length of this list is an important parameter that will concentrate the search process on small or larger areas.
- Intermediate-term, so-called aspiration criteria. It can be employed to bias moves towards promising areas.
- Long-term. It can be introduced to encourage broader exploration.

3. GRASP method

GRASP (**Greedy Randomized Adaptive Search Procedure**) is a multi-start algorithm.

In each iteration, there are two steps. In the first one, so-called **construction**, a feasible solution is built using a randomized greedy heuristic. At each iteration of the heuristic, a list of candidate elements is formed. The element to be added to the partial solution is randomly chosen among the best candidates. The list of the best candidates is known as restricted candidate list. It can be limited either by the number of elements or by their quality. The second strategy is the most used; they both have parameters to be tuned. In the second step, a **local search procedure** is run with the previous solution as the initial one. After a given number of iterations, the procedure is stopped and the best solution found is returned.

4. Variable Neighbourhood Search

It consists in the exploration of dynamically changing neighbourhoods for a given solution.

Initially, a set of neighbourhood structures is defined. Afterwards, an initial solution is generated. Then a cycle of three steps starts: a) shaking, b) local search and c) move. In the **shaking step**, a solution is randomly selected in the neighbourhood of the current solution. The selected solution is used to apply a **local search procedure** to generate another solution. If this last solution is better than the first one, we update it and the cycle starts again in the initial neighbourhood. In the opposite case, the algorithm **moves to the next neighbourhood** and the cycle starts there.

This metaheuristics can be designed without parameters.

5. Guided Local Search

This metaheuristic dynamically **changes the objective function** optimized by a local search, according to the found local optima.

A set of **features** is defined; each one represents a characteristic of a solution. A **cost** and a **penalty** are associated with each feature. The penalties are initialized to zero and they are updated when a local optimum is reached. The augmented objective function of a solution is calculated as the objective function plus the sum of the penalties related to the features of the solution multiplied by a parameter.

Large values of this parameter diversify the search and small values intensify it.

6. Iterated Local Search

It starts **randomly choosing a solution**. Subsequently a **local search** is applied to get a new solution. Then a loop is initiated until some stopping criterion is satisfied. In the loop, the current solution is **perturbed** and a **new local search** is employed to get another solution. If some previously defined acceptance criterion is satisfied, this new solution will be considered as the current one. The extreme criteria are accepting only improving solutions or anyone. Finally, the algorithm returns the best found solution.

This acceptance criterion enables controlling the trade-off between intensification and diversification.

- Population-based metaheuristics

1. **Evolutionary computation**

This is a wide group that includes algorithms inspired by the **Darwinian principles** of nature's capability to evolve living beings well adapted. They simulate the evolution of individuals by means of **processes of selection, recombination and mutation**; which allow getting better individuals or solutions.

There are several iterations (generations) where a set of candidate solutions (population) is capable of reproducing and receives pressure that causes natural selection (survival of the fittest). The new individuals are created via combination of different individuals (recombination). The offspring can have mutations, which boosts diversity. Then, the fitness of the resulting solutions is calculated and a selection strategy is employed to decide which ones will be maintained into the next generation.

These metaheuristics can have a large number of parameters: crossover and mutation rates, crossover and mutation operators, fitness function, etc.

2. **Swarm intelligence**

This term also covers many procedures. They include a population of **agents interacting locally with one another and with their environment**. These agents are capable to jointly perform complex tasks.

The most popular algorithms are: **ant colony optimization, particle swarm optimization and bee colony optimization-based algorithms**.

For instance, the first one is based on the ants' behaviour. When starting to look for food, these insects search randomly in the area surrounding their nest. While moving, they deposit a chemical pheromone trail in order to mark favourable path to guide other ants. Consequently, the shortest path between the nest and the food source will finally attract more ants as it will present a higher concentration of pheromone. The ants' decision of which path should they follow is influenced by the amount of pheromone, which is formally expressed by an equation expressing a probability that contains some parameters.

2.2 Parameter Setting Problem

2.2.1 Definition

As it has been seen, most metaheuristics include parameters to be tuned. However, the problem of studying the best setting for them had not been intensively studied until this century. In the past, the first authors that tried to shed light on this problem elaborated some **guidelines** for the practitioners. They recommended some specific values for the parameters based on their own experiments; metaheuristics were seen as **robust problems solvers** that exhibit approximately the same performance over a range of problems (see Eiben et al. (1999)). It was usual **reusing** the parameters utilized by other scientists who had faced similar problems. In general, the procedure used to set the parameter values was **not reported**. Gradually, the scientific community became convinced that each problem requires a proper fit of its parameters. Over recent years there has been an increasing concern in this subject and authors, nowadays, tend to follow a methodological procedure and to publish it.

The formal definition of the problem proposed by Ries (2009) is introduced. Let there be an algorithm A with k parameters, each having a domain θ_j $(j = 1, \ldots, k)$, and let θ be a vector of specific values for each parameter in the parameter space Θ. **The Parameter Setting Problem consists in**, for a specific problem instance, **finding a set of optimal parameter values** $\theta^* \in \Theta$ such that:

$$\forall\, \theta \in \Theta:\ \theta^* \succ \theta \tag{1}$$

2.2.2 Approaches

The research dedicated to analyze this problem can be classified into two main **approaches**: Parameter Control Strategies and Parameter Tuning Strategies. In the remainder of the section, the main differences and a relatively new approach will be introduced.

- **Parameter Control Strategies** (PCS)

These strategies aim for an instance-specific fine-tuning of the parameters. This is done by changing the basic algorithm introducing mechanisms to **control and adapt the parameter values**.

As the execution of a metaheuristic proceeds, it receives information that tries to employ to adapt the parameter values and find a better solution regarding the one that would obtain by keeping the parameter values unchanged.

One disadvantage is the complication of modifying the algorithm that can lead to increment the computational time needed.

- **Parameter Tuning Strategies** (PTS)

If it can be assumed that there is a **fixed robust set of parameter values** that provides the best possible results, once found, it can be used to solve a particular instance and even different instances of the same problem. In other words, the practitioners who follow these strategies obtain a set of parameter values that they do not change during the execution of the algorithm when solving one instance and reuse to apply to other instances.

In general, researchers select several representative instances and study which parameter values works better and use them later to solve other instances.

Although it is not necessary to change the algorithm, finding the adequate set can be a difficult and time consuming task requiring statistical methods.

- **Instance-specific Parameter Tuning Strategies** (IPTS)

This approach aims to combine the best of the previous strategies: the instance-specific tailoring of the metaheuristic parameters (as in the PCS) and the simplicity of executing the algorithm with a fixed set of parameter values (as in the PTS).

The methodology consists in developing an algorithm (independent of the main one) able to **extract relevant information from the characteristics of an instance** and use it to **return a recommended set of parameter values** to solve the instance. The algorithm needs a training phase.

2.2.3 Approach selection

It is complicated to choose just one approach, as the three have different and interesting features.

- The PTS are a good option for robust algorithms and the unique that does not require developing a new algorithm or modifying the main one, but researchers have to master statistical techniques. They are the easiest and fastest to use once an appropriate set of parameter values is found.
- The PCS are an alternative when there is no knowledge about the robustness of the algorithm or if there are evidences of the lack of it. They require coding skills.
- The IPTS have several advantages: when the metaheuristic is not robust, they produce better solutions than the PTS, and they also incorporate knowledge from a sample of different instances (the ones used during the training phase) while the PCS have information from just one and have to repeat this entire knowledge gathering for each instance, so need much computational time. However, it also presents some important disadvantages: there are not many references yet and it can be difficult to collect information about the characteristics of the instances (they have to be easily measureable and summarize the differences between instances that explain the different optimal sets of parameter values for each one). Moreover, it requires statistical learning skills.

2.3 Statistical and optimization techniques

In this section, some popular statistical and optimization techniques will be briefly introduced. These concepts are frequently employed by researchers to select parameter values for metaheuristics. Besides, they will be the cornerstone of the proposed methodology.

2.3.1 **Regression Models**

In general, researchers need to analyze data in depth to better understand the problem they are facing or to propose a procedure to tackle it. If they work with correlated variables, it is usually interesting to model and explore their relationship.

- Introduction

Regression analysis is a statistical procedure that data analysts employ when there is a single variable y that depends on k independent variables $x_1, \dots, x_k$. Sometimes, the analyst will know exactly the manner in which the selected variables are related, that is, $y = \phi(x_1, \dots, x_k)$. However, it is usual that the exact model remains unknown and, in that case, he will have to choose a function to approximate ϕ.

- The fitted model

The most basic function represents the linear regression model:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k \ + \ \varepsilon \qquad (2)$$

The parameters $\beta_0, \dots, \beta_k$ are called regression coefficients. The parameter $\beta_j \ (1 \leq j \leq k)$ represents the expected change in response per unit change in x_j when all the remaining independent variables are held constant. ε is an error term.

The model equation can be rewritten in terms of observations as

$$y_i = \beta_0 + \sum_{j=1}^{k} \beta_j x_{ij} + \varepsilon_i \qquad i = 1, \dots, n \qquad (3)$$

The parameters are estimated by the method of least squares, which minimizes the sum of the squares of the errors. The least squares function is

$$L = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{k} \beta_j x_{ij})^2 \qquad (4)$$

Once the parameters are estimated, the fitted values can be calculated as

$$\hat{y}_i = \hat{\beta}_0 + \sum_{j=1}^{k} \hat{\beta}_j \ x_{ij} \qquad i = 1, \dots, n \qquad (5)$$

The differences between the observations and their corresponding fitted values are called residuals, it can be noted as

$$e_i = y_i - \hat{y}_i \qquad i = 1, \dots, n \qquad (6)$$

- Assumptions and model adequacy checking

The model errors are assumed to be normally and independently distributed random variables with mean 0 and variance σ^2. This implies that

$$y_i \sim N(\mu_i, \sigma^2) \tag{7}$$

where μ_i is the expected value of y_i, and that the observations are mutually independent.

As the errors are unobserved, a residual analysis is usually performed to verify the hypothesis of the model. It can also be useful to suggest model improvements, propose alternatives models and detect atypical observations. There are many graphical tools to assess this diagnosis, the most basic are:

- Residuals versus fitted values
- Normal probability plot of residuals
- Residuals versus explanatory variables (considered or not in the model)

- Goodness of fit

There are several measures to rate the goodness of fit of one model. The prime one is the coefficient of multiple determination. It indicates the amount of reduction in the variability of y obtained by using the regressor variables $x_1, \dots, x_k$ in the model. It is calculated as

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{8}$$

where $\bar{y}$ represents the mean response.

R^2 is not a good tool to compare nested models as adding a variable to the model will always increase R^2, regardless of whether the additional variable is statistically significant. For this reason, statisticians developed an adjusted R^2 statistic defined as

$$R_{adj}^2 = 1 - \left(\frac{n-1}{n-k-1}\right)(1 - R^2) \tag{9}$$

Another inconvenient of R^2 is the possibility for models that have large values of R^2 to yield poor predictions (problem of overfitting). A properly measure to detect this problem is an approximate R^2 for prediction, given by

$$R_{pred}^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_{(i)})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{10}$$

where $\hat{y}_{(i)}$ represents the prediction of the observation i calculated from a model with $n - 1$ observations, all but i.

- Hypothesis testing in multiple regression

- Test for significance of regression

This is a technique to determine if a linear relationship exists between y and $x_1, \dots, x_k$. The hypotheses are:

$$\begin{aligned} &H_0: \beta_1 = \cdots = \beta_k = 0 \\ &H_1: \beta_j \neq 0 \text{ for at least one } j \end{aligned} \tag{11}$$

The test procedure for H_0 is to compute

$$F_0 = \frac{\sum_{i=1}^{n}(y_i - \bar{y})^2/k}{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2/(n-k-1)} \tag{12}$$

and to reject the null hypothesis (H_0) if F_0 exceeds $F_{\alpha,k,n-k-1}$.[2]

- Tests on individual regression coefficients

The hypotheses for any individual regression coefficient are:

$$\begin{aligned} H_0 &: \beta_j = 0 \\ H_1 &: \beta_j \neq 0 \end{aligned} \tag{13}$$

The test statistic is

$$t_0 = \frac{\hat{\beta}_j}{se(\hat{\beta}_j)} \tag{14}$$

where se represents the standard error. The null hypothesis is rejected if $|t_0| > t_{\frac{\alpha}{2},n-k-1}$.[3]

- Other models

The measures of goodness of fit and the residual analysis can suggest the suitability of contemplating more complex models. For instance, models with quadratic terms

$$y = \beta_0 + \sum_{j=1}^{k}\beta_j x_j + \sum_{j=1}^{k}\beta_{jj} x_j^2 + \sum_{j=1}^{k-1}\sum_{l=j+1}^{k}\beta_{jl} x_j x_l + \varepsilon \tag{15}$$

can be adequate.

2.3.2 **Design of Experiments**

- Introduction

Experimenting is an essential part of the scientific method. It enables scientists and engineers to discover or confirm something about a process or system. Montgomery (2007) defines an experiment as a test or series of tests in which purposeful changes are made to the input variables of a process or a system so that researchers may observe and identify the reasons for changes that may be observed in the output response.

A suitable experimental design is indispensable to obtain conclusions in an efficient and reliable way. Since experiments involve data that are subject to experimental errors, statistical methods are the only objective approach to analysis.

[2] It is the $100(1-\alpha)$ percentile of the F distribution with $k, n-k-1$ degrees of freedom.

[3] It is the $100(1-\frac{\alpha}{2})$ percentile of the t distribution with $n-k-1$ degrees of freedom

The features of the faced problem and the available resources determine the best design to apply. The most important characteristic to account for is the objective. The main kinds of objectives are:

1. Determining which variables are most influential on the response variables.
2. Determining where to set some controllable process variables so that the response variable is almost always near the desired nominal value.
3. Determining where to set the process variables so that variability in the response variable is small.
4. Determining where to set the process variables so that the effects of some uncontrollable variables are minimized.

There are three basic principles of experimental design:

a. Randomization. The allocation of the experimental material and the order in which the individual runs of the experiment are to be performed need to be randomly determined. This is done because statistical methods require observations to be independently distributed random variables and randomization usually makes this assumption valid.
b. Replication or independent repeat of each factor combination. Replication allows obtaining an estimate of the experimental error. A second advantage is that if the sample mean is used to estimate the true mean response for one of the factor levels in the experiment, having independent repetitions permits to precise the estimate of this parameter.
c. Blocking. It is a technique employed to reduce the variability transmitted from nuisance factors, in which the experimenter is not interested.

- Basic approaches

There is one strategy of experimentation widely used called the best-guess approach. It consists on selecting a combination of factors based on the experimenter criteria, test them and use these results to decide the next step to take.

Another important strategy is known as the one-factor-at-a-time approach. This procedure recommends selecting a baseline set of levels, then successively varying each factor over its range with the other factors held constant at the baseline level. The principal disadvantage is that it fails to consider any possible interaction between the factors.

In the following subsections, other approaches, more elaborated and efficient, are commented.

- **The Factorial Design**

In a Factorial Design, all possible combinations of the levels of the factors are investigated. For instance, if there are a levels of factor A and b levels of factor B, each replica includes $a \cdot b$ treatment combinations.

A special case of the general Factorial Design commonly applied in research work is the 2^k Factorial Design. It analyzes an experiment with k factors, each at only two levels. This design provides the smallest number of runs with which k factors can be studied in a Factorial Design. This characteristic makes it extremely useful in the early stages of experimental work, when many factors are likely to be investigated. When using this design, the experimenter assumes that the response is approximately linear over the range of the factor levels chosen.

- The 2^2 Design

A representation of this design is shown in Figure 2.1. By convention, the effect of a factor is denoted by a capital Latin letter, so "A" refers to the effect of factor A, "B" refers to the effect of factor B and "AB" refers to the effect of the interaction between A and B. The low and high levels of A and B are denoted by "-" and "+", respectively. The four treatment combinations are represented by lowercase letters. The low (high) level of any factor in a treatment combination is denoted by the absence (presence) of the corresponding letter. (1) is used to denote both factors at the low level.

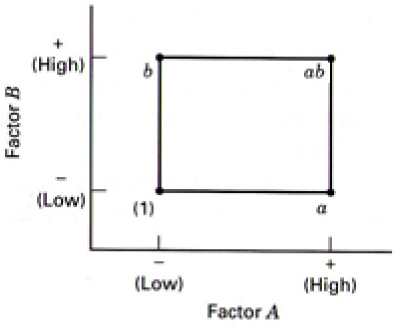


Figure 2.1. Representation of a 2^2 Design.

- Calculating effects

The effect of a factor, also called main effect, is defined as the change in response produced by a change in the level of the factor. The interaction AB is the average difference between the effect of A at the high level of B and the effect of A at the low level of B.

The easiest way to estimate effects is using a table of treatments and effects (Table 2.1). It provides the scheme to calculate contrasts. The steps to construct the table are:

1. List the treatments in the first column.
2. Fill the second one with positive signs.
3. Write "+" in the columns of main effects if the level of the correspondent factor in that treatment is high, otherwise write "-".
4. Complete the columns of interactions by multiplying the columns of the individual factors.

Treatment\Effects	I	A	B	AB
(1)	+	-	-	+
a	+	+	-	-
b	+	-	+	-
ab	+	+	+	+

Table 2.1. Treatments and effects for a 2^2 Design.

The contrasts are obtained from the columns:

$$\begin{aligned}\text{Contrast}_{\text{A}} &= -(1) + a - b + ab\\ \text{Contrast}_{\text{B}} &= -(1) - a + b + ab\\ \text{Contrast}_{\text{AB}} &= (1) - a - b + ab \qquad (16)\end{aligned}$$

As there are $4n$ observations in n replicates, the contrasts are the mean of $4n/2$ observations less the mean of the other $4n/2$. In order to calculate the effects, each contrast expression is divided by $2n$. For instance:

$$AB = \frac{(1) - a - b + ab}{2n} \qquad (17)$$

- Regression model

The regression model associated to this design is:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \varepsilon \qquad (18)$$

where $x_1$and x_2 are coded variables that represent the different levels of each factor. $x_1$is -1 (1) if the level of the factor A is low (high), and, similarly, x_2 represents the factor B. The relationship between the natural and the coded variables is:

$$x_1 = \frac{Factor\ \ A - (Factor\ \ A_{high} - Factor\ \ A_{low}\,)}{(Factor\ \ A_{high} - Factor\ \ A_{low}\,)/2} \qquad (19)$$

- The 2^3 Design

This design of 8 treatment combinations can be displayed as a cube (Figure 2.2).

