Algorithmic assistance to the optimization process in Vehicle Routing Problems

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur
(Dr.-Ing.)

von Dipl.-Ing. M.Sc. Yasel José Costa Salas
geb. am 19.03.1982 in Nuevitas, Kuba

genehmigt durch die Fakultät für Maschinenbau
der Otto-von-Guericke-Universität Magdeburg

Gutachter:
Prof. Dr.-Ing. habil. Prof. E. h. Dr. h. c. mult. Michael Schenk
Prof. Dr.-Ing. Dr. h. C. Norge Isaías Coello Machado
PD Dr. rer. nat. habil. Juri Tolujew

Promotionskolloquium am 24.April 2013

Acknowledgements

Men always fight in life for earn virtues and the acceptation of their similar. But definitely one of the greatest virtues that one must not dispense is the gratitude. For that reason, this important moment in my life bring me the formidable memories of gratitude for those especial persons that support me and made truth this fabulous dream.

Starting from the farthest place (Cuba):

- Thanks to my scientific father and supervisor since my first studies, Prof. Dr. C. René Abreu Ledón. The most intelligent and precise professional that I have ever known.
- Thanks to Prof. Dr. C. Carlos Machado Oses for his faith in my abilities and for the most important, introduce me to my father and mentor “Prof. Dr.-Ing. Dr. h. c. Norge Isaías Coello Machado. The person how has taught me many thing in the academy and practical life.
- Thanks to my department colleagues in the UCLV essential part of my professional training.

Gratefulness for my second countrymen (Germans):

- I definitely must begin with the person who really is responsible of this dream made truth, Frau. Dr.-Ing. Elke Glistau. Manager, mentor and second mother of my stays in OVGU (Germany). For her my highest sense of respect, affection and deep gratitude.
- Thanks to my supervisor PD Dr. rer. nat. habil. Juri Tolujew for his confidence from the beginning. For the time spent in revision and his wise advices.
- Thanks to my dear friends and young colleagues Dr.-Ing. Sebastian Trojahn, Dr.-Ing. Tobias Reggelin, Dipl.-Math. Annegret Brandau, M.Sc. Til Hennies (from my first group of German students), Dipl.-Wirtsch.-Ing. Fabian Behrendt and M.Sc. Markus Koch (always gentile friend), for their support and friendship.
- One especial acknowledgement to all my German students since 2009. They really have improved me as educator and also remind me my greatest time in Germany, my second land.

For all this especial persons, my gratitude, forever.

“The hard part begins now; one can be a doctor, the hardest part is to look like a doctor”

Yasel J. Costa Salas

Central University “Marta Abreu” of Las Villas
Faculty of Industrial Engineering and Tourism
Department of Industrial Engineering
Camajuani Road Km 5.5
Santa Clara

Email: yaselcs@uclv.edu.cu
URL:http://www.fiit.uclv.edu.cu

Abstract

The combinatorial optimization, as a scientific paradigm, has a significant influence on the increasement of the effectiveness of any logistic decision, with particular emphasis on vehicle routing decisions. Although these decisions have been widely studied, research on vehicle routing optimization mostly focused on the empirical application of the solution methods (exact or approximate), already established in the scientific literature, or on providing new methods for such purposes. However, in both cases, the greatest contributions are addressed to the design, improvement and application of optimization methods, a priori unknowing, how effective these methods can be, considering the complexity feature of the problem in a multivariate context. Therefore, the general methodologies to carry out the optimization process for such decisions lack of an integrative approach, which allows to check the relevancy degree of the proposed methods.

To avoid the mentioned inadequacies, a conceptual model and procedure have been proposed in this thesis. Both proposals involved the assistance of decision-making in vehicle routing optimization. In this sense, the major research contributions are summarized in the conception of the optimization process into three stages, the design and modification of meta-heuristic algorithms based on Ant Colony Optimization (ACO) and the application of some robust statistic techniques in decision-making.

Finally, the proposed algorithms were successfully applied to a realistic case study: route planning for the repair of electrical breakdowns in Cuban territory, set up in the city of Santa Clara.

Contents

Figures

Tables

Acronyms

ACO: Ant Colony Optimization

M-ACS: Multi-type Ant Colony System

VRP: Vehicle Routing Problem

mTSP: Multiple Traveling Salesman Problem

ACS: Ant Colony System

ANOVA: Analysis of Variance

AS: Ant System

BB: Branch and Bound

BC: Branch and Cut

BCP: Branch-and-Cut-and-Price

BP: Branch and Price

BPC: Branch-and-Bound-and-Price

BWAS: Best-Worst Ant System

CG: Column Generation

CP: Cutting Plane

CVRP: Capacitated Vehicle Routing Problem

DA: Discriminant Analysis

DP: Dynamic Programming

DVRP: Dynamic Vehicle Routing Problem

GA: Genetic Algorithm

GIE: Global Index of Effectiveness

GIS: Geographical Information Systems

GP: Goal Programming

GPS: Global Positioning Systems

IH: Insertion Heuristic

KB: Knowledge Base

KSP: Knapsack Sharing Problem

LIP: Linear Integer Programming

LS: Local Search

MACS: Multiple Ant Colony System

MDVRP: Multiple Deposits Vehicle Routing Problem

MMAS: MAX-MIN Ant System

NP: nondeterministic polynomial bounded

NUE: National Union of Electricity

P: Polynomial bounded

PSO: Particle Swarm Optimization

RBAS: Rank-Based Ant System

SA: Simulated Annealing

SECs: Subtour Elimination Constraints

SH: Saving Heuristic

SVRPs: Stochastic Vehicle Routing Problems

TS: Tabu Search

TS-ACO: Two-Stage Ant Colony Optimization

TSP: Traveling Salesman Problem

WT: Work-team

VRPST: Vehicle Routing Problem with Stochastic Times

VRDs: Vehicle Routing Decisions

VRPB: Vehicle Routing Problem with Backhaul

VRPBTW: Vehicle Routing Problem with Time Windows and Backhaul

VRPPD: Vehicle Routing Problem with Pickup and Delivery

VRPSC: Vehicle Routing Problem with Stochastic Customers

VRPSD: Vehicle Routing Problem with Stochastic Demands

VRPTW: Vehicle Routing Problem with Time Windows

Chapter 1

Introduction

1.1 Motivation

Decision-making, in any practical or theoretical problem, requires of knowledge in order to execute effective decision. The effectiveness[1] is a result of systematic process with defined elements that are managed in a sequence of detailed steps. In this sense, decision-making based on optimization methods is the most used solution approach to assure the effectiveness and therefore, success for the managerial organizations. Cuban companies bet for such strategies, essentially for those that allow to optimize limited resources [*PERFECCIONAMIENTO EMPRESARIAL*, 2007]. Furthermore, they are searching the proper mathematical and computational tools which support hard decisions, considering the current complexities imposed by the competitive managerial environment.

In Logistics, an important group of these hard decisions are involved and its suitable management can represent the main competitive advantage of any enterprise. The decision-making related with logistic processes can determine either success or failure, in most of cases. There exist two types of decisions when logistic is planned: design and optimization, being the optimization decisions the harder to figure out in mathematical and computational matters.

The optimization, as a process, consists of finding the best values of the variable for a particular criterion or, in other contexts, the best decisions for a particular measure of performance [Baker, 2011]. Other well-accepted concept is proposed in Venkataraman [2009], where the optimization process is defined as a search of the best objective operating within a set of constraints.

The combinatorial optimization and its solution methods have been one of the most studied scientific subjects in Operation Research literature [Costa Salas et al., 2011]. Various methodologies have been developed for the optimization process. One of the well-know methodologies is showed in Venkataraman [2009] and Yang [2010], who suggest that the primary aspects to consider in optimization process are summarized in the following steps:

[1] In this thesis the effectiveness (efficacy + efficiency) is measured by two performance indicators: solution quality (efficacy) and computation time (efficiency) of the algorithms.

1. Define a set of decisions.
2. Define an objective.
3. Establish the conditions that constraints must satisfy.
4. Chose a mathematical model.

On the other hand, authors such as Abid [2008], Ravindran [2008] y Blumenfeld [2009] argue that, the general methodology designed for decision-making in Operation Research is appropriate, as a framework, to carry out the optimization processes. In opposition to the above methodology, the framework is composed of six steps.

In both described methodologies, the steps are addressed, essentially, to the formalization and construction of optimization models, excluding relevant aspects that can be decisive on increasing the effectiveness. Within the main theoretical limitations encountered in the methodologies are:

1. The absence of multivariate learning process capable to associate the complexity feature of the decisions with its solution methods. Therefore, the analyzed methodologies present a limited *proactive* approach.
2. The definition of one or more objectives is strictly related with the mathematical construction of the model, without considering that the previous selection of an appropriate optimization method can be crucial for the effectiveness in decision-making.
3. The absence of steps capable to check, after the optimization process, how relevant (*relevancy degree*[2]) the proposed optimization methods were, regarding the established performance indicators.
4. Sensitivity analysis in optimization has been addressed as experimental approach, without establishing specific methodologies.
5. The lack of an *integrative* approach, considering the missing steps mentioned in the above limitations and the steps included in currents methodologies.

Moreover, similar theoretical limitations are evident in one of the most common optimization decisions in logistics planning, transportation decisions. The costs associated with such decisions are the most significant within the logistic costs [Tseng et al., 2005]. The transportation problems involve an important group of decisions, ranging from the vehicle selection to the route planning (Vehicle Routing Problem, VRP).

Vehicle Routing Problems (VRPs) are a great family of problems which have been extensively studied by different authors, usually specialists in the areas of Operations Research and Logistics. The optimization methods to these combinatorial problems can be classified as either exact or approximate, heuristics and metaheuristics are the most used algorithmic approach within the last classification.

[2] In this thesis the *relevancy degree* is expressed proportionally to effectiveness achieved by the proposed algorithms, considering the external conditions as well.

Several recent researches have shown the effective use of approximate algorithms on real large-scale VRPs [Kytöjoki et al., 2007, Sim et al., 2009]. On the other hand, there are valuable solutions applying exact algorithms to small dimension VRPs [Archetti et al., 2007; Iori et al., 2007, Andersen et al., 2011]. However, consequently with the theoretical limitations, these studies are only focused on model construction and the empirical test of optimization methods for VRP [Li et al., 2007, Belfiore et al., 2009], excluding a previous and posterior analysis in the optimization process, which allows to predict relevant solution methods according to the complexity of the problems.

The previous analyses in optimization are only dedicated to determine the complexity class *P* (polynomial bounded) or *NP* (nondeterministic polynomial bounded) [Choi & Tcha, 2007, Hashimoto et al., 2006]. In particular, the VRPs belong to *NP* class, because when the problem scale grows; the space search of the problem grows according to non-polynomial function (exponential, factorial). Thus, this family of problems is among the hardest combinatorial problem.

In the literature, authors such as Jiang et al. [2008], Torresani et al. [2008], Chen et al. [2008], Talbi [2009], Jozefowiez et al [2007] suggest to resolve *NP* combinatorial problem, such VRPs, using approximate methods, especially when the problem grows dimensionality, due to an increase in the number of nodes (number of customers to visit in a graph). This conception shows the univariate approach that present the analysis of complexity, ignoring other important features of complexity such as, fleet size, time windows, fleet type and others, which can lead to solve the problem using either exact or approximate methods.

Although several algorithmic proposals have been applied for VRPs, using both exact and approximate optimization methods, there are some real-life conditions that impose to design and modify such approaches in order to carry out an effective decision-making process in realistic context. Furthermore, we consider also the statement of the theorem No free lunch (Wolpert & Macready, 1997), which suggests that there is no method that guarantees to be better than others for any problems. Therefore, this leads to develop new solution methods in optimization fields.

The solution of combinatorial problems in Cuba, particularly the VRPs, presents the same above mentioned inadequacies. However, the design and application of optimization methods for such problems have not received the same attention as in the international context. Decision-making is characterized by an empirical approach, prevailing the excessive control in the ineffective planned routes. In this sense, a real-life case study is indentified; it is "Route planning to repair electrical breakdowns in power networks". The company involved in this decision-making process needs to deal with two decision scenarios (normal weather conditions and after hurricanes), for which the optimization methods are unknown.

The theoretical limitations and the univariate approach in the complexity of VRP, both identified in the case study, were the main motivation for this research.

1.2 Research goal and Contributions

The aim of this thesis consists of assisting the optimization process related with Vehicle Routing Decisions (VRDs), for such propose a new *conceptual model* and general *procedure* is presented in order to increase the effectiveness in decision-making. The design of general procedure involves the main research contributions, which are summarized in the conception and integration of three stages (previous, during, and after optimization) for the optimization process in a multivariate context and in the exploration of new application of the proposed tools. More specifically, other contributions of this thesis belong to the following research topics.

Knowledge Discovery

In this research a Knowledge Base (KB) is proposed. It represents a set of real-life solution of VRPs, showing in each sample (case) the best experiences in algorithmic approach for optimization. Two classifiers (Discriminant Analysis and the decision tree algorithm C4.5) are trained on KB, aim to predict relevant categories of optimization methods (exact or approximate). A novel methodology for estimating training-set size of KB is applied, adapting this methodology from other application context (Cancer classification problems).

Ant Colony Optimization

A further contribution of this thesis is a new approach called Multi-type Ant Colony System (M-ACS) based on ACO, which falls under the umbrella of the meta-heuristic techniques. The algorithm uses multiple artificial ant colonies in order to solve the Multiple Traveling Salesman Problem (mTSP), which is the basic formulation of our case study; each colony represents a set of possible global solutions. The colonies cooperate among them, sharing its experiences through “frequent” pheromone exchange.

The algorithm performance is compared with the results of the efficient heuristic of Lin-Kernighan reported in Dazhi and Dingwei [2007], using benchmark problems form literature. Dazhi and Dingwei [2007] proposes the Lin-Kernighan heuristic based on the transformation described in Tang et al. [2000], for that reason we defined M-ACS according to this transformation as well.

Computational complexity of M-ACS, as an algorithm family of ACO, is analyzed, yielding an overall time complexity of $O(n^2)$. Finally, a sensitivity analysis is developed for all ACO strategies in the research.

Computational Implementations

The computational implementations developed in this thesis, specifically *VRP Solution classifier* and *ANTRO version 2.0*, are the main commercial software proposed for assisting the decision-making process described in the case study. Especially in *ANTRO versions 2.0* all realistic complexities of the case study are conceived (unexpected breakdown, priority level of the breakdowns, and the probabilistic time for repair).

1.3 Outline of the Thesis

This thesis is structured as follows:

Chapter 2 provides the reader with main concepts and background information about the work presented in the thesis. First, the basic concepts, frameworks and methodologies of optimization process are introduced, analyzing the integration characteristics of studied approach as well. The Vehicle Routing Problem and its extensions are described in the context of combinatorial optimization. Next, we summarized the most used algorithms based on the *two* categories of optimization methods (*exact* and *approximate*), making special emphasis on the metaheuristic approaches.

The chapter ends with an important discussion about the main characteristics of decision-making related with VRPs in Cuba. Some remarkable statistics summarize the types of research developed for VRPs solution.

In Chapter 3 we present the novel conception and integrative approach of optimization process based on the proposal of conceptual model. Here the goal of the model, potentialities, inputs, its process and outputs are explained. The main part of the chapter is dedicated to describe the *three* stages of the general procedure. The first stage comprises the essential aspects of the *Knowledge Base* construction and the formulation of classifiers (Discriminant Analysis and C4.5) for the knowledge discovery process. In the second stage some specific algorithms according to are analyzed their “remarks to be applied”. Both stages are supported in a computational implementation, which is also described in this chapter. The chapter concludes with the third stage called post-optimization, where various statistical tests are provided to analyze the relevance of proposed optimization algorithms in the earlier stages and the sensitivity of algorithm user-defined parameters.

The application of general procedure to real-life case study is reported in Chapter 4. We use all stages of the procedure to solve a realistic decision-making process entitled “Route planning to repair electrical breakdowns in power networks”. Experimental results of the proposed algorithms are obtained, using both benchmark datasets from literature and data of the real instance. The influence of the user-defined parameters is examined for all approximate algorithmic approaches.

A performance analysis based on the simulation of some problems conforms the last part of the chapter. The Global Index of Effectiveness (*GIE*) is determined applying the stages of the general procedure.

Finally, in Chapter 5, we summarize the main contributions of this thesis and outline directions for future research.

Chapter 2

Optimization Theory, Vehicle Routing Problem and its solution approaches

The combinatorial problems, such as VRP and its extensions, have become a widely accepted field within applied mathematics. This chapter provides the theoretical support of all subjects treated in the thesis.

We start with the discussion of various concepts of optimization and analyze the current methodologies proposed for decision-making in optimization process. Then, we give an overview of the fundamental aspects of optimization theory, computational complexity and post-optimization analysis. Here, the necessity of integration and proactiveness in the optimization process is studied as well.

In the larger part of this chapter we examine some extensions of the VRP, focusing on those which are studied in other chapters of this thesis. In this sense, some particular characteristics of VRP are analyzed, especially the dynamicity, which is most related with the mathematical formulation of the case study presented in Chapter 4. Subsequently, an extensive survey of the optimization methods (both exact and approximate) is developed. Metaheuristic algorithms, specially those based on Ant Colony Optimization (ACO) are deeply examined, due to its close connection with the main contributions of this research.

Finally, we end this chapter presenting an overview of the decision-making related with the vehicle routing in Cuba. The solution approaches adopted by the Cuban enterprises are outlined in descriptive statistic environment, indicating that the optimization methods have been poorly treated in decision-making.

2.1 The optimization process in decision-making

Many well-known authors have studied the optimization as an essential scientific activity in decision-making. One of the most recent definitions of optimization is provided by Baker [2011], whom optimization consists of finding the best values of the variable for a particular criterion or, in other contexts, the best decisions for a particular measure of performance. Undoubtedly, the optimization is in any subject of decision-making [Venkataraman, 2009]. In fact, it is considered an idea as old as mankind itself [Ho et al., 2007], and every evolution process in nature reveals that it follows optimization [Diwekar, 2008].

In the optimization, an analytical visualization of the decision is usually given before its adoption [Hillier & Lieberman, 2010; Joshi & Moudgalya, 2004]. This proves the proactive approach that presents the optimization as scientific technique.

One comprehensive conceptualization is provided by Bartholomew-Biggs [2008], who defines the optimization as the seeking values for certain designs or control which minimize (or sometimes maximize) an objective function. In Griva et al. [2009] the optimization models are considered as the attempt to express, in mathematical terms, the goal of solving a problem in the “best” way.

Mustafi [2007] suggests that in the optimization, as in Operation Research, the main features of the problem are examined, data are collected or generated and subsequently quantitative analysis are carried out using “appropriate” mathematical methods and statistical principles. In this thesis we give more importance to the relevant (appropriate) selection of solution methods, it will be deeply addressed in later chapters.

On the other hand, in Klemeš et al. [2010] the optimization is expressed as a solution of the mathematical model. Therefore, the optimization is justly considered as the mathematical programming of the given problem. These authors summarize the characteristics of optimization problems in the following aspects:

- Optimization criterion: Minimization or maximization.
- Presence of constrains: Restricted or unrestricted.
- Linearity of the functions: Linear or nonlinear.
- Type of variables: Discrete or continuous.

The former research propose relevant algorithms considering indistinctly each characteristic previously described, which indicates a limited integrative approach in decision-making related with algorithm selection.

In accordance with Rao [2009], the optimization is the act of obtaining the best result under given circumstances and for such purposes there is not a single method available for solving all optimization problems efficiently. Here, similar characteristics to those of Klemeš et al. [2010] are suggested.

Antoniou & Lu [2007] present other interesting definition, considering the optimization as, if it is possible to measure and change what is “god” or “bad”. In addition, the author of this thesis consider either the optimization as the process of obtaining the “best”, if it is possible to know what is “relevant”, in terms of relating the complexity features of an optimization problem with its solution methods.

The authors consulted in the scientific literature; identify optimization, in most cases as an essential action in decision-making, which attempts to find the best solution, or the optimal solution to the problem under certain considerations. The fundamental difference between research resides in the conception and establishment of methodologies to carrying out the optimization process.

In reviewing the literature, we find many general methodologies to develop an optimization process in decision-making. The common proposal found point out to decrease solution time in decision-making [Sodhi & Tang, 2010]. There exist some of these methodologies defined into three steps. In Diwekar [2008] a classical example of these methodologies is proposed, where the first step consists of understanding the system, which is well-known in mathematical terminology as the problem modeling. The second step is based on finding a measure of effectiveness, while the third involves degree of freedom analysis and applying a proper optimization algorithm to find the solution. Other authors such as Venkataraman [2009] and Yang [2010] pose similar methodologies for optimization, although much more oriented towards the definition of a set of alternatives and compliance of conditions or constraints to which these alternatives are subjected.

According to Giorgi et al. [2004] optimization process can be implemented following three steps, which may be often formalized as follows:

a) The behavior of a system (in the most general meaning) depends on some variables, some of them beyond the control of the decision maker (these are named the "data of the problem) and the other ones under his control (these latter are true variables of the problem, variables usually described by a vector $x \in R^n$.
b) The various alternative possibilities for the decision maker are described by a set $S \in R^n$: so one has to choose, in an optimal way, a vector $x^0 \in S$ or more than one vector, in case of a problem with several solutions.
c) Let us consider the case of $f: D \subseteq R^n \to R$. i.e. the case of scalar function. Then the choice of vector $x^0 \in S$ is considered an optimal one and x^0 solves the optimization problem when it is:

$$\mathrm{f}(\mathrm{z}^0) \leq \mathrm{f}(\mathrm{x}), \qquad \text{for each } \mathrm{x} \in \mathrm{S} \qquad [2.1]$$

Undoubtedly, the methodologies described are mainly intended to build the optimization model, with emphasis on the knowledge of the system based only on the variables involved in developing the mathematical model, excluding out of the analysis, another important group of variables that describe the complexity of the problem and its relationship with the solution algorithms.

Moreover, Abid [2008], Ravindran [2008], Blumenfeld [2009] y Kandiller [2007] suggest that the general methodology followed in Operation Research is fully applicable to develop the optimization process. The steps involved in this methodology are described as

1) Formulate the problem.
2) Construct a model of the system.
3) Select a solution technique.
4) Obtain a solution to the problem.
5) Establish controls over the system.
6) Implement the solution.

The methodology proposed in Operation Research is more comprehensive that the other mentioned (three steps), considering the number of steps involved. However, this methodology presents the same inadequacies that the described above, due to the absence of a learning process, which makes possible to establish, starting form problem formulation (Step 1) or the construction of the model (Step 2), the relevant algorithms for the problem solution (linked to Step 4). Another limitation, in the sense of achieving greater global effectiveness in decision-making, is referred to the control and implementation of the solution. These steps are based solely on performance analysis of the proposed optimization method, obviating the relevance that such proposed method presents.

In research reported by Deb [2004], Belegundu & Chandrupatla [2011] and Ravindran & Ravindran [2008] a framework (see Figure 2.1) is defined for solving optimization problems. Here, Mustafi [2007] and Hillier & Lieberman [2010] propose similar frameworks to the previously mentioned, although, the evaluation and implementation of optimization solution are incorporated. The frameworks as the methodologies described exclude steps that allow the choice of the optimization method mainly from the need for optimization and definition of variables. The design variables, illustrated in the framework of the Figure 2.1 are referred to the decision variable of the optimization model; these variables are not related with the complexity of the problem.

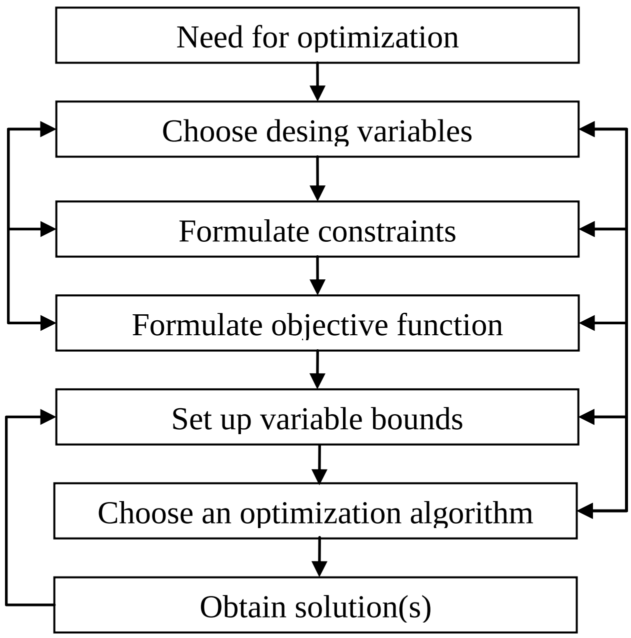


Figure 2.1: Framework for the optimization process

Source: Deb [2004]

The worst mistakes in optimization process are deeply analyzed by Jeżowski & Thullie [2009]. These authors argue that the inconsistent problem representation into the mathematical model is one of the worst failures in decision-making, disregarding that an inadequate solution methods can be as serious as the aforementioned mistake.

The major weaknesses, considering the analysis of all previous researches, is observed before (pre-optimization) and after (post-optimization) the construction of optimization model. For these reasons, in the following sections of this chapter, we propose to examine in detail, the main issues in both moments of the optimization process, with particular emphasis on the optimization process related with the vehicle routing.

2.1.1 Computational complexity in optimization problems

In general, the computational complexity involves two measures of complexity: spacial-complexity and time-complexity [Boudali et al., 2005]. Recent experimental results, reported in some research about complexity theory, have proved that the spacial-complexity is irrelevant in the complexity analysis, due to the computation capacities of the current computers, which are able to admit any large-scale instances of decision problem (such optimization problems). For this reason, we will focus only on time-complexity analysis, with special stress in the VRPs.

The complexity of any decision problem can be expressed through the total combinations in its search space [Wilhelm et al., 2008]. There are two basis categories of decision problem: the class P of decision problems that can be solved by a polynomial-time algorithm, and the class NP that can be solved by non-deterministic polynomial-time algorithm. The time-complexity of any algorithm is measured by a time-complexity equation that gives, depending on the instance size, the maximal run-time for the algorithm to solve an instance. The size of a problem instance reflects the amount of data to encode an instance in a compact form [Stützle, 1998]. An illustrative example of the class P is given when the minimum value is searched in a list of n figures. Let t the time to check each figure of the list, then, the problem can be solved in the worst-case with time-complexity expressed into $O(t \cdot n)$.

On the contrary, in the Traveling Salesman Problem (TSP) [Dantzig et al, 1954], simple variant of the VRPs [Dantzig & Ramser, 1959], the finding the optimal solution in the worst-case of computational complexity can be expressed into $O[t \cdot (n-1)!]$ (see Figure 2.2). The combinations in the search space of this combinatorial problem are determined according to a factorial function.

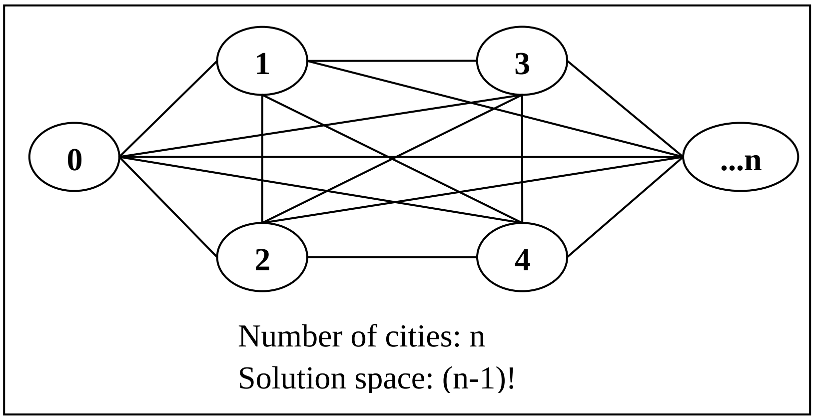


Figure 2.2: Representation of the classical TSP

Some authors such as Kim et al. [2009], Wang & Lu [2009], Shanmugam et al. [2010], Archetti et al. [2011] and Meihua et al. [2011] argue that VRPs belong to the *NP*-hard category, since we cannot expect an exact algorithm to solve a given instance to optimality in polynomially bounded computation time.

In accordance with the reviewed literature, in this thesis the complexity analysis as the main study before optimization is identified. Furthermore, we consider that the complexity classes *P* and *NP* have been linked with optimization methods, whether exact and approximate respectively. A good example of the latter is shown in Gilbert [1992], Mihelič & Robič [2005] and Bourgeois et al. [2009], where the complexity categories are directly associated with the already established optimization methods. In this sense, Yi & Kumar [2007], Ai & Kachitvichyanukul [2009a], Jozefowiez et al. [2008] and Eksioglu et al. [2009], suggest that relevant solution of a *NP* problems is achieved, specifically in the VRPs, when the approximate methods (mostly heuristic and meta-heuristics) are used.

Other studies conducted by Rajan & Mohan [2004] and Dréo [2006] propose that the exact methods are most suitable for solving problems of *P* complexity class. Undoubtedly, in both cases (associating exact methods to *P* problems or approximate methods to *NP* class) a direct association of the complexity category is carried out. The direct association, as a decision rule to determine which solution method will be relevant in optimization process, may be inaccurate, depending on the problem size. For instance, some non large-scale VRP extensions are efficiently solved with exact methods [Ropke & Cordeau, 2009]. Hence, the problem size has a significant influence in the relevance of the optimization methods.

On the other hand, in Cordeau & Laporte [2003, 2007] are reported empirical studies to determine, based on instance size (number of cities in the graph) of one VRP extension, which specific algorithm, within approximate methods, must be used to obtain the highest quality solution. This empirical study is limited to analyze the relevance of the optimization methods in a univariate context, excluding other important variables that must be considered. Here, Contreras & Fernandez [2012] suggest to take some other variables into consideration, although the variables are not directly conceived to analyze the relevance of the optimization methods in multivariate context.

The pre-optimization analysis described in this section is summarized in two fundamental aspects: the first is closely linked *to* the *selection* of the *optimization method* according to the category of computational complexity, *disregarding the real size of the instance.* The second aspect *involves experimental studies* that relate the *optimization method* with the *true scale* of the problem, which is expressed by only one variable (number of nodes or size of the graph). The limitations presented by both sides will be studied in the next chapter of this research.

2.1.2 Analysis of post-optimization

The post-optimality analysis in optimization models has been studied in many research [Jan, 2007; Kara, 2011 and Moghaddam & Usher, 2011]. These studies focus mostly on the posteriori-influence of the performance indicators in optimization model, considering changes in its data inputs, or fixed parameters of the algorithms [Gál & Greenberg, 1997].

The sensitivity of the optimization model parameters has been the most common field of research referred to post-optimization in the current issues, mainly conditioned by the implementation of algorithms which are pseudo-random nature, such as approximate algorithms, with special emphasis on the meta-heuristics. According to Belgacem & Hifi [2008], sensitivity analysis of an optimal solution consists of computing the ranges within the parameters of an instance of a given problem may vary without altering the optimal solution at hand. However, we consider that the sensitivity is also a common way to check the reliability of the model, even if their parameters are not completely known, which occurs whenever a new optimization algorithm is proposed.

In the more frequent optimization problems of the Supply Chain (VRPs), the sensitivity analysis is a pivotal issue [Gudehus & Kotzab, 2009] and [Chetouane et al., 2012]. The VRPs are one of the most studied when analyzing the sensitivity of the parameters involved in the optimization algorithms to solve them, usually in the approximate methods. However, in the literature are appraised sensitivity studies for both methods (see Table 2.1), which means that sensitivity analysis is still interesting, even when the algorithms are exacts.

Table 2.1: Some sensibility analyzes in the VRPs

Exact methods	**Approximate methods**
Lucas & Chhajed [2004]	Reimann et al. [2003]
Fancis et al. [2006]	Stephan [2006]
Ordóñez et al. [2007]	Quadrifoglio & Dessouky [2008]
Prescott-Gagnon et al. [2009]	Karlaftis et al. [2009]
Rottkemper et al. [2012]	Costanza et al. [2011]

The sensitivity analysis has involved other optimization problems, see Ghosh et al. [2006] and Belgacem & Hifi [2008], these research suggest a perturbation analysis for the classical problem of the backpack (Knapsack Sharing Problem, KSP), which is other of the well-known classic models widely studied in logistic decisions.

Moreover, in Bakirli et al [2011] is developed a sensitivity analysis for one the problem later discussed in this thesis, the classification problems. These authors propose the sensitivity analysis of the incremental genetic algorithm parameters such as crossover probability, mutation probability, elitism and population size.

The computational implementations have been developed for sensitivity analysis; these have been widely accepted by researchers who study combinatorial optimization. The DAKOTA (Design Analysis Kit and Terascale Appplication), developed by Young et al. [2004], is one of the commercial software used by the scientific community.

The duality theory [Murty, 2009], provides important contributions to the post-optimization analysis. This theory allows to obtain more mathematical-economic information of the primal optimization model [Griva et al., 2009 and Löhne, 2011]. After analyzing several recent researches of the theory of duality, we found that this issue has received greater attention when exact algorithms are implemented (see Brian [2008]).

Although many recent research have been addressed to the post-optimization studies, either through sensitivity analysis or the theory of duality, the problem of verifying the relevance in the optimization methods have been poorly treated. Hence, we argue that such verification after the optimization process would make possible to assert or in some cases redefine the type of optimization method (exact or approximate).

2.2 The Vehicle Routing Problem

The Vehicle Routing Problem (VRP) indicates a generic name given to a large family of combinatorial problems related with the delivery (or pick up) of personnel and goods, for which is responsible a fleet of vehicles [Golden et al., 2008]. In general, VRPs involve two very complex decisions: the fleet size determination and the proper routes that the vehicles should follow [Godinho et al., 2008; Costa Salas et al., 2010]. According to José [2009], determining the optimal fleet size of vehicles, used to serve a set of customers distributed in a graph, is one of the most complex decisions with greater economic impact [Hoff et al., 2010]. However, Ballou [2004] y Ai & Kachitvichyanukul [2009b] argue that no matter how complex the decisions were, the aim is to improve customer service by finding the best paths on a road network.

Many authors, including Torres Gemeil et al. [2003], Golden et al. [2008] and Minis et al. [2010], identify the term “Vehicle Routing” as a set of commercial, financial and legal relationships, intended to provide aggregate value of place, time and possession to the supplier products, according to customer expectations and ensuring significant competitive advantages.

In Garza Ríos [2001], Voudouris et al. [2008] and Wenning [2010] the distribution process is recognize as the more complex process that companies have to dealing with, due to the diversity of customers and the real-life constraints in the current context.

The first solution approach (linear programming) for a primitive variant (truck dispatching) of VRP was reported in Dantzing & Ramser [1959], specifically the algorithm was proposed for the optimum routing of a fleet of gasoline delivery trucks between a bulk terminal and a large number of service stations supplied by the terminal. Subsequently, in Clarke & Wright [1964] the first really effective algorithm is introduced: the popular *savings algorithm*. Thus, a wide area of research has endured and grown to the present day. Basically, the main contribution in the issue can be defined according to two trends, on one side, to discover new extensions of VRPs, mostly incorporating features of real-life problems, and on the other side, to find novel algorithms for a more efficient problem solving.

In general, the advances in this field are largely due to the technological development [Kalcsics & Nickel, 2008], the computing power [Du et al., 2007, Chang et al., 2009], this aspect that has reduced the run times of the algorithms [Miguel Andres, 2010]. Moreover, the development of GIS (Geographical Information Systems) [Matis, 2008], the communication equipments (wireless technology) and Global Positioning Systems (GPS)

[Cheung et al., 2008, Larsen et al., 2008], are essential for adequate interaction models and algorithms with planners, and at the same time, with the drivers of the vehicles.

There are a wide range of VRP real-life applications [Mester & Bräysy, 2007; Miguel Andres, 2007; Savelsbergh & Song, 2007; Zäpfel & Bögl, 2008; Jotshi et al., 2009; Berbeglia et al., 2010; Costa Salas et al., 2012], they are mostly involved in the following service systems:

- Food, raw material and fuel distribution.
- Urban transportation.
- Delivery and/or pick up of the correspondence.
- Emergency services.
- Cab services.
- Maintenance or repair services (examined in this thesis as a case study).

The solution of real-life case studies based on the VRPs benchmark models is not the only motivation of the scientific community. These combinatorial problems belong to the complexity class of *NP*-hard [Bianchessi & Righini, 2007]. Hence, the academic motivation lies in the impossibility of finding an optimal solution for instance size measured by a polynomial time-complexity function. This problem has been recognized as one of the seven mathematical problems of this century [Smale, 1998].

2.2.1 Main characteristics of the VRPs

The characteristics of the VRPs have been studied as the most significant subject in distribution management. A classical VRP consists of a number of geographically scattered customers, each of them requires a specified weight (or volume) of goods to be delivered (or picked up). Then, a fleet of identical vehicles dispatched from a single depot is used to deliver the required goods and once the delivery routes have been completed, the vehicles must return to the depot. Each vehicle can carry a limited weight and only one vehicle is allowed to visit each customer. The solution to problem consists of finding a set of delivery routes which satisfy the above requirements at minimal total cost. The Figure 2.3 shows a typical scenario of the classical VRP.

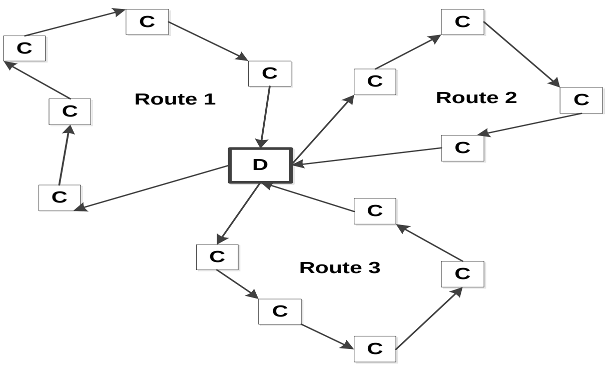


Figure 2.3: Graphical representation of the classical VRP

Based on practical requirements, such as the customer characteristics [Golden et al., 2008], number of depots [Ho et al., 2008] and fleet type [Paraskevopoulos et al., 2008], various extensions of the basic VRP can be found in literature. According to Fernandez [2006], the VRPs present particular characteristics compare with other combinatorial problems (see Figure 2.4).

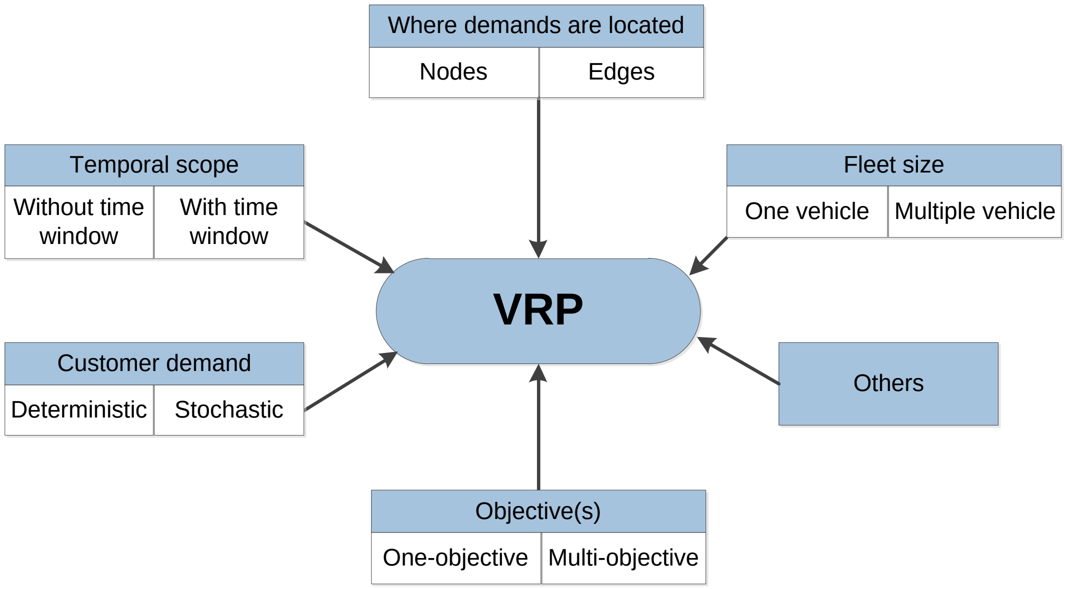


Figure 2.4: Scheme of the particular characteristics in VRPs

Moreover, Toth & Vigo [2002], Helbing [2007], Uyar & Technology [2008], Becher [2008], Eksioglu et al. [2009] and Xie & Levinson [2011] describe the main characteristics of customers, depots and vehicles, which generate some of the well-known variants of the VRP, these variants are fully discussed in the next section.

- **Customers:** Every customer has a demand (goods or services) that must be satisfied by any of the vehicles. Also, an especial condition can be established when a customer is visited by more than one vehicle. Customers can be suppliers in some cases. Other particular situation occurs when the set of customers demands with earliest and latest time deadlines, which is called *time windows*.
- **Depots:** Usually, the route developed by the vehicles start and end in the depot(s), rarely times, the vehicles depart or end in other specific node of the graph (e.g. the driver house or some customer). A single depot is frequent in the VRPs, even though, multiples depots can appear in some variants. When the problem comprises multiple deposits could be that each depot has a fleet of vehicles previously assigned.

  The *time windows* can be either defined in the depot(s). Furthermore, in some cases must be considered the time used for loading, unloading or preparing a vehicle before that the routes start.
- **Vehicles:** The vehicle capacity can be expressed in different way, such as weight and/or volume. In general, each vehicle has an associated fixed cost, which is set up when the vehicle is used. A variable cost is also defined, usually, is proportional to the traveled distance performed by the vehicle. Regularly, each vehicle has a limited

time during the schedule. When the vehicles of the fleet have the same or similar attributes (e.g. efficiency and capacity) the fleet is called *homogeneous*, otherwise is named *heterogeneous*. Eventually, the legal regulations impose some restrictions related with the maximum time or speed that a vehicle can stay on the road.

The goal in the VRPs can be summarized as follows:

- Minimize the total time of transportation.
- Minimize de total traveled distance.
- Minimize the sum of the customer lead times.
- Minimize the fleet size.

2.2.2 Extensions of the Vehicle Routing Problem

Several variants of the classical VRP have been proposed in the literature; basically extensions of VRP are resulting from the real application contexts. These variations are generated by addition of variables and constraints. In this section we will examine the most popular extensions of the VRP, concentrating more attention for those with similar characteristics to the case study presented in this thesis.

The traveling salesman problem (TSP), studied in Gutin & Punnen [2002], is probably the most widely studied combinatorial optimization problem and has attracted a large number of researchers for a long time. Intuitively, it is the problem faced by a salesman who wants to find, starting from his home town, a shortest possible trip [Papadakos et al., 2011] through a given set of customer cities and to return to its home town. In the original TSP model the depot is not clearly defined, since it can be located in any customer position. Furthermore, the customers do not have associated *time windows* constraints [Öncan et al., 2009]. Most problems in goods distribution are related to the VRP, which is a generalization of the TSP [Tatarakis & Minis, 2009; Meisel, 2011]. In the Appendix A.1 some of the more complex variants of the TSP are described.

The Multiple Traveling Salesman Problem (mTSP), in spite of being a generalization of TSP, has been less studied. In this problem, more than one salesman is allowed to be used in order to visit some cities just once [Dazhi & Dingwei, 2007]. Then, the mTSP consists of finding tours for all *m* salesmen, which start and end at the depot. As a result, each intermediate node is visited exactly once and the total cost of visiting all nodes is minimized [Bektas, 2006]. This extension will be deeply studied in this thesis due to its great similarity to our case study [Costa Salas et al, 2012].

The Capacitated Vehicle Routing Problem (CVRP) in an extension of the mTSP, where the customers have a known demand and are visited by a homogeneous fleet of vehicles with limited capacity and initially located at a central depot [Christiansen & Lysgaard, 2007]. Contrary to the TSP or mTSP, in CVRP the number of routes is unknown. Here, the objective is to minimize total traveled distance, such that each customer is serviced exactly once (by a single vehicle) [Sungur et al., 2008] and total load on any vehicle associated with a given route does not exceed vehicle capacity [Barreto et al., 2007].

According to Desaulniers et al. [2008], the Vehicle Routing Problem with Time Windows (VRPTW) is one of the most studied in the literature. In this variant, both limited capacity of the vehicles and *time windows* for each customer are considered. The *time windows* is defined for each customer and consists of a start time (ready time) and end time (due time), within which the customer must be served [Azi et al., 2007]. Time windows can be considered hard when the delivery to the customer is developed out of time limits [Hsu et al., 2007]. Otherwise, they are considered soft but with a penalty in the objective function [Zeng & Wang, 2010].

Another variant of the basic VRP formulation is the Vehicle Routing Problem with Backhaul (VRPB); see Pereira & Tavares [2008]. This extension considers that after the vehicle executes the last delivery, it can visit one or more suppliers, picks up inbound products (backhauling) from the suppliers and carries them back to the depot [Goetschalckx, 2011]. In some formulation of the VRPB, a *time windows* or time interval is found at each customer/supplier location in order to constraint the time service. Then, this extension is called the Vehicle Routing Problem with Time Windows and Backhaul (VRPBTW) [Zhong & Cole, 2005].

The Vehicle Routing Problem with Pickup and Delivery (VRPPD) is a variant of CVRP, in which clients require both pickup and delivery [Festa, 2010], this process are performed simultaneously. The three well-known types of Pickup and Delivery Problem (PDP) are described in Barnhart & Laporte [2007], one is single-commodity PDP, where a single type of good is either picked up or delivered at each node of the graph. The second type involves the pickup and delivery process of two goods (two-commodity PDP) and the third (n-commodity) occurs when each commodity is associated with a single pickup node and a single delivery node.

There exists a small family of problems within VRPs, in Chaovalitwongse et al. [2009] is defined as the Stochastic Vehicle Routing Problems (SVRPs). The name is given to this group of problems due to the random behavior that one or more variables have [Novoa & Storer, 2009]. There are three classical problems belonging to this family:

1) The Vehicle Routing Problem with Stochastic Demands (VRPSD), in which the customers have random demands [Goh & Tan, 2009].
2) The Vehicle Routing Problem with Stochastic Customers (VRPSC), where the customers appears in the graph according to a probability value [Barnhart & Laporte, 2007].
3) The Vehicle Routing Problem with Stochastic Times (VRPST), which occurs when the service time or travel time are random [Cao et al., 2008].

Several depots can be established in VRP (e.g. in Ho et al. [2008]), when the vehicles depart from multiples depots the problem is denominated the Multiple Deposits Vehicle Routing Problem (MDVRP). In Toth & Vigo [2002] is presented a detailed survey of all described extensions of the VRP.

The Dynamic Vehicle Routing Problem (DVRP) is other of the well-known VRP generalization [Hanshar & Ombuki-Berman, 2007]. However, we confer a tremendous importance due to the similarity of this variant with the case study discussed in this thesis. The term DVRP is referred to a wide range of problems where the required information is not given to the decision-maker, but is revealed concurrently with the decision-making process [Demazeau et al., 2009]. In Kreowski et al. [2010] the DVRP considering that the solution is not a static set of routes is analyzed. On the contrary, the routes should evolve, involving the unexpected request (see Figure 2.5).

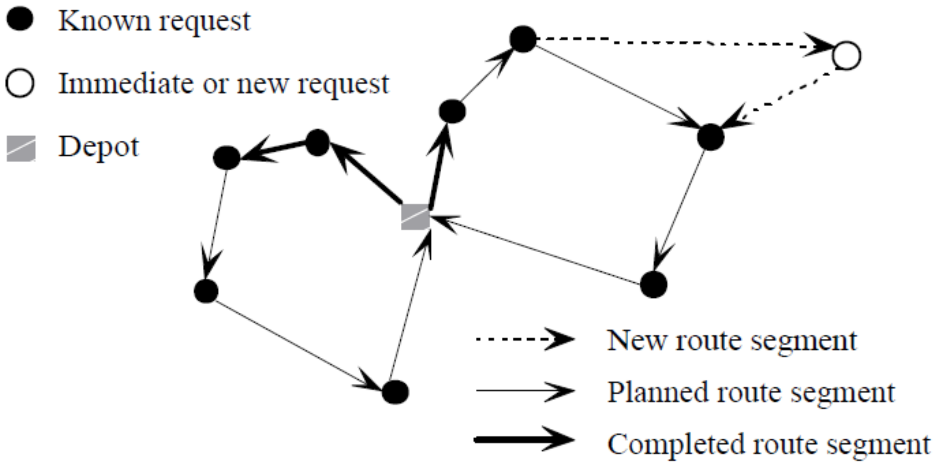


Figure 2.5: View of DVRP scheme

Moreover, Larsen et al. [2007] argue the in DVRP not all information relevant to the planning of the routes is known by the planner when the routing process begins. Therefore, the information can change after the initial routes have been constructed. In Garrido & Riff [2010] are described the main discriminat features between the static and dynamic extensions of the VRP. Furthermore, a comprehensive study about the solution strategies is detailed.

2.3 Solution approaches for the VRPs

Due to the academic and practical importance of VRPs, many algorithms for their solution have been devised. Undoubtedly, hundreds of papers have been devoted to the algorithmic approaches of the several variants of the VRP. The optimization methods to combinatorial problems, particularly to VRPs, can be classified as either *exact* or *approximate* [Gilbert 1992; Golden et al., 2008; Scholz, 2011]. The main differences between *exact* and *approximate* are depicted in Appendix A.2. This comparison is carried out considering the most used indicators of performance in the optimization methods: the quality of the solutions, the computation time, the flexibility and complexity in implementation.

2.3.1 Exact algorithms

The exact approaches to VRP have been well studied since the first solution presented in this filed by Dantzig & Ramser [1959]. The basic principle of an exact algorithm is based on enumerate the full solution space. Therefore, such algorithms can be infeasible due to a significant growth in the exponential size of the solution space. Despite this success, on many extensions of VRPs the performance of exact methods is suitable.

Several experimental results can be found about exact algorithms in VRPs. Based on these results, some decision rules have been proposed related with the VRP instance size. In this sense, Toth & Vigo [2002], Francis & Smilowitz [2006] and Barnhart & Laporte [2007] consider that the exact methods are efficient when the number of customers is approximately below 50 (nodes). However, some other experimental results (see Table 2.2) proof the opposite. The main reason of this contradiction is due to the univariate conception of the problem size (determined only by the number of nodes in the graph).