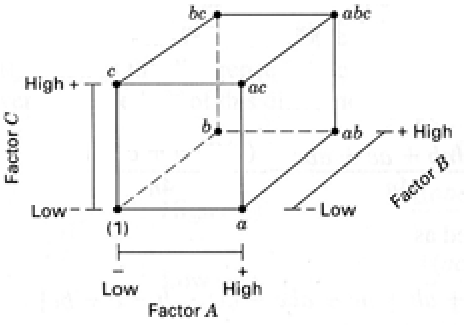


Figure 2.2. Representation of a 2^3 Design.

The table of treatments and effects is shown in Table 2.2.

Treatment\Effects	I	A	B	AB	C	AC	BC	ABC
(1)	+	-	-	+	-	+	+	-
a	+	+	-	-	-	-	+	+
b	+	-	+	-	-	+	-	+
ab	+	+	+	+	-	-	-	-
c	+	-	-	+	+	-	-	+
ac	+	+	-	-	+	+	-	-
bc	+	-	+	-	+	-	+	-
abc	+	+	+	+	+	+	+	+

Table 2.2. Treatments and effects for a 2^3 Design.

- The 2^k Design

The previous design can be automatically generalized for more factors. For an effect AB...K, the contrast can be obtained by developing the right-hand side of:

$$\text{Contrast}_{\text{AB}\dots\text{K}} = (a \pm 1)(b \pm 1)\dots(k \pm 1) \tag{20}$$

The sign in each set of parentheses is negative if the factor is included in the effect and positive if the factor is not included. When finished expanding the equation, “1” is replaced by (1). For instance, in a 2^5 design, the contrast for ABCD would be

$$\text{Contrast}_{\text{ABCD}} = (a-1)(b-1)(c-1)(d-1)(e+1) = abcde + cde + bde + ade + bce + ace + abe + e + abcd + cd + bd + ad + bc + ac + ab + (1) - a - b - c - abc - d - abd - acd - bcd - ae - be - ce - abce - de - abde - acde - bcde \tag{21}$$

- A single replicate of the 2^k Design

The 2^k Design consists of $n2^k$ observations with $n2^{k-1}$ degrees of freedom. k degrees of freedom are associated to main effects, $\binom{k}{2}$ to two-factor interactions, $\binom{k}{3}$ to three-factor interactions and so on.

For even a moderate number of factors, the total number of treatments is large. Sometimes, running several replicates can be infeasible. However, if the experimenter chooses an unreplicated design, there is no internal estimate of error.

- Linearity

As it has been commented, the 2^k Design assumes linearity in the factor effects. Nevertheless, introducing interaction terms to a first-order model, the resulting model

$$y = \beta_0 + \sum_{j=1}^{k} \beta_j x_j + \sum_{j=1}^{k-1} \sum_{l=j+1}^{k} \beta_{jl} x_j x_l + \varepsilon \tag{22}$$

is capable of representing some curvature in the response function. In some cases, a more complex model as

$$y = \beta_0 + \sum_{j=1}^{k} \beta_j x_j + \sum_{j=1}^{k} \beta_{jj} x_j^2 + \sum_{j=1}^{k-1} \sum_{l=j+1}^{k} \beta_{jl} x_j x_l + \varepsilon \tag{23}$$

can be needed. To estimate these new parameters, the experimenter can add center points to the 2^k Design. This procedure incorporates n_c replicates run at the points $x_i = 0$ $(i = 1, \ldots, k)$.

- **Two-Level Fractional Factorial Design**

As the number of factors in a 2^k Factorial Design increases, the number of runs also does. On the other hand, the experimenter usually has a limited amount of resources and the runs are expensive in economic and/or computational terms. For example, an unreplicated 2^5 Design requires 32 runs. But only 5 of the 31 degrees of freedom of this design are related to main effects; 10 correspond to two-factor interactions. The remaining 16 degrees of freedom are associated with higher interactions.

The basic idea of the Two-Level Fractional Factorial Design is that if some interactions can be supposed to have negligible effects, information about the others can be obtained by running only a fraction of the complete factorial experiment.

The success of this design stems from three facts:

1. The sparsity of effects principle. It means that the analyzed process is likely to be driven mainly by some of the main effects and low-order interactions.
2. The projection property. Fractional Factorial Designs can be projected into stronger designs in the subset of significant factors.
3. Sequential experimentation. The runs of several Fractional Factorials Designs can be assembled sequentially.

These designs are classified in three groups:

1. Resolution III Designs: designs in which no main effect is aliased with any other main effect. In other words, all main effects can be individually estimated.
2. Resolution IV Designs: designs in which no main effect is aliased with any other main effect or with any two-factor interaction.
3. Resolution V Designs: designs in which no main effect or two-factor interaction is aliased with any other main effect or two-factor interaction.

- The one-half fraction of the 2^k Design or **2^{k-1} Design**

A factor or a set of them is called generator of a particular fraction if it divides a given full 2^k Factorial Design. For example, from a 2^3 Factorial Design, an experimenter could fractionate the number of runs by selecting only the treatment combinations that had a "+" in the ABC column. ABC would be its generator. Designers would call

$$I = ABC \tag{24}$$

the defining relation of the design. This relation is used to determine the aliases. In the case of the main effect A, multiplying it by the defining relation yields the aliases of A

$$AI = AABC = A^2BC \ \rightarrow A = BC \qquad (25)$$

So when the experimenter was estimating A, he would be really estimating $A + BC$.

If, later on, the experimenter wanted to estimate only A, he could run the observations of the other one-half fraction (corresponding to the defining relation $I = -ABC$). He would obtain the estimate of $A - BC$ and combining the two estimates, he could get A.

As for the projection property, Figure 2.3 shows the projection of a 2^{3-1} Design into three 2^2 Designs.

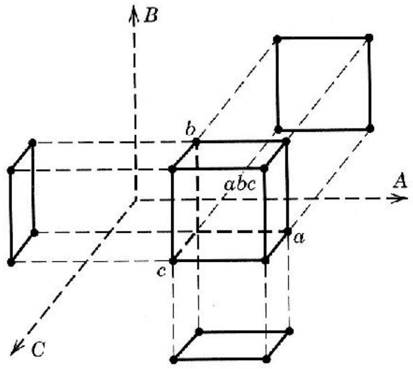


Figure 2.3. Projection of a 2^{3-1} Design into three 2^2 Designs.

- The general 2^{k-p} Fractional Factorial Design

This design requires the selection of p independent generators. Each effect has 2^{p-1} aliases. Researchers should choose the generators in such a way that effects of potential interest were not aliased with each other. The 2^{k-p} Design projects into either a full Factorial or a Fractional Factorial in any subset of $r \leq k - p$ of the original factors.

- **The Central Composite Design**

This is the main class of design for fitting a second-order model. It consists of a 2^k Factorial or Fractional Factorial of resolution V design with n_F runs, $2k$ axial runs and n_c center runs. A representation for $k = 2$ and $k = 3$ is shown in Figure 2.4. This design has two parameters: the distance of the axial runs from the design center (α) and the number of center points.

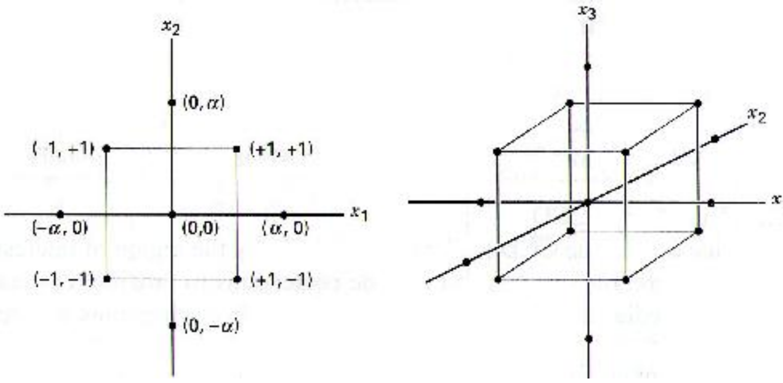


Figure 2.4. Central Composite Designs for two and three factors, respectively.

An important property of the designs for fitting a second-order model is rotatability. It means that the variance of the predicted response is the same at all points x that are at the same distance from the design center (provided x is in the analyzed region). That is to say, the variance of predicted response is constant on spheres. A Central Composite Design is made rotatable by the choice of α. Montgomery (2007) suggests setting $\alpha = {n_F}^{1/4}$. He also recommends executing three to five center runs.

- Sequential experimentation

A practical approach when the experimenter does not know which model will properly fit the relationship between his variables consists of:

1. Applying a 2^k Factorial (or Fractional Factorial) Design
2. Analyzing the goodness of fit.
3. If there are evidences of lack of fit, introduce the additional runs needed to obtain a Central Composite Design.

- Testing for lack of fit

In order to detail the techniques currently utilized by the scientific community to test for lack of fit, it is necessary to introduce some concepts of Analysis of Variance (ANOVA). It will be done by describing the procedure for a design with 2 factors.

  - Basic description of ANOVA

ANOVA is a set of statistical methods to study the differences between means of several groups.

Suppose there is a response variable y that is thought to depend on two factors: A (with a levels) and B (with b). Let y_{ijk} denote the result of a test where i (j) represent a specific level of the factor A (B) and k is the replication number. Consider the following notations:

$$y_{i..} = \sum_{j=1}^{b}\sum_{k=1}^{n} y_{ijk} \qquad \bar{y}_{i..} = \frac{y_{i..}}{bn} \qquad i = 1,\dots,a \tag{26}$$

$$y_{.j.} = \sum_{i=1}^{a}\sum_{k=1}^{n} y_{ijk} \qquad \bar{y}_{.j.} = \frac{y_{.j.}}{an} \qquad j = 1,\dots,b \tag{27}$$

$$y_{ij.} = \sum_{k=1}^{n} y_{ijk} \qquad \bar{y}_{ij.} = \frac{y_{ij.}}{n} \qquad i = 1,\dots,a \qquad j = 1,\dots,b \tag{28}$$

$$y_{...} = \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n} y_{ijk} \qquad \bar{y}_{...} = \frac{y_{...}}{abn} \tag{29}$$

Then, the measure of variability called total corrected sum of squares (SS_T) may be written as:

$$\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}\left(y_{ijk} - \bar{y}_{...}\right)^2 = bn\sum_{i=1}^{a}(\bar{y}_{i..} - \bar{y}_{...})^2 + an\sum_{j=1}^{b}\left(\bar{y}_{.j.} - \bar{y}_{...}\right)^2 + n\sum_{i=1}^{a}\sum_{j=1}^{b}(\bar{y}_{ij.} - \bar{y}_{i..} - \bar{y}_{.j.} - \bar{y}_{...})^2 + \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}(y_{ijk} - \bar{y}_{ij.})^2 \tag{30}$$

In the previous expression, the total sum of squares has been partitioned into a sum of squares (SS) due to factor A, a sum of squares due to factor B, a sum of squares due to the interaction between A and B, and a sum of squares due to error. Expressed symbolically,

$$SS_T = SS_A + SS_B + SS_{AB} + SS_E \tag{31}$$

A sum of squares divided by its degrees of freedom is a mean square (MS).

In this case, three hypothesis tests can be formulated: one for each main effect and one for the interaction. For instance, the hypothesis regarding factor A are:

H_0: there are no differences due to factor A
H_1: there are differences due to factor A

If the null hypotheses are true, then MS_A, MS_B, MS_{AB} and MS_E, all estimate σ^2. Furthermore, the ratios MS_A/MS_E, MS_B/MS_E and MS_{AB}/MS_E are distributed as F with $a-1$, $b-1$ and $(a-1)(b-1)$ numerator degrees of freedom, respectively, and $ab(n-1)$ denominator degrees of freedom. The correspondent critical regions are the upper tail of the Fdistributions.

ANOVA table summarizes the procedure (Table 2.3).

Source of variation	Sum of squares	Degrees of freedom	Mean square	F_0
Factor A	SS_A	$a-1$	$MS_A = SS_A/(a-1)$	MS_A/MS_E
Factor B	SS_B	$b-1$	$MS_B = SS_B/(b-1)$	MS_B/MS_E
AB Interaction	SS_{AB}	$(a-1)(b-1)$	$MS_{AB} = SS_{AB}/[(a-1)(b-1)]$	MS_{AB}/MS_E
Error	SS_E	$ab(n-1)$	$MS_E = SS_E/[ab(n-1)]$	
Total	SS_T	$abn-1$		

Table 2.3. Table ANOVA for a replicated 2^2 Design.

This analysis is usually implemented with programmes as R or Minitab. It requires assumptions of normality, independence and homogeneity of variance.

- ANOVA in testing for lack of fit

Once the basic partitioning of the total sum of squares for the first-order model has been done,

$$SS_T = SS_A + SS_B + SS_E \tag{32}$$

the sum of squares due to error can be partitioned into a sum of squares due to the interaction, a sum of squares due to pure quadratic and a sum of squares due to pure error. In order to do that, the experimenter needs to augment the 2^k Design by n_c center points.

Regarding to the interaction, a lack-of-fit statistic can be built dividing the sum of squares due to interaction by an estimation of error. It can be used to reject the hypothesis about the lack of differences due to interaction. If the effect was (not) negligible, one would observe a small (big) value.

If there is a quadratic curvature in the response function, it can be estimated as the difference between the average response of the points in the factorial portion of the design ($\bar{y}_F$) and the average response of the center points ($\bar{y}_C$). If β_{11} and β_{22} are the coefficients of the pure quadratic terms x_1^2 and x_2^2, then $\bar{y}_F - \bar{y}_C$ is an estimate of $\beta_{11} + \beta_{22}$. The sum of squares associated with the null hypothesis

$$H_0\colon \beta_{11} + \beta_{22} = 0 \tag{33}$$

can be calculated as

$$SS_{(Pure\ quadratic\ )} = \frac{n_F n_C (\bar{y}_F - \bar{y}_C)^2}{n_F + n_C} \tag{34}$$

The statistic is calculated by dividing this sum of squares by an estimate of the error. As before, a big value would indicate lack of fit related to quadratic terms.

To illustrate the procedure, the table for a replicated 2^2 Design is shown in Table 2.4.

Source of variation	Sum of squares	Degrees of freedom	Mean square	F_0
Factor A	SS_A	$a-1$	$MS_A = SS_A/(a-1)$	MS_A/MS_E
Factor B	SS_B	$b-1$	$MS_B = SS_B/(b-1)$	MS_B/MS_E
Error	SS_E	$ab(n-1)$		
(Interaction)	$SS_{(I)}$	1	$MS_{(I)} = SS_{(I)}/1$	$MS_{(I)}/MS_{(PE)}$
(Pure quadratic)	$SS_{(PQ)}$	1	$MS_{(PQ)} = SS_{(PQ)}/1$	$MS_{(PQ)}/MS_{(PE)}$
(Pure error)	$SS_{(PE)}$	$ab(n-1)-2$	$MS_{(PE)} = SS_{(PE)}/[ab(n-1)-2]$	
Total	SS_T	$abn-1$		

Table 2.4. Table ANOVA for a replicated 2^2 Design. Partitioning of the sum of squares due to error into a sum of squares due to interaction, a sum of squares due to pure quadratic and a sum of squares due to pure error.

More information about regression analysis and design of experiments can be found in Montgomery (2007) and Myers and Montgomery (2012).

2.3.3 **Multi-Objective Optimization**

In many optimization problems, there are several important objectives that should be considered. They are known as multiple-objective optimization problems and can be formulated as follows:

$$\begin{aligned} &Minimize_{\ x}\ F(x) = [F_1(x), F_2(x), \dots, F_k(x)]^T \\ &\quad subject\ to\colon \\ &\quad g_j(x) \le 0 \qquad j = 1, \dots, m \\ &\quad h_l(x) = 0 \qquad l = 1, \dots, e \end{aligned} \tag{35}$$

where k is the number of objective functions, m is the number of inequality constraints and e is the number of equality constraints. $x \in \mathbb{R}^n$ is a vector of decision variables, where n represents the number of independent variables x_i. $F(x) \in \mathbb{R}^k$ is a vector of objective functions $F_i(x): \mathbb{R}^n \rightarrow \mathbb{R}^1$.

A common approach to solve these problems detailed in Caramia et al. (2008) is the weighted-sum or scalarization method. It minimizes a positively weighted sum of the objectives. In mathematical terms:

$$\begin{aligned} &\underset{x}{Minimize} \ \textstyle\sum_{i=1}^{k} \gamma_i F_i(x) \\ &\quad subject\ to: \\ &\quad\ \textstyle\sum_{i=1}^{k} \gamma_i = 1 \\ &\quad\ g_j(x) \leq 0 \qquad\qquad j = 1, \dots, m \\ &\quad\ h_l(x) = 0 \qquad\qquad l = 1, \dots, e \\ &\quad\ \gamma_i \ \geq 0 \qquad\qquad\quad i = 1, \dots, n \end{aligned} \tag{36}$$

If the objective functions are measured in different units, transforming data is a good idea. It facilitates the use and interpretation of results. There are several approaches in the literature to transform functions. Marler et al. (2004) highlight a robust one known as normalization, which propose the following transformation:

$$F_i^{trans}\ (x) = \frac{F_i(x) - F_i^0}{F_i^{max}\ - F_i^0},\ F_i^0 > 0 \tag{37}$$

where

$$F_i^0 = \underset{x}{Minimize}\ \{F_i(x) | x \in \mathbb{R}^n\} \tag{38}$$

and

$$F_i^{max}\ = \underset{1 \leq j\ \leq k}{Maximize}\ F_i\left(x_j^*\right) \tag{39}$$

where x_j^* is the point that minimizes the jth objective function.

3. Literature review

Many authors have carried out investigations to solve the Parameter Setting Problem over the last few decades. In this section, the main references are presented. It is interesting to highlight the huge variety of methods analyzed in this reduced list.

3.1 Parameter Tuning Strategies

One of the most referenced proposals was written by Xu et al. (1998), they elaborated a tree growing and pruning method based on **statistical tests**, specifically, the Friedman and Wilcoxon tests, to compare the performance of runs executed with different parameter values.

A different strategy was suggested by Park et al. (1997), they applied a nonlinear response surface optimization method based on a **simplex design** as a procedure without much human intervention.

Ramos et al. (2005) made use of **logistic regression** to model the relationship between the probability of obtaining an optimal result and two parameters to tune. Bartz-Beielstein et al. (2004) also applied regression analysis, they built a method consisted of a **tree based regression**, which outstands for its simplicity and easiness to interpret. Moreover, it does not require any assumption regarding the distribution of the response variable.

Another common procedure is using a **metaheuristic** to choose the parameters of other one. In this case, logically, it is needed to select a priori the values of the first metaheuristic. It was performed, for example, by Tewolde et al. (2009).

Undoubtedly, the most widespread technique is a combination of statistical methods: **design of experiments, analysis of variance and regression models** (see Parsons et al. (2013), Gunawan et al. (2011), Coy et al. (2000) or Kim et al. (1996)). Using these techniques, Ridge et al. (2007) put forward a method to combine two response surface models (based on solution quality and elapsed time) with **desirability functions**. They used a minimum run resolution V Circumscribed Central Composite Design. In this same group, an interesting investigation was performed by Pongcharoen et al. (2006), who employed a one-ninth Fractional Factorial Design embedded within a full **Latin Square**. From a statistical and theoretical point of view, the most remarkable paper is one presented by Czarn et al. (2004); the authors brought up the issues of blocking to analyze the variation due to seeds, test individual parameters and interactions, power analysis, etc.

3.2 Parameter Control Strategies

Eiben et al. (1999) provided a classification of the different methods for setting parameter values valid for any metaheuristic. Their second objective was surveying several forms of control studied by the evolutionary computation community. They compared PTS and PCS, backing the second ones, reasoning that a run of an evolutionary algorithm is an intrinsically dynamic, adaptive process: "It is intuitively obvious that different values of parameters might be optimal at different stages”. However, they also highlighted the learning cost involved in the main control mechanisms and questioned whether it is worthwhile. They concluded that the literature needed to be increased and focused on, among several items, analysis of interactions between parameters.

To deeper into this issue, the book of Lobo et al. (2007) is recommended. It contains several chapters written by renowned authors about parameter setting in evolutionary algorithms. It shows the different approaches during the last 30 years, definitions, classifications, applications, etc.

3.3 Instance-specific Parameter Tuning Strategies

The most noteworthy work came from the thesis of Ries (2009) which relaunched these strategies. She proposed a multi-objective parameter tuning (to maximize solution quality while minimizing computational time) based on **fuzzy logic** to solve the Travelling Salesman Problem in two case studies, one employing a Guided Local Search and the other using a Genetic Algorithm.

Previously, Hutter et al. (2006) had proposed employing machine learning techniques to build models to predict the runtime of search algorithms to solve hard combinatorial problems. Specifically, their approach consisted in learning a function that took as input both features of an instance and parameter configuration of an algorithm, and returned sufficient statistics to estimate the runtime of the algorithm. They used this function to predict which parameter settings would result in the lowest run-time for a given instance. The authors uphold their strategy by indicating that the reactive approach[4] is less general since the implementation is typically tightly coupled to a specific algorithm.

Pavón et al. (2009) developed a methodology to tune the parameters of a genetic algorithm for solving the Root Identification Problem. Each problem instance was dealt with a specific **Bayesian network** and **case-based reasoning** was used as the framework integrator for the different instances from the same problem. As a result, they designed a mechanism that recommended a parameter configuration according to the characteristics of the problem instance at hand and past experience of similar instances.

In the same direction, an important contribution was made by Dobslaw (2010), who introduced the idea of working with an artificial **neural network** and, after a phase of training, use it as an oracle; in other words, requesting for recommended parameters values while submitting the instance and its characteristics as inputs.

[4] Called PCS in this master’s degree thesis.

4. Existing methodology

In order to illustrate the differences between approaches and study in more detail the state of the art, in this Chapter three methodologies are described. They are representative of each approach and are constantly referenced.

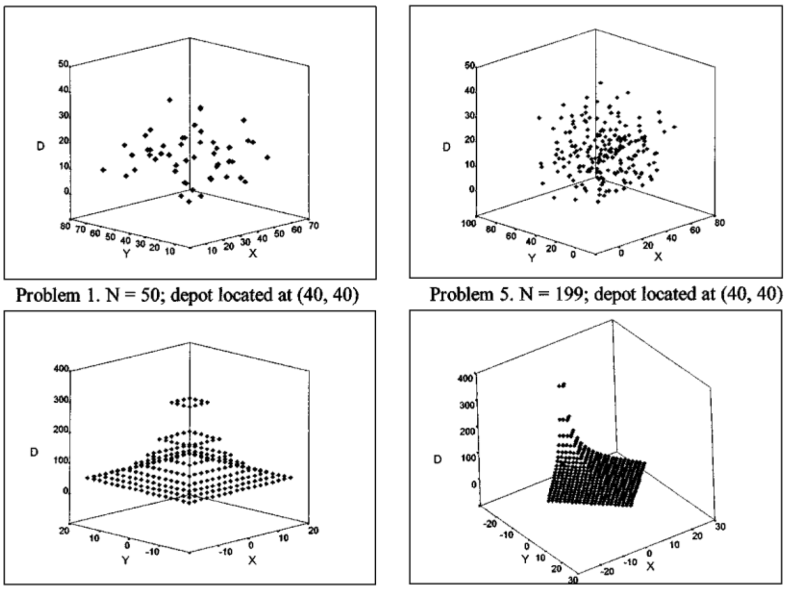
4.1 Parameter Tuning Strategies

Selected paper:
Coy, S. P., Golden, B. L., Runger, G. C. and Wasil, E.A. 2000. Using Experimental Design to Find Effective Parameter Settings for Heuristics. *Journal of Heuristics,* 7, 77-97.

In this paper, the authors explain their proposal based on **statistical design of experiments** and **gradient descent**. They apply their procedure to solve 19 Capacity-constrained and 15 Capacity-constrained and Route-length-constrained Vehicle Routing Problems. Two vehicle routing heuristics are implemented. Altogether, they perform four experiments.

Their procedure has four steps:

- Step 1. Select a **representative subset of problems** to analyze from the entire set of problems.
For instance, Figure 4.1 shows the selection of problems for the first experiment (first local search heuristic and set of 19 Capacity-constrained Vehicle Routing Problem).



Problem 1. N = 50; depot located at (40, 40)

Problem 5. N = 199; depot located at (40, 40)

Problem 28. N = 320; depot located at (0, 0)

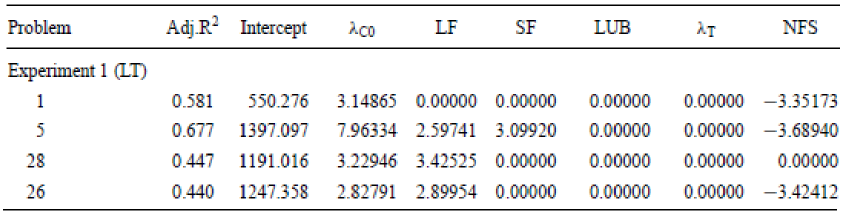
Problem 26. N = 483; depot located at (0, -21)

Figure 4.1. Scatterplots of the subset of Vehicle Routing Problems. Adapted from Coy et al. (2000). Demand of each customer is represented on the vertical axis and its location on the other two axes.

- Step 2. Select the starting level of each parameter, the **range** over which each parameter will be varied, and the amount to change each parameter.
This can rely on the scientific experiences or a pilot study.

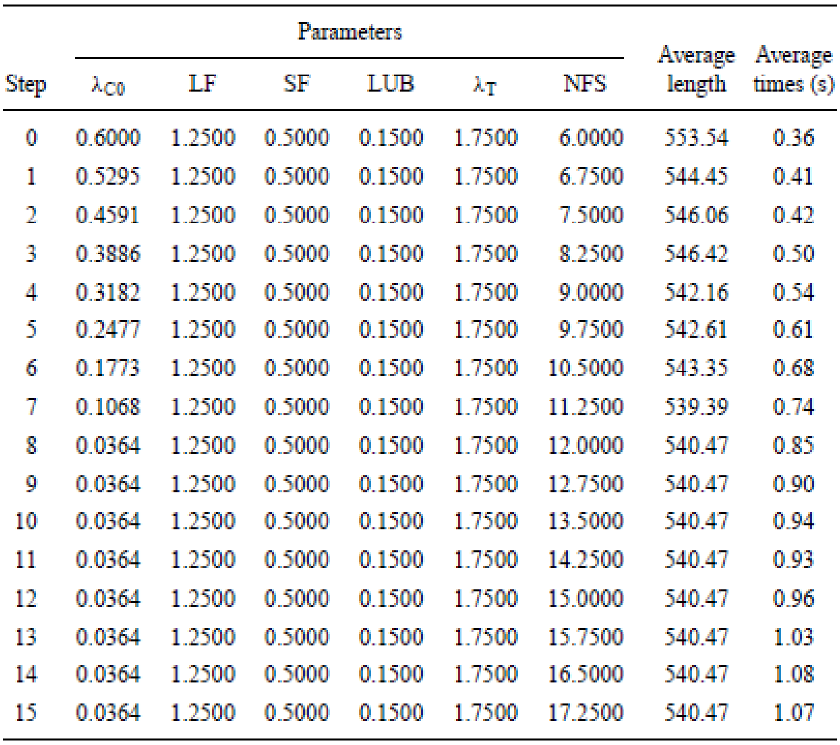
- Step 3. Select **good parameter settings** for each problem in the analysis set using design of experiments.

For the first experiment, they choose a $2^{6\text{-}1}$ **Fractional Factorial Experimental Design** with 5 **replications**. Once the executions according to the design are run, a **linear estimate of the response surface** is used to determine how to set the parameter values (see Table 4.1). They find the **path of steepest descent** on the response surface. The taken steps are indicated in Table 4.2.



Problem	Adj.R^2	Intercept	λ_{C0}	LF	SF	LUB	λ_T	NFS
Experiment 1 (LT)								
1	0.581	550.276	3.14865	0.00000	0.00000	0.00000	0.00000	−3.35173
5	0.677	1397.097	7.96334	2.59741	3.09920	0.00000	0.00000	−3.68940
28	0.447	1191.016	3.22946	3.42525	0.00000	0.00000	0.00000	0.00000
26	0.440	1247.358	2.82791	2.89954	0.00000	0.00000	0.00000	−3.42412

Table 4.1. Linear estimate of the response surface (first experiment). Adapted from Coy et al. (2000). All models are significant at the 0.01 level. A zero indicates that the parameter is not significant at the 0.05 level.



	Parameters							
Step	λ_{C0}	LF	SF	LUB	λ_T	NFS	Average length	Average times (s)
0	0.6000	1.2500	0.5000	0.1500	1.7500	6.0000	553.54	0.36
1	0.5295	1.2500	0.5000	0.1500	1.7500	6.7500	544.45	0.41
2	0.4591	1.2500	0.5000	0.1500	1.7500	7.5000	546.06	0.42
3	0.3886	1.2500	0.5000	0.1500	1.7500	8.2500	546.42	0.50
4	0.3182	1.2500	0.5000	0.1500	1.7500	9.0000	542.16	0.54
5	0.2477	1.2500	0.5000	0.1500	1.7500	9.7500	542.61	0.61
6	0.1773	1.2500	0.5000	0.1500	1.7500	10.5000	543.35	0.68
7	0.1068	1.2500	0.5000	0.1500	1.7500	11.2500	539.39	0.74
8	0.0364	1.2500	0.5000	0.1500	1.7500	12.0000	540.47	0.85
9	0.0364	1.2500	0.5000	0.1500	1.7500	12.7500	540.47	0.90
10	0.0364	1.2500	0.5000	0.1500	1.7500	13.5000	540.47	0.94
11	0.0364	1.2500	0.5000	0.1500	1.7500	14.2500	540.47	0.93
12	0.0364	1.2500	0.5000	0.1500	1.7500	15.0000	540.47	0.96
13	0.0364	1.2500	0.5000	0.1500	1.7500	15.7500	540.47	1.03
14	0.0364	1.2500	0.5000	0.1500	1.7500	16.5000	540.47	1.08
15	0.0364	1.2500	0.5000	0.1500	1.7500	17.2500	540.47	1.07

Table 4.2. Steps of the path of steepest descent on the response surface (first experiment). Adapted from Coy et al. (2000).

Step 4. Combine the settings obtained in Step 3 to obtain high-quality parameter values.
They **average** the results.

The authors conclude that the obtained results confirm the effectiveness in terms of solution quality of their methodology.

4.2 Parameter Control Strategies

Selected paper:
Lessmann, S., Caserta, M. and Arango, I.M. 2011. Tuning metaheuristics: A data mining based approach for particle swarm optimization. *Expert Systems with Applications,* 38, 12826-12838.

Most metaheuristics are **iterative algorithms**: given an initial solution, neighbouring solutions are evaluated and used to improve the present one. With the data obtained as the execution of the metaheuristic proceeds, the authors propose to construct a prediction model capable of estimating suitable parameter values for a subsequent iteration by means of regression. Before using an estimated model, it is needed to analyze if it shows high forecasting accuracy.

Lessman et al. consider several methods for **regression**:
- MLR: **linear regression model**. It is vulnerable to overfitting in high-dimensional settings.
- SWMLR: **stepwise linear regression model**. It overcomes problems of overfitting.
These two models assume an additive and linear relationship between attributes and targets.
- LSSVM: **least-square support vector machine**. It augments the objective of MLR by incorporating a ridge-penalty to prevent overfitting. It can accommodate kernel functions that facilitate nonlinear regression models to be built.
- REGFOR: **random regression forests**. This type of model constructs a large number of regression trees and averages their predictions to form a forecast.
- Naïve approach: it always considers an identical setting for each parameter.

They apply a Particle Swarm Optimization for solving the Water Supply Network Planning Problem. In this metaheuristic, each particle (or bird) has three associated vectors that form the particle's signature: a position vector that represents a potential solution, the flying velocity vector and the best position vector. The velocity depends on the differences between the best position ever reached by the particle or the position of the leader of the swarm and the actual position of the particle. The researchers focus on two parameters that weight these differences in the equation used to calculate the new particles' velocity (c_1 and c_2). They also analyze a parameter that will have influence on the exploration and exploitation capacities of the particles ($Vmax_{factor}$).

The architecture of the proposed system to determine the parameters is illustrated in Figure 4.2. There is an **initialization phase** with I cycles; for each one the information of each particle (signature) and the parameters used is stored. Later on, the data from successful moves is used to build 3 regression models, one per parameter. The following cycle employs them to decide the parameters to use and once all particles have moved, the algorithm appends the data from successful moves to update the fitted models. This is done for all the subsequent cycles.

Table 4.3 shows the results in terms of R^2 for predicting PSO parameters using all available data. It indicates that the results of LSSVM and REGFOR are better (their R^2 values are around 0.5 for c_2 and $Vmax_{factor}$). The authors conclude that these methods appear to be a viable approach.

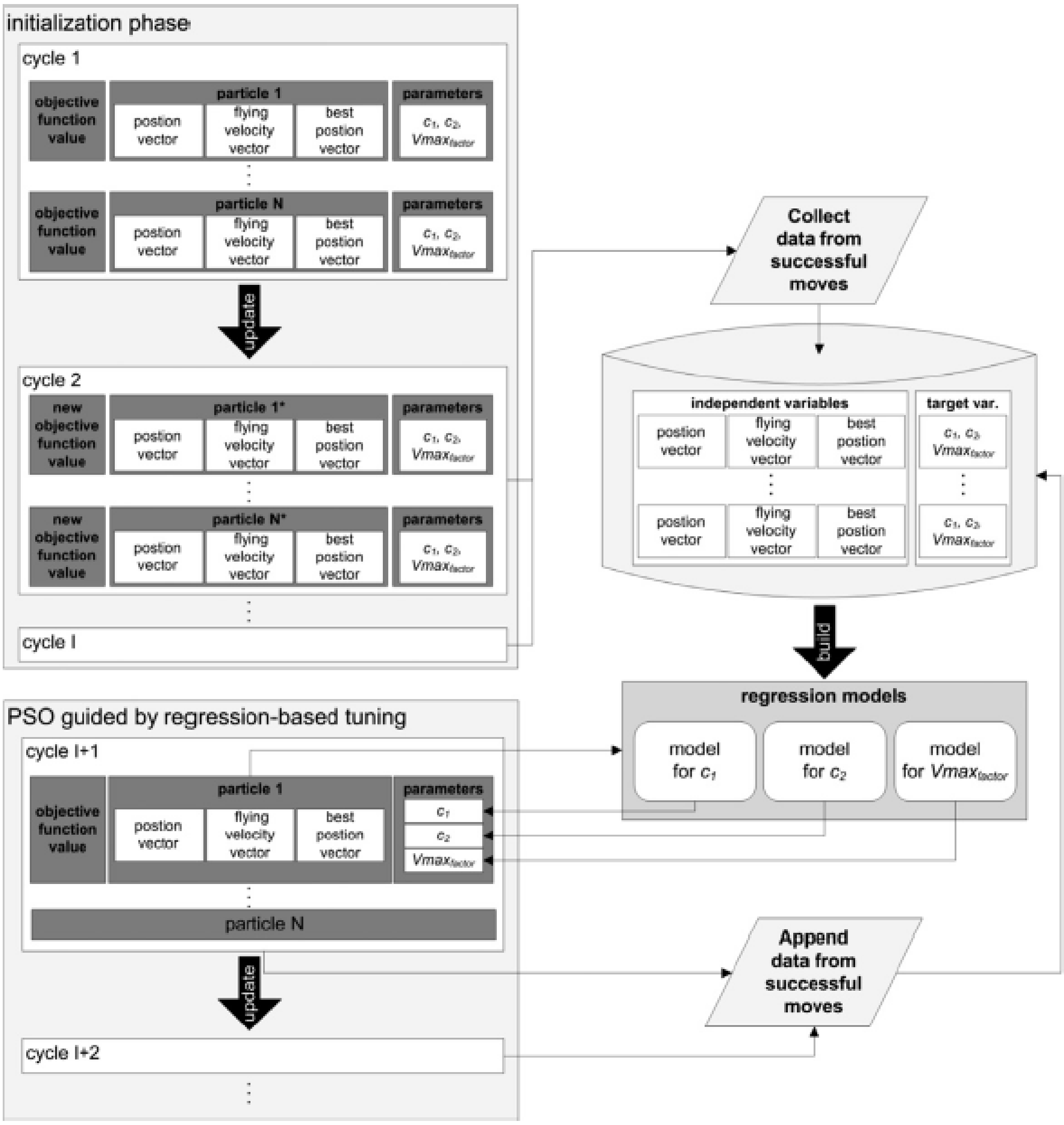
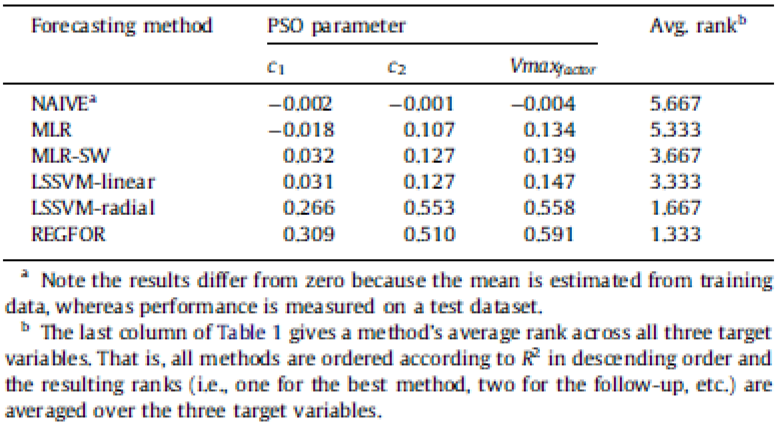


Figure 4.2. Architecture of the system proposed to tackle the Parameter Setting Problem in Lessmann et al. (2001).