Table 2.2: Suitable results of exact approaches in the VRPs

Researches	**Problem size**	**Algorithm(s)**	**Other constraints**
Oppen et al. [2010]	Larger than 100	Column Generation (CG)	Time windows and random variables
Dumas et al. [1991]	Larger than 100	Branch-and-Price (BP)	Multiple depots, pickup and delivery and heterogeneous fleet
Bélanger et al. [2006]	Larger than 250	Branch-and-Bound-and-Price (BPC)	Time windows and random variables
Calvete et al. [2007]	Larger than 75	(GP) Goal Programming	Soft time windows and multiple objectives
Ropke & Cordeau [2009]	Less than 100	Branch-and-Cut-and-Price (BCP)	Time windows and pickup and delivery
Righini & Salani [2008]	Between 50 and 100	(DP) Dynamic Programming	Time windows, random customers and heterogeneous fleet
Baldacci et al. [2012]	Larger than 100	Column-and-Cut- Generation and Branch-and-Cut-and-Price	Limited capacity and time windows

The figures in Table 2.2 show that exact algorithms can be effective when the problem size exceeds 50 nodes, which proves that a priori is not trivial to predict, following a unique criterion in the problem size, what family of algorithms will be effective to solving VRP. This issue will be examined in the next chapters of the present research.

As we previously mentioned, the classical way to obtain optimum solutions to the VRPs is to evaluate all possible solutions in feasible solution area. However, there are other strategies to find guaranteed optimum solutions. In this sense, the well-known Branch and Bound (BB), examined in Toth & Vigo [2002], is one of the most studied exact algorithms in literature Farahani et al. [2011]. The BB procedure is more efficient that enumerate all possibilities in solution space. In BB is only enumerated the possibilities by eliminating large classes of solutions using domination or feasibility arguments, straightforward or implicit enumeration [Nemhauser & Wolsey, 1998]. In Padberg & Rinaldi [1991] an improvement of the BB by adding cut and planes forming the formidable Branch and Cut (BC) is proposed.

On the other hand, when the Column Generation (CG) algorithm, proposed by Vanderbeck & Wolsey [1996], is combined with BB, it results into the efficient algorithm called Branch and Price (BP) [Christiansen & Lysgaard, 2007]. The BP, according Baldacci et al. [2008], is considered the most efficient of exact algorithms in literature, with which the present author agrees.

Several variants of the VRP have been solved using the described exact approaches. In Bard et al. [2002] is applied the BC to the VRPTW, while Christiansen & Lysgaard [2007] addressed the VRPSD applying BP but based on the Dantzig-Wolfe decomposition. There exist other exact strategies to solve VRP based on *Lagrangean relaxation* [Jing-Quan, 2009] and Dynamic programming [Righini & Salanil, 2009].

A comprehensive survey of exact methods for VRPs can be found in Laporte [1992] and more recently in Kallehauge [2008], in which the application of the exact algorithms during the last three decades is examined.

Despite the inefficiency of the exact methods in large-scale VRP instances, see Marinakis & Migdalas [2007] and Brito et al. [2009], these algorithms still show suitable results to solve combinatorial problems. Especially for the VRPs, exact strategies are widely accepted and easily understood by decision-makers.

2.3.2 Approximate algorithms

The approximate approaches differ essentially from exact ones as they cannot guarantee to find optimal solutions in finite time. But, for optimization problems, they often find high quality solutions much faster than exact algorithms and are able to successfully attack large instances [Stützle, 1998]. These algorithms have been the most studied in the issue of the VRPs [Laporte & Martín, 2007; Chen et al., 2008]. Approximate algorithmic approaches in VRP can be classified into four categories: *heuristics*, *metaheuristics*, *approximation* and *trial and error*. In the Appendix A.3 the main algorithms according to each of aforementioned category are presented. However, in this section we will focus on the study of the heuristics and metaheuristics approaches, as they are the most popular and efficient within the approximate category [Dondo & Cerdá, 2007].

The *heuristics* were one of the first solution approaches to VRP. Despite its inefficacy to guarantee the optimal solution, the heuristics algorithms can solve large instances of VRP, through a limited exploration of solution space in negligible time [Cordeau et al., 2005].

In Michalewicz & Fogel [2004] one of the most accepted concept of heuristic is proposed, it is defined as a technique (consisting of a rule or a set of rules) which seeks (and hopefully finds) good solutions at a reasonable computational cost.

Various heuristics algorithms have been addressed to VRPs in literature, starting by the one proposed by Clarke & Wright [1964], which is named the *saving algorithm*. Later, Mole & Jameson [1976] and Christofides [1979] introduce the famed algorithm well-known as the *insertion heuristic*. These heuristics can be classified as constructive algorithms which generate solutions from scratch by adding to an initially empty solution components in certain order until a solution is complete.

Another efficient heuristic approaches are examined in Laporte et al. [2000], where the *two-phase* heuristic [Bramel & Simchi-Levi, 1995], *cluster-first* and *route-second* algorithm [Fisher & Jaikumar, 1981] and the *sweep* algorithm [Wren, 1971] are analyzed in details. Recent experimental results reported in Imran et al. [2009], Yazgi Tütüncü et al. [2009] and Na et al. [2011] prove that such approach are still used with great acceptance and success.

The most widely and successfully applied approximate algorithms are *local search* algorithms [Stützle, 1998; Aarts & Lenstra, 2003]. In general, they are *iterative improvement* methods that start from some given solution and try to find a better solution [Ibaraki et al., 2008] in an appropriately defined *neighborhood* [Hemmelmayr et al., 2009] of the current solution. In case a better solution is found it replaces the current solution and the local search is continued right from this point [Stützle, 1998].

The solutions in local search are typically obtained by simple procedures called moves. In most of cases, they are very effective heuristics (see Kytöjoki et al. [2007]). However, the main disadvantage of this algorithm is that it may stop at poor quality local minima. Therefore, Gendreau & Potvin [2010], Zapfel et al. [2010] and the present author consider that local search should be subordinated to the *metaheuristics* algorithms in order to obtain better results.

2.3.2.1 Metaheuristics for the VRPs

The results either theoretical or experimental obtained by applications of *metaheuristic* algorithms are often quite impressive. These solution approaches are effectively superiors than classical heuristics [Toth & Vigo, 2002]. Furthermore, they carry out a much more intelligent search process compared with other mentioned methods. Several definitions of the *metaheuristics* have been provided by well-known authors in this field. Some of them, specifically Osman & Laporte [1996], Dorigo & Stützle [2004], Geiger et al. [2009], Zapfel et al. [2010], and Gendreau & Potvin [2010], agree that a *metaheuristic* can be defined as follows:

A meta-heuristic is an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions. It may manipulate a complete (or incomplete) single solution or a collection of solutions at each iteration. The subordinate heuristics may be high (or low) level procedures, or a simple local search, or just a construction method.

There exist three groups of metaheuristic algorithms to solve the VRPs, the *swarm algorithms* [Marinakis et al., 2010], the *evolutionary algorithms* [Talbi, 2009] and *immune algorithms* [Hu et al., 2009]. Those framed in swarm intelligence are bio-inspired on the behavior of the real animals and insects such as ants [Forestiero et al., 2008; Dressler & Akan, 2010], bees [Karaboga & Akay, 2009], termites [Khereddine & Gzara, 2011], birds [Marinakis & Marinaki, 2010] and fishes [Ma, 2010]. The individual agents of a swarm behave without supervision and each of these agents has a stochastic behavior due to her perception in the neighborhood. Local rules, without any relation to the global pattern, and interactions between self-organized agents lead to the emergence of collective intelligence called swarm intelligence [Karaboga & Akay, 2009].

When we examine the swarm algorithms is mandatory, due to its formidable results in VRPs, lead us to analyze, the set of algorithms based on the Particle Swarm Optimization (PSO) [Kennedy & Eberhart, 1995] and the family of Ant Algorithms (AAs) based on Ant Colony Optimization (ACO) [Dorigo & Gambardella, 1997]. Yet, this type of approximate algorithms will not further be discussed here. However, both approaches have been studied by many researchers, mainly ACO for the static extensions of the VRPs [Chen, 2007a; Chen, 2007b; Colorni & Roizzoli, 2007]. Some other applications of ACO for dynamic variants can be found in Yi & Kumar [2007] and Runka [2009]. Moreover, successfully application of PSO for VRP, in particular for the one extension with *time windows*, is reported in Ai & Kachitvichyanukul [2009a].

The *evolutionary algorithms* (studied in Mester et al. [2007]) are inspired in the natural evolution process, where the more adapted individuals survive (prevail the best solution) while the weakest subject tend to become extinct (solution with poor contribution to the objective function). Multiple variants of VRPs have been solved using evolutionary approaches, predominantly with the family of Genetic Algorithms (GAs) [Alvarenga et al., 2007; Hanshar & Ombuki-Berman, 2007; Lau et al., 2010]. Immune algorithms are a new computational intelligence paradigm, which take inspiration from the immune system. In this area suitable solutions in Hu et al. [2009] and Zhang & Wu [2010] are reported, specifically for real-life instances of distribution problems.

The recent advances in metaheuristics tend to the hybridization [Flisberg, 2009], in most cases combining metaheuristics in order to improve the best results achieved for some extensions of the VRP [Chen et al., 2010]. On the other side, the *hyper-heuristics* constitute another progress in the field of approximate methods. A hyper-heuristic is a high-level algorithm, which generates or chooses a set of low-level heuristics in a common framework, to solve the problem at hand [Garrido & Riff, 2010]. To our knowledge, the most satisfactory results in hyper-heuristic have been obtained by Garrido & Castro [2009] and Garrido & Riff [2010].

2.3.2.2 Ant Colony Optimization

In this section we examine the family of algorithms based on the well-known Ant Colony Optimization (ACO). The main reason to study this metaheuristics separately is due to its linking with the scientific contributions of this thesis. Contrary to the previous section, here ACO is described in detail considering the following elements: the characteristics of the search process, some prominent variants of AAs and the main procedure followed by the artificial ants in route construction.

As we mentioned before, the ACO [Dorigo et al., 1999; Dorigo & Stützle, 2004] belongs to the swarm intelligence branch [Bonabeau, 1999]. Equality to swarm intelligence, in ACO a model of social insect behavior is applied (ants), which work in a distributed manner, without centralized control and self-regulating [Dušan, 2008; Fountas, 2010].

The ACO metaheuristic is inspired by the behavior of the real ants [Stützle et al., 2010]. In general, real ants have developed an efficient way of finding the shortest path from a food source to their nest without using visual information [Merz, 2000]. When the ants takes the path to a food source, they depose a specific substance called pheromone, which is a chemical substance that ants may lay down in varying quantities to mark a path. The ants establish communication exchanging information via pheromones [Dorigo et al., 2006]. If new ants arrive at a point where they have to decide on one or another path, the ants, in most cases, will follow the path where the pheromone intensity is higher. However, the insolated ants move essentially at random. Thus, the ants use the environment as a communication medium, this way of indirect communication is known in literature as *stigmergetic*.

The AAs obtain the solution by a probabilistic construction, which is based on the representation of the solutions in a graph. Hence, each edge of the graph represents a component of the global solution. In probabilistic construction, each artificial ant usually selects the move to expand the state taking into account two values: the *attractiveness*, which indicates the apriori desirability of that move and the *pheromone trail* level, which indicates how good the choice in the past has been [Dorigo & Stützle, 2003; Dorigo & Stützle, 2010]. In the context of the VRPs, the *attractiveness* value is computed by some heuristic that indicate a measure of desirability related with distance, traveled time or vehicle capacity exploitation.

Stimulated by both the attractive natural inspiration of real ants and the interest to solve combinatorial problems (such as VRPs) as efficiently as possible, several variants of AAs have been proposed. Some modifications of the first algorithms have lead to new remarkable variants of AAs, particularly for those modifications addressed to improve the intelligence of the search process and the ability to avoid becoming trapped in local optimum [Al-Ani, 2005]. The most studied variants of AAs in literature are listed below.

- Ant System (AS) [Dorigo et al, 1996].
- Ant Colony System (ACS) [Dorigo & Gambardella, 1997].
- MAX-MIN Ant System (MMAS) [Stützle & Hoos, 2000].
- Rank-Based Ant System (RBAS) [Bullnheimer et al, 1999].
- Best-Worst Ant System (BWAS) [Cordón et al, 2000].

In Cuba, the algorithms based on ACO have been subject of modification as well. The Two-Stage Ant Colony Optimization (TS-ACO) proposed by Gomez et al. [2008] and [Puris et al., 2010] is one of the main contribution to this field of swarm intelligence. Furthermore, in that sense a contribution of the present author is reported in Costa Salas et al. [2012], but this topic will be analyzed in detail in Chapter 4.

General Framework of Ant Colony Optimization

The basic mode to running ACO for any discrete combinatorial problem is described in Dorigo & Stützle [2003], Dorigo & Stützle [2004] and Dorigo et al. [2006]. Based on above researches, we summarize the main steps of the general framework followed in the ACO metaheuristic.

The ACO algorithms are essentially construction algorithms. Therefore, in every algorithm iteration, each ant constructs a solution to the problem by traveling on a construction graph. Each edge of the graph, representing the possible steps the ant can make, has associated two kinds of information that guide the ant movement:

- *Heuristic information*, which measures the heuristic preference of moving from node i to node j, of traveling the edge a_{ij}. It is denoted by η_{ij}. This information is not modified by the ants during the algorithm running.
- *Artificial pheromone trail information*, which measures the "learned desirability" of the movement and mimics the real pheromone that natural ants deposit. This information is modified during the execution of the algorithm depending on the solutions found by the ants. It is denoted by τ_{ij}.

ACO is structured into three main functions (see **Procedure 1**) [Dorigo & Stützle, 2004; Mullen et al., 2009]. AntSolutionsConstruct( ) performs the solution construction process as it is previously described. Artificial ants move through adjacent states of a problem according to a transition rule, iteratively building solutions. PheromoneUpdate( ) performs pheromone trail updates. This may involve updating the pheromone trails once complete solutions have been built, or updating after each iteration.

In addition to pheromone trail reinforcement, ACO also includes pheromone trail evaporation. Evaporation of the pheromone trials is included to help ants 'forget' bad solutions that were learned early on in the algorithm run. Implementation could be as simple as reducing all pheromone trials by a set amount after each epoch. DeamonActions( ) is an optional step in the algorithm which involves applying additional updates from a global perspective (there exists no natural counterpart). An example could be applying additional pheromone reinforcement to the best solution generated (known as offline pheromone trail update).

Procedure 1. The ACO metaheuristic

```
ParameterInitialisation
WHILE termination conditions not met do
  ScheduleActivities
    AntSolutionsConstruct()
    PheromoneUpdate()
    DeamonActions() optional
    END ScheduleActivities
END WHILE
```

The stop criterion (termination conditions) [Dorigo & Stützle, 2003] in ACO can be established as follows:

- According to the number of iteration.
- If some pre-defined value of objective function is reached.
- According to a deadline or pre-defined running time.
- When a maximum number of evaluation in the objective function is reached.

Based on the detailed analysis of ACO, we conclude that the algorithms based on Ant Colony Optimization consist of three main stages (repeated throughout iterations): Solution construct, evaluate the solution and deposition-updating of pheromone.

ACO with multiple colonies

Several improvements have been developed in ACO. The approaches of multiple colonies in ACO are one of the most efficient according to the literature for VRPs (e.g. for VRPTW reported in Gambardella et al. [1999]). In general, this approach is based on the cooperation between colonies, which is performed by exchanging information through pheromone updating. In the search process performed by each colony, the best experiences (best found solutions) are shared with other colonies. The AAs with multiple colonies have been addressed to large-scale of VRP [Aljanaby, 2010]. Despite the efficiency of this approach, the researches in this area of ACO have been poorly treated, particularly in VRP. Just few of them are involved in the application parallel programming, and some multi-colonies algorithms are used to optimize multiple objectives functions simultaneously. To our knowledge, in multiple colonies of AAs, each colony only deals with the components (parts of the solution) of the global solution, the colony do not represents a global solution, therefore, in the cooperation process the experiences about the components of the solutions are exchanging.

The Multiple Ant Colony System (MACS) introduced by Gambardella et al. [1999] is considered the first algorithmic proposal to the VRPs. Here, the MACS is proposed to the VRPTW, this algorithm (MACS-VRTW) is organized with a hierarchy of artificial ant colonies designed to successively optimize a multiple objective function: the first colony minimizes the number of vehicles (ACS-VEI*)* while the second colony minimizes the traveled distances (ACS-TIME). The two colonies evolve in parallel exchanging information when a better solution of number of vehicles is found. A similar approach to the above is recently reported in Ortega et al. [2009]. Moreover, in Gómez Díaz [2010] is examined an ACO multi-colonies for solving the Feature Selection Problem (FSP), the novel idea consisted in applying this approach in a context of multiple source of data.

The approaches based on multiple colonies, according to the present author, can be summarized into three strategies: the first is related with parallel programming, in the second the multi-objective approach is analyzed and the third the colonies cooperate exchanging meta-information (multiple source of data). Finally, according with the revised literature in this thesis, some possible variants of multiple colonies have not been treated, for example, considering each colony as a set of global solutions (one per each ant) of any undertaken VRP. Therefore, the exchanged information represents the acquired experience of each colony about the construction of the global solution, not about a single component of the global solution.

2.4 The Vehicle Routing Problem in Cuba

The transportation is considered of one the most important decision in logistics. In general, the distribution process implies one of the higher costs in the enterprises and its role in the customer service is crucial [Fink & Rothlauf, 2008]. In this sense, Tseng et al. [2005] argues that the transportation involves a considerable part of the logistics costs. Hence, the necessity to increase the effectiveness in the transportation process, where the VRPs constitute the main optimization decision, becomes a pivotal issue.

The optimization decisions are carried out considering limited resources, for example, financial, required time, raw materials, among others. The optimization scenarios can be harder in the context of a developing country, such as Cuba. Therefore, the ineffectiveness in decision-making is a luxury that such countries cannot afford. For that reason, the optimization process, especially the VRPs, should be supported on effective methods, which provide to the decision-makers with feasible solutions in reasonable time.

In Cuba, the transportation processes are carried out under the following characteristics: a much diversified fleet of vehicles [Faedo, 1999], the decrease of transport flow in the urban sector and the distribution of any load (e.g. goods and raw material for whatever industry or service facility). Although some investments have been developed, specifically in the railroad sector, the transportation infrastructure (pickup and delivery center, harbors, train stations, bus stations, network roads and airports) still requires larger investments, which is typical in many countries of the so-called third world. However, in this current scenario there exist some high-priority areas, for example, the primary industry, the transportation in tourism and the international aviation.

Moreover, some small investments have been performed to acquire vehicle fleets of high technology, which have guaranteed the supply and distribution in important new companies [CUBALSE, CIMEX S.A., ABATURITH and the National Union of Electricity (NUE)] in the country. On the contrary, the load truck services provided by the main company of the country, Truck Center of the Transport Ministry, have decreased significantly [Sendas, 2009].

The Cuban transportation process is developed almost entirely by petroleum-derived (fuel-oil or gasoline) [Johannesburgo, 2002]. That is why, the urgent need to use rationally this valuable and limited resource, through the proper exploitation of transportation means.

Transportation decisions in Cuba are made according to two levels of hierarchy [Johannesburg, 2002]. The first level involves the macro-economic decision of the transportation process, for instance, decisions of national importance (material movement of heavy industry, large-scale fuel distribution, passenger movement throughout the all country and so forth). The Council of Ministers is in charge of these decisions dealing with transportation. In the second level, those transportation decisions that take place in the province and its corresponding cities are analyzed. These decisions are made by the local authorities of each territory. For both levels of hierarchy, the decision-making is supported by the technical advisors of Cuban Transport Ministry. Recently, these technical advisors have examined the main inadequacies in the transportation process [MITRANS, 2008]. The most important are listed below.

1) The transportation routes and vehicle fleet are inefficiently planned; therefore the exploitation of fleet load capacity does not exceed the 50% in most cases.
2) In the urban transportation, there are empty return tours even when the bus stops are almost full.
3) There are unnecessary tours in managerial sector when the neighbor enterprises organize trips, one per each one, to other enterprises for some similar paper work.
4) The distribution supply are disused what bears to a great conglomeration in the dispatching centers.
5) The trip objectives are not achieved in the 40% of the cases, therefore fuel and engine recourses are consumed in vain.
6) The specialized transportation companies do not guarantee an efficient decision-making processes due to their solutions is mainly based on the experience, disregarding the current scientific techniques for such proposes.
7) Some goods, which were sent from the cities, should be collected in the main province warehouse and then should be returned to their origin places in order to be consumed.
8) There exist delays in the proper restoration of the essential services (electric service, waste collection and supply of raw material to rebuilding) after natural disaster, such as hurricanes.

Clearly, the mentioned inadequacies are the results of an empirical approach adopted by the decision-maker. In particular, the transportation process requires to be organized considering the inherent characteristics of the Cuban context, such as huge financial limitations, a reduced and heterogeneous fleet of vehicles and the hard environmental conditions due to natural disasters.

Despite these successes, some Cuban researchers have studied the transportation decision, devoting the major contributions to the control methodologies in transportation management. In this sense, most researches propose some control measure based on the well-known technology of Global Positioning System (GPS) [Del Valle, 2006]. In Cuba, the GPS technology is used *offline*, which provokes that route control process has to be performed after vehicle returns to the dispatching center. The control process based on the *offline* GPS technology does not allows to developed effective actions in the transportation management, only in the traveled route, which in most cases is planned disregarding the studied optimization algorithms, it can be checked when the vehicles end their journey.

Based on an extensive analysis of scientific literature, the present author concludes that most important contributions to transportation management (specifically in VRP) in Cuba are related with control methodologies (see descriptive analysis in Figure 2.6).

To date, the application of either exact or approximate algorithms to VRPs seems to be a pending task for the Cuban managerial organizations. According to our search, fewer solutions are found in this area (see Garza Ríos & Gonzáles Sánchez [2004] and López

Milán et al. [2004]), while much more are dedicated to established control measures in transportation process (see in MITRANS [1984], UDECAM [1999], Ruiz González [1999], CETRA [2001], MITRANS [2005], Resolución 249 de la UDECAM [2005] and Del Valle [2006].

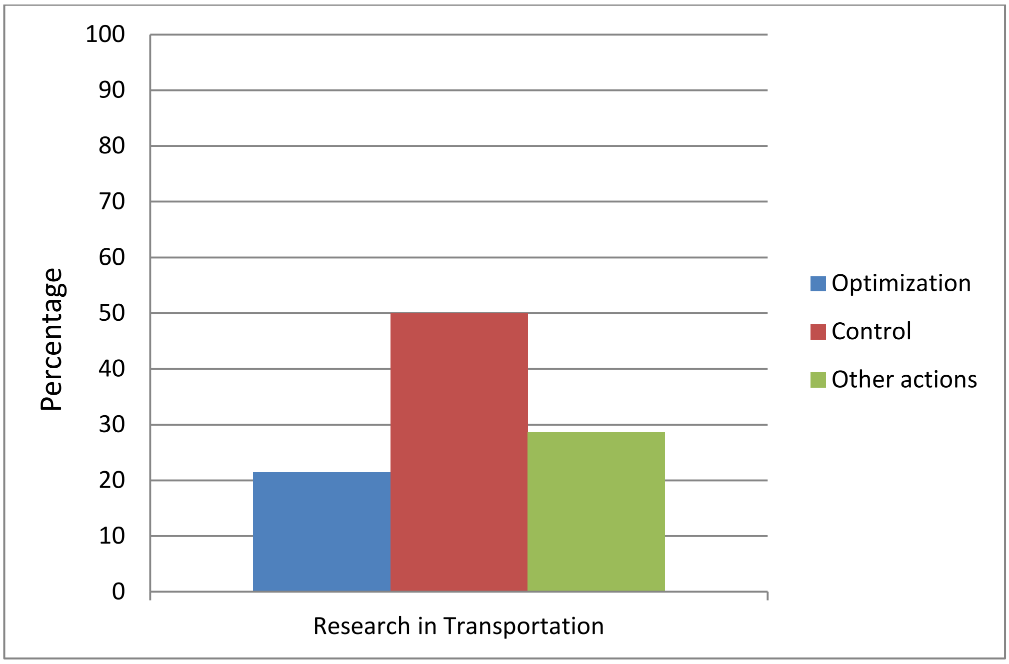


Figure 2.6: Descriptive analysis of the main VRP solution approaches in Cuba

The vehicle routing decisions, as most difficult decision in transportation management, should be supported, in most cases, by efficient mathematical and computational techniques, which the Cuban managerial organizations have not frequently applied. In this sense, the Cuban enterprises require of technical assistance in order to improve their performance in decision-making.

2.5 Summary

In this chapter, many definitions of optimization have been provided and then some critical analyses have been developed. Due to the importance for this research, we have studied the main methodologies to carry out the optimization process, identifying their inadequacies, both the univariate analysis and the necessity of the integrative approach. Moreover, we have examined the computational complexity in the optimization problems, which is recognized as the main analysis previous to the optimization process, while the sensitivity analysis becomes the fundamental issue in the post-optimization.

Furthermore, we have presented the main features of one of the most widely studied combinatorial problems, the Vehicle Routing Problem. Here, some well-know extensions of the VRP have been described. Recent researches have shown that the algorithmic approaches for solving the VRPs can be classified as either exact or approximate. Therefore, we have discussed the main algorithms for both optimization categories. In particular, the metaheuristics algorithms, especially ACO, have been described in the application context of the VRPs. Finally, we have analyzed the VRP solution approaches in Cuba, proving that the algorithmic approaches, studied in the present chapter, have not been often applied by Cuban decision makers.

Chapter 3

A conception and solution approach for assisting the optimization in VRPs

A priori it is often not obvious how optimization methods perform when are applied to specific Vehicle Routing Problems. That is why, they have to be evaluated and compared empirically in most cases. In this chapter a new conception of the optimization process is presented based on conceptual model, specifically for the VRPs. We propose this novel conception designing the optimization process in VRP according to three medullar and integrated moments: decision-making before optimization, during optimization and after to the optimization of any VRP. The second part of the chapter is dedicated to establish the procedure, which allows to apply the theoretical ideas proposed in the conceptual model through some computational implementation.

Some specific methodologies are also presented in sections of the chapter, in particular, to estimate the training-set size (based on Mukherjee et al. [2003]) in the Knowledge Base and to analyze the parameter sensitivity of any defined algorithm.

3.1 Conceptual model for optimization process in VRPs

The new conception of optimization process, according to the conceptual model (see Figure 3.1), is introduced in order to *increase the effectiveness in decision-making* based on an *integrative* and *proactive* approach. For such purposes, the conceptual model considers to structure the optimization process into three integrated stages (previous, during, and after optimization). Furthermore, a learning process is also considered, which determines the relevant optimization methods, both exact and approximate, for a given Vehicle Routing Problem. The learning process permits to predict these appropriate optimization methods based on two conditions: the internal conditions, which define the VRP complexity (number of nodes, fleet size, number of objectives, customer demand, fleet type and time windows), and the external conditions (decision hierarchy, decision scope and available time), which can affect the relevance degree provided by the learning process and therefore, they can have a significant influence in the decision-making effectiveness. The conceptual model is defined as a theoretical-methodological support to the decision-making in VRP optimization. Hence, the graphical representation of the

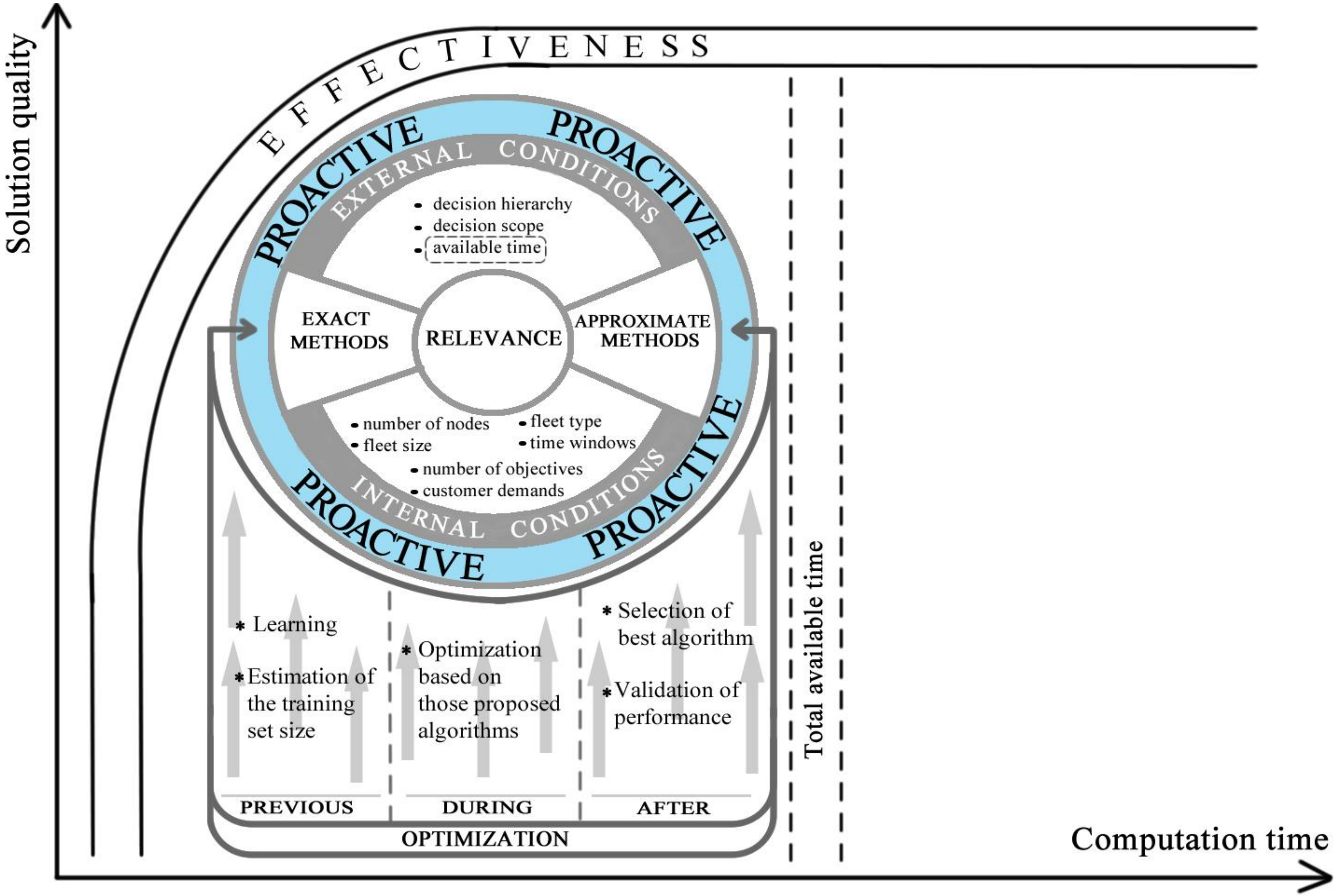


Figure 3.1: Conceptual model for the optimization process in the VRPs

optimization process is illustrated as classical solution space of an optimization problem, where the effectiveness measure is given by three performance indicators: solution quality, computation time and the real time in which the decision-making occurs.

Optimal solution, in our new conception of VRP optimization is reached, when the effectiveness takes the higher values according to the performance indicators. The trajectory in optimum pursuit starts with the integration of the three stages in the optimization. Then, other three derived processes (learning, algorithms proposal and validation) allow to guide the search to promising areas of the solution space, where exact or approximate optimization methods can be properly applied for the VRPs. Subsequently, if the method prediction considers both internal and external conditions, the highest values of effectiveness are achieved. Finally, two important *premises* should be considered when the conceptual model is conceived for any variant of VRP, we define these below.

1) Theoretical knowledge and practical experience in the issue concerned to combinatorial optimization problems, specifically VRP and its solutions methods.
2) For the implementation of the proposed conceptual model, we suggest that the users should possess the following resources:
    - Access to the VRP solutions which are reported in literature (mainly for learning process).
    - Acceptable computation capability for the algorithm runs and statistical tests.
    - Some VRP solutions require professional equipment, for instance, to establish the communication between the driver and dispatching center (e.g. the case study analyzed in Chapter 4), or any equipment of Geographical Information System (GIS).

When the optimization process in VRP is conceived according to the proposed conceptual model, some significant *potentialities* are provided to the decision-making. The most important can be stated as follows:

1) Gives a scientific-practical conception of the optimization process related with the VRPs, identifying its three required stages, and also considering a proactive approach in terms of effective and relevant optimization solutions.
2) Provides an integrative approach of the VRP decision-making for any managerial organization, conceiving the close relationship between the results of its stages (previous, during, and after optimization).
3) Includes a set of scientific tools, such as Mathematical Statistics and Machine Learning for Knowledge Discovery and fit test of random variables existing in optimization models. Furthermore, involves classical Operation Research (exact methods) and Artificial Intelligence algorithms (approximate methods) to solving the optimization problems.
4) Presents flexibility attributes since when the internal and external conditions change, all stages results and its derived process can be updated.

5) The planning and execution of the optimization process according to the proposed conceptual model is in accordance with the current logic followed in VRP decision-making.
6) The conception of VRP optimization process can be extended to other combinatorial optimization problems existing in logistics planning.

To achieve the main objective of the conceptual model in the optimization process the following *inputs* should be defined:

- List of possible variables which describe the feature of complexity.
- Expert judgment about inclusion or not of the aforementioned variables in the learning process.
- Domain values of each defined variable in the Knowledge Base (KB).
- The feature of complexity for any VRP instance (new case to classify).
- The total available time to carry out the Vehicle Routing Decision.
- Constant or data values of a given VRP instance.
- Discriminat characteristics of the specific optimization algorithms (within the exact and approximate category).
- Parameters values of the proposed optimization algorithms.

The above *inputs* are involved in the following *conceptual model processes*:

- Learning (Knowledge Discovery in VRP solutions).
- Estimation of the minimum training-set size in KB.
- Optimization based on those proposed algorithms.
- Selection of the best algorithm according to the performance indicators.
- Validation and simulation of the performance in all optimization stages.

Some of the main *outputs* after running the previous processes can be defined as follows:

- Value of the minimum training-set size in KB.
- Prediction of relevant optimization method for a given VRP instance.
- The best route, objective function value and the computation time of the proposed algorithms.
- The Global Index of Effectiveness (*GIE*) achieved after the simulation of performance.

As we mentioned before, the conceptual model introduces a new theoretical conception which requires to be implemented experimentally to a given VRP. In this sense, the next section of the current chapter presents the practical solution to this issue. A procedure is proposed as algorithmic instrumentation of the *conceptual model processes*.

3.2 Procedure for optimization process in the VRPs

The proposed procedure (see Figure 3.2) is structured into three stages. In the first one, a previous optimization analysis is performed, where the main idea is to define algorithmic steps that allow the relevant selection of the optimization method (either exact or approximate). The second consist of specific algorithm election based on two patterns: the remark to be applied of each algorithm and the results of classification, provided by the computational implementation developed for such propose in this chapter. Finally, in the third stage a comprehensive post-optimization analysis is carried out, which includes performance indicator analysis for the relevance validation and sensibility analysis of those parameters that can be defined by procedure users.

Stage I: *Pre-optimization in the VRP*

3.2.1 Learning process for the method selection in VRPs

In this stage, the main variable involved with the VRP complexity will be defined. These variables are used in the learning process in order to classify the relevant solution of a given VRP instance. Subsequently, a Knowledge Base is designed based on samples of the real-life VRP solutions. For classifying the VRP relevant solutions two well-known data meaning algorithms are introduced, they are the Discriminant Analysis and the C4.5. Then, the minimum training-set of the Knowledge Base is estimated when the solution categories (exact or approximate) are obtained by each classifier. Another important result of this stage is the computational implementation that includes both classification algorithms in the learning process. This implementation is a friendly interface for users who need to estimate relevant solution to any VRP instance.

3.2.1.1 Setting predictive variables in the learning process

As we mentioned in Chapter 1, some solution approaches indicate that exact methods should be used in small-scale VRP instances. On the contrary approximate algorithms should be applied to large-scale datasets. In both cases, the instance dimensionality is based on the number of nodes defined in the VRP graph, which demonstrated be imprecise when some recent researches have been used the exact algorithms with successfully results in not small-scale VRP problems (over 100 nodes in the graph). Therefore, the objective of this procedure step is to define other representative variables, which describe the problem complexity or dimensionality in a multivariate context.

The set of predictive variables is defined as the result of two decision rules: the first is based on an extensive literature review, where those variables suggested by Fernández [2006] emerged as the predominant in our experiment. Subsequently, in the second decision rule, the resulting variables (see Table 3.1) form literature are analyzed by a work-team (*wt*) which determines the final inclusion of predictive variables in the subsequent classification experiment.

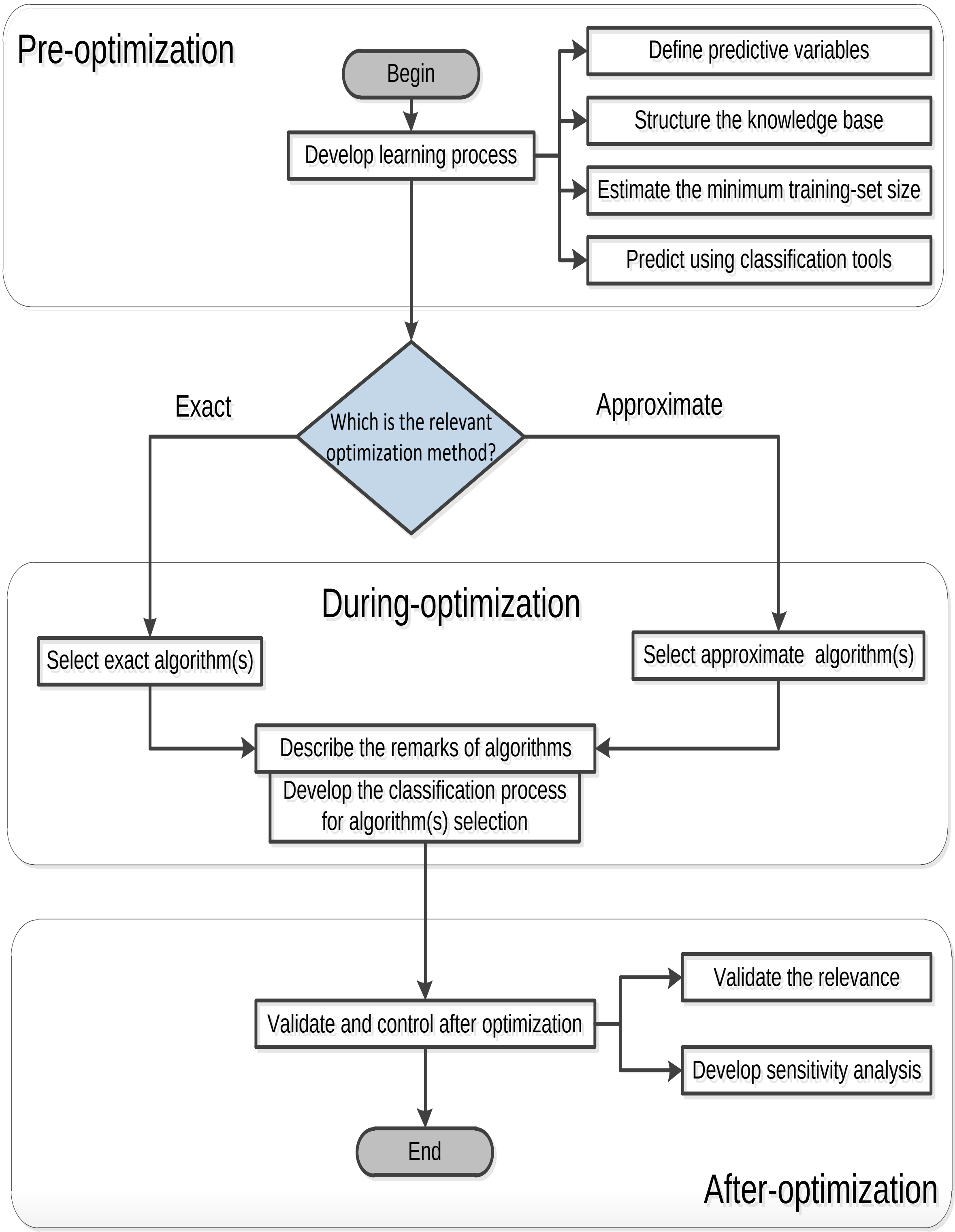


Figure 3.2: Procedure for assisting the decision-making in the VRPs

Based on our experiences with the work-teams and some other related scientific works in this area, the final inclusion of the predictive variables was the result of eight (8) expert judgments. The work-team members establish its criterion respect to include or not the predictive variables in the learning process.

Inevitably, when a set of opinions are given by persons, it is necessary to measure the achieved consensus in such judgments. In this sense, we introduce the Equation 3.1 which determines the Index of Consensus (IC) achieved in the definition of the predictive variables.

$$IC_i = \left(1 - \frac{SD_i}{SD_L}\right) \times 100 \qquad [3.1]$$

where IC_i is determined, regarding to the inclusion of each predictive variable i, by the ratio (see Table 3.1) between the standard deviation of expert judgments for each variable i (SD_i), and the highest possible value of standard deviation (SD_L) according to the scale used in the experiment (see Table 3.2).

Table 3.1: Expert judgments in the inclusion of predictive variables

	Work-team (*wt*)									
Predictive variables	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Average**	$\boldsymbol{IC}$
Number of nodes	5	5	5	5	5	5	5	5	5.000	100
Fleet size	4	4	4	4	4	4	4	5	4.125	86.77
Number of objectives	5	5	5	5	5	5	5	5	5.000	100.00
Time windows	4	4	4	4	3	4	4	4	3.875	86.77
Customer demand	5	5	4	5	5	5	5	5	4.875	86.77
Fleet type	5	5	5	4	5	5	5	5	4.875	86.77

The criterion for variable inclusion is set on a discrete scale from 1 to 5, where 1 indicates the minimum degree of agreement with the inclusion of the analyzed variable, and the value 5 indicates otherwise.

Table 3.2: The highest possible value of standard deviation for the defined scale

wt size	7	8	9	10	11	12	13	14	15
SD_L	2,673	2,673	2,635	2,635	2,611	2,611	2,594	2,594	2,582

From the figures of Table 3.1, it is clearly understandable that all the analyzed variables can be included in the learning process, since the Index of Consensus exceed the 85% in all predictable variables and the average values of expert judgments exceed the significant number of 3.75 in all variables. The acceptance of predictive variables has been the result of successfully empirical studies in other Cuban doctoral thesis (see Abreu Ledón [2004]).

3.2.1.2 Structure of the Knowledge Base

As we discussed in previous sections, each case[3] in Knowledge Base (KB) represents a real-life solution given to any variant of VRP. The Appendix A.4 shows the formal structure of KB, where the optimization methods, predictive variables and the respective reference of every case are reported. The optimization methods (exact or approximate) and specific algorithms applied to every sample is well proved through mathematical and/or statistical test. In most of cases the VRP solution relevance is checked by solution quality and computation time as performance indicators.

To assist the decision-making regarding KB data management, classification quality and graphical outputs a computational implementation is developed in this section. The *VRP solution classifier* (see the main interface in the Figure 3.3) is encoded in *java* and its computational requirements are quite reasonable. In the Appendix A.5 the software utilities are deeply described. However in this section those utilities related with the data-information management will be explained in details.

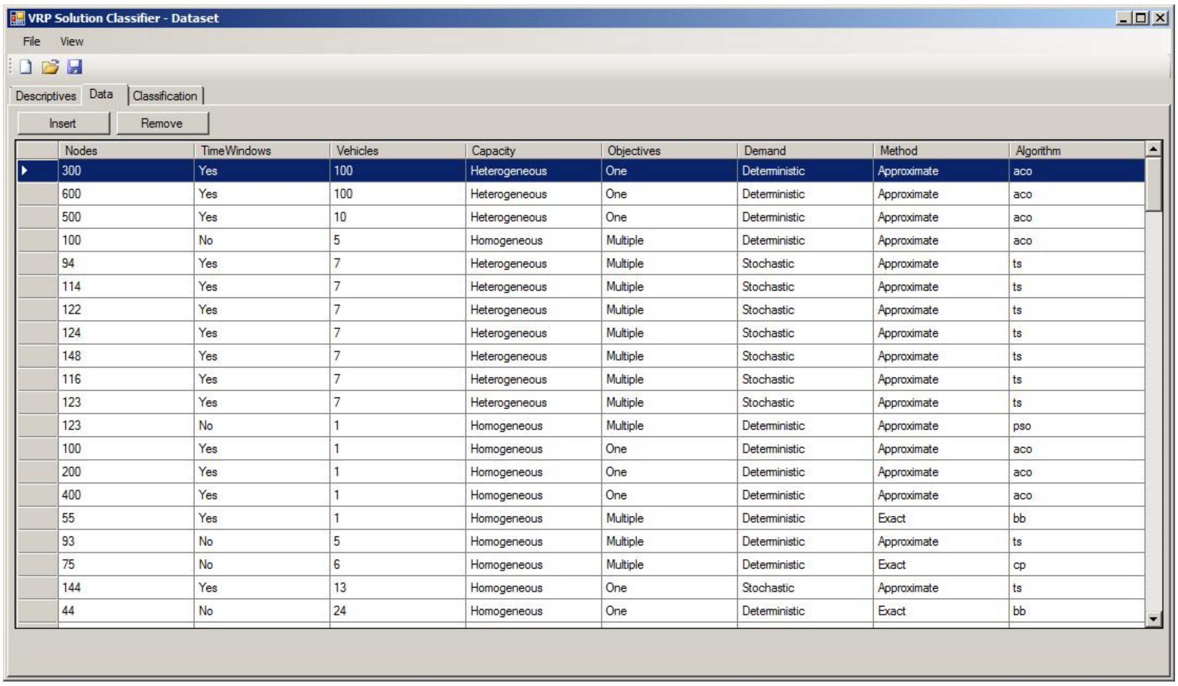
Figure 3.3: Main interface of the software VRP solution classifier

[3] In some sections of this thesis is called sample, includes the classification variable (optimization method) and its respective predictive variables.

In fact, the first module of *VRP solution classifier* is closely related with what we mentioned above. This module is called *Descriptives-Data* module (see Figure 3.4), which provides the following utilities:

- Add or delete any sample of the Knowledge Base.
- Modify any predictive or classification variables of each sample.
- Descriptive analysis of all variables separately, either continuous as discrete variables.

The KB reported in Appendix A.4 consists of 42 exact VRP solutions and 125 belong to the approximate optimization category. Regarding to predictive variables (see predictive variable domain in Table 3.3), particularly in the customer demand, 130 cases can be defined with deterministic demand, while 37 customer demands are reported stochastic in KB.



Nodes	TimeWindows	Vehicles	Capacity	Objectives	Demand	Method	Algorithm
300	Yes	100	Heterogeneous	One	Deterministic	Approximate	aco
600	Yes	100	Heterogeneous	One	Deterministic	Approximate	aco
500	Yes	10	Heterogeneous	One	Deterministic	Approximate	aco
100	No	5	Homogeneous	Multiple	Deterministic	Approximate	aco
94	Yes	7	Heterogeneous	Multiple	Stochastic	Approximate	ts
114	Yes	7	Heterogeneous	Multiple	Stochastic	Approximate	ts
122	Yes	7	Heterogeneous	Multiple	Stochastic	Approximate	ts
124	Yes	7	Heterogeneous	Multiple	Stochastic	Approximate	ts
148	Yes	7	Heterogeneous	Multiple	Stochastic	Approximate	ts
116	Yes	7	Heterogeneous	Multiple	Stochastic	Approximate	ts
123	Yes	7	Heterogeneous	Multiple	Stochastic	Approximate	ts
123	No	1	Homogeneous	Multiple	Deterministic	Approximate	pso
100	Yes	1	Homogeneous	One	Deterministic	Approximate	aco
200	Yes	1	Homogeneous	One	Deterministic	Approximate	aco
400	Yes	1	Homogeneous	One	Deterministic	Approximate	aco
55	Yes	1	Homogeneous	Multiple	Deterministic	Exact	bb
93	No	5	Homogeneous	Multiple	Deterministic	Approximate	ts
75	No	6	Homogeneous	Multiple	Deterministic	Exact	cp
144	Yes	13	Homogeneous	One	Stochastic	Approximate	ts
44	No	24	Homogeneous	One	Deterministic	Exact	bb

Figure 3.4: The Descriptives-Data module in VRP solution classifier

On the other hand, in KB are registered 92 VRP cases where the time windows were solved, on the contrary 75 does not considered these constrains. In general, 78 become of multiobjective approach, while 89 researches only considered one objective in the VRP instance. The fleet type is composed by 100 real-life solution with homogeneous fleet of vehicles and 67 otherwise. The number of nodes in the graph and the fleet size is examined in all samples of the KB, showing a proper distribution between large, medium and small-scale of VRP instance.

In summary, the learning process is based on a Knowledge Base composed by 167 cases, which are the result of many important contribution reported in literature. However, a priori is difficult to assure that KB size is large enough in order to carry out a proper prediction of the optimization method. Therefore, the size of the KB, in particular the training-set size should be verified before any prediction of solution category. In this sense, the next section the methodology is examined, which allows estimating the minimum training-set size according to the error prediction.

3.2.1.3 Estimating the minimum training-set size in the KB

In this section we present a validation process in the Knowledge Base, specifically to estimate the adequate size of the training-set. This estimation is unquestionably necessary due to its direct influence in the classification quality. The KB is always divided into two basic sets: the training-set, which is used to train a given classifier and the validation-set, where the classifier predictions are compared with the real values of this mentioned set. Then, the classification accuracy is determined based on the above comparison. Intuitively, the prediction accuracy depends on the classifier type and the training-set used in the learning process.

As shown before in this stage, we introduce the Discriminant Analysis (DA) and the C4.5 algorithms to classify the relevant solution of VRP instances. Therefore, the estimation of training-set is based on these classifiers, even when they are not described in this section (that will take place in the next section).

The problem of estimating minimum training-set, according to the performance of both defined classifiers, is solved based on the methodology reported in Mukherjee et al. [2003]. In this methodology (see Appendix A.6), the training-set size is studied as a function of classification accuracy by building empirical scaling models called *learning curves*. The proposed methodology applies learning curves to estimate the empirical error rate as a function of training-set size for a given classifier and dataset (see Equation 3.2). Usually, the *learning curves* are characterized by inverse power-laws:

$$e(n) = an^{-\alpha} + b \qquad [3.2]$$

where $e(n)$ is the expected error rate given by n samples. The variable a denotes the learning rate, a represents the decay rate and the Bayes error b which indicates the minimum error rate achievable.

These *learning curves* have been developed for other application context, such as several cancer classification problems in *Bioinformatics* field [Mukherjee et al., 2003]. In fact, the *Support Vector Machine* [Vapnik, 1998] is used as the classifier. However, the classifiers DA and C4.5 are not found in training-set size estimation, which is considered as a practical contribution of this thesis.