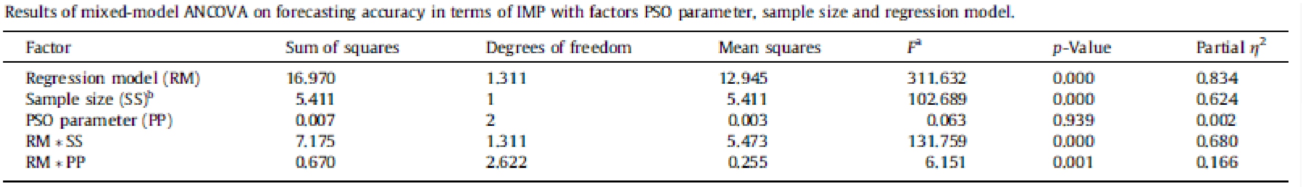
Forecasting method	PSO parameter			Avg. rank[b]
	c_1	c_2	$Vmax_{factor}$	
NAIVE[a]	−0.002	−0.001	−0.004	5.667
MLR	−0.018	0.107	0.134	5.333
MLR-SW	0.032	0.127	0.139	3.667
LSSVM-linear	0.031	0.127	0.147	3.333
LSSVM-radial	0.266	0.553	0.558	1.667
REGFOR	0.309	0.510	0.591	1.333

[a] Note the results differ from zero because the mean is estimated from training data, whereas performance is measured on a test dataset.

[b] The last column of Table 1 gives a method's average rank across all three target variables. That is, all methods are ordered according to R^2 in descending order and the resulting ranks (i.e., one for the best method, two for the follow-up, etc.) are averaged over the three target variables.

Table 4.3. Contrasting of alternative forecasting methods in terms of R^2. Adapted from Lessmann et al. (2011).

A mixed-model **analysis of covariance** is considered to study whether observed differences across regression models, PSO parameters and dataset sizes are statistically significant. The dependent variable is an integrated measure of model performance (IMP) constructed by the authors. Table 4.4 confirms the difference of the performance between the methods of regression model.



Results of mixed-model ANCOVA on forecasting accuracy in terms of IMP with factors PSO parameter, sample size and regression model.

Factor	Sum of squares	Degrees of freedom	Mean squares	F^a	p-Value	Partial η^2
Regression model (RM)	16.970	1.311	12.945	311.632	0.000	0.834
Sample size (SS)^b	5.411	1	5.411	102.689	0.000	0.624
PSO parameter (PP)	0.007	2	0.003	0.063	0.939	0.002
RM * SS	7.175	1.311	5.473	131.759	0.000	0.680
RM * PP	0.670	2.622	0.255	6.151	0.001	0.166

Table 4.4. Results of the mixed-model ANCOVA. Adapted from Lessmann et al. (2011).

Afterwards, a **learning curve analysis** is undertaken. Figure 4.3 contains the output for the regressions of the parameter c_1. The IMP statistic measures the relative performance of a forecasting model compared to the best model. This plot reveals that REGFOR provides highest forecasting accuracy on average and is the least sensitive towards dataset size. The conclusions for the study of the others parameters are similar, although, for the c_2 the model that gives the best results is the LSSVM-radial. The second is REGFOR. They repeat the analysis in terms of the mean absolute percentage error (MAPE) to provide a clearer view on the absolute importance of dataset size (Figure 4.4).

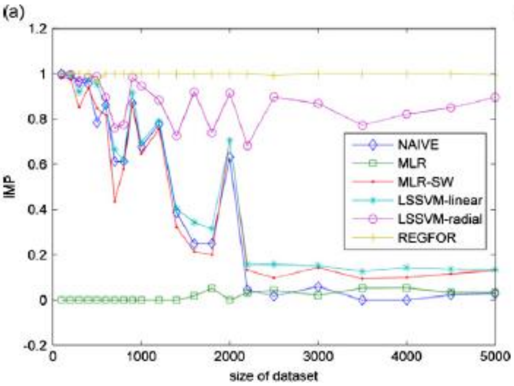


Figure 4.3. Forecasting accuracy in terms of the performance measure IMP for c_1. Adapted from Lessman et al. (2011).

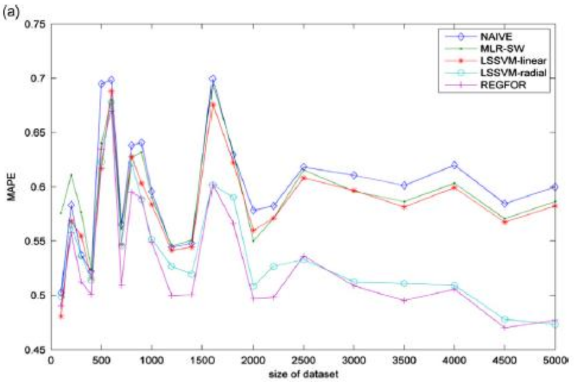


Figure 4.4. Forecasting accuracy in terms of the mean absolute percentage error (MAPE) for c_1. Adapted from Lessman et al. (2011).

4.3 Instance-specific Parameter Tuning Strategies

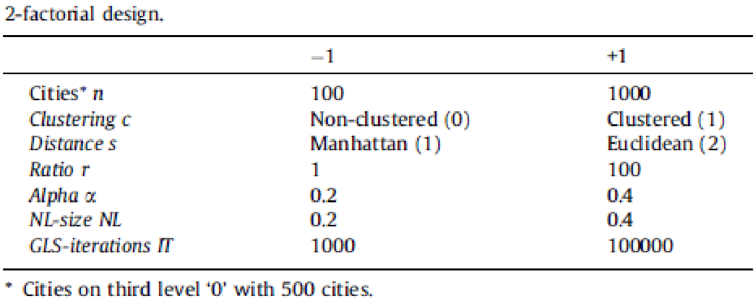
Selected paper:
Ries, J., Beullens, P. and Salt, D. 2012. Instance-specific multi-objective parameter tuning based on fuzzy logic. *European Journal of Operational Research*, 218, 305-315.

The authors propose a two-step methodology to select the parameter values for the metaheuristic Guided Local Search applied to the Symmetric Travelling Salesman Problem.

They choose the following instance characteristics to assess their relation with the performance of the metaheuristic: size of instance (number of cities) (n), distance metric (s), level of clustering (c) and shape of the overall area in which the vertices are distributed (r).

The algorithm-specific parameters considered are:
- α $(0 < \alpha < 1)$ defining the amount of penalisation applied to long edges
- Neighbour list size $NL(0 < NL < 1)$, a percentage reducing the number of considered instance vertices within the Local Search method
- Number of iterations $IT(0 < IT \leq 240000)$

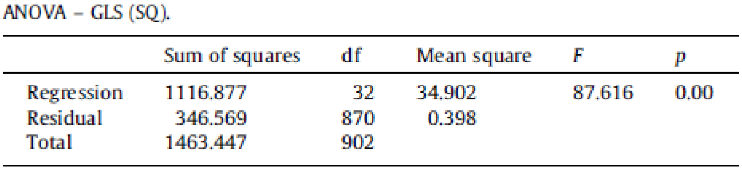
Firstly, a statistical analysis is performed. The researchers set up a full **Factorial Design** with 7 factors (Table 4.5). There are 192 classes of combinations. For each class, 6 instances are created with a random instance generator. This generator takes the instance characteristics as inputs. Then, each instance is solved with respect to the combination of algorithm-specific parameter assigned. Once the solution quality and computational time are noted, a multiple regression analysis is conducted for each one of the dependent variables (solution quality and computational time). **ANOVA** results confirm the predictive power of the fitted models (Table 4.6). They study the statistical significance of the main and the two-way interaction effects.


2-factorial design.

	−1	+1
Cities* *n*	100	1000
Clustering c	Non-clustered (0)	Clustered (1)
Distance s	Manhattan (1)	Euclidean (2)
Ratio r	1	100
Alpha α	0.2	0.4
NL-size NL	0.2	0.4
GLS-iterations IT	1000	100000

* Cities on third level '0' with 500 cities.

Table 4.5. Description of the full Factorial Design employed to tackle the Travelling Salesman Problem. Adapted from Ries et al. (2012).


ANOVA – GLS (SQ).

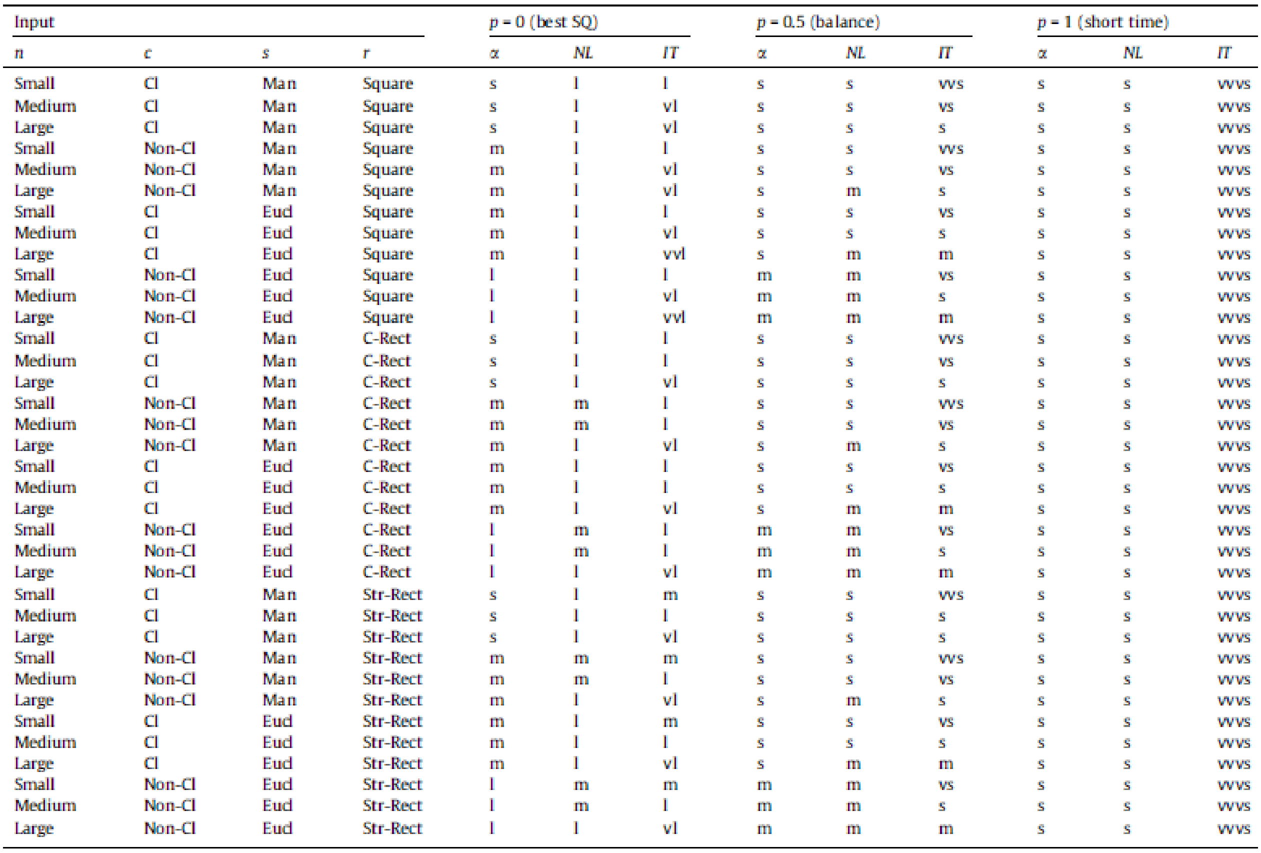
	Sum of squares	df	Mean square	*F*	*p*
Regression	1116.877	32	34.902	87.616	0.00
Residual	346.569	870	0.398		
Total	1463.447	902			

Table 4.6. ANOVA results for the model on the solution quality. Adapted from Ries et al. (2012).

The second step is based on **fuzzy logic**. It is a rule-based approach that allows partial set-membership. The instance based-information and decision maker preferences ($p(0 \leq p \leq 1)$, where $p = 0$ represents the aim of focusing on the best possible quality, and $p = 1$ the desire to look for a reasonable solution in the shortest time possible) are mapped on membership sets which are described by linguistic terms as "very small", "small", etc. Relationships between inputs and outputs are captured in a fuzzy logic system. Rules take the following basic form:

$$IF\ input_1\ AND\ input_2\ AND \ldots AND\ input_i\ THEN\ input_k\ =\ large \qquad (37)$$

Table 4.7 shows the complete rule-base, which represents the insights proportioned by the previous statistical analysis. So for a new instance, one should observe the characteristics of it and look for the adequate row in the first 4 columns and write down the parameter values to use depending on his preferences.



Input				p = 0 (best SQ)			p = 0.5 (balance)			p = 1 (short time)		
n	c	s	r	α	NL	IT	α	NL	IT	α	NL	IT
Small	Cl	Man	Square	s	l	l	s	s	vvs	s	s	vvvs
Medium	Cl	Man	Square	s	l	vl	s	s	vs	s	s	vvvs
Large	Cl	Man	Square	s	l	vl	s	s	s	s	s	vvvs
Small	Non-Cl	Man	Square	m	l	l	s	s	vvs	s	s	vvvs
Medium	Non-Cl	Man	Square	m	l	vl	s	s	vs	s	s	vvvs
Large	Non-Cl	Man	Square	m	l	vl	s	m	s	s	s	vvvs
Small	Cl	Eud	Square	m	l	l	s	s	vs	s	s	vvvs
Medium	Cl	Eud	Square	m	l	vl	s	s	s	s	s	vvvs
Large	Cl	Eud	Square	m	l	vvl	s	m	m	s	s	vvvs
Small	Non-Cl	Eud	Square	l	l	l	m	m	vs	s	s	vvvs
Medium	Non-Cl	Eud	Square	l	l	vl	m	m	s	s	s	vvvs
Large	Non-Cl	Eud	Square	l	l	vvl	m	m	m	s	s	vvvs
Small	Cl	Man	C-Rect	s	l	l	s	s	vvs	s	s	vvvs
Medium	Cl	Man	C-Rect	s	l	l	s	s	vs	s	s	vvvs
Large	Cl	Man	C-Rect	s	l	vl	s	s	s	s	s	vvvs
Small	Non-Cl	Man	C-Rect	m	m	l	s	s	vvs	s	s	vvvs
Medium	Non-Cl	Man	C-Rect	m	m	l	s	s	vs	s	s	vvvs
Large	Non-Cl	Man	C-Rect	m	l	vl	s	m	s	s	s	vvvs
Small	Cl	Eud	C-Rect	m	l	l	s	s	vs	s	s	vvvs
Medium	Cl	Eud	C-Rect	m	l	l	s	s	s	s	s	vvvs
Large	Cl	Eud	C-Rect	m	l	vl	s	m	m	s	s	vvvs
Small	Non-Cl	Eud	C-Rect	l	m	l	m	m	vs	s	s	vvvs
Medium	Non-Cl	Eud	C-Rect	l	m	l	m	m	s	s	s	vvvs
Large	Non-Cl	Eud	C-Rect	l	l	vl	m	m	m	s	s	vvvs
Small	Cl	Man	Str-Rect	s	l	m	s	s	vvs	s	s	vvvs
Medium	Cl	Man	Str-Rect	s	l	l	s	s	s	s	s	vvvs
Large	Cl	Man	Str-Rect	s	l	vl	s	s	s	s	s	vvvs
Small	Non-Cl	Man	Str-Rect	m	m	m	s	s	vvs	s	s	vvvs
Medium	Non-Cl	Man	Str-Rect	m	m	l	s	s	vs	s	s	vvvs
Large	Non-Cl	Man	Str-Rect	m	l	vl	s	m	s	s	s	vvvs
Small	Cl	Eud	Str-Rect	m	l	m	s	s	vs	s	s	vvvs
Medium	Cl	Eud	Str-Rect	m	l	l	s	s	s	s	s	vvvs
Large	Cl	Eud	Str-Rect	m	l	vl	s	m	m	s	s	vvvs
Small	Non-Cl	Eud	Str-Rect	l	m	m	m	m	vs	s	s	vvvs
Medium	Non-Cl	Eud	Str-Rect	l	m	l	m	m	s	s	s	vvvs
Large	Non-Cl	Eud	Str-Rect	l	l	vl	m	m	m	s	s	vvvs

Table 4.7. Rule-base for the fuzzy system. Adapted from Ries et al. (2012).

The researchers concluded that their approach is consistently faster than a traditional non-instance-specific parameter tuning strategy without significantly affecting solution quality.

5. Proposed methodology

5.1 Introduction

The proposed methodology to solve the Parameter Setting Problem is inspired by Monteiro et al. (2013). These researchers applied several concepts of design of experiments as factorial designs, response surface methodology and desirability functions, to choose the parameter values of a hybrid algorithm employed to work out the Capacitated Vehicle Routing Problem. Their proposal follows the parameter tuning approach.

As an extension of the previous work, this master's degree thesis aims to provide **more evidences** about the adequacy of using statistical techniques to solve the Parameter Setting Problem. Furthermore, our work presents a **general methodology** (including validation) and some contributions in the basic procedure. They can be summarized as:

1) **Sequential experimentation**. A Factorial Design will be implemented and, only if the corresponding fitted model shows lack of fit, a second-order Design will be applied. So the numbers of runs will be reduced.
2) An exhaustive **analysis of the results**.
3) Taking account of **computational time**. Although our methodology is focused on objective solutions, the computational time required to find them will also be considered.

The rest of this chapter is structured in two sections, the first one proposes a procedure for parameter fine-tuning of metaheuristics and the second establishes a general methodology which introduces some instructions to validate the selected set of parameter values and analyze the results that it provides.