The training-set sizes and all variables described by Equation 3.2 are estimated according to the selected methodology (see Appendix A.6). As we mentioned before, the estimation uses the DA and C4.5 as the classifiers that are trained with 15, 30, 45, 60, 75, 90, 105, 120, 150, 180 sample sizes of our KB (see Appendix A.7). The Figure 3.5 shows the learning curves obtained by applying both classifiers with the above sample sizes.

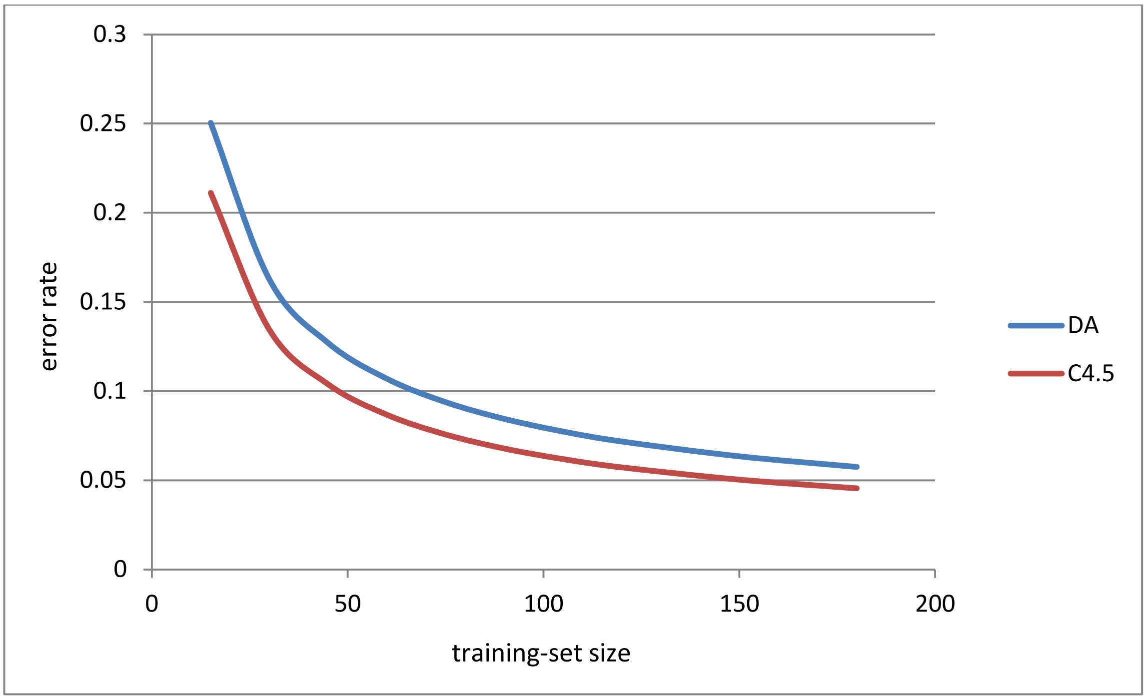


Figure 3.5: Learning curves obtained with DA and C4.5 classifiers

From the results achieved in Appendix A.7 and *learning curves* shown in Figure 3.5, we concludes that KB sample size is large enough to conform training-set, which allows an accurate learning process. Considering an achievable error rate of 10% at most, the minimum training-set required when DA is used should be superior to the 105 cases. Furthermore, under the same achievable error rate value and sample sizes, the C4.5 requires less training-set size (superior to the 60 cases).

3.2.1.4 Classification tools for predicting solutions in VRP

The classification process presented in this section is supported by two well applied algorithms in data mining, Discriminat Analysis and C4.5. Both algorithms are used to predict the relevant solution of a given VRP instance. The prediction is concerned with a set of independent variables (predictive variables) which characterize the instance complexity. In the Table 3.3 the classification variable are shown and the independent variables defined for both classifiers. These variables were selected in previous section, although Table 3.3 presents the variables in a classification context including the domain values of each one.

Algorithms election was based on the classification problem size (sample size) and the number of classes (two optimization categories in our problem). Furthermore, the proposed classification algorithms have been reported as the top 10 algorithms in data mining [Wu et al., 2008]. From Artificial Intelligence techniques very efficient algorithms for large-scale classification problems are introduced, such as *Artificial Neural Networks* examined in Ripley [2008].

Table 3.3: The set of variables defined in classification process

Variables	Type of variable	Domain values
Optimization method (Y)	Dependent	Exact / Aproximate
Number of nodes (X_1)	Independent	Discrete-finite
Time windows (X_2)	Independent	Without / With - time windows
Fleet size (X_3)	Independent	Discrete-finite
Fleet type (X_4)	Independent	Homogeneous / Heterogeneous
Number of objectives (X_5)	Independent	One-objective / Multi-objective
Customer demand (X_6)	Independent	Deterministic / Stochastic

Discriminant Analysis

The Discriminant Analysis (DA) is the most commonly used statistical technique to solve classification problem. Here, the objective of the analysis is to provide a tool for predicting which optimization method of a new case is most likely to fall into, based on a set of useful predictive variables.

The concept of DA involves forming linear combinations of independent (predictor) variables (all of Table 3.3), which becomes the basis for category classifications. The linear combination of predictive variables is given by Equation 3.3. Thus, the relevant optimization category according to DA can be determine as

$$Y = b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot x_3 + b_4 \cdot x_4 + b_5 \cdot x_5 + b_6 \cdot x_6 \qquad [3.3]$$

So, given a set of values for the predictive variables ($x_1 \dots x_6$) and its respective weight ($b_1 \dots b_6$), we can determine the linear combination function of dependent variable (Y) capable to discriminate between the optimization categories (exact or approximate methods).

For the Discriminat function, we define three statisticians for the definitive variable inclusion (*stepwise*). The statistician F based on the *Wilks´ lambda*, the *Mahalanobis Distance* and the minimum F*-ratio*. These statistical expressions are explained below.

- **Statistician *F* based on *Wilks´ lambda* (λ):** this statistician can be used as stepwise in the multiple regressions, in which a hierarchy inclusion is established for the independent variables. The statistical expression of F is defined by

$$F = \left(\frac{n-g-p}{g-1}\right) \cdot \left(\frac{1-\frac{\lambda_{p+1}}{\lambda_p}}{\lambda_p}\right) \qquad [3.4]$$

where n denotes the total number of cases over all the categories (groups). The value g indicates the number of groups, which are two in our experiment (exact and approximate), and the term p is given by the number of discriminating variables in the experiment. λ_p and λ_{p+1} represent the *Wilks´ lambda* values before and after of variable inclusion respectively.

- **Statistician *H* based on Mahalanobis Distance (MD):** In this case, at each step those predictive variables are incorporated, which maximize the MD [Mahalanobis, 1936] between the closest groups. Then, the multivariate distance between e (exact) and a (approximate) can be determined as

$$H_{ab}^2 = (n-g)\cdot\sum_{i=1}^{p}\sum_{j=1}^{p} w_{ij}{}^{*}\cdot\left(\bar{x}_i^{(e)}-\bar{x}_i^{(a)}\right)\cdot\left(\bar{x}_j^{(e)}-\bar{x}_j^{(a)}\right) \qquad [3.5]$$

where $w_{ij}{}^{*}$ is a value of inverse variance-covariance matrix intra-groups. The variables $\bar{x}_i^{(e)}$ and $\bar{x}_i^{(a)}$ denote the mean of group e and a in the *i-th* independent variable, while $\bar{x}_j^{(e)}$ and $\bar{x}_j^{(a)}$ represent the either the mean for each respective group e and a in the *j-th* predictive variable.

- **Minimum *F*-ratio:** The variable inclusion is based on the minimum ratio value of *F* (see Equation 3.6) for both groups (e and a). In fact, this statistician consists on weighting the Mahalanobis Distance according to the group size, n_1 denotes the size of group e and n_2 denotes the size of group a.

$$F = \frac{(n-p-1)+(n_1\cdot n_2)}{p\cdot(n-2)+(n_1\cdot n_2)}\cdot H_{ab}^2 \qquad [3.6]$$

As can be seen, in this stage the variable inclusion is given by the expert judgment and previous statisticians. We consider necessary the inclusion of variables according the expert judgment, since they introduce diversity in the classification process. The diversity consists on the introduction of predictive variable, which cannot be identified by any *stepwise* process.

The classifier C4.5

Classification algorithms expressed as decision tree are widely studied in literature. The C4.5 [Quinlan et al., 1993] is a descendant of CLS [Hunt et al., 1966] and ID3 [Quinlan, 1979]. Furthermore, is identified as one of the most influential that has been widely used in the data mining community [Wu et al., 2008].

In this research the C4.5 is implemented in order to carry out the classification process. As we mentioned before, the previous optimization stage requires of classification algorithms that allow predicting relevant solutions according to a set of complexity characteristics. In this sense, the C4.5 offers the classification solution in a friendly and much understandable way (decision tree). The decision tree can be easily interpreted, although not so trivial to construct. In the Pseudocode 3.1 the decision tree construction is described in detail. Then, the main equations to compute the tree are given below.

Pseudocode 3.1: Decision tree construction in the C4.5 classifier

Inputs of the C4.5 classifier

A: set of attributes (independent variables)
V: set of possible values in the domain of attributes
S: training-set according A, V
N: root node in the decision tree

```
C4.5 [S, A, (N)]
If (A ≠ ∅) or all cases in S belongs to the same class Then
  class-node(N) = class-majority(S)
Else
  A_i = best-attribute(A) [see Equation 3.7]
  N ← A_i
  For each v in A_i
    B = create-node(A_i, v)
    Branch(N) = Branch(N) + B
    S_i = {s ∈ S|value (e, A_i) = v}
    C4.5 [S_i, A – {A_i}, B]
Return N
```

The best-attribute is determined according to Equation 3.7, which is called the *GainRatio* measure (GR).

$$GR(A_i) = \frac{G(S, A_i)}{-\sum_{i=1}^{c} \frac{|S_i|}{|S|} \cdot log_2 \frac{|S_i|}{|S|}} \qquad [3.7]$$

where $GR(A_i)$ denotes de *GainRatio* of attribute "A_i". The *Gain* of attribute "A_i" is defined as $G(S, A_i)$. Here, S_i represents the number of cases belonging to the class "i", which has "c" possible values, and S the sample size of the training-set.

The *Gain* of one particular attribute "A_i" is given by the Equation 3.8, which is based on the expected reduction in *entropy* generated by the attributes. More precisely, the information gain, $G(S, A_i)$ of *an* attribute A**,** relative to a collection of cases S**,** is defined as

$$G(S, A_i) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \cdot Entropy|S_v| \qquad [3.8]$$

where $Values(A)$ is the set of all possible values for attribute A, and S_v, is the subset of S for which attribute A has value v. The first term in Equation 3.8 is called the *entropy* of the original collection S. It can be computed as

$$Entropy(S) = \sum_{i=1}^{c} -p_i \cdot log_2 p_i \qquad [3.9]$$

where p_i is the proportion of S belonging to class "i". Note that the logarithm is still base 2 because entropy is a measure of the expected encoding length measured in *bits*. Note also that if the target attribute can take on c possible values, the entropy can be as large as $\log_2 c$.

Computational implementation of proposed classifiers

As can be seen, in the current stage (previous optimization) two efficient classifiers have been introduced: Discriminat Analysis and C4.5. The classifiers are implemented in the *Classification* module of the *VRP solution classifier*. Classification results in the mentioned computational module (see Figure 3.6) are shown in both analytical (see text box underlined in red) and graphical (decision tree) form.

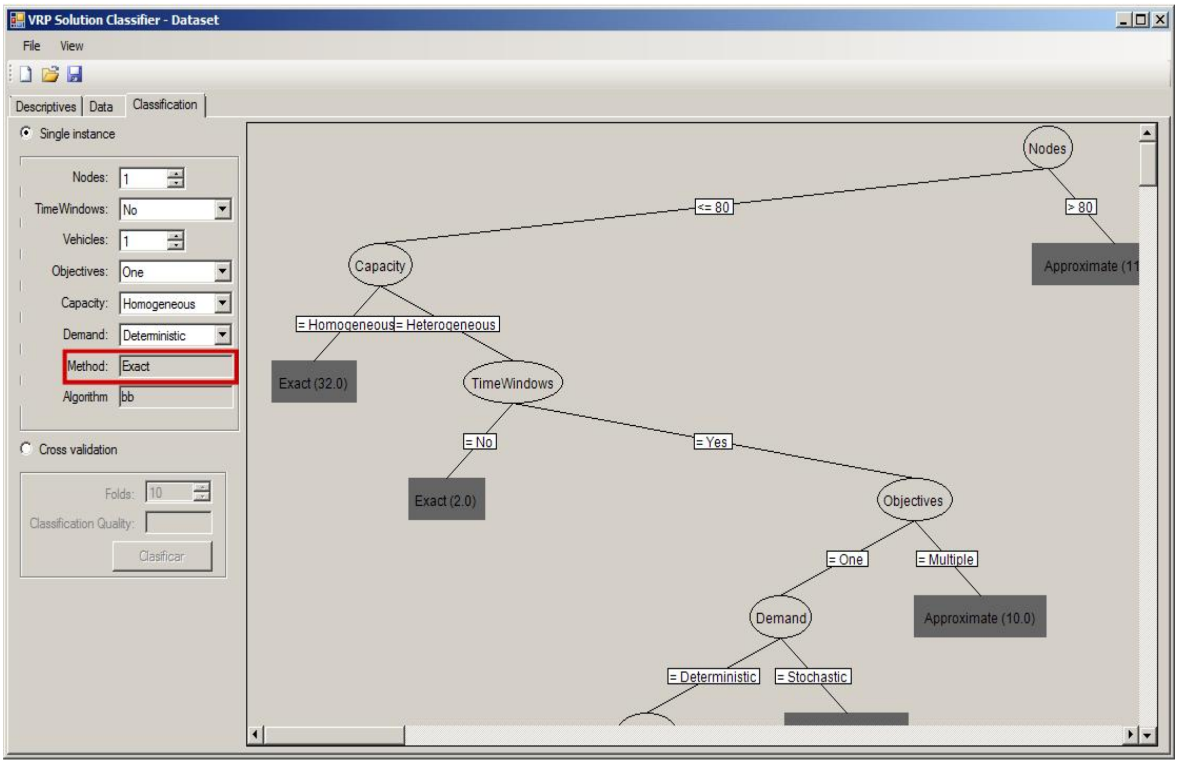


Figure 3.6: View of the Classification module in VRP solution classifier

The definitive classification result in both forms is taken from the most accurate classifier. This, intuitively, means that internally the computational implementation has encoded the described classifiers. However, the displayed results are obtained by the classifier with best *classification quality* (see Figure 3.7). Therefore, the value of *classification quality* displayed in module (see text box underlined in blue) becomes the best result achieved by classifiers.

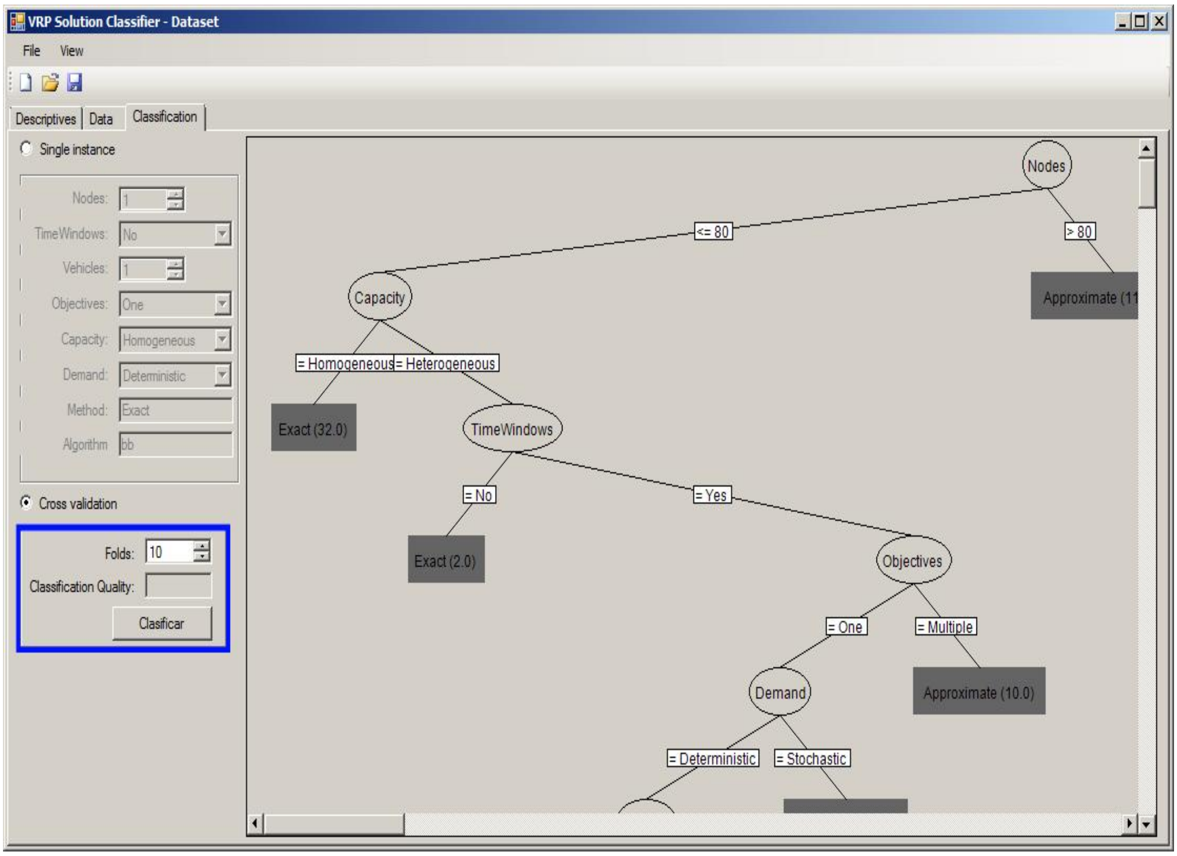


Figure 3.7: The classification quality in VRP solution classifier

The *classification quality* is estimated according to the *cross-validation* method (see Figure 3.7). Its mathematical expressions and the method steps are described as follows:

1) Split the Knowledge Base into approximately "k" folds.
2) Select randomly one of the k-th folds and then use it as validation-set. The remaining folds are used as training-set.
3) Compute the prediction errors (PE) according to Equation 3.10, for all k-th randomly, selected as validation-set in the former step.

$$PE = Prob(Y \neq \hat{Y}) \qquad [3.10]$$

4) Determine the average of prediction errors between all defined k-th folds and finally compute the *classification quality* (CQ) based on the Equation 3.11.

$$CQ = (1 - \overline{PE}) \times 100\% \qquad [3.11]$$

In summary, the first stage of the proposed procedure allows to predict the appropriate optimization method, whether as exact or approximate, according to the main characteristics of the previously defined complexity in VRPs. The creation of KB is essential in this stage, facilitating the training of two classifiers for an efficient prediction of optimization method.

Stage II**:** *During-optimization in the VRP*

3.2.2 Optimization methods in the VRPs

In this stage the main algorithms within the categories (exact and approximate) of optimization methods for the VRPs are described and analyzed. The aim in this procedure stage consists on supporting the decision-making related with the selection of the specific algorithms for the optimization process. As we mentioned before, it is quite difficult the *a priori* selection of a specific algorithm for solving the VRP variants. Hundreds of researches have been addressed to prove the individual efficiency of the algorithms to solve VRP variants. For such reasons, is either difficult to establish accurately the most efficient algorithm for each given variant of VRPs. However, some decision rule can be defined, with acceptable precision, in order to guide the decision-making for such purposes. In this sense, the current stage suggests supporting the described issue according to two approaches, the *remarks to be applied* a particular algorithm and the algorithm *classification results*. We analyze both approaches in the present stage, first describing the algorithms and its remarks to be applied within each optimization category (exact and approximate), and then, another classification process is proposed, but in case the dependent variable denoted the specific algorithms.

3.2.2.1 Selecting exact algorithms for VRPs

In this section we present the description of some well-know exact algorithms to the VRPs, as well as the main remarks to be applied such algorithms. After the prediction of optimization method is necessary to choose the proper algorithm(s) within each category of the optimization methods. Therefore, we analyze some algorithms (see Table 3.4) belonging to the exact category, describing how they can be applied to VRPs and some discriminant characteristics that support the selection of algorithm(s).

Table 3.4: Some exact algorithms for the VRPs

Linear Integer Programming (LIP)	
Algorithm description	Remarks to be applied
▪ Decision variables mostly represent binary values, where 1 denotes that the arcs are included in the route a value 0 otherwise ▪ One objective function is established in order to visit all nodes with minimum traveled distance ▪ In general can be used to modeling the most of VRP variants (TSP, CVRP, VRPTW, among others)	▪ Linearity in the model functions: objective functions and constrains ▪ Determinism in the components of the problem ▪ Is difficult to apply to dynamic variants of VRPs

Dynamic Programming (DP)	
▪ Solve the VRP variants by a sequence of decisions using dummy resources ▪ The algorithm assigns states to each vertex and each state includes a resource consumption ▪ The algorithm repeatedly extends each state to generate new states. ▪ The states generation must consider the constraints of the studied VRP variant.	▪ The Optimality principle: For any subsequence of the optimal sequence in the global problem should be either optimal in its associated sub-problem ▪ Is mostly used when the fleet size is fixed ▪ Its efficiency depends on the reduction in the state number

As we explained in Chapter 2, the first exact approaches are based on the enumeration of the full solution space. However, in order to increase efficiency, all modern exact methods use pruning rules to discard parts of the search space in which the (optimal) solution cannot be found. These approaches are doing an *implicit enumeration* of the search space. In the Table 3.5 we present a good example of this, the well-know Branch and Bound (BB) algorithm.

Table 3.5: The most used exact algorithm for VRP extensions

Branch and Bound (BB)	
Algorithm description	Remarks to be applied
▪ The algorithm deals with the solution space as a tree, where the search is doing by branching ▪ In the branching process the optimality of each branch is analyzed, bounding those where the optimality is unreachable	▪ For some variant of VRPs with hard constraints, are required relaxation functions ▪ The defined tree must contain as levels as possible routes ▪ Sometimes it is difficult to establish the lower bound: Initial solution ▪ Also sometime it is difficult to determine the upper bound: feasible and optimal solutions

Certainly, more than one exact algorithm can be applied to a given VRP variant. In fact, there are some cases where it is difficult to discriminate within the exact optimization category. That is why, the application of more than one algorithm, when the conditions permit, results much suitable.

3.2.2.2 Selecting approximate algorithms for VRPs

Similarly to the above mentioned, in this section we describe the main approximate algorithms used in VRPs. Here, three subcategories within approximate method are analyzed separately: the classical *heuristics*, *metaheuristics*, and *error and trials* techniques. We start with the description and remarks to be applied of heuristics (see Table 3.6).

Table 3.6: The classical heuristic algorithms in VRPs

Algorithm description	Remarks to be applied
Clarke and Wright Saving algorithm	
▪ It is based in the successive subtour combination according to a saving function ▪ The algorithm consists on a constructive heuristic in which the solution are computed and stored considering the shortest distances between all pairs of demand points	▪ The performance succeed better when the number of routes is minimum ▪ Can takes considerable time depending of the computation of maximum saving value (complexity of saving function)
Nearest Neighbor (NN)	
▪ Consists of the successive insertion of the nearest customers ▪ The tours, then, are constructed choosing one after the other the customer ▪ This procedure is iterated until all the customers have been served	▪ Sometimes the modeling of closeness is not easy to define ▪ Can be encoded in a few lines ▪ Mostly, at the end of the construction process long arcs have to be added to the tour
Local Search (LS)	
▪ It is based on the iterative exploration of neighborhoods of solutions, trying to improve the current solution by local changes ▪ The type of local changes that may be applied to a solution is defined by a neighborhood structure	▪ Requires of a good initial solution ▪ Its performance depends of an appropriate neighborhood structure ▪ The obtained solutions are -by definition- only local optima

As can be seen in Chapter 2, the common disadvantage of heuristics algorithms is that they cannot escape from local optima. In this sense, metaheuristic algorithms have been proposed in order to guide the described heuristics towards better global solutions. Here, we start with the most popular bioinspired metaheuristics, according to the present author (see Table 3.7). However, we defined some other that can be predicted as a proper algorithm by the classification process described in the next section.

Table 3.7: Prediction analysis of some bioinspired metaheuristics

Genetic Algorithms (GA)	
Algorithm description	Remarks to be applied
▪ Generate initial population of solutions (chromosomes) ▪ Modify the random solutions through evolutionary operators: the selection, crossover and mutation ▪ Define fitness to rank the solutions ▪ Repeat the generation runs till a stop criterion is reached	▪ In some combinatorial problems it is necessary to define a particular representation of the solution into chromosome (e.g. PMX crossover operator in VRP) ▪ The algorithm performance depends largely on how good (solution quality) were the solution in the initial population ▪ The fitness is not trivial to define in some cases (e.g. in multi-objective approach)
Ant Colony Optimization (ACO)	
▪ In general, the solutions are constructed probabilistically preferring to use solution components with a high pheromone trail and a promising heuristic information ▪ The pheromone trails are associated components (e.g. the arcs connecting the cities in VRPs) ▪ In most of algorithms of ACO the pheromone is locally and globally updated	▪ The algorithm performance can be improved when the initial pheromone value is not obtained by random generation (e.g. using the NN heuristic) ▪ Sometime, the pheromone updating mechanism lead to a easy solution trap into local optimum (mainly the multi-colonies approach can avoid this) ▪ In most of ACO´s variants, the erroneous definition of the parameters reduces significantly the algorithm effectiveness

Another efficient group of metaheuristics classified as the trajectory methods is analyzed in this section. This group involves the efficient Tabu Search (see Table 3.8) and the Simulated Annealing [José, 2009; Lin et al., 2009]. The important distinction of these metaheuristics compared with those from the Table 3.7, is whether they follow one single search trajectory corresponding to a closed walk on the neighborhood graph or large jumps in the neighborhood graph are allowed.

Table 3.8: Analysis of Tabu Search in algorithm selection

Tabu Search (TS)	
Algorithm description	Remarks to be applied
▪ Is based on the systematic use of a memory to guide the search process ▪ Typically uses an aggressive local search that in each step tries to make the best possible move from current to a neighboring solution next ▪ A solution that has been recently visited is included in a tabu list and therefore will not be considered as a candidate for the next solution to visit	▪ The strategy of neighborhood search depends on the VRP variant, in some cases it is difficult to define ▪ The maintenance of the tabu list and the searching within the list is often too time consuming to be practical ▪ The tabu conditions may be too restrictive and they may forbid moving to attractive, unvisited solutions

Finally, we show some remarks related with other widely used techniques in practical application context of VRPs, the Simulation Models (see Table 3.9). The simulation techniques are mostly used when the set of possible solution alternatives is considerably small, due to the significant resources (financial and humans) engaged in each possible solution. Therefore, the solution space should be reduced by mean of other techniques before using simulation. When the solution space is reduced, some promising areas are eliminated, therefore, the optimum value is difficult to achieve.

Table 3.9: Some remarks of the Simulation Models

Simulation techniques	
Technique description	Remarks to be applied
▪ Consist of the real-world imitation process over artificial systems ▪ Mostly, each alternative solution implies a simulation scenario ▪ The model boundaries have to be defined in the artificial systems ▪ The similarity between the real system and the simulation model should be proved	▪ Simulations of large system are limited by this sequentiality, since a modest number of events can be simulated ▪ Solutions require of substantial humans and computational recourses ▪ Can solve real problems where the analytical solutions have not been found yet ▪ Can be used for those problems where the solution based on real experimentation is quite expensive

In the previous two sections many well-know algorithms belonging to the optimization categories studied in this thesis were described. However, others have not been analyzed in the context of "algorithm selection" developed in this stage. Despite this fact, we introduce in the next section a learning process, in which the new classification variable is expressed by the most specific algorithms of both optimization categories (exact and approximate).

3.2.2.3 Learning for supporting the selection of algorithms

To date, finding the best algorithm for any VRP instances is often a difficult task to carry out. If we search an accurate prediction of the best algorithm for a given VRP dataset, the performance values of all the existing algorithms should compared on an exhaustive statistical experiment. However, some good experiences can be adopted to guide such decision-making considering a reasonable accuracy value. In this sense, we propose to assist the decision-making developing a learning process, which considers the same independent variables described in Table 3.3. The main difference regarding the previous learning process lies in the classification variable (dependent variable). In this case, the dependent variable denotes the algorithm with best performance, which means adding to each case of Knowledge Base its corresponding algorithm.

The prediction process of "proper algorithm" is implemented in the *Classification* module of the *VRP solution classifier* (see the classification result in Figure 3.8). Consistently, with its category of optimization method (exact or approximate), the "proper algorithm" is predicted for any new case of VRP.

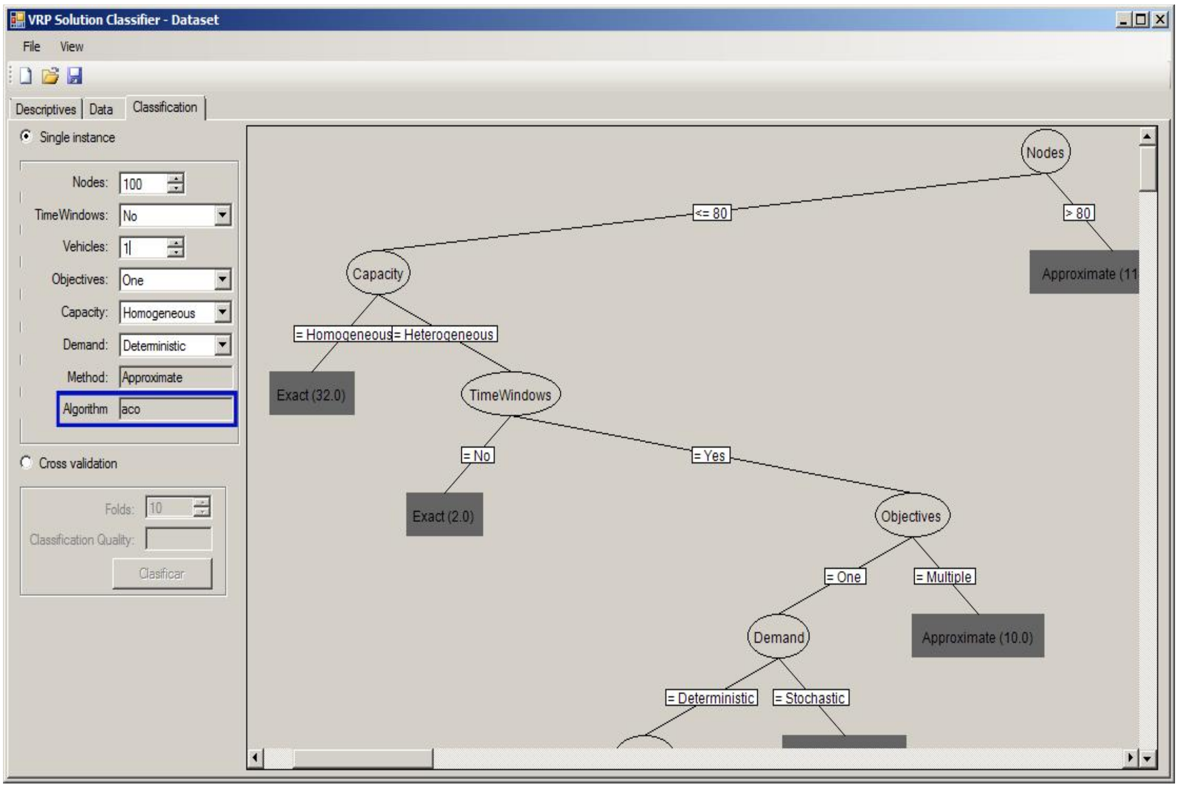


Figure 3.8: Predicting "proper algorithms" in the VRP solution classifier

To conclude, during the optimization process it can be selected and then applied many algorithms (either exact or approximate). But, determining which presents the best performance for a given VRP instance is still a difficult problem to solve. However, following the two approaches proposed in this procedure stage (remarks to be applied and classification results as the decision rules), the decision-making will be developed much more effective.

Stage III: *After-optimization in the VRP*

3.2.3 Validation and control processes in post-optimization

The post-optimality analyses after solving the combinatorial problems have been widely treated in the literature. However, these are mostly addressed to examine the sensitivity for those input data defined in the optimization model, and to estimate the proper parameter settings of the algorithm applied (control). As we mentioned in Chapter 2, still much work has to be invested in analyzing the influence of parameter combinations in the algorithm performance, especially in metaheuristics algorithms. In addition, we propose in this stage of the procedure an experimental study to check the *relevancy degree* achieved by the proposed algorithms (validation).

3.2.3.1 Validation of relevance in optimization process

In this section we study how to measure the relevance achieve by those algorithmic approaches proposed in the previous stages. Clearly, some indicators of relevance (see Table 3.10) are defined in order to validate if the optimization categories (exact or approximate) and its associated algorithms (some of them are analyzed in *Stage II*) were appropriately chosen (with highest *relevancy degree*) in the two previous stages. Undoubtedly, the indicators of relevance include the algorithm performance, excepting the complexity in the implementation. Therefore, a relevant solution implies that the solution is also effective. Consider this fact; we will later make emphasis in determining the global effectiveness (see Chapter 4) after applying all solution approaches.

To apply the current stage more than one algorithmic proposal from the previous stages is required, whether exact or approximate. Otherwise it will be very difficult to validate the relevancy degree and therefore, determine the global effectiveness. The possible comparisons between optimization categories can be carried out according to the indicators of relevance defined in Table 3.10.

Excepting the complexity in the implementation, the resulting indicators of relevance (which coincide with the performance indicators) can be measured on a numerical scale, which allows applying *statistical tests* to perform comparisons between the algorithms. The *statistical test* can be classified either as parametric or nonparametric *test*. Parametric tests require assumptions such as: *normality*, *randomness* and *homocedasticity*. These assumptions are proved by classical statisticians [Larson-Hall, 2009], therefore they will not be analyzed in this thesis. In the Appendix A.8 the main parametric tests (with its corresponding software) that can be used to compare two or more algorithm performances are described, while the Appendix A.9 shows some nonparametric tests also useful to compare two or more algorithm performances.

Table 3.10: Indicators of relevance in the post-optimality

	Indicators
Between approximate algorithms	▪ Computation time ▪ Solution quality ▪ Complexity in the implementation
Between exact algorithms	▪ Computation time ▪ Complexity in the implementation
Between exact and approximate algorithms	▪ Computation time ▪ Solution quality ▪ Complexity in the implementation

As a conclusion, the statistical analysis is crucial to determine: which algorithm will solve the given VRP instance with higher solution quality or with lower computation time or simply proves that there are not significant differences between the algorithms according to the performance indicators.

In addition to the statistical analysis we examine some external conditions which have appreciable influence in the *relevancy degree* and give rise to different decision variants. The external conditions can change the relevance course in the decision-making. For instance, in most cases, the decisions involved in the VRPs are operative decision (hierarchy). Therefore the total available time to develop such decision is quite limited. No matter how “relevant” an algorithm were according to the internal conditions, the real condition of decision-making demand to apply the algorithms with smaller time (under the total available time). So far, only the internal conditions (see conceptual model in Figure 3.1) have been considered, since such conditions were defined as predictive variables in the learning process. Decision-making in post-optimization should consider both internal and external conditions. In this sense, we propose to analyze the possible decision variants (see Appendix A.10) that may appear when is combined the total available time, as an external condition, with all the results of previous stages (considering internal conditions in the learning process).

As can be seen in Appendix A.10, the performance indicators and one of the external conditions are involved in various statistical experiments. The first column points out to the set of decision variants after obtaining the possible results from two previous stages. Then, column 2, 3 and 4 show the possible comparison results using the statistical tests for both performance indicators (solution quality and computation time) and one external condition (total available time). Finally, in the last column the relevant decision that should be made is suggested.

3.2.3.2 Sensitivity analysis in the post-optimality

In this section a methodology to estimate the sensibility of those parameters that can be defined in the optimization algorithms is provided. Our methodology (see Figure 3.9) establishes a sequence of steps, which use both the parametric and nonparametric test as the main inferential statistical techniques. Indistinctly, the sensitivity in the parameters can be analyzed defining a precise value or in a range of values. Moreover, the algorithms performance indicators constitute the response variable in which the influence of parameter variations is examined.

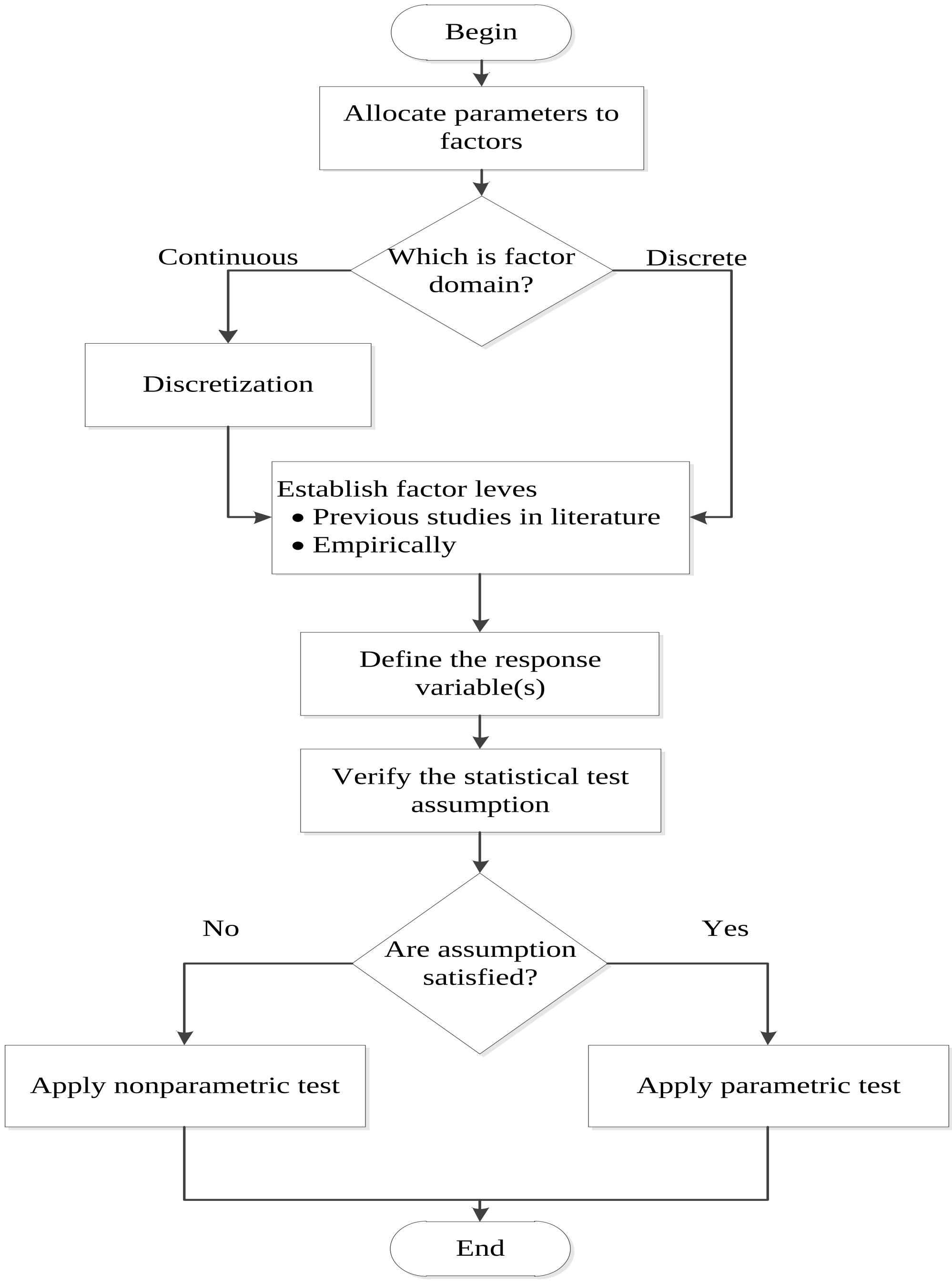


Figure 3.9: Methodology followed in the sensitivity analysis

The methodology consists of five steps, starting with the allocation of parameters to statistical factors. Then, some strategies are defined to operate with both continuous and discrete factors. Furthermore, the possible response variable(s) can be designed in the step number 3. The assumptions of statistical test are verified in the subsequent step, and finally a general framework is given to applying the statistical test (parametric or nonparametric). A detailed explanation of the steps is given as follows:

1) **Allocate parameters to the statistical factors:** In general, more than one parameter is defined when some optimization algorithm, either exact or approximate, is applied to the VRP. We consider that each parameter established in the optimization algorithms will be considered in the next steps as statistical factor and its influence in the response variable(s) should be studied. Then, this factor is divided into the set of levels below.
2) **Establish the factor levels:** The factors can be expressed in both discrete and continuous domain. For the continuous factors it should be implemented a discretization procedure (e.g. see in Engle & Gangopadhyay [2010]), due to the time consumed in the experiment. On the other side, the discrete levels can be tested almost entirely. However, for both type of factor domain previous studied in literature should be examined before setting the factors empirically.
3) **Define the response variable(s):** Then, having specified the statistical factors and their corresponding levels it is necessary to define the response variable(s) in the experiment. Concretely, in the application context of the VRP we will study the performance indicators as possible response variables.
4) **Verify the statistical test assumptions:** Any statistical test requires the satisfaction any particular assumptions. Specially, the parametric test based on the *Analysis of Variance* (ANOVA) may not have appropriate type I error when certain assumptions are violated. There are three major assumptions that should be satisfied to use ANOVA *F* test: randomness and independence, normality, and homogeneity of variance. To prove such assumptions can be used classical test reported in literature (e.g. the *Runs* test, *Kolmogorov-Smirnov* test and Levene´s test). Therefore we will not explain these tests in detail. Eventually, the assumptions are not satisfied, which make inappropriate the application of parametric tests. Here, we will consider applying the nonparametric test for the sensitivity analysis.
5) **Apply the proper statistical test:** After verifying the assumptions it can be applied as either parametric or nonparametric test. For both tests the following aspects should be defined:

   - Definition of the *hypotheses*.
   - Select the proper *statistician*.
   - Establish the *critical region*.
   - Decide the nonrejection or acceptance of the null hypothesis.

   Finally, the same statistical tests defined in the previous section can be applied to the sensitivity analysis, the parametric tests (see Appendix A.8) and the nonparametric tests (see Appendix A.9).

3.3 Summary

In this chapter we have introduced a new conception of the optimization process for the Vehicle Routing Problems, which is based on the integration of three stages: previous optimization, during optimization, and after optimization. The novel conception proposes a set of processes (leaning, estimation, optimization, selection, and validation) focused on the assurance and increase of the effectiveness in decision-making. In addition, many algorithmic approaches are defined in a procedure, which allows implementing the new conception provided by the conceptual model. In this sense, we have adopted a methodology to estimate the minimum training-set in the learning process, in which the relevant optimization category (exact or approximate) is predicted for a given VRP instance. The minimum training-set can be used by two proposed classifiers: Discriminat Analysis and the C4.5 algorithm. We have described both classifiers, which are also encoded in the computational implementation called *VRP solution classifier*.

When predicting the optimization category, we have examined some popular algorithms from each category. Subsequently, an algorithm selection process is supported by two decision rules (remarks for be applied and classification results). Finally, the *relevancy degree* (as measure of effectiveness) of each algorithmic proposal can be validated by means of some statistical tests, which are described for the conclusive sensitivity analysis in the algorithm parameters.

Chapter 4

Route planning to repair electrical breakdowns in power networks

The number of publications is still growing in the field of real-life VRP solutions, mostly, when the application contexts involve primary services for the population (e.g. emergency service, waste collection and electric service). In this chapter we present a practical contribution to this, providing an algorithmic solution to the case study mentioned in former chapters. We analyze the process of route planning to repair a set of electrical breakdown that appear in Cuban power networks. For such analysis, we propose the entire application of the procedure described in Chapter 3. As a result of procedure application, we introduce a new algorithm for solving the case study, which is set up in the city of Santa Clara. The Multi-type Ant Colony System consists in our approximate algorithmic contribution based on ACO. In addition to this contribution, a computational implementation (ANTRO version 2.0) is designed for supporting the decision-making in the dispatch center.

This chapter is concerned with two scenarios of decision-making in the cases study: the route planning in normal weather conditions and the fleet dispatching to repair electrical breakdown after natural disasters, specifically hurricanes. For both scenarios the procedure and various statistics techniques are implemented. We end this chapter determining the Global Index of Effectiveness (*GIE*) by means of instance simulation.

4.1 The case study: background and motivation

The issue of repair the electrical breakdowns in electricity distribution networks has been treated in literature [Tajnsek et al., 2011]. However, the main contributions are addressed to develop new technologies in order to make much more efficient the distribution networks. Furthermore, in some other the proper size of power network [Wang & Cheng, 2008] and the system reliability [Borges & Falcão, 2006] are studied. Regarding optimization decision, the common researches are focused on minimizing the network size, and in particular cases the multi-objective optimization are proposed, where the network size and the system reliability are optimized simultaneously [Falaghi, 2009].

Inevitably, the power networks can be subject of often breakdowns, which have to be repaired as soon as possible. Sometimes, the number of breakdowns reaches impressive values, particularly after natural disasters, such as hurricanes. Obviously, to repair such breakdowns both human and material resources in order to reestablish so valuable service (the electricity) are required. However, facilitating the proper sequence to repair and the quick departure of these resources towards the breakdown place could be crucial in the decision-making. In general, for repairing the breakdown is disposed of limited fleet of vehicles, which transport those specialist and necessary resource to the repair.

When the repair sequence is planned, interesting constraints can be visualized. For instance, not all breakdowns have the same priority. Mostly, it depends on the region where the breakdown took place and the voltage level existing in the network line. Depending on the breakdown priorities, different repair time can be consumed for the repair activities. Another difficult situation occurs when an unexpected breakdown appears after dispatching the fleet of vehicle to the repairing process. Interestingly, the planning of repair sequence (route planning) in power networks resembles many of the VRP extensions described in Chapter 2.

In Cuba, the route planning to repair electrical breakdown is carried out under harder conditions, this is largely due to network distribution type (not underground lines), weather conditions and limited resources. The hurricane season comprises six months of the year. Hence, on average, the power networks undergo severe damage twice a year.

As can be seen, the issue described in the case study reveals two attractive characteristics: the complexity of VRP extension (multiple variant simultaneously) and the particular features of the case study in the Cuban conditions. Therefore, the next sections of this chapter will be dedicated to the solution of case study applying the procedure proposed in Chapter 3.

4.2 Experimental results of the procedure application

In this section we present the experimental results obtained by the procedure application to the case study. Basically, we use five instance (moments of route planning) of the case study, which represents 5 real moments of decision-making in the dispatching centre of the Electric Company (Branch Santa Clara). The five datasets comprise both described scenarios in the route planning, during normal weather conditions and after hurricanes. Numerical input data used in this section were provided by the main office of the Santa Clara dispatching center, which is responsible for the route planning to repair the electrical breakdowns in the whole territory of Santa Clara.

Applying Stage I: *Pre-optimization in the case study*

4.2.1 Prediction of the relevant optimization methods

We start with the main characteristics of the case study instances (see Table 4.1). In general, the figures in Table 4.1 show the domain values of the predictive variables, which were defined in the classification process. During these moments of route planning, the fleet size is ranging between 3 and 7 vehicles. There are two (I-170 and I-220) instances belonging to the second scenario (after hurricanes). Furthermore, the fleet is considered homogeneous, since it is composed of identical vehicles with the same capacity of human and material resources. The customer demand is defined as stochastic based on the random behavior in the repair time.

Table 4.1: Values of predictive variables in the case study instance

	Instances				
Predictive variables	**I-32**	**I-94**	**I-142**	**I-170**	**I-220**
Number of nodes	31	93	141	169	219
Fleet size	3	5	5	6	7
Number of objectives	One-objective				
Time windows	Without time windows				
Customer demand	Stochastic				
Fleet type	Homogeneous				

The Appendix A.11 offers a synthesis of the distance matrix between nodes where the breakdowns appear. This matrix is classified as symmetric because of the negligible increases of distances for reasons of traffic rules, street direction and detours. In addition, a breakdown priority level (see the first column of the figure in Appendix A.11) is considered for route planning. The three priority level are established according to the voltage level, being the electrical breakdowns that occur in 220KV and 33KV lines of the first priority level, the second priority level for those which occur in 4KV lines, and the third in electrical lines with voltage level under 4KV (more frequent).

Basic mathematical formulation of the case study

The route planning to repair the electrical breakdown can be basically formulated as Multiple Traveling Salesman Problem (mTSP), due to some appreciable similarities with this well-known theoretical variant of the VRPs. The similarities reside in the classical dispatching of a homogeneous fleet of vehicles (with the technical staff to repair), to which a set of nodes (breakdowns) in the graph is assigned. Similar to the mTSP, the breakdowns are once visited by the vehicles and each breakdown can be visited by just one vehicle (salesman). The other particular characteristics of the case study (the occurrence of an unexpected breakdown and the priority level) will be examined in next sections, specifically when the algorithmic approaches are proposed.

Formally, the mTSP can be defined on a graph $G = (V, A)$, where V is the set of n nodes (vertices) and A is the set of arcs (edges). Let $C = (c_{ij})$ be a cost (typically distance) matrix associated with A. The matrix C is said to be symmetric when $c_{ij} = c_{ji}$, $\forall(i,j) \in A$ and asymmetric otherwise. The aim of this discrete combinatorial problem is to find m routes (one for each salesman), which start and end in a same node (depot or dispatching center in the case study). Each salesman has to visit a node once and a node can be visited by just one salesman.

Several integer programming formulation have been proposed for the mTSP in literature, the most commonly used one is the assignment-based integer programming formulation [Bektas, 2006]. In this mathematical description, the mTSP is usually formulated using an assignment-based double-index integer linear programming formulation. The decision variable can be defined as follows:

$$x_{ij} = \begin{cases} 1 & if\ arc\ (i,j)\ is\ used\ on\ a\ route, \\ 0 & otherwise. \end{cases} \quad [4.1]$$

The general formulation of assignment-based integer programming of the mTSP can be given as follows:

$$minimize \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \cdot x_{ij} \quad [4.2]$$

s.t.

$$\sum_{j=2}^{n} x_{1j} = m, \quad [4.3]$$

$$\sum_{j=2}^{n} x_{j1} = m, \quad [4.4]$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j = 2, \dots, n, \quad [4.5]$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad i = 2, \dots, n, \quad [4.6]$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \quad \forall S \subseteq V \setminus \{1\}, S \neq \emptyset, \quad [4.7]$$

$$x_{ij} \in \{0,1\}, \forall (i,j) \in A \quad [4.8]$$

The Equation 4.2 describes the fact that the objective of the problem is the minimization of the sum of the associated costs (distance) for each arc (i,j). The constraints 4.3 and 4.4 ensure that exactly m salesmen depart form and return back to node 1 (the dispatching center). Expressions 4.5 and 4.6 represent the classical assignment constraints. Finally, constraints 4.7 are used to prevent subtour-s (Subtour Elimination Constraints, SECs).