5.2 Methodology to choose parameter values of a metaheuristic

A 6-step procedure based on statistical methods is described herein. It is assumed that the experimenter has defined a problem and has chosen the metaheuristic that will employ to tackle it. Among the family of potential problem instances, he has to decide whether it is worthwhile concentrating on all or only on a group of them with a set of specific characteristics. This is an important point because if the analyzed instances are homogeneous, it will be easier finding a good set of parameter values for all of them. The term 'good set' will be more precisely described later. In particular, the practitioner needs to select a number of benchmark instances to solve. If he wanted to study a heterogeneous group of instances, it would be recommendable to split it into homogeneous subgroups and implement the procedure on each one.

The first step involves choosing a **subset of the benchmark instances**. Obviously, it has to be representative regarding the main features because the experimenter, based on it, will extract information to identify a good set of parameter values for the subset and this set will be used to solve all the benchmark instances. In this step, it is assumed that similar instances will have a similar (or equal) optimum set of parameter values.

Afterwards, the second step requires selecting the **range** over which each parameter can be set. Some experience or knowledge about the problem and the metaheuristic is essential for that. If needed, one pilot study can be carried out to assess the performance of the problem for different parameter values. The ranges should be large enough to cover the set of values that can give the best objective solution with a high probability. On the other hand, a smaller range would allow the experimenter to describe more accurately, with the same resources, the relationship between the parameter values and the objective solution.

For each instance of the subset, the third, fourth and fifth steps are implemented as follows.

The third step consists of **designing an experiment**. Initially a Factorial Design at two levels of resolution V is studied (see section 2.3.2 for more details). Each metaheuristic parameter is considered a factor and the extreme values of its range define its high and low levels. According to this design, the algorithm is executed once for each combination of factors. The computational time and the objective solution of all runs are stored.

Afterwards, in the fourth step, two **regression models** (introduced in section 2.3.1), one for the computational time and another for the objective solution, are fitted on the factors (called variables in this context). In case of being required, for example if there were non-significant variables, a posteriori power analysis[5] would be performed. Based on these outputs, it could be necessary going back to the third step and modifying the design for building a stronger one[6]. Then, the goodness of fit of each model is calculated. It could be indispensable changing the design by adding some center runs to estimate interactions and some axial runs to estimate quadratic terms (it would be a Central Composite Design). Before continuing, the models assumptions should be checked. If they were violated, further actions as transforming the variables should be taken.

In order to obtain the parameter values, in the fifth step a **bi-objective optimization** (defined in section 2.3.3) is performed with the two normalized estimating regression functions by using the weighted-sum method.

Finally, the sixth step is **averaging** the parameter values of each instance. It is expected that this mean value will provide a good set of parameter values for solving all the benchmark instances.

A summary of the procedure is outlined in Figure 5.1.

[5] The power of a statistical test is the probability that it correctly rejects the null hypothesis when the null hypothesis is false. In the case of multiple regression, one can analyze the following test:

$$H_0: R^2 = 0$$
$$H_1: R^2 > 0$$

where R^2 is the coefficient of determination. In other terms, this test studies whether there are enough evidences on the data to reject the hypothesis that the independent variables do not explain the variability of the response variable. Consequently, a power analysis will give us the probability of rejecting the null hypothesis, which depends on the sample size, the confidence level and the effect size considered.

[6] A stronger design is one with more runs, and accordingly, with more power. For instance, a stronger design would be an unreplicated complete or replicated Factorial Design.

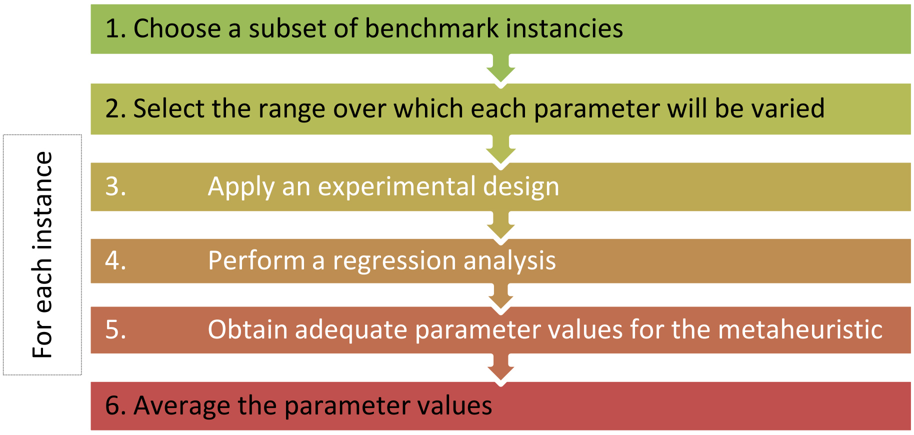

1. Choose a subset of benchmark instancies
2. Select the range over which each parameter will be varied
3. Apply an experimental design
4. Perform a regression analysis
5. Obtain adequate parameter values for the metaheuristic
6. Average the parameter values
For each instance

Figure 5.1. Outline of the procedure for parameter fine-tuning.

5.3 General methodology

An extended proceeding is described below in order to validate the obtained set of parameter values and analyze the results provided by it.

Initially, a set of parameter values $\hat{\theta}$ is chosen as has been exposed in the precedent section.

Later on, each benchmark instance of the subset used to select $\hat{\theta}$ is solved with it and with different sets of parameter values $\theta_i, i = 1, \ldots, \mathcal{N}$, (equally spaced, randomly selected or relatively close to $\hat{\theta}$ according to the Euclidean distance). To assess the performance of $\hat{\theta}$ in a specific instance regarding the other sets, the associated overall objective results[7] are analyzed. Given a decision level parameter p_1 $(0 < p_1 < 1)$, if the proportion of overall objective results below the one provided by $\hat{\theta}$ is lower[8] than it, $\hat{\theta}$ is considered a good set of parameter values for that benchmark instance. Once the experimenter has examined all the benchmark instances of the subset, he can reckon the proportion of them in which $\hat{\theta}$ has been classified as a good set. $\hat{\theta}$ is validated by comparing this proportion with a predefined parameter p_2 $(0 < p_2 < 1)$; if the proportion is upper, then the experimenter has enough evidence of the quality of $\hat{\theta}$ to go on to test it with other instances in the next step.

If $\hat{\theta}$ was not validated, the process should be readjusted and restarted. This readjustment can be done in several ways, some decisions that can be contemplated are: check for the homogeneity of the benchmark instances, increase the number of instances in the chosen subset, adapt the ranges, build a stronger design, etc. The best strategy is problem dependent. As a consequence, the choice of one should be based primarily on the opinion of the experimenter, who will have acquired valuable knowledge from the previous steps.

Once a set of parameter values has been labelled as valid, it is applied for solving the other benchmark instances. To examine the effectiveness of the procedure, if possible, the overall objective solutions (one per instance) should be compared with others reported in the

[7] The term 'overall objective result' refers to the weighted sum of the analyzed variables (objective solution and computational time), once they have been standardized.

[8] For a minimization problem.

literature by performing a t-test for paired samples[9]. The level of confidence $(1-\alpha)$ usually recommended is 0.95. If the means did not differ significantly, it could be classified as a satisfactory outcome as the proposed methodology, automated and generalizable to other problems, would have been proof to be competitive. If the results were unsatisfactory, the procedure should be modified and reinitiated.

It is useful to note that, since the available resources are usually limited, the possible readjustments should be also limited (the limit is denoted as $\mathcal{K}$). Consequently, this process could end without a satisfactory set of parameter values. In this case, the set which provided on average the best overall objective results would be accepted.

A scheme of the proposed methodology is shown in Figure 5.2.

[9] It is a location test used when comparing two sets of measurements to assess whether their population means differ. It assumes that the differences between pairs are normally distributed.

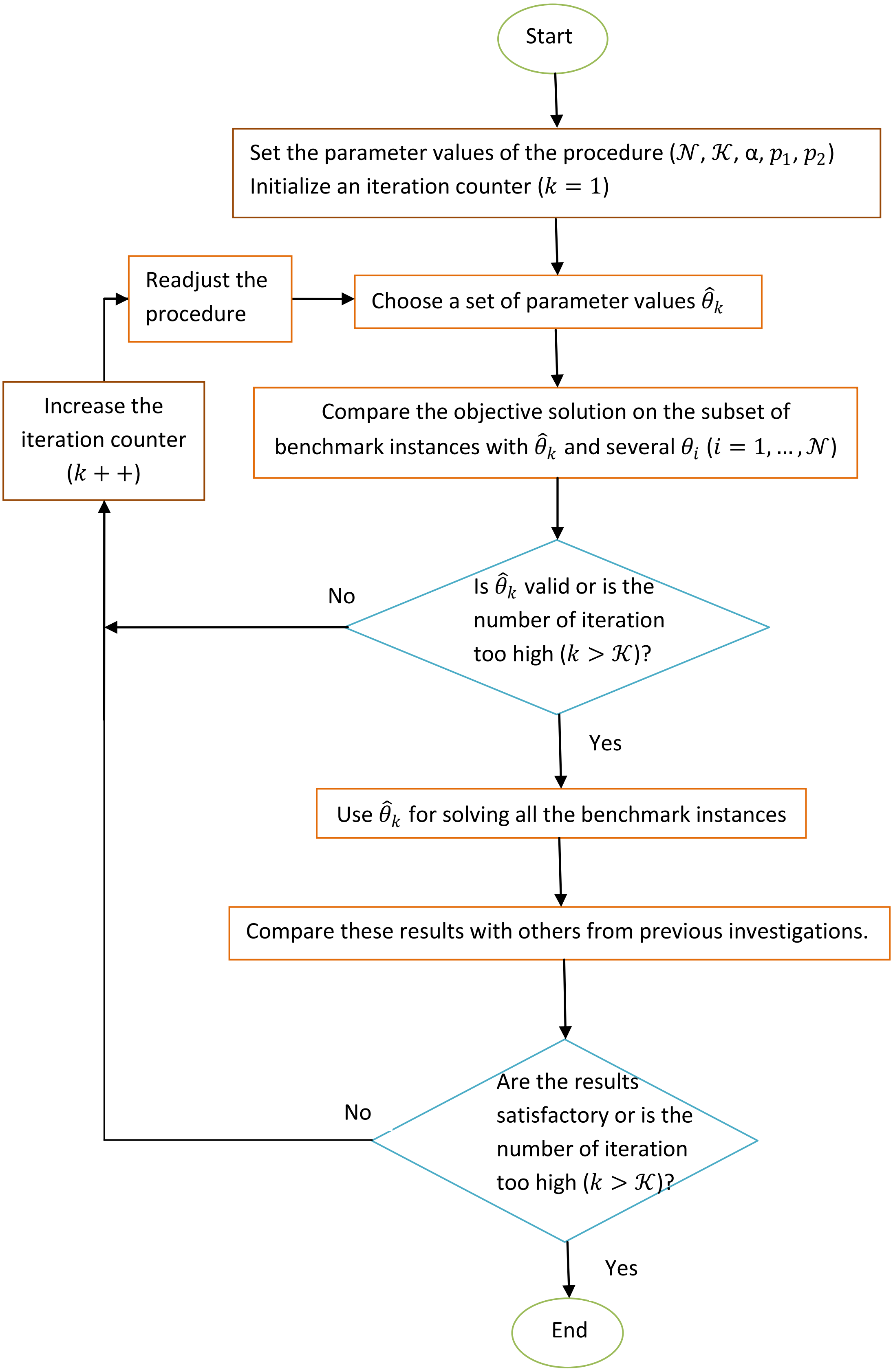

Start
Set the parameter values of the procedure ($\mathcal{N}$, $\mathcal{K}$, α, p_1, p_2)
Initialize an iteration counter ($k = 1$)
Readjust the procedure
Choose a set of parameter values $\hat{\theta}_k$
Increase the iteration counter ($k++$)
Compare the objective solution on the subset of benchmark instances with $\hat{\theta}_k$ and several θ_i ($i = 1, \dots, \mathcal{N}$)
No
Is $\hat{\theta}_k$ valid or is the number of iteration too high ($k > \mathcal{K}$)?
Yes
Use $\hat{\theta}_k$ for solving all the benchmark instances
Compare these results with others from previous investigations.
No
Are the results satisfactory or is the number of iteration too high ($k > \mathcal{K}$)?
Yes
End

Figure 5.2. Flowchart representing the proposed methodology.

6. Experimental results and data validation

In order to test the efficiency of the proposed methodology, a hybrid algorithm for solving **the Multi-Depot Vehicle Routing Problem** (which will be described in the following section) has been chosen. The algorithm was developed for a previous project carried out by the *Distributed, Parallel and Collaborative Systems* research group of the *Open University of Catalonia.* The title of the paper describing it is **"Combining Biased Randomization with Iterated Local Search for Solving the Multi-depot Vehicle Routing Problem"** (Juan et al. (2014)).

In the first section of this chapter, the problem and the algorithm are introduced, the following section describes the experiment, and the last one provides the results.

6.1 Description of the problem and the project

1. The Multi-Depot Vehicle Routing Problem (MDVRP) with a limited number of identical vehicles per depot

This problem is an interesting variant of the well-known Capacitated Vehicle Routing Problem (CVRP), which consists in planning routes to service a number of customers with a homogeneous fleet of vehicles that has a maximum capacity. The main objective of the CVRP is to find a feasible solution that minimizes the total transportation cost.

The MDVRP integrates an **allocation problem**, in which the customers are assigned to one depot, with several **CVRPs**, one per depot. In the tackled case, there is also a maximum number of vehicles associated to each depot. Figure 6.1 shows the common representation of a solution for this problem.

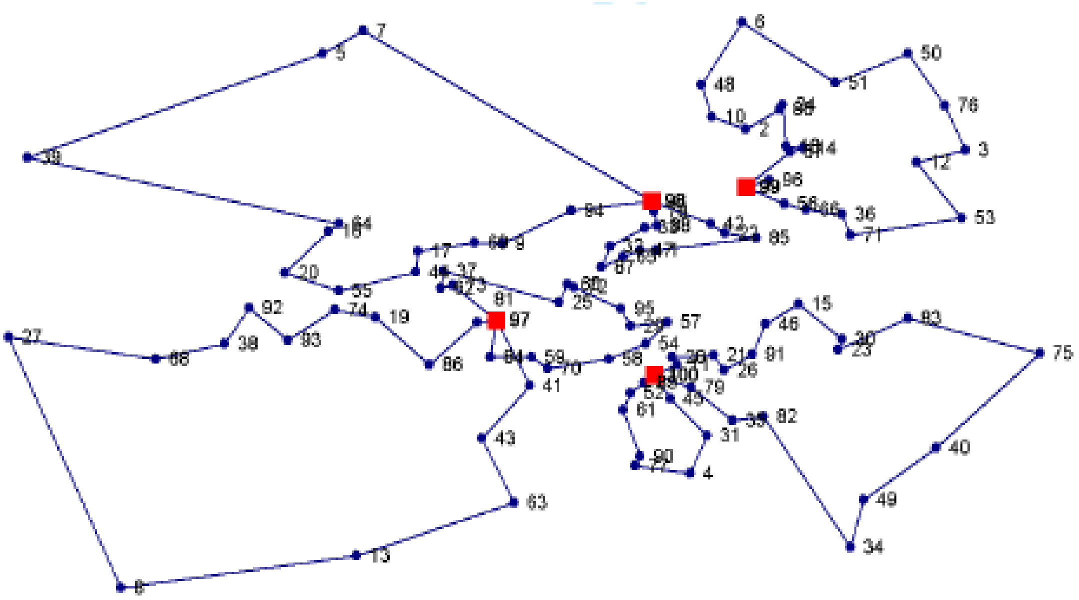


Figure 6.1. Graphical solution of a MDVRP with 4 depots (squares) and 96 customers (circles).

2. The project

- The algorithm

Initially, a **priority list** of potentially eligible customers (nodes) is computed for each depot. The lists are sorted according to a distance-based criterion which is depot-dependent. Then, they are **randomized through a geometric distribution** and are used to allocate customers to depots.

Once a **depots-nodes allocation map** is designed, an initial solution is obtained by individually solving each routing problem using a version of the **Clarke & Wright's Savings (CWS) heuristic**[10]. In short, the CWS technique starts building an initial solution in which each route include just one customer. Afterwards, the heuristic considers the possibility of merging two routes if the total cost is reduced. This operation is repeated in order to find a good (and feasible) solution, so-called base solution. For this project, a **biased-randomized version** was developed; while the original version seeks always the best possible merging, this one lists and sorts all the options, and apply a biased randomization using a geometric distribution to select one. Therefore, multiple solutions can be obtained.

In the next phase, an **Iterated Local Search** procedure is implemented. The base solution is **perturbed** by reallocating a concrete percentage of customers and computing a new solution. The base solution is updated if the new solution is better. If this last one is also better than the best solution found so far, it is improved by means of a fast **local search process**, and then the base and best solutions are updated with the values of the obtained solution. On the other hand, if the solution after the perturbation is worse than the current one, an acceptance criterion is contemplated and, consequently, the current base can be modified. This phase ends after a fixed number of iterations. During the procedure, the top best solutions are stored as promising solutions.

Finally, each one of these top solutions is improved with a **post-optimization process**.

- Parameters of the algorithm

There are three parameters to be tuned in this algorithm:

- bM: the parameter of the geometric distribution employed to assign nodes to depots.
- bR: the parameter of the geometric distribution used to select edges (or mergers) in the CWS heuristic.
- p^*: the percentage of nodes that are reallocated in the ILS phase, when perturbing the base solution.

Because of its definitions, these parameters only take values between 0 and 1.

- Benchmark instances

33 MDVRP benchmark instances were solved. Table 6.1 lists their identification name and some basic features: number of customers, vehicles available at each depot, number of

[10] The description of this heuristic can be read in Clarke et al. (1964).

depots, maximum route length allowed, vehicles maximum capacity and mean distance between nodes.

Instance	# nodes	# vehicles per depot	# depots	Max. route length[11]	Capacity	Mean distance
p01	50	4	4	∞	80	29.45
p02	50	2	4	∞	160	29.45
p03	75	3	5	∞	140	35.92
p04	100	8	2	∞	100	43.48
p05	100	5	2	∞	200	43.48
p06	100	6	3	∞	100	43.48
p07	100	4	4	∞	100	43.48
p08	249	14	2	310	500	116.69
p09	249	12	3	310	500	116.69
p10	249	8	4	310	500	116.69
p11	249	6	5	310	500	116.69
p12	80	5	2	∞	60	78.96
p13	80	5	2	200	60	78.96
p14	80	5	2	180	60	78.96
p15	160	5	4	∞	60	97.21
p16	160	5	4	200	60	97.21
p17	160	5	4	180	60	97.21
p18	240	5	6	∞	60	143.17
p19	240	5	6	200	60	143.17
p20	240	5	6	180	60	143.17
p21	360	5	9	∞	60	176.25
p22	360	5	9	200	60	176.25
p23	360	5	9	180	60	176.25
pr01	48	1	4	500	200	47.50
pr02	96	2	4	480	195	57.55
pr03	144	3	4	460	190	76.97
pr04	192	4	4	440	185	81.35
pr05	240	5	4	420	180	94.94
pr06	288	6	4	400	175	117.17
pr07	72	1	6	500	200	60.30
pr08	144	2	6	475	190	72.35
pr09	216	3	6	450	180	97.58
pr10	288	4	6	425	170	116.35

Table 6.1. List and description of the benchmark instances.

[11] Some instances do not have a restriction about the maximum route length allowed. In order to implement the algorithm and apply some statistical techniques, a sufficiently large value was assigned when needed (5000).

The principal reasons to choose these characteristics are that they are informative, useful to differentiate subgroups, easy to obtain/calculate and usually employed in the literature.

Other features could have been selected. For instance, Coy et al. (2000) analyze, when solving the CVRP, the distribution of customers (it is classified as random or symmetric), the distribution of demand (random or clustered) and the location of the depot. Ries et al. (2009) studies a measure of clustering for the Travelling Salesman Problem.

6.2 Description of the experiment

0.1 **Technical details**

The statistical analysis has been performed with **R** (version 2.15.0), a freely available language and environment for statistical computing and graphics. It is widely used among statisticians and other scientists. It is an implementation of/ influenced by S, a statistical interpreted language. R supports procedural and object-oriented programming. Compared with other popular statistical software, this one presents two major advantages: it is extensible through functions and extensions, and it has a very active community which creates and shares packages (including helping manuals), solves doubts, organizes conferences and courses, etc.

Concretely, the following packages have been used:

- *rsm* (Lenth (2009)): it provides several functions to facilitate classical response-surface methods.
- *cluster*: it includes cluster analysis methods.
- *scatterplot3d:* it helps to visualize data in 3 dimensions.

The **G*Power** program[12] (Faul et al. (2009)) has been employed to do the power analysis. This software was developed by researchers of the *Universität Düsseldorf* (Germany) to compute statistical power analyses for many different *t* tests, *F* tests, χ^2 tests, *z* tests and some exact tests. It is free, user-friendly (it is easy to use and there are many tutorials), based on one of the most referenced books of this topic, Cohen (1988), and has been used in several research (for instance, in Czarn et al. (2004)).

In this project, we have focused our analysis on the *F* test for multiple regression[13].

0.2 **Fitness landscape analysis**

Before starting to implement our methodology, it is worthwhile to study the search space and the objective function of our problem. As it has been commented before, the parameter values range between 0 and 1. Figures 6.2 and 6.3 illustrate the fitness landscape for two instances, using a specific seed, a fixed value for the third parameter[14] and running the algorithm for 11 equally spaced values (ranging from 0 to 1) of the first parameter and 11 of the second one. The selected instances, p03 and p09, are one of the easiest (requires less

[12] It can be downloaded from: http://www.gpower.hhu.de/

[13] This test indicates whether the parameters explain a positive percentage of the variability of a response variable.

[14] As it will be explained later, we will focus our study on the first two parameters.

computational time to be solved) and the most difficult instances, respectively. It can be concluded that these instances have rugged fitness landscapes.

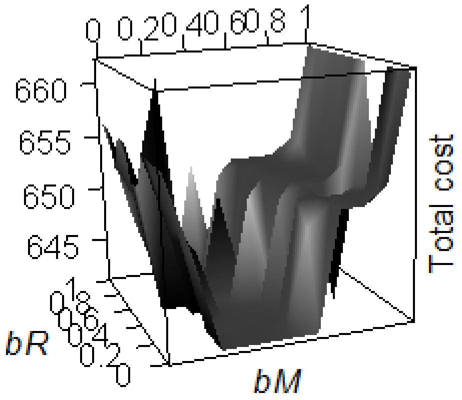


Figure 6.2. Surface plot of the objective solution (total cost) versus bM and bR for the instance p03.

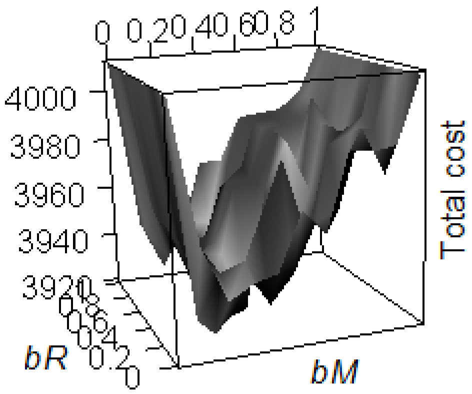


Figure 6.3. Surface plot of the objective solution (total cost) versus bM and bR for the instance p09.

1. **Selection of a representative subset of instances**

This first step was performed by using the **Partitioning Around Medoids**[15] (PAM) algorithm. The introduced input was the data in Table 6.1 standardized. The results were k clusters of homogeneous instances and one medoid for each cluster. These medoids made up the subset of instances. Accordingly, each cluster of instances was represented by one instance, independently of the cluster size.

- Value of k

Table 6.2 shows the relation between the number of groups and the **average silhouette**[16], a distance-based measure. Usually, one is interested in maximizing the average silhouette without having a high number of clusters, which would difficult the interpretation and, in our case, would suppose to analyze many groups. Taking into account these considerations, a recommended value for k is 4. The obtained clusters with 4 groups are displayed in Table 6.3.

[15] The PAM algorithm is a k-medoids clustering method that starts from an initial set of k medoids and iteratively replaces one of them by one of the non-medoids if it improves the total distance of the resulting clustering. In this analysis, the Euclidean distance is used.
An excellent book of statistical learning, including clustering methods, has been written by Hastie et al. (2009).

[16] In mathematical terms, the silhouette of an object i is:

$$s(i) = \frac{b(i) - a(i)}{\max(b(i), a(i))}$$

where, $a(i)$ is the mean distance between i and the objects from the same cluster, and $b(i)$ is the mean distance between i and the objects from other clusters.

k	Average silhouette
2	0.258
3	0.303
4	0.391
5	0.372
6	0.352
7	0.402
8	0.393
9	0.422

Table 6.2. Analysis of the number of clusters to include in the PAM algorithm.

Medoids	Clusters
p07	p01, p02, p03, p04, p05, p06, p07, p12, p15
p09	p08 , p09, p10, p11
p19	p18, p19, p20, p21, p22, p23, pr06, pr10
pr03	p13, p14, p16, p17, pr01, pr02, pr03, pr04, pr05, pr07, pr08, pr09

Table 6.3. Clustering of the benchmark instances.

2. **Selection of ranges**: pilot study

The 4 instances selected with the PAM algorithm were solved to gather knowledge about the problem we were facing. **4 equally spaced values** were analyzed for each parameter: 0.125, 0.375, 0.625 and 0.875. Each instance was solved with 4 different seeds[17], as it has been shown that they can have an important influence on the solutions of algorithms including metaheuristics (see Juan et al. (2014), Monteiro et al. (2013), and Czarn et al. (2004)). Hence, a total of 256 runs per chosen instance were executed. The objective solutions for different seeds were aggregated by calculating the lower endpoint of the confidence interval for the population mean[18]. This measure will be denoted as 'Total cost (eCI)'.

The results for the instance p07 are illustrated in a 3D-scatterplot (Figure 6.4). Each aggregated observation is represented by a point, the axes give information of its parameter values and the colour of the point reveals the rank of the total cost (eCI), the light yellow tones indicate

[17] The number of seeds, as the number of parameter values, was decided based on the computational time.

[18] In statistics, the confidence interval for a population mean is calculated with this formula:

$$\bar{X} \pm t_{1-\frac{\propto}{2},n-1} \cdot \frac{s}{\sqrt{n}}$$

Where $\bar{X}$ represents the sample mean, α the confidence level, n the sample size, s the sample deviation and the term $t_{1-\frac{\propto}{2},n-1}$ is the $100(1-\frac{\propto}{2})$ percentile of the t distribution with $n-1$ degrees of freedom.
In this case, n is the number of seeds (4) and α is 0.05, a usually recommended value.
Employing this measure is better than using only the mean because we want to find a good solution, not a good mean; at the end, we will just store the minimum of the objective solutions found with the selected set of parameters values. Therefore, by taking into account the standard deviation, we will choose parameter values that provide a low total cost on average and/or that have a high variance, which means that for some seed the total cost will be relatively far from the mean.
Another good option would be to select the minimum value, it could be better if we worked with few seeds and all the analysis was performed with the same ones.
In this project, we selected the first one because it is a more robust measure.

the worst outputs and the dark red tones highlight the best. It can be concluded that the parameter p^* **does not influence the total cost**. We expected to observe an effect of this parameter; however, the absence of it could be due to the predominance of the other studied effects. This pattern is repeated in all 4 instances. Therefore, from here, we considered a fixed value for this parameter.

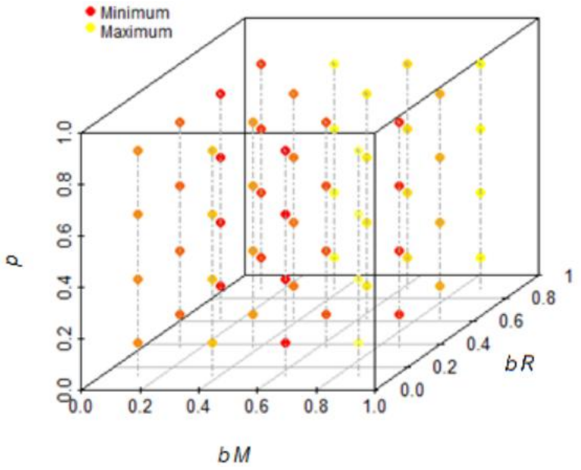


Figure 6.4. Scatterplot of total cost (eCI) versus algorithm parameters for the instance p07.

The next step consisted on better studying the relation between the parameters bM and bR. We solved the selected instances with 25 sets of parameters values as Figure 6.5 indicates. 4 seeds were considered, so the solutions of 100 runs were stored. Then the values for other sets were estimated by linear interpolation.

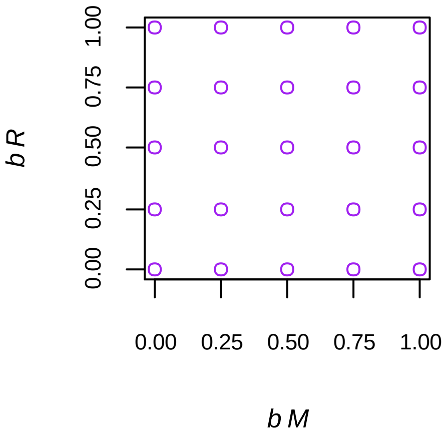
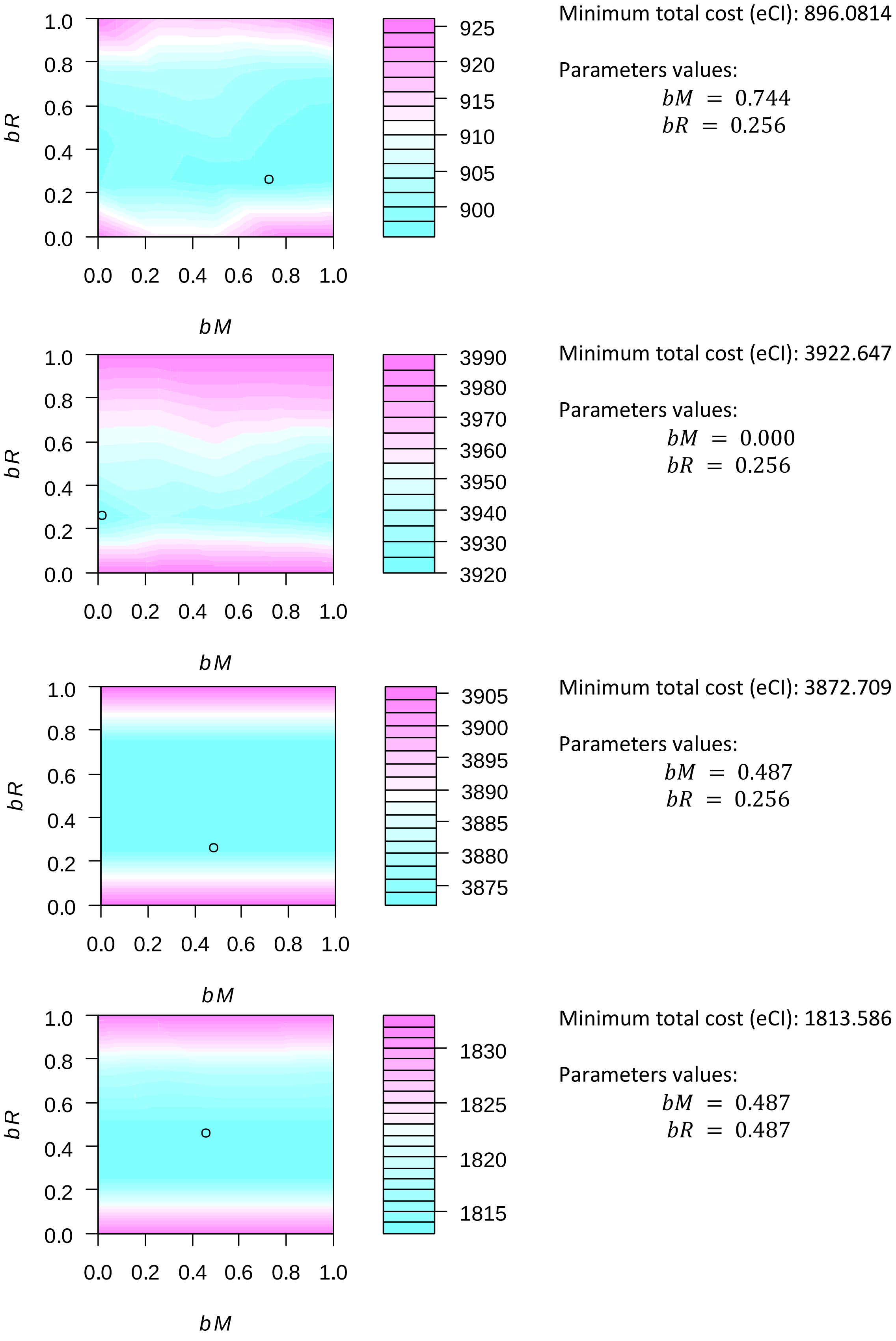


Figure 6.5. Sets of parameters values analyzed in the pilot study.

Figure 6.6 show the results for each instance. It can be concluded that the second parameter, bR, influents more than the first one. The results of the instance p19 do not change for a wide set of values of both parameters, that is to say, the algorithm is very robust for that instance.

Instead of choosing a range, the minimum values were used in the next step.



Minimum total cost (eCI): 896.0814

Parameters values:
$bM = 0.744$
$bR = 0.256$

Minimum total cost (eCI): 3922.647

Parameters values:
$bM = 0.000$
$bR = 0.256$

Minimum total cost (eCI): 3872.709

Parameters values:
$bM = 0.487$
$bR = 0.256$

Minimum total cost (eCI): 1813.586

Parameters values:
$bM = 0.487$
$bR = 0.487$

Figure 6.6. Left: contour plots of total cost (eCI) versus bM and bR. From top to bottom: instance p07, p09, p19 and pr03. The minimum values are indicated with a circle. Right: information about the exact value of the minimum and the parameter values where they are found.

3. **Estimation of optimal values**

- Design of the experiment

A **Central Composite Design** of 2 factors was implemented; the replicates were based on different seeds. Considering the computational time availability, the difficulty of the problem and the power analysis[19], which was realized before estimating the regression models (i.e. “a priori”), a number of 4 replicates was initially chosen. Figure 6.7 represents the design for the first instance of the subset. It is centred on the values estimated in the previous section. The difference between points was set at 0.05 for both parameters[20].

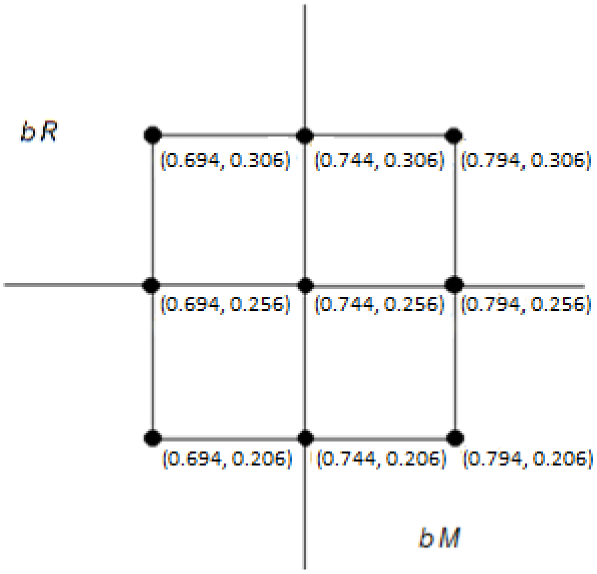


Figure 6.7. Central Composite Design for the instance p07. We decided to set $\alpha = 0$ to ensure that we did not set values greater than 1 or smaller than 0. Although this would not have happened with this instance, we prefer to use the same design for all instances. Hence, the resulting design is a 3^2 Factorial Design.

- Estimations

Figure 6.8 shows the objective solutions for instance p09; each point represents the lower endpoint of the confidence interval for the population mean calculated with 4 observations (replicates) and the black asterisks are the mean for each value of bM. The lines are not parallel, so it can be deduced that there is an interaction between the parameters. From these combinations, we will select $bM = 0.05$ and $bR = 0.256$ in order to minimize the total cost (eCI) (left plot). However, to minimize the computational time, we would choose $bM = 0.05$ and $bR = 0.206$ (right plot).

[19] For a large effect size, two predictors and a sample size of 36 (4 seeds · 9 observations per seed), at the 95% confidence level, the power is 0.87. The large effect size is the proposed in Cohen (1988).

[20] It was based on several experiments. It had to be big enough to obtain different solutions for each point but sufficiently small, to work in the neighborhood of the central point, where a local minimum is expected to be.