Application of the classifiers to the case study instances

In this section the proposed classifiers are applied (Discriminant Analysis and C4.5), which allows to predict the relevant optimization category (exact or approximate) for the case study instance. As we mentioned before, we design the VRP solution classifier in order to obtain the classification results from both classifier as efficient as possible. The values of predictive variables are introduced (see Figure 4.1) in the Descriptives-Data module of the designed computational implementation.

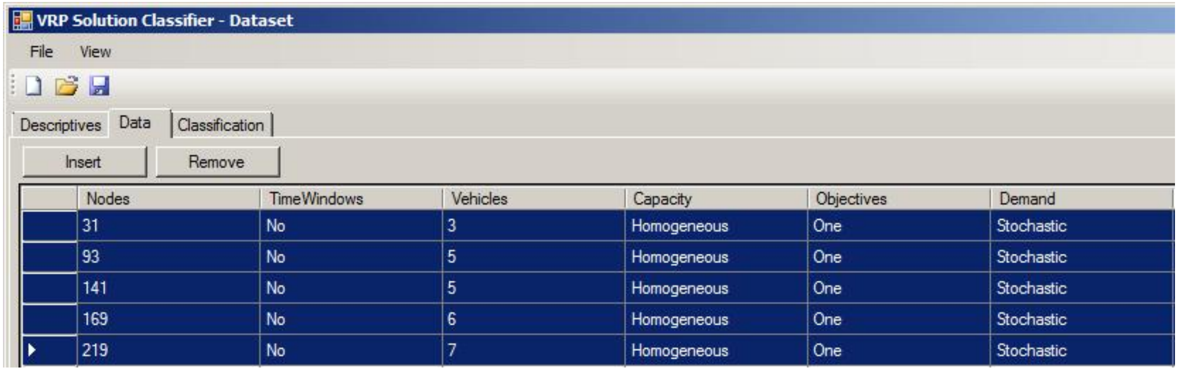


Figure 4.1: Introducing predictive variables in the Descriptives-Data module

As described in Chapter 3, the classification results can be displayed according to the analytical value in the textbox interface and following the proper branch in the decision tree. The analytical value of classification, see (approximate as the relevant category for the instance I-220) in the Figure 4.2, is easily obtained by executing the interface button “classify”. However, to understand the classification results in the decision tree one should follow the discriminant branch showed in the Appendix A.12.

A prototypical example of the aforementioned is the classification of the dataset I-220; one should follow the proper discriminant order: number of nodes > 80 and finally in the blue rectangle is defined the classification result with its corresponding class proportion (see Appendix A.12). Also, the other instances (see Table 4.2) are easy to follow throughout the branches of the classification tree

Table 4.2: Classification results of the case study instances

Instances	Classification results
I-32	Exact method
I-94	Approximate method
I-142	Approximate method
I-170	Approximate method
I-220	Approximate method

Concluding, after introducing the value of predictive variables into Descriptives-Data module, the five instances of case study have been classified (see Table 3.2), resulting the I-32 as the only one with exact classification value. Therefore, the approximate optimization algorithms should be considered in the remaining instances. Therefore, in the subsequent sections are selected some algorithms from both optimization categories.

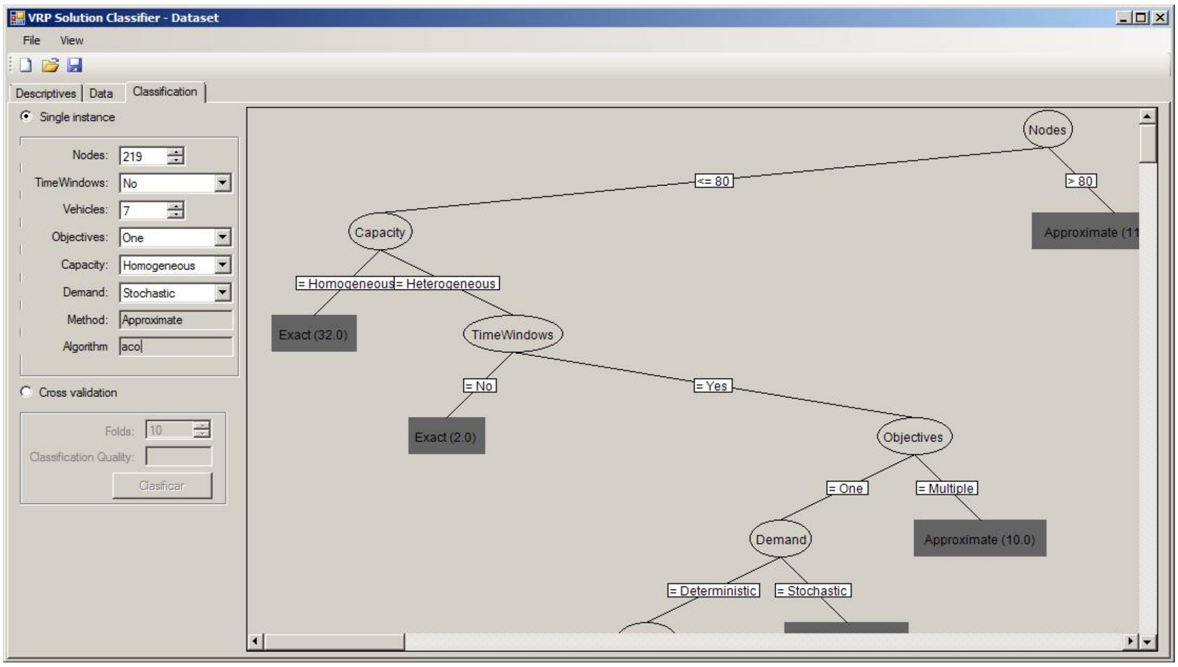


Figure 4.2: Classification result of the instance I-220

The classification quality values (see Table 4.3) are obtained with the cross-validation method. In this case, they are defined as fold sizes 10, 15 and 25 samples. The figures of classification quality in Table 4.3 are the percentage result of 10 runs of cross-validation method.

Table 4.3: Classification quality values using the cross-validation method

	Runs										
Folds	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Av.**
10	94.01	95.21	94.61	93.41	95.81	92.22	92.81	91.62	91.02	90.42	93.11
15	94.61	92.81	91.62	93.89	90.04	95.45	91.23	90.64	94.62	93.24	92.82
25	92.22	93.41	94.61	94.01	95.16	90.02	93.89	90.42	93.42	90.21	92.74

The average (Av.) values of classification quality are considered “quite accurate”, since in all fold sizes average exceeds the 90%. To our knowledge, there is not a benchmark value for this classification experiment. However, in most of studied researches, when any classifier achieves 90% of classification accurate, is considered an effective classifier.

Applying Stage II: *During-optimization in the case study*

4.2.2 Selection of relevant exact algorithm to the case study

In this section we analyze the possible exact algorithm(s), which can be applied to that instance of the case study with exact classification (I-32). Considering the instance characteristics and then examining the "remarks to applied" of the Branch and Bound algorithm, we decide to use this efficient exact algorithm in this case study instance. Other important reason to this election is based on the classification result reported in Appendix A.13, where the classifiers indicate the Branch and Bound as the relevant algorithms to the instance I-32.

The Branch and Bound (BB) algorithm has been applied to the mTSP [Husban, 1989]. There exit several good examples of source code of the BB to implement. Therefore, in this chapter we solely present the algorithm description (Pseudocode 4.1) to the case study and the computational results.

The Pseudocode 4.1 shows the steps of the BB considering the priority level of each breakdown. For such constraints an array in the distance matrix is proposed, which allows as considering the priority level the logical scheduling of the breakdown repair.

Pseudocode 4.1: The Branch and Bound algorithm to the case study

Step 1

Assign in the distance matrix C

$c_{ij} = c_{ji} = 0 \quad \forall j \; with\, priority\; degree\; 1$

$c_{ij} = c_{ji} = 0 + \varepsilon \quad \forall j \; with\, priority\; degree\; 2$

Initiate $Z_0 = -\infty$

Determine the upper bound $U(C)$ by a descending order arrangement of $t_i c$

$t_i = \max_j \{C_{ij}; C_{ij} \neq \infty\}$

$$U(C) = \sum_{i=1}^{n-m} t_i$$

$t = t_{[n-m+1]}$

Step 2

Find the link with the largest entry in the updated matrix to branch on (p, q)

$C_{pq} = \max_{i,j} \{C_{ij}; C_{ij} \neq \infty\}$

Step 3

Split C into X (the set of all solutions including the link (p, q), and $\bar{X}$, the set of all solutions not including (p, q)

Compute the upper bound on $\bar{X}$

$U(\bar{X}) = U(C) - C_{pq} + max\; \{t, r\}$

$r = \max_{j \neq q} C_{pj}; C_{pj} \neq \infty$

Develop the matrix describing $\bar{X}$ from the matrix describing C by setting $C_{pq} = \infty$

Step 4

Develop the matrix describing X by:

$C_{pj} = -\infty \quad \forall j; j \neq p$

$C_{iq} = \infty \quad \forall i; i \neq p$

$C_{ij} = \infty \quad If\ the\ link\ (i,j)\ will\ create\ a\ circuit$

Step 5

Compute $U(X)$

Begin

If $t_i = -\infty$ Then

$t_i \leftarrow v_{iq}$ (where (i,q) is a link in the partial solution so far obtained)

Else

If $l < n - m$ and $U(X) > Z_0$ Then

$C = X$

$Z_0 = U(X)$

Go to Step 2

Else

If $l = n - m$ Then

the matrix describing X has a unique solution

Else

If $U(X) \leq Z_0$

Go to **Step 6**

End If

End

Step 6

If x contains a single solution or $U(X) \leq Z_0$ Then

backtrack to the smallest subset with $U(X) > Z_0$

$C = x$

Go to **Step 2**

If $x = \emptyset$ Then

$X = optimal\ solution$

End If

Despite the multiple commercial software used (CPLEX, GAMS and LINDO) for the VRP variants, we implemented the above pseudocode in *java* programming language. The main reason of the BB implementation resides in the subsequent performance comparison between algorithms, which require of homogeneous conditions (all algorithms should be implemented in the same programming language).

4.2.3 Election of approximate algorithm(s) to the case study

The classification results obtained with the application of the first procedure stage showed that approximate algorithm(s) should be applied for most (4 instances) of case study instances. In this sense, the present section examines the possible approximate algorithms that can be applied to the case study. Clearly, we propose the ACO algorithms for solving the instances classified with approximate optimization category. As previous section, the election of this specific family of algorithms is based on the “remarks to be applied” such algorithms and the result of the classification process (see Appendix A.14). Another important fact is the well-known efficiency of the multi-colonies approach in ACO, which has been analyzed in Chapter 2. Therefore, in this section we present two algorithms of ACO´s family, the classical Ant Colony System (ACS) and a new multi-colony approach called Multi-type Ant Colony System (M-ACS).

4.2.3.1 ACS for solving the case study instances

To solve the case study, the artificial ants construct solutions by successively choosing breakdowns to visit, until each breakdown has been visited. For the selection of a (not yet visited) breakdown two aspects are taken into account in the classical ACS: how good was the choice of the breakdown before (τ_{rs}, pheromone trails) and how promising is the choice of that breakdown (η_{rs}, measure of desirability). In the original ACS given by Dorigo and Gambardella (1997), each ant k moves form present node r to the next node s using a pseudorandom rules, which are given by

$$s = \begin{cases} arg\,max_{u \in J_k(r)}\{[\tau(r,u)] \cdot [\eta(r,u)]^{\beta}\} & if \quad q \leq q_0 \quad (exploitation) \\ S & otherwise \quad (exploration) \end{cases} \qquad [4.9]$$

$$S{:}\, p_k(s) = \begin{cases} \dfrac{[\tau(r,s)] \cdot [\eta(r,s)]^{\beta}}{\sum_{u \in J_k(s)}[\tau(r,u)] \cdot [\eta(r,u)]^{\beta}} & if \quad s \in J_k(s) \\ 0 & otherwise \end{cases} \qquad [4.10]$$

where q is random number uniformly distributed in $[0\ldots1]$, q_0 is user-defined parameter ($0 \leq q_0 \leq 1$), and S is a random variable selected according to the probability given in Equation [4.10]. Moreover, $J_k(s)$ is the set of breakdowns that remain to be visited by the ant k positioned in the breakdown r, the parameter β determines the relative importance ($\beta > 1$) of pheromone versus measure of desirability (e.g. distance).

The parameter q_0 determines the relative importance of exploitation versus exploration: every time an ant in breakdown r has to choose a breakdown s to move to, it samples a random number $0 \leq q \leq 1$. If $q \leq q_0$ then the best edge, according to Equation [4.9], is chosen (exploitation), otherwise an edge is chosen according to Equation [4.10] (biased exploration).

As we mentioned in Table 3.7, often the initial value of pheromone (τ_0) is often defined as a random number uniformly distributed [0…1]. However, in this thesis it is used the Nearest Neighbor (NN) heuristic to set up the initial pheromone value. Thus, the initial pheromone can be computed as follows:

$$\tau_0 = \frac{1}{L_0} \qquad [4.11]$$

where L_0 is the total traveled distance, which is achieved by the NN heuristic.

The pheromone updating of ACS includes the same rules (local and global updating rules) given by Dorigo and Gambardella (1997). According to Dorigo and Gambardella (1997), local updating rule, see Equation 4.12, is applied to change pheromone level of edges while building a solution.

$$\tau_{(r,s)}^{new} = (1-\rho)\tau_{(r,s)}^{old} + \rho\tau_0 \qquad [4.12]$$

where ρ is defined as evaporation coefficient (with $0 \leq \rho \leq 1$), thus the trail evaporation is given by $(1 - \rho)$. Moreover, in each iteration of the algorithm, the global updating rule is applied to those arcs that conform the best tour of the fist iteration. The rule is described as follows:

$$\tau_{(r,s)}^{new} = (1-\alpha)\tau_{(r,s)}^{old} + \frac{\alpha}{L_{best}} \quad \forall(r,s) \in BestSol \qquad [4.13]$$

where L_{best} is total travel distance of the so far best solution $BestSol$.

Subsequently, in the Pseudo-code 4.2, the general procedure of ACS is presented; the procedure includes also the global pheromone update mechanism between and a nested procedure called the **new-ant-solution** in the pseudo-code.

Pseudocode 4.2: The ACS algorithm

Initialize the parameters
Obtain the initial solution (ψ^{nn}) using NN heuristic
  $\psi^{gb} \leftarrow \psi^{nn}$
  $L_{gb} \leftarrow L_{nn}$
Initiate the pheromone trail
  For each (r,s)
    $\tau(r,s) = (n \cdot L_{gb})^{-1}$
  End For
Do Until End_condiction = True

While all ant have built a complete solution
For each ant k
Build a solution ψ^k using (**new-ant-solution**)
If $L_k \leq L_{gb}$ **Then**
$L_{gb} = L_k$
$\psi^{gb} \leftarrow \psi^k$
End If
End For
End While
Update the global pheromone trail using the Equation [4.13]
Loop

The nested procedure called **new-ant-solution** (see Pseudocode 4.3) details how the ants built every component of the problem solution. For better understanding of both pseudocodes proposed in ACS, it is necessary to consider the proposed transformation of mTSP in to classical TSP. That is to say, we defined the pseudocodes based on the transformation reported in Tang et al. [2000]. The transformation is called the adding virtual city method. This method suggests adding a virtual city (breakdown in our case study) for each salesman (vehicle with technical staff to repair), where infinite cost is assigned to virtual-to-virtual distances and zero cost is assigned between virtual cities and the other cities. The main reason to apply such transformation resides on the subsequent performance comparisons that will take place in this section. We compare the performance of the proposed approximate algorithms in this section with some benchmark dataset of TSP, in which efficient heuristics algorithms based on the mentioned transformation have been applied.

Pseudocode 4.3: The new-ant-solution algorithm

Initialize the parameters
Locate ant k in depot
Initialize traveled distance: $L^k \leftarrow 0$
While (Ant k has not completed its solution)
Compute the desirability:

$$\eta(r,s) = \frac{1}{C_{r,s}} \cdot \frac{RT_s}{TAT} \cdot PD_s$$

$$C_{r,s} = max\,(1, c_{r,s})$$

$$RT_s = \begin{cases} U(45,60) & if\ priority\ degree\ is\ 1 \\ U(25,35) & if\ priority\ degree\ is\ 2 \\ U(10,20) & if\ priority\ degree\ is\ 3 \end{cases}$$

Select next node s using expression [4.9] or [4.10]
Update the local pheromone trail $\tau(r,s)$ according to Equation [4.12]

Update the tour: $\psi^k \leftarrow \psi^k + \langle s \rangle$
Update traveled distance: $L^k \leftarrow L^k + d_{ij}$
End While

$C_{r,s}$: Road distance between the breakdown r and the breakdown s.
RT_s: Repair time consumed by the breakdown s.
TAT: Total available time of the vehicles in charge of the breakdown repair.
PD_s: Priority level of the breakdown s ($0 \leq PD_s \leq 1$).

Summing up, the ACS described above can be applied either or as mTSP as TSP using the previously mentioned transformation. However, the desirability formula is completely addressed to the case study, where the priority level of each electrical breakdown and the repair time as well are considered.

4.2.3.2 M-ACS for solving the case study instances

The Multi-type Ant Colony System (M-ACS) proposed in this thesis is based on the following idea: let be CO a set colonies, representing each of them a set of global solutions of the problem (mTSP or transformed to TSP). Each colony obtains a set of global solutions (each ant of the colony represents a solution to full mTSP) using an Ant Colony System (ACS) algorithm and during the route construction the different colonies cooperate, sharing experience through “frequent” pheromone exchange. However the different types of ants are also involved in a competition process, which is based on the fact that the ants are repulsed by the pheromone of ants that belong to other colony (other type of ants). Combining both mechanism (collaboration as well as competition), a set of global solutions can be reached for all colonies (better exploration process as a main advantage), selecting the best solution after the last iteration. It is important to note that the multi-type approach differs from the one proposed in Nowé et al. (2004), where each type builds a part of the solution and the different parts were disjoint. A typical application is finding a set of disjoint paths in a graph. In our M-ACS the pseudo-random-proportional rule either considers the experience earned by each colony. The state transition rules are given by

$$s = \begin{Bmatrix} arg\,max_{u \in J_k(r)}\{[\tau(r,u)] \cdot [\eta(r,u)]^{\beta} \cdot [\phi_a(r,s)]^{\gamma}\} & if \quad q \leq q_0 \\ S & otherwise \end{Bmatrix} \qquad [4.14]$$

$$S: p_k(s) = \begin{cases} \dfrac{[\tau(r,s)] \cdot [\eta(r,s)]^{\beta} \cdot [\phi_a(r,s)]^{\gamma}}{\sum_{u \in J_k(s)}[\tau(r,u)] \cdot [\eta(r,u)]^{\beta} \cdot [\phi_a(r,u)]^{\gamma}} & if \quad s \in J_k(s) \\ 0 & otherwise \end{cases} \qquad [4.15]$$

where $\phi_a(r,s)$ indicates the average value of pheromone in the edge (r,s) taken from the other colonies, excluding the pheromone trail of colony a (current colony), after some number of iteration (F). Another parameter defined in the M-ACS is γ, which denotes the sensibility of each ant for using its own colony experience ($\gamma = 0$) or also the experience of the remaining colonies ($\gamma > 0$).

The proposed algorithm (M-ACS) presents significant features of swarm intelligence, contrary to the classical ACS, in the M-ACS a set of colonies cooperate in order to provide a better solution. The cooperation process, inspired by Nowé et al. (2004), consists on the exchange of pheromone trails reached by the ants that belong to each colony. Each colony deals with two matrixes of pheromone trails: the first one contains the pheromone trail of its own ants, and the second matrix denotes the pheromone trails reached by the ants of remaining colonies.

The frequent pheromone exchange is performed after a number of iteration F, where F is a user-defined parameter and can be established dividing the total number of iteration N in equal amount or as the user decides. Finally, the frequent pheromone exchange can be computed as follows:

$$\phi_a(r,s) = \frac{\sum_{co \in CO;\, co \neq a} \phi_{co}(r,s)}{CO - 1} \qquad [4.16]$$

where index a indicates the current colony, which performs the pheromone update, taking the average pheromone values of the other colonies, excluding its own pheromone trail. Subsequently, in the Pseudocode 4.4, the general procedure of M-ACS is presented; the procedure also includes the pheromone exchange mechanism between all colonies and the **new-ant-solution** procedure**.**

Pseudocode 4.4: The M-ACS algorithm

Initialize parameters
Obtain the initial solution (ψ^{nn}) using NN heuristic
  $\psi^{gb} \leftarrow \psi^{nn}$
  $L_{gb} \leftarrow L_{nn}$
Initiate the pheromone trail
  For each (r,s)
    $\tau(r,s) = (n \cdot L_{gb})^{-1}$
  EndFor
Do Until $IT = N$
  If $IT \,\%\, N = F$ **Then**
    Exchange the pheromone between all colonies according to Equation 4.16
  End If
  For each colony a
    For each ant k
      Build a solution ψ^k using (**new-ant-solution**) (substitute the expressions [4.9] and [4.10] by the expressions [4.14] and [4.15] respectively

If $L_k \leq L_{gb}$ **Then**

$L_{gb} = L_k$

$\psi^{gb} \leftarrow \psi^k$

End If

End For

End For

Update the global pheromone trail using the Equation [4.13]

$IT = IT + 1$

Loop

IT: Iteration.

N: Total number of iteration.

Experimental results of M-ACS using benchmark problems

Some computational experiences are presented in this section in order to evaluate the performance of the new approach previously described. Algorithm runs have been carried out on a personal computer equipped with an Intel Pentium processor 1.6 GHz and 1 GB of ram memory. The Multi-type ACS has been coded *java*.

The M-ACS was tested on six benchmark problems described in TSPLIB [236]. These problems have been originally solved with several approaches for the classical TSPs. Furthermore, we compare the M-ACS performance, using the mentioned datasets, with Lin-Kernighan heuristic reported in Dazhi and Dingwei (2007). The mentioned instances range from 124 to 783 cities and the number of the salesman used is 3, 5 and 7 respectively. The Table 4.4 summarizes the benchmark problem information, where the first indicates that last three problems are asymmetric TSPs. Columns 2-3 show the problem codes and the scale respectively. The other columns show the function objective values of Lin-Kernighan heuristic and the M-ACS (average of 10 runs) for all the salesmen used (M).

Table 4.4: The performance of M-ACS in the benchmark problems

			M-ACS			Lin-Kernighan heuristic		
Type	Codes	Scale	M = 3	M = 5	M = 7	M = 3	M = 5	M = 7
Symmetric	bier127	127	95934	87915	80345	95592	87562	80283
	ts225	225	117452	113570	110551	117960	113562	110656
	rat783	783	8668	8626	8534	8708	8650	8597
Asymmetric	kro124p	100	33765	32271	30907	33655	32247	30915
	ftv170	171	2482	2341	2263	2498	2368	2272
	rgb443	443	2604	2541	2466	2621	2555	2489

The parameter setting for M-ACS is the following: $q_0 = 0.8$, $\beta = \gamma = 1$, $\alpha = \rho = 0.1$, a small number of ants for each colony, e.g. 10 ants. From a previous statistical study we define 3 colonies, and every 10 iterations, 10% of the total number of iteration (100), the pheromone exchange is carried out for all benchmark problems reported in Table 4.4.

Starting from figures of Table 4.4, we obtained significant differences between the results achieved by M-ACS and Lin-Kernighan heuristic results. The significant differences were ensured by mean of Wilcoxon coefficient as nonparametric statistical test. Furthermore, it is important to observe that the Lin-Kernighan heuristic provide better results when the problem scale is smaller. Moreover, the practical computational time of the approach that we propose has been quite small for 100 iterations.

Complexity analysis of M-ACS

The time complexity of ACO algorithms is mainly based on its search strategies, where a set of m ants develop a tour construction with complexity $O(n^2)$ until a number of iterations is reached. The pheromone trails are stored in a matrix with $O(n^2)$ entries (one for each edge) as in all ACO strategies [Dorigo and Stützle, 1999]. In M-ACS a set of CO colonies is defined, each colony represents a subgroup of the total number of ants m. In the computational analysis this total number of ants is the important parameter and not the number of colonies. This is because the pheromone exchange between the colonies, which only is performed every 10% of the iterations, takes $O(n^2)$ as well and therefore does not increase the complexity of the standard pheromone updates within each colony. Yielding an overall time complexity of $O(n^2)$.

The Lin-Kernighan algorithm has a worst-time computational complexity of $O(n^3)$ for the TSP (or mTSP transformed) [Helsgaun, 2009]. Thus, the worst-time complexity of the proposed algorithm proves to be competitive in term of computational time compared with the efficient Lin-Kernighan heuristic.

Treatment of unexpected breakdowns in the route planning

In all the algorithmic approaches (exact or approximate) described in this chapter, it has been examined the particularities of the case study. However, the emergence of an unexpected breakdown has not been treated yet. Eventually, some new breakdown of any priority level can appear after planning all route of the fleet. Therefore, the unexpected breakdown should be feasibility inserted with the minimum cost.