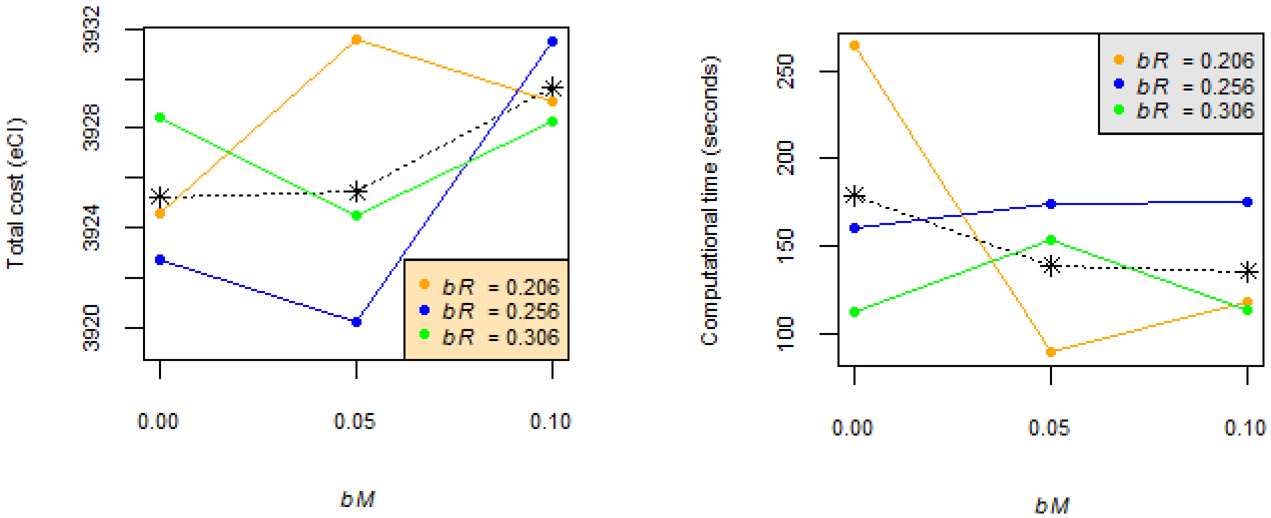


Figure 6.8. Results for the instance p09. Left: total cost (eCI). Right: computational time in seconds.

In multi-objective optimization there is usually a trade-off as in this case, the best solution depends on the response variable that is being considered. A possible solution consists in reckoning a weighted mean by assigning a weight to each response variable. In this master's degree thesis, we will just focus on the total cost due to its major importance. Just for illustration purposes, we present the bi-objective problem for the instance p09 and a possible solution (Figure 6.9).

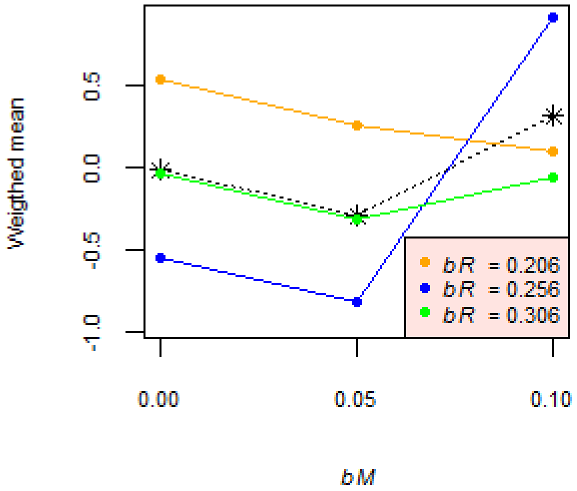


Figure 6.9. Bi-objective optimization. Results for the instance p09 (standardized data). Weights: 0.6, total cost (eCI) and 0.4, computational time.

This experiment was carried out also for the instances p07 and pr03. However, it did not make sense for the instance p19, as its variance near the minimum is too low. The minimum values, for each analized instance, were found at:

Medoid	Set
p07	$bM = 0.794, bR = 0.256$
p09	$bM = 0.050, bR = 0.256$
p19	$bM = 0.487, bR = 0.487$
pr03	$bM = 0.437, bR = 0.206$

Table 6.4. Best set of parameter values for each cluster of the benchmark instances.

At this point, we tried to estimate two regression models, one per response variable, with the data of each instance. Nevertheless, the goodness of fit of the obtained models (both linear and quadratic) was extremely low. Therefore, we opted for proposing the previous sets of parameter values (Table 6.4).

The final step was averaging the parameter values. As the contourplots suggest that some instances are much more robust that others, we decided to give more weight[21] to the most variable. The proposed set of parameter values was: $bM = 0.517$, $bR = 0.284$.

6.3 Results analysis

Following the next step of the proposed methodology, we tested the proposed set on the instances of the subset and compared the results with those provided by other sets (9 random sets were considered). Each instance was solved for 4 different seeds and the solution with the lowest total cost was stored. The results[22] are summarized in Table 6.5. Considering the number of sets and the ties, our set can be classified as valid; it provides better results that the ones expected due to random.

	Rank
p07	5
p09	1
p19	5.5 (1-10)
pr03	3.5 (2-5)

Table 6.5. Rank of our results for the instances of the subset compared with 9 random sets of parameter values. In case of ties, the parentheses reveal which ranks present them.

Table 6.6 shows the results when solving all instances with the proposed sets of parameter values.

	PS	Rank	RS1	RS2	RS3	RS4	RS5	RS6	RS7	RS8	RS9
p01	578.2	8	576.9	578.2	578.2	578.2	578.2	576.9	576.9	576.9	576.9
p02	476.7	5	473.9	476.7	476.7	476.7	476.7	476.7	473.9	476.7	476.7
p03	641.2	5.5	641.2	641.2	641.2	641.2	641.2	641.2	641.2	641.2	643.7
p04	1015.5	3	1011.0	1019.0	1020.3	1018.2	1019.8	1030.7	1030.8	1011.0	1036.6
p05	752.9	1	755.7	755.0	762.4	755.0	763.4	766.3	762.4	765.0	767.8
p06	882.1	4	880.4	882.2	880.4	882.2	883.0	881.5	887.2	884.3	886.1
p07	899.8	5	902.2	898.0	898.9	898.0	897.8	904.3	901.6	900.6	905.6
p12	1319.0	6.5	1319.0	1319.0	1319.0	1319.0	1319.0	1319.0	1319.0	1319.0	1319.0
p15	2562.9	4	2562.9	2562.9	2562.9	2562.9	2562.9	2564.0	2564.0	2564.0	2564.0
p08	4441.2	1	4441.5	4443.6	4453.9	4446.5	4449.0	4457.4	4468.3	4442.8	4477.5
p09	3922.3	1	3928.3	3927.9	3961.1	3925.8	3930.1	3960.3	3960.3	3960.3	3967.3

[21] Concretely, the coefficient of variation (which is the ratio of the standard deviation to the mean) of each instance was calculated. The weight of an instance was defined as the coefficient for that instance divided by the sum of all.

[22] The expected rank due to random would be 5.5. In the case of a very robust solution, for example for the instance p19, it would also be 5.5.

p10	3680.0	2	3692.3	3690.4	3699.1	3679.6	3690.7	3701.2	3709.6	3699.0	3713.6
p11	3574.8	1	3594.4	3581.4	3582.5	3589.0	3591.4	3594.0	3594.0	3594.0	3594.0
p18	3825.8	4	3825.8	3823.7	3825.8	3825.8	3825.8	3825.8	3825.8	3825.8	3825.8
p19	3869.6	5.5	3869.6	3869.6	3869.6	3869.6	3869.6	3869.6	3869.6	3869.6	3869.6
p20	4091.5	7	4091.5	4091.5	4091.5	4091.5	4091.5	4085.9	4080.3	4091.5	4085.9
p21	5660.7	2	5658.4	5678.4	5690.8	5678.4	5678.4	5690.8	5690.8	5690.8	5690.8
p22	5801.0	3	5803.9	5803.9	5801.0	5806.4	5801.0	5801.0	5805.7	5801.0	5806.4
p23	6140.0	3	6140.0	6145.6	6145.6	6145.6	6145.6	6140.0	6140.0	6145.6	6140.0
pr06	2706.1	2	2713.0	2711.8	2716.8	2699.9	2715.1	2719.9	2725.4	2711.7	2730.8
pr10	3021.4	1	3027.1	3027.0	3056.9	3082.4	3037.5	3033.6	3037.8	3047.9	3056.7
p13	1319.0	5.5	1319.0	1319.0	1319.0	1319.0	1319.0	1319.0	1319.0	1319.0	1319.0
p14	1360.1	5.5	1360.1	1360.1	1360.1	1360.1	1360.1	1360.1	1360.1	1360.1	1360.1
p16	2585.4	3.5	2585.4	2585.4	2585.4	2585.4	2585.4	2586.1	2586.1	2585.4	2586.1
p17	2725.8	6.5	2725.8	2725.8	2725.8	2725.8	2725.8	2725.8	2720.2	2725.8	2720.2
pr01	861.3	3	861.3	861.3	861.3	861.3	861.3	861.3	861.3	861.3	867.8
pr02	1316.0	8	1314.2	1314.4	1316.0	1309.0	1314.4	1310.7	1316.0	1310.7	1316.0
pr03	1813.6	3.5	1813.6	1812.3	1813.6	1813.6	1813.6	1813.6	1818.5	1813.8	1821.4
pr04	2089.9	3	2107.5	2103.6	2086.4	2104.4	2100.0	2104.4	2124.8	2087.7	2101.9
pr05	2368.6	3	2360.5	2373.1	2371.3	2375.4	2365.9	2372.0	2373.2	2370.7	2380.6
pr07	1109.2	4.5	1109.2	1109.2	1109.2	1109.2	1109.2	1109.2	1109.2	1109.2	1110.4
pr08	1678.9	3	1678.9	1678.9	1679.9	1678.9	1678.9	1680.6	1682.9	1679.9	1686.8
pr09	2152.0	5	2143.1	2175.5	2149.2	2147.2	2147.2	2173.6	2178.2	2164.0	2187.4

Table 6.6. Table of results. It includes our results, the ranks and the results provided by 9 random sets. PS: Proposed set. RS: Random set. Gray: instances of the subset. Purple: worse results compared with our results. Green: better results. Orange: equal results. Data rounded to one decimal place.

This data is more concisely presented in Table 6.7, which shows the average rank.

	Average rank
33 inst.	4
29 inst.	4

Table 6.7. Average rank of our results for all benchmark instances and same measure without considering the four instances of the subset.

The coloured cells of Table 6.6 reveal that there is no set of the 9 randomly generated that performs 'always' better than our one and, at the same time, there is no set that performs 'always' worse; the better values are not concentrated in a column, and there is no clear pattern. Therefore, it seems that the best set of the 10 reported is highly instance-dependent. Even though, our proposed set obtains an average rank of 4, performing especially well for the instances of the second and the third clusters.

Finally, we compared our results with the ones provided by the set of parameter values that the authors of the project proposed[23]. As before, each instance was solved for 4 seeds and

[23] In order to be able to compare results, the ones from the paper were recalculated to adjust them to the number of seeds (4), number of iterations and computer used in this master's degree thesis.

only the solution with the lowest total cost was stored. The differences are shown in Table 6.8. The mean difference is -0.08%, indicating that the results obtained with the proposed methodology are significantly[24] better.

	OR	PR	Difference (%)
p01	578.2	578.5	-0.06%
p02	476.7	473.9	0.59%
p03	641.2	641.2	0.00%
p04	1015.5	1020.0	-0.44%
p05	752.9	760.3	-0.98%
p06	882.1	882.6	-0.06%
p07	899.8	897.0	0.31%
p12	1319.0	1319.0	0.00%
p15	2562.9	2562.3	0.03%
p08	4441.2	4448.1	-0.16%
p09	3922.3	3929.9	-0.19%
p10	3680.0	3680.4	-0.01%
p11	3574.8	3578.7	-0.11%
p18	3825.8	3823.7	0.05%
p19	3869.6	3869.6	0.00%
p20	4091.5	4091.5	0.00%
p21	5660.7	5684.3	-0.41%
p22	5801.0	5803.9	-0.05%
p23	6140.0	6145.4	-0.09%
pr06	2706.1	2714.5	-0.31%
pr10	3021.4	3038.9	-0.58%
p13	1319.0	1319.0	0.00%
p14	1360.1	1360.1	0.00%
p16	2585.4	2589.4	-0.15%
p17	2725.8	2725.8	0.00%
pr01	861.3	861.3	0.00%
pr02	1316.0	1314.4	0.12%
pr03	1813.6	1813.8	-0.01%
pr04	2089.9	2092.3	-0.11%
pr05	2368.6	2367.7	0.04%
pr07	1109.2	1109.2	0.00%
pr08	1678.9	1678.9	0.00%
pr09	2152.0	2152.4	-0.02%

Table 6.8. Comparison between our results and the ones provided by the set proposed in the project. OR: our results. PR: project results.

[24] The Wilcoxon signed Rank test with continuity correction was applied as the data was not normal. This test is a non-parametric test for paired data to analyze differences between population mean ranks. The p-value was 0.01.

It is important to note that these are **preliminary results**. In order to confirm our results, it would be necessary to repeat this comparison with a higher number of seeds.

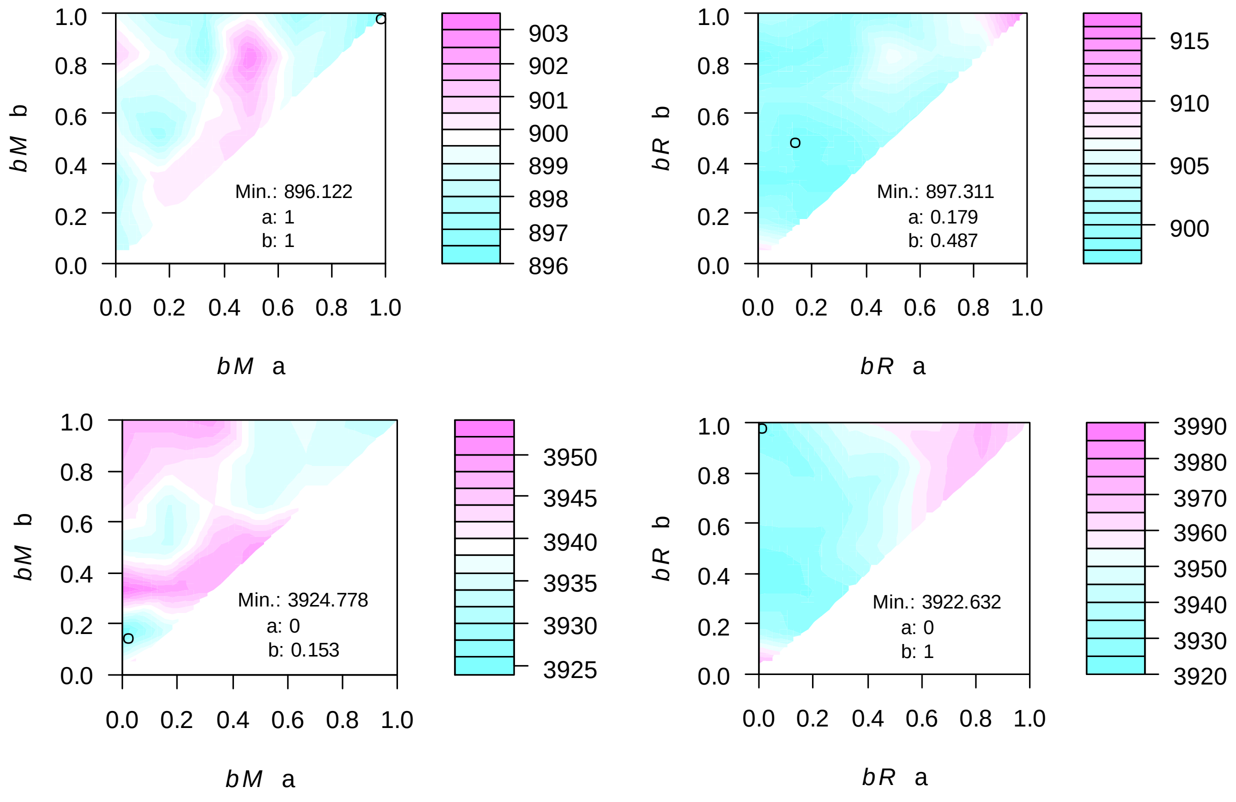
- The set proposed by the authors of the project

The fine-tuning process performed by the authors provided this proposal:

$$bM \sim U\ (0.5, 0.8)$$
$$bR \sim U\ (0.1, 0.2)$$
$$p \sim U\ (0.1, 0.2)$$

Instead of choosing a constant value, a Uniform distribution was selected for each parameter. So at each iteration of the algorithm execution, three values are simulated using these distributions. As the number of iterations considered in the hybrid algorithm is relatively high, this procedure incorporates more diversity of results, which can lead to better solutions.

Based on this idea, some runs were executed to try to improve our results. The third parameter, as in the previous experiment, was set constant. To simplify the analysis, it was assumed that there was no interaction[25] between the first parameter (bM) and the second one (bR). Therefore, two sets of runs were studied. The first one had fixed and equal values for the distribution of the second parameter (i.e. $bR \sim \mathrm{U}\ (\mathrm{a_{bR}}, \mathrm{b_{bR}})$, where $\mathrm{a_{bR}} = \mathrm{b_{bR}}$). Similarly, the values of the first distribution were set for the second set of runs ($bM \sim \mathrm{U}\ (\mathrm{a_{bM}}, \mathrm{b_{bM}})$, where $\mathrm{a_{bM}} = \mathrm{b_{bM}}$). The 4 instances of the subset, 4 seeds and 7 equidistant values (ranging from 0 to 1) were analyzed. The solutions are shown in Figure 6.10.



[25] Figure 17 does not indicate the existence of it, which makes this assumption more realistic.

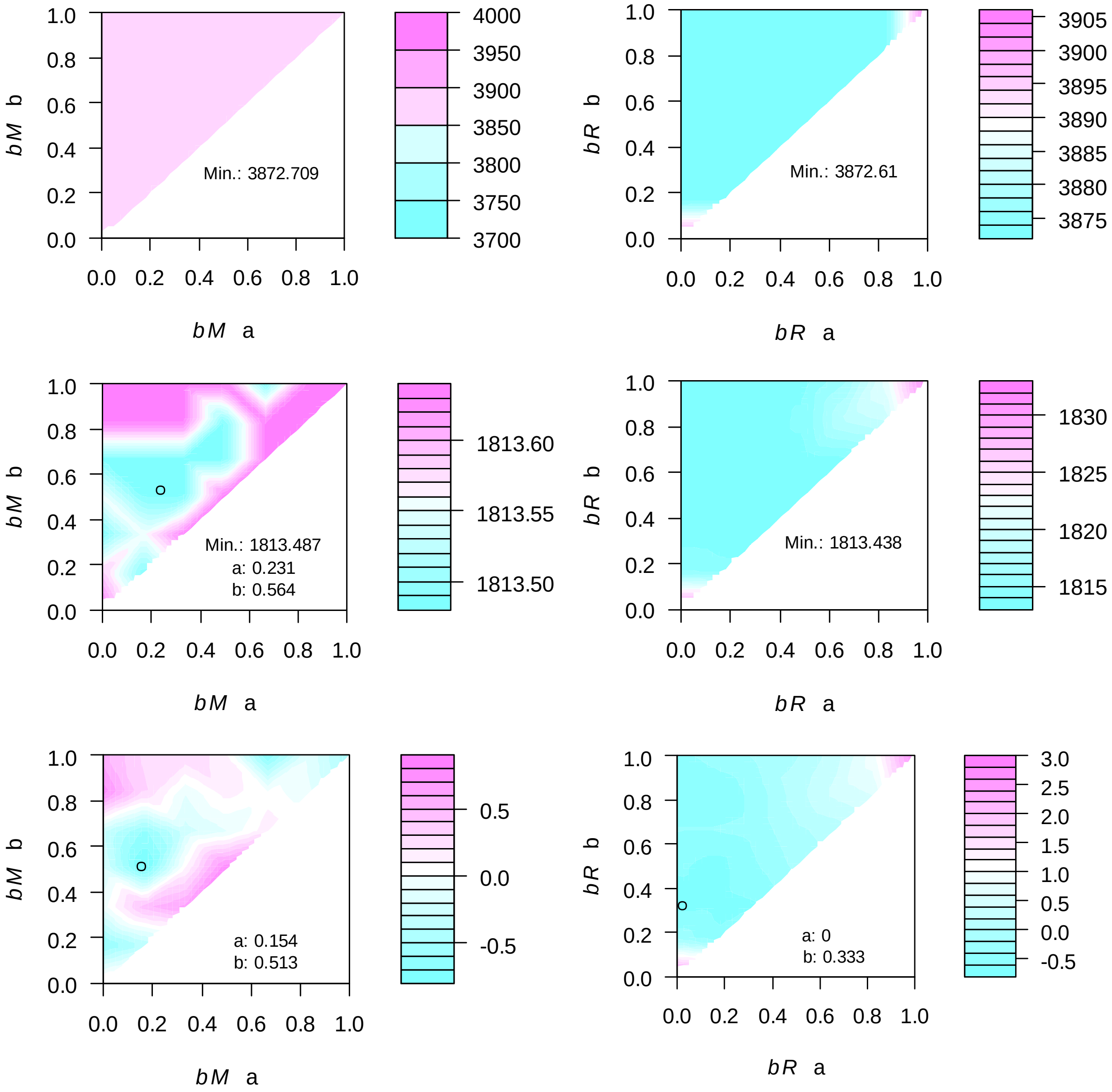


Figure 6.10. Left: contour plots of total cost (eCI) versus the parameters of the distribution of bM for a fixed bR. From top to bottom: instance p07, p09, p19, pr03 and all four instances[26]. The minimum values are indicated with a circle. Right: contour plots of total cost (eCI) versus the parameters of the distribution of bR for a fixed bM.

Afterwards we compared the results obtained with constant parameter values with the ones provided by this new proposal, $bM \sim U\,(0.154, 0.513)$ and $bR \sim U\,(0, 0.333)$. Table 6.9 shows the differences between results.

[26] For drawing the fifth plot, the data of the four instances was gathered, once standardized. The linear interpolation used for this average the duplicate values. As there are differences between the plots of the instances, using only the average of the parameter values proposed for each one could have lead to inefficient results. For example, if we averaged the proposed sets for the instances 07 and 09, the result would provide poor values for both parameters. This is the reason why we opted for this method.

	OR (UD)	OR	Difference (%)
p01	578.2	578.2	0.00%
p02	476.7	476.7	0.00%
p03	641.2	641.2	0.00%
p04	1011.5	1015.5	-0.39%
p05	755.0	752.9	0.28%
p06	880.4	882.1	-0.19%
p07	897.0	899.8	-0.31%
p12	1319.0	1319.0	0.00%
p15	2562.9	2562.9	0.00%
p08	4441.9	4441.2	0.02%
p09	3925.5	3922.3	0.08%
p10	3677.4	3680.0	-0.07%
p11	3583.1	3574.8	0.23%
p18	3825.8	3825.8	0.00%
p19	3869.6	3869.6	0.00%
p20	4085.9	4091.5	-0.14%
p21	5678.4	5660.7	0.31%
p22	5801.0	5801.0	0.00%
p23	6145.6	6140.0	0.09%
pr06	2702.6	2706.1	-0.13%
pr10	3035.1	3021.4	0.45%
p13	1319.0	1319.0	0.00%
p14	1360.1	1360.1	0.00%
p16	2589.3	2585.4	0.15%
p17	2720.2	2725.8	-0.20%
pr01	861.3	861.3	0.00%
pr02	1309.0	1316.0	-0.53%
pr03	1813.6	1813.6	0.00%
pr04	2100.5	2089.9	0.51%
pr05	2378.5	2368.6	0.42%
pr07	1109.2	1109.2	0.00%
pr08	1678.9	1678.9	0.00%
pr09	2147.5	2152.0	-0.21%

Table 6.9. Comparison between the results obtained with constant values and with parameters following Uniform distributions. OR: our results. OR (UD): our results (Uniform distributions).

Although the new results are better for some instances, the average difference (0.01%) does not indicate a general improvement.

- Final comments of the experiment

The performed experiment has enabled us to show some statistical approaches to tackle the Parameter Setting Problem. In order to improve it, design of experiments could be applied when tuning the parameter of the distributions. Furthermore, it would be fruitful to **test several distributions** to find the most suitable (e.g. Normal or Log-normal distributions). Other interesting change would consist in setting some values for the distribution that boost the diversity of results (e.g. $bM \sim \mathrm{U}\ (0{,}1)$) and modifying the algorithm to **store data** about the selected parameter and the solution for each iteration. After an initial phase, we would know which parameter values provide the best values and we could use them for the next iterations. This could be considered a Control Parameter Strategy.

7. Conclusions and future research

7.1 Main conclusions

This master's degree thesis has addressed the Parameter Setting Problem in metaheuristics focusing on the use of different statistical techniques.

Metaheuristics are potent procedures for solving NP-hard problems. They can provide **sufficiently good solutions in a relatively short amount of time**. Their use has grown during the last decades in a high number of fields. Despite being easy to adapt to different problems, there are many kinds of metaheuristics. The reason is explained by the multidisciplinary theorem called "No free lunch". In this context, it states that there is no metaheuristic that outperforms the others in all the problems. An introduction of metaheuristics has been shown in section 2.1. Studying the literature, it has been seen that, in the past, researches did not use to clearly report how they chose the parameter values of the metaheuristics. However, it has changed, and nowadays, we find an increasing interest for this problem as the influence of these values on the performance of metaheuristics has been proven. In fact, there are **several approaches**: the Parameter Tuning Strategies, which aim to find a robust set of parameter values for a group of instances by using statistical techniques, the Parameter Control Strategies, based on using the information gathered during the solving of an instance to adapt the parameter values, and the Instance-specific Parameter Tuning Strategies, which employ a statistical learning technique to obtain a recommended set which depends on the instance features. The definition of the Parameter Setting Problem, the different approaches, and their comparisons, have been presented in Section 2.2.

Considering the **diversity of statistical tools** used by researchers to solve the PSP, Chapter 3 has reviewed some of the main contributions of the academic community. Almost all the existing methodologies use **Regression Models and Design of Experiments**. Chapter 4 has outlined one representative investigation (paper) of each approach.

A methodology including concepts of Regression Models, Design of Experiments and Multi-objective Optimization can be developed to obtain a **general and scientific methodology** to tackle the PSP. It has been reported in Chapter 5.

It can be also concluded that the proposed statistics-based methodology provides promising solutions. Chapter 6 describes the application of our methodology. Indisputably, the outputs highlight the **importance/need of further research** on this topic.

In summary, this master's degree thesis has tried to shed light on the Parameter Setting Problem analyzing the existing approaches and the techniques used so far. The main contributions of it are:

- A general methodology based on Regression Models and Design of Experiment which include some validation procedures. Its steps seek to provide an efficient procedure. Our proposal is necessarily flexible, as the different features of each problem can make some options more adequate than others.

- A first approximation to tackle the Parameter Setting Problem. It has been illustrated with a real application which has been useful to analyze the benefits and pitfalls of the proposed methodology.

7.2 Limitations and future research topics

The proposed methodology presents some limitations which allow us to propose a range of future possibilities.

- The main limitation of this work has been the difficulty to model the relation between the parameters and the objective solution of the metaheuristic with quadratic regression models.

However, the adequacy of this technique has been proven in many investigations. We conclude that the rugged fitness landscape of our problem could need other approaches as **polynomial regressions of higher degree** or **non-parametric methods,** such as K-nearest neighbours regression, in order to obtain a better adjustment. Any proposed methodology has to be flexible enough to be capable of tackling all sorts of problems, so this issue could be more studied.

- Due to the huge variety of problems in which metaheuristics are employed, it is essential to test each methodology on **several problems.**

- Another part that could be improved in our methodology would be the clustering phase. It would be helpful to develop some procedure to determine which features plays a major role in splitting a group of instances into subgroups with similar optimum sets of parameter values. Our approach was naive in this respect, as it considered that all the analyzed features had the same role/weight.

- We have focused on a problem with only three parameters. Indeed, researchers usually analyze three or four factors. However, sometimes it is desirable to study **tens of factors** that can affect a system simultaneously. In those cases, a Full Factorial Design can be too expensive. Other popular and more effective designs are: the Fractional Factorial, the Central Composite, the Box-Behnken (formed by combining 2^k Factorials with Incomplete Block Designs, they can be very efficient in terms of required runs and they are either rotatable or nearly rotatable) and the D-Optimal (in which the practitioner decides the numbers of runs a priori, its objective is to minimize the variance across all regression coefficients). Another interesting design which is investigated in Méndez-Vázquez et al. (2013) consists in generating a full Factorial Design as an initial enumeration, clustering the listed runs and retrieving the $k-$medoids associated to the k clusters. Only these medoids are run, consequently the number of required runs is dramatically decreased. All these designs enable us to investigate relations with more parameters, more interactions and usually more realistic.

Other potential lines of future work are discussed in the remainder of this section.

- In our opinion, the Instance-specific Parameter Tuning Strategies constitute a new approach that requires much further investigation. It is a field where statisticians could apply a wide range of techniques. For instance, **Principal Component Analysis** could be used to convert the variables describing the main features of each instance into linearly uncorrelated variables while reducing the number of variables, and to analyze the contribution of each original variable to explain the variability of the data. Working with uncorrelated variables would avoid potential problems of multicollinearity. One more example, as it was proposed from a theoretical point of view in Dobslaw (2010), **neural networks** could be trained with features of some instances and their objective solutions for a set of combinations of parameter values. Then, networks could be used as an oracle to return a proposal of parameter values for new instances. In this same line, as suggested by Pavón et al. (2009), concepts of **Bayesian networks** could help to improve the oracle by feeding it with new information provided by the new instances and their resolution.

In conclusion, statisticians could play an even more important role in tackling the Parameter Setting Problem. It is a real problem, which despite having been studied in recent years, **lacks of a methodology commonly accepted** by the scientific community.

References

Adenso-Díaz, B. and Laguna, M. 2006. Fine-Tuning of Algorithms Using Fractional Experimental Designs and Local Search. *Operations Research,* 54 (1), 99-114.

Bartz-Beielstein, T. and Markon, S. 2004. *Tuning Search Algorithms for Real-World Applications: A Regression Tree Based Approach.* Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2004, 1, 1111-1118, Portland, OR, USA.

Birattari, M. 2005. *Tuning Metaheuristics: A machine learning perspective*. Springer.

Boussaïd, I., Lepagnot, J. and Siarry, P. 2013. A survey on optimization metaheuristics. *Information Sciences*, 237, 82-117.

Caramia, M. and Dell'Olmo, P. 2008. *Multi-objective Management in Freight Logistics*. Springer

Clarke, G. and Wright, J.W. 1964. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research,* 12 (4), 568-581.

Cohen, J. 1988. *Statistical Power Analysis for the Behavioral Sciences.* Lawrence Erlbaum Associates.

Coy, S. P., Golden, B. L., Runger, G. C. and Wasil, E.A. 2000. Using Experimental Design to Find Effective Parameter Settings for Heuristics. *Journal of Heuristics,* 7, 77-97.

Czarn, A., MacNish, C., Vijayan, K., Turlach, B. and Gupta, R. 2004. Statistical Exploratory Analysis of Genetic Algorithms. *IEEE Transactions on Evolutionary Computation*, 8 (4), 405-421.

Dobslaw, F. 2010. *A Parameter Tuning Framework for Metaheuristics Based on Design of Experiments and Artificial Neural Networks.* Proceedings of the International Conference on Computer Mathematics and Natural Computing 2010, Rome, Italy.

Eiben, A.E., Hinterding, R. and Michalewicz, Z. 1999. Parameter Control in Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 3 (2), 124-141.

Faul, F., Erdfelder, E., Buchner, A. and Lang, A. 2009. Statistical power analyses using G*Power 3.1: Tests for correlation and regression analyses. *Behavior Research Methods,* 41 (4), 1149-1160.

Gunawan, A., Chuin Lau, H. and Lindawati. 2011*. Fine-Tuning Algorithm Parameters Using the Design of Experiments Approach*. Proceedings of the 5th International Conference on Learning and Intelligent Optimization, 278-292. Rome, Italy.

Hastie, T., Tibshirani, R. and Friedman, J. 2009. *The Elements of Statistical Learning. Data Mining, Inference, and Prediction.* Springer.

Hutter, F., Hamadi, Y., Hoos, H. H. and Leyton-Brown, K. 2006. *Performance Prediction and Automated Tuning of Randomized and Parametric Algorithms*. Proceedings of the 12th International Conference on Principles and Practice of Constraint Programming, 213-228, Nantes, France.

Juan, A. A., Pascual, I., Guimarans, D. and Barrios, B. 2014. Combining Biased Randomization with Iterated Local Search for solving the Multi-Depot Vehicle Routing Problem. *International Transactions in Operational Research*. DOI: 10.1111/itor.12101

Kim, Y-D., Lim, H-G. and Park, M.-W. 1996. Search heuristics for a flowshop scheduling problem in a printed circuit board assembly process. *European Journal of Operational Research,* 91, 124-143.

Lenth, R. V. 2009. Response-Surface Methods in R, Using rsm. *Journal of Statistical Software*, 32 (7), 1-17.

Lessmann, S., Caserta, M. and Arango, I.M. 2011. Tuning metaheuristics: A data mining based approach for particle swarm optimization. *Expert Systems with Applications,* 38, 12826-12838.

Lobo, F.J., Lima, C.F. and Michalewicz, Z. 2007. *Parameter Setting in Evolutionary Algorithms.* Springer.

Marler, R. T. and Arora, J. S. 2004. Survey of multi-objective optimization methods for engineering. *Struct Mulltidisc Optim,* 26, 369-395.

Méndez-Vázquez, Y. M., Ramírez-Rojas, K. K. and Cabrera-Ríos, M. 2013. *The Search for Experimental Design with Tens of Variables: Preliminary Results.* Proceedings of the 2013 Winter Simulation Conference, 2654-2665, Washington, United States.

Monteiro. A., Oliveira. T. and Juan. A. A. 2013. *O Planeamento de Experiências no aperfeiçoamento de Metaheuristicas*. Master's degree thesis. Universidade Aberta.

Montero, E., Riff, M.C. and Neveu, B. 2014. A beginner's guide to tuning methods. *Applied Soft Computing,* 17, 39-51.

Montgomery, D.C. 2009. *Design and analysis of experiments,* 7th edition. John Wiley & Sons.

Myers, R.H. and Montgomery, D.C. 2002. *Response Surface Methodology. Process and Product Optimization Using Designed Experiments,* 2nd edition. John Wiley & Sons.

Park, M-W. and Kim, Y-D. 1998. A systematic procedure for setting parameters in simulated annealing algorithms. *Computers & Operations Research,* 25 (3), 207-217.

Parsons, R. and Johnson, M.E. 1997. A case study in experimental design applied to genetic algorithms with applications to DNA Sequence Assembly. *American Journal of Mathematical and Management Sciences,* 17 (3-4), 369-396.

Pavón, R., Díaz, F., Laza, R. and Luzón, V. 2009. Automatic parameter tuning with a Bayesian case-based reasoning system. *Expert Systems with Applications,* 36, 3407-3420.

Pongcharoen, P., Chainate, W. and Thapatsuwan, P. 2006. Exploration of Genetic Parameters and Operators through Travelling Salesman Problem. *ScienceAsia,* 33, 215-222.

Ramos, I.C.O., Goldbarg, M.C., Goldbarg, E.G. and Neto, A.D.D. 2005. *Logistic Regression for Parameter Tuning on an Evolutionary Algorithm.* Proceedings of the IEEE Congress on Evolutionary Computation*,* CEC 2005, 2, 1061-1068, Edinburgh, UK.

Ridge, E., and Kudenko, D. 2007. *Analyzing Heuristic Performance with Response Surface Models: Prediction. Optimization and Robustness.* Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, 150-157. London, England.

Ries, J. 2009. *Instance-Based Flexible Parameter Tuning for Meta-Heuristics using Fuzzy-Logic*. Thesis. British library. Electronic Theses Online Service.

Ries, J., Beullens, P. and Salt, D. 2012. Instance-specific multi-objective parameter tuning based on fuzzy logic. *European Journal of Operational Research*, 218, 305-315.

Talbi, E. 2009. *Metaheuristics: from design to implementation*. John Wiley & Sons.

Tewolde, G.S., Hanna, D.M. and Haskell, R.E. 2009. *Enhancing Performance of PSO with Automatic Parameter Tuning Technique*. Proceedings of the Swarm Intelligence Symposium, 2009, 67-73. Nashville, Tennessee, United States.

Xu, J., Chiu, S. Y. and Glover, F. 1998. Fine-tuning a Tabu Search Algorithm with Statistical Tests. *International Transactions in Operational Research,* 5(3), 233-244.