As we analyzed in Chapter 2, the dynamicity in VRPs becomes one of the most difficult extension to be solved in such combinatorial problems. In this section we introduce a framework (see Figure 4.3) for supporting the unexpected breakdown insertion, which is based on the solution approach described in Runka [2009].

The proposal consists of two integrated modules (dispatcher and optimizer module), in which a sequence of static VRP problems is created. Dispatcher module initializes all the data structures, controls the time, handle the occurrence of all breakdowns (pending breakdowns of unexpected breakdowns), provide to the **Optimizer** module the input data and update the routes according to the results of the Optimizer module. On the other hand, the Optimizer module is responsible for solving the static problems generated by the other module. The static problem solution can be developed with any of the proposed algorithms (BB, ACS, and M-ACS).

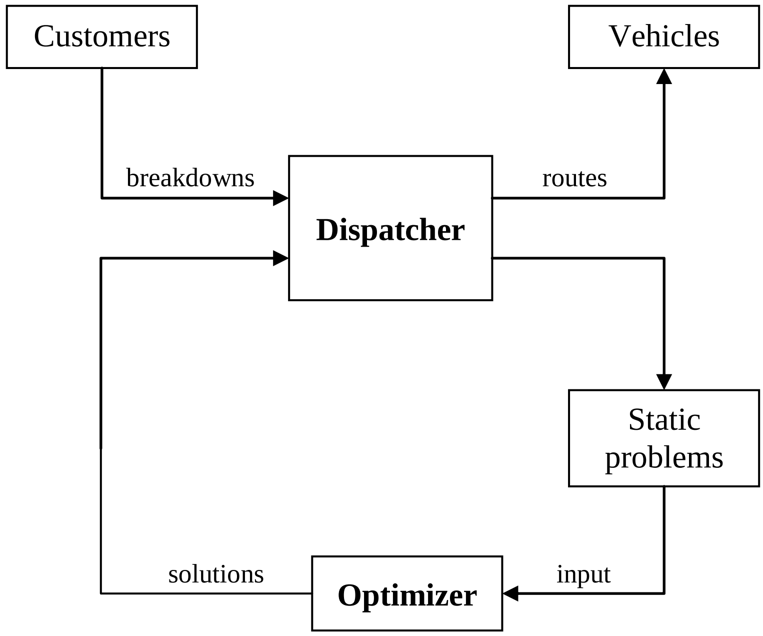


Figure 4.3: Framework to dispatching the unexpected breakdowns

In the Pseudocode 4.5 the main actions suggested by the proposed framework are explained in details. The pseudocode show the steps that should be followed when some unexpected breakdowns occurs.

Pseudocode 4.5: The insertion procedure for the unexpected breakdowns

---

Initialize ()
    $time \leftarrow 0$
    Locate the vehicles at dispatching center
    $PendingBreakdowns \leftarrow InitialBreakdowns()$
Do until While $PendingBreakdowns = \{\}$
    $breakdown\ arrival\ time \leftarrow time$
    Create the static problem with the following breakdowns:
        $Pendingbreakdowns + Newbreakdowns\ (breakdown\ arrival\ time)$
        $Solution = Optimizer\ module\ (static\ problems)$
        $time = breakdown\ arrival\ time$
    Update the route of the vehicles
    Update the PendingBreakdowns
Loop

The software ANTRO version 2.0

In this section we introduce a computational implementation called ANTRO in its version 2.0. The software ANTRO (see in Figure 4.4) consists on a useful implementation for supporting the route planning process in the repair of electrical breakdowns. This product makes possible a flexible interaction between the dispatcher and those case study instances. The software has only encoded the approximate algorithms (ACS and M-ACS) described in this chapter. As main result of the software application, the route can be planned disregarding the unexpected breakdown occurrence, and also the unexpected breakdowns can be considered in the route planning process.

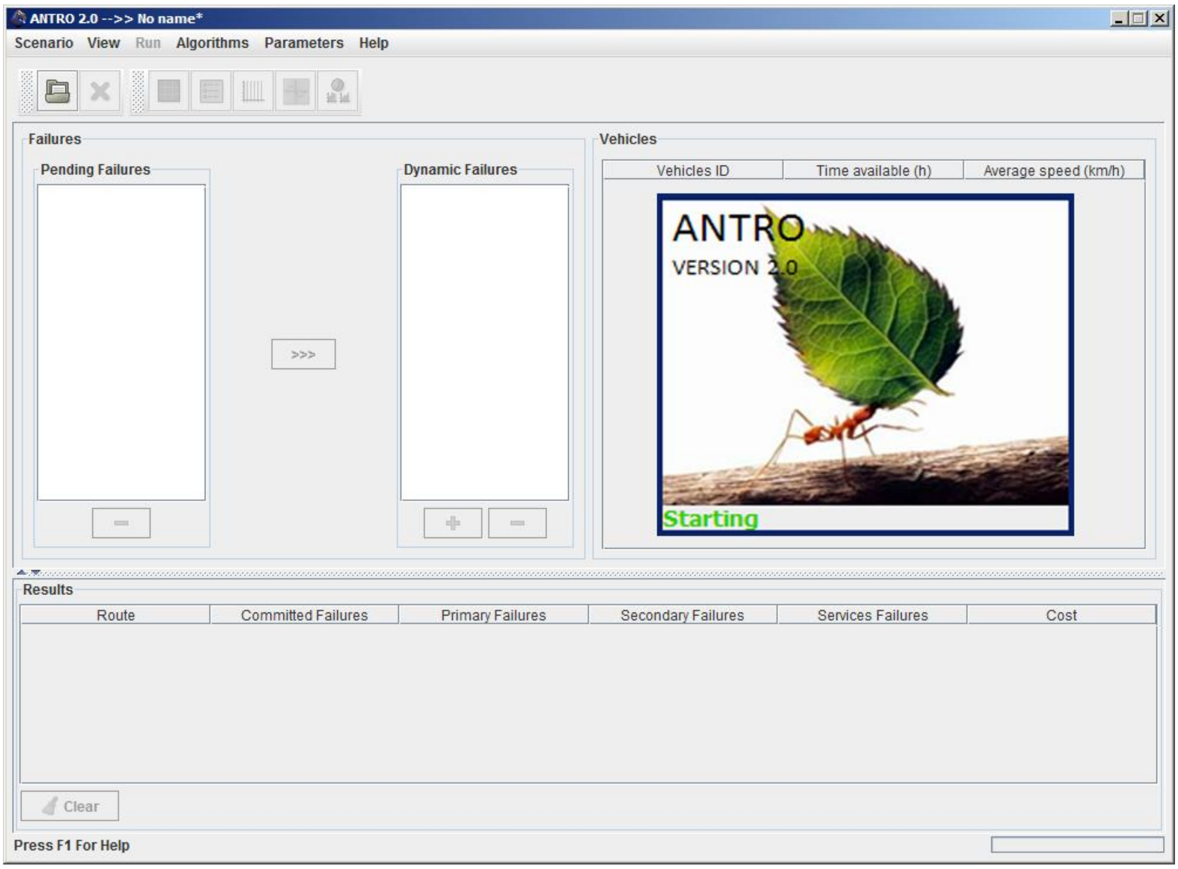


Figure 4.4: The application ANTRO version 2.0

The Appendix A.15 shows a detailed description of all ANTRO utilities: the introduction of breakdowns in different list (pending or dynamic[4]), the definition of other input data (repair time, breakdown priority level, and parameters for both algorithms and geographical coordinate of any unexpected breakdown) and visualization of various results (both analytical and graphical).

In the Appendix A.16 the source code of the ANTRO implementation appears, also implemented in *java* language programming. Moreover, both algorithms have been applied to the case study instances described in Table 4.1. After using the ANTRO software, we

[4] The dynamic breakdowns (failures) mean the same as unexpected breakdowns.

show the best achieved results by ACS and M-ACS in the Appendix A.17 and Appendix A.18 respectively. However, this result will be deeply discussed in the application of the third procedure stage (next section), also analyzing the exact results achieved by Branch and Bound algorithm in the post-optimization process.

***Applying Stage III**: After-optimization in the case study*

4.2.4 Validation of the solution relevance in the case study

In this section we measure the achieve relevance after applying the all algorithmic approaches described in this chapter. The application of the previous stages suggested the implementation of exact algorithms (specifically BB) to I-32 and approximate algorithms (ACS and M-ACS) to the remaining instances. Here, is applied the third stage of the proposed procedure, which implies the analysis of performance indicators of those algorithms designed for the case study. In order to establish the performance comparison between the algorithms, both algorithmic proposals (exact and approximate) of this chapter have been applied to the five real-life instances.

In the Table 4.5 the solution quality are depicted according to four (4) strategies, three of them are proposed in this chapter (BB, ACS and M-ACS), and the fourth is take from dispatching center records, which means the empirical solution (total traveled distance) provided by the dispatchers for these instances. The figures obtained by the ACO strategies are the average result of ten (10) algorithm runs.

Table 4.5: Results of the performance indicators applying all solution approaches

Instances	ACS	M-ACS	Branch and Bound	Dispatching center (DC)
I-32	45.08	44.76	40.53	61.64
I-94	100.03	99.02	98.00	114.53
I-142	88.75	87.03	86.33	103.80
I-170	141.93	141.88	139.41	172.38
I-220	177.12	174.84	173.01	201.47

Based on the previous performance values, we introduce the statistical technique application in order to test the significant differences between the proposed algorithms. Since some statistical assumptions are not satisfied, we applied the Wilcoxon test (see Table 4.6) to compare and rank the solution approach performances. As it can be inferred from figures of Table 4.6, in the instance I-32, the solution quality reported by BB is significantly superior compared to approximate algorithms. Still, in the other instances the performance of BB and ACO algorithms obtained different results (with better performance the BB). Interestingly, the M-ACS (our algorithmic proposal) performs significantly higher within the approximate optimization category, using all the instances in the same statistical experiment. While the solution provided by the dispatching center (the real solution that was implemented) proved to be the worst according to the solution quality.

The computation time is also analyzed in this section. In the case of instance I-32, there are no significant differences between BB and approximate algorithms based on ACO. However, after running both the exact and approximate approaches for remaining instances, evident significant differences have been reported. In particular, the time consumed by the BB exceeds the 25 minutes, while the ACO algorithms report a few seconds of computation time. On the other side, the dispatching center establishes that decision-making in route planning should be developed in 20 minutes at most, which mean that the computation times of BB are unsuitable in some instances (excluding the I-32).

Table 4.6: Performance comparison between all solution approaches

Instances	**Comparisons**	***p-value***	**Null hypothesis**
I-32	BB vs. ACS	0.020	Rejected (R)
	BB vs. M-ACS	0.035	R
I-94	BB vs. ACS	0.051	Accepted (A)
	BB vs. M-ACS	0.061	A
I-142	BB vs. ACS	0.055	A
	BB vs. M-ACS	0.063	A
I-170	BB vs. ACS	0.055	A
	BB vs. M-ACS	0.062	A
I-220	BB vs. ACS	0.042	R
	BB vs. M-ACS	0.058	A
All	M-ACS vs. ACS	0.043	R
All	M-ACS vs. DC	0.044	R
All	BB vs. DC	0.043	R

Starting from the analysis in the previous performance indicators, we examine the relevance degree based on the following statement:

1) The classification result of the instance I-32 (classifier predicted an exact method) is totally coherent with performance results. Here, the decision-making is identified as the decision variant number 6 [Vart6] (see Appendix A.10), where there exist significant differences between algorithm solution qualities (obviously the BB performs better). Furthermore, all computation times consumed by both optimization methods (exact and approximate) reached similar and acceptable results according to the total available time established by the dispatching center (at most 20 minutes).
2) The remaining instances, for which approximate algorithms based on ACO have been initially applied, due to the classification results of previous stages, the

relevance degree is also appreciable. Certainly, the classification results (approximate algorithms for all instances) are in accordance with the one indicated by the performance values. Even when the BB provides better solution quality in these instances, the differences between the algorithm computation times leads to avoid exact solution for such instances. Specially, when the time consumed by the BB is significantly superior to the total available time established for the decision-making (see decision variant number 9 in the Appendix A.10).

3) The solution approaches proposed in this chapter, either exact or approximate, provide a considerable improvement to the route planning in the case study. Therefore, the dispatching center should adopt the algorithmic proposals derived of the procedure application in the route planning.

4.2.5 Analyzing the sensitivity of algorithm parameters

In this section we determine the numerical sensibility that can experiment the parameters defined in the algorithmic proposal. Specifically, we analyze the sensitivity in the approximate algorithms (ACS and M-ACS); since a few studies of sensitivity have been reported in real-life case study applying such algorithms based on ACO. Another reason to study the sensibility of ACO parameter (applied to our case study) is subsisted in the randomness nature presented of these bioinspired algorithms, particularly observed in the pseudo random rules for route construction.

As described in Chapter 3, the sensitivity analysis is developed in this research according to the methodology depicted in Figure 3.9. The application results of each step of the mentioned methodology are discussed below.

1) **Allocating parameters to the statistical factors:** In the ACS algorithm are defined at least 5 parameters, however in this section we propose a statistical experiment analyzing solely three (3) of them, q_0, α, and β. According to Zabala [2005], some experimental studies have proved that such parameters have the greatest influence in the algorithm performance. On the other side, for the M-ACS, the parameters FI, q_0, and γ are defined as statistical factors. Unfortunately, there not exists any precedent of sensitivity analysis in M-ACS. Therefore, the parameter designation is totally empirical.

2) **Establishing the factor levels:** All factors defined in previous step are continuous. Therefore we propose the discretization of each one. In the case of ACS we adopted the discrete values defined by Zabala [2005], in which the values 0.6, 0.7 and 0.8 to q_0; 0.1, 0.3 and 0.5 to α are designated. Finally the values 1, 2 and 3 are the levels of factor β. Empirically, we designate the following levels for the M-ACS: 5%, 15% and 25% to FI; the same levels of q_0 defined in ACS; and the values 1, 2, 3 are defined as the levels of factor γ.

3) **Defining the response variable(s):** In this sensitivity analysis we adopt as response variable the algorithm solution quality. The computation time is not analyzed due to the obvious similarity in the time consumed after arbitrary changes of algorithm parameters.

4) **Verifying the statistical test assumptions:** Before applying the statistical tests (parametric or nonparametric, we check some assumption of the parametric tests. We start with the normality, which is successfully verified using the Kolmogorov-Smirnov test (*p-value* equal to 0.56). Subsequently, the randomness is proven due to the significance value (0.68) of Durbin-Watson test. The homogeneity of variance is also tested, obtaining 0.71 of Levene significance. Therefore, a parametric statistical test (for instance ANOVA) can be applied in the next step.

5) **Applying the proper statistical test:** Finally, we introduce the Analysis of Variance (ANOVA) to determine the numerical sensibility of the ACO parameters. The results of sensitivity analysis in ACS are showed in the Table 4.7, where the significance of each factor and its respective interactions are depicted in the last column (*p-value*s). While the Figure 4.5 gives a graphical perspective of the factor influence in the algorithm solution quality.

Table 4.7: The ANOVA results to sensitivity analysis in the ACS

Analysis of Variance for FO

Source	*Sum of Squares*	*Df*	*Mean Square*	*F-Ratio*	*P-Value*
A:q0	65.0324	1	65.0324	4.21	0.0440
B:alfa	711.843	1	711.843	46.06	0.0000
C:beta	45.6136	1	45.6136	2.95	0.0903
AA	7.78809	1	7.78809	0.50	0.4801
AB	0.2601	1	0.2601	0.02	0.8972
AC	9.26188	1	9.26188	0.60	0.4415
BB	41.8308	1	41.8308	2.71	0.1045
BC	38.1718	1	38.1718	2.47	0.1206
CC	68.0944	1	68.0944	4.41	0.0395
blocks	41.1246	2	20.5623	1.33	0.2710
Total error	1066.27	69	15.4532		
Total (corr.)	2095.29	80			

The *p-values* of previous table indicate that q_0 and α have individual influence in the solution quality of ACS. The objective function (minimize total traveled distance) reach the worst value when such factors are increased (see Figure 4.5). On the other hand, the parameter β has a quadratic influence; in particular, the minimum solution quality is reached when the factor takes the medium value of its level range.

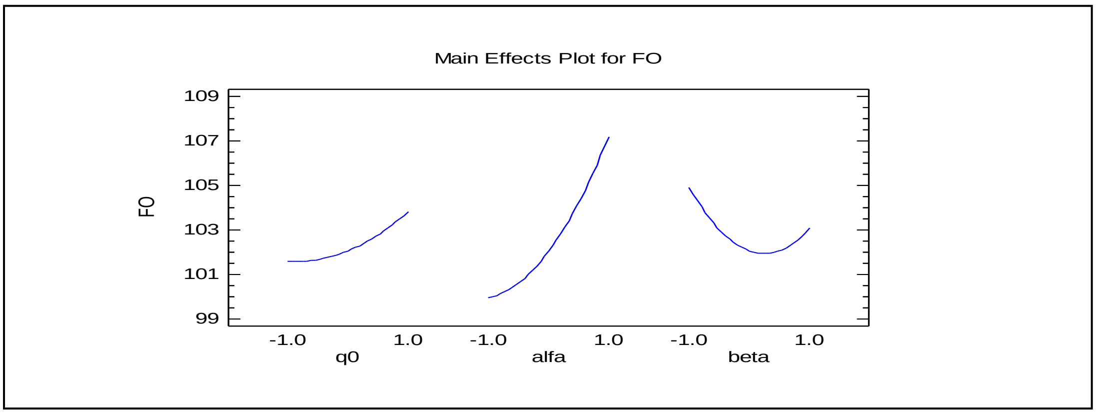


Figure 4.5: Parameter influence in the solution quality using ACS

The ANOVA results to M-ACS are showed in the Table 4.8, which indicates that individual factors [A (FI), B (γ) and C (q_0)] have not influence in the solution quality achieved by the algorithm.

Table 4.8: The ANOVA results to sensitivity analysis in the M-ACS

Analysis of Variance for Costo

Source	*Sum of Squares*	*Df*	*Mean Square*	*F-Ratio*	*P-Value*
A:FI+block	1.60856	1	1.60856	0.38	0.5414
B:Gamma+block	0.000474074	1	0.000474074	0.00	0.9916
C:q0+block	12.2503	1	12.2503	2.87	0.0948
AA	9.66045	1	9.66045	2.26	0.1371
AB	0.156025	1	0.156025	0.04	0.8490
AC	0.00934444	1	0.00934444	0.00	0.9628
BB	0.791002	1	0.791002	0.19	0.6682
BC	1.38847	1	1.38847	0.33	0.5704
CC	27.5117	1	27.5117	6.44	0.0134
Blocks	33.1498	2	16.5749	3.88	0.0253
Total error	294.602	69	4.2696		
Total (corr.)	381.129	80			

Despite the no influence of the individual factor of M-ACS, the factor q_0 presents a quadratic influence on the solution quality when the M-ACS is applied to the case study. Equally to β in ACS, the minimum solution quality is reached when the factor takes the medium value of its level range.

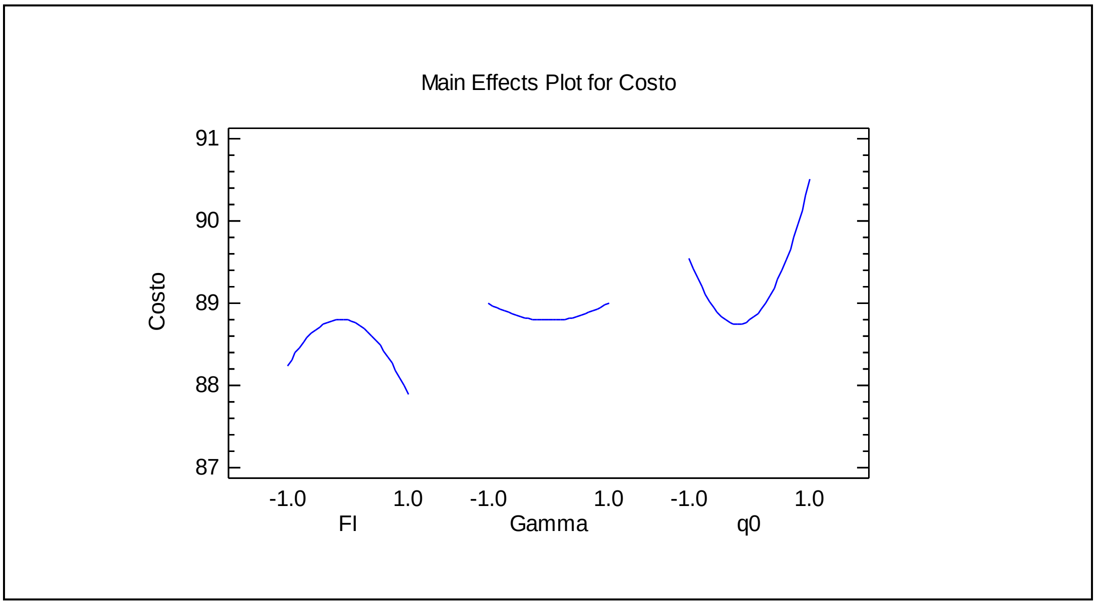


Figure 4.6: Parameter influence in the solution quality using M-ACS

4.3 Effectiveness analysis simulating enough set of instances

The procedure application in former sections has been developed according to five real-life instances of the case study. Based on these instances, the relevance degree was successfully validated due to in all cases the classification results were in accordance with the expected results of performance indicators. However, such results are not considered conclusive to really assure if the proposed procedure guarantees a suitable relevance degree and therefore effectiveness in decision-making. Considering this fact, we generate, and subsequently are analyze, a considerable set of case study instances, which are conceived using the numerical simulation of its predictive variables (number of nodes, fleet size, number of objectives, customer demand, fleet type and time windows). Clearly, we propose to analyze the relevance and effectiveness in the generated instances based on the following three steps.

Generation of case study instances using the simulation: In this step is used the numerical simulation to generate each component (expected value of the predictive variable) of the case study instance. Due to the few real-life instances our case study (at most 20), we studied the probabilistic behavior of all predictive variables in order to know the best fit of each variable. After the best probabilistic function is determined for each instance component, the instances are created predicting the expected values of every probabilistic function. As result of the above, the Table 4.9 shows the probabilistic distribution followed by the predictive variables.

Table 4.9: Result of probabilistic fit test in the predictive variables

Predictive variables	Probabilistic distribution
Number of nodes	Binomial (490,0.35)
Fleet size	Poisson (8.44)
Breakdowns of first priority (P1)	Poisson (9.12)
Breakdowns of second priority (P2)	Geometric (0.04)
Breakdowns of third priority (P3)	Binomial (380,0.29)

In summary, we generate 127 instances based on the probabilistic distribution defined above. The main characteristics of the generated instances are depicted in the Table 4.10. As can be seen, the instances values are well balanced according to the number of nodes and fleet size. Furthermore, in each instance is showed the number of nodes (breakdowns) according to the priority level [the first priority (P1), second (P2) and the third priority level (P3)], considering either a reasonable balance between both analyzed scenarios, during normal weather conditions and after hurricanes).

Table 4.10: Main descriptive characteristics of the generated instances

Instances	Number of nodes	Number of breakdowns according to the priority level: P1	P2	P3	Fleet size
1-20	23-97	77	203	844	100
21-40	102-193	196	472	2276	180
41-65	18-97	97	218	995	100
66-90	26-92	109	252	1050	100
91-110	251-396	435	861	5480	170
111-127	252-449	338	777	4998	204

Description of solution approaches: The solution approaches described in this step consist of three possible decision-making situations. In the first one (1), we propose to obtain the experimental results (solution quality and computation time) after applying all procedure in the above generated instances. The second situation (2) is based on the same solution approach (apply all procedure stages) but permuting the classification labels (where is defined **exact** replace with **approximate** and vice versa). For both former situations, are used the approximate algorithms (ACS and M-ACS) and the exact algorithm (BB) in cases that the classification process suggests approximate or exact category respectively. In the third situation (3), we propose to use the solution approach of the dispatching center for all generated instances. Here, we assume that the dispatching center carry out the route planning according to Nearest Neighbor heuristic, which either considers the priority level. The experimental results of all described solution approaches are reported in the Appendix A.19, such results were determined for all generated instances.

Estimation of the Global Index of Effectiveness (GIE): As described in Chapter 1, the effectiveness is the sum of efficacy and the efficiency. In this thesis we assume the solution quality as the efficacy reached by any solution approach, while the efficiency is measured by the computation time that consumes such solution approach. Considering the above, we introduce the Equation 4.17, which determines the GIE based on the performance indicator results reached for those solution approaches described in the previous step. Finally, the GIE is given by

$$GIE_{yz} = \sum_{\forall i \in SQS_{yz}} (SC_{yi} - SC_{zi}) + \sum_{j \in SCT_{yz}} (CT_{yj} - CT_{zj}) \qquad [4.17]$$

where GIE_{yz} denotes the GIE of solution approach y respect to the solution approach z. The variables SC_{yi} and SC_{zi} represent the reached solution qualities (measure in time units), when the solution approaches y and z, respectively, are applied to the instance i. Furthermore, the variables CT_{yj} and CT_{zj} indicate the computational time consumed by the solution approach y and z, respectively, when such solution approaches are applied to the instance j. The term SQS_{yz} denotes the set of instances, in which the solution approaches y and z have reported significant difference in the solution qualities. While SCT_{yz} denotes the set of instances, in which the solution approaches y and z have reported significant difference in the computation times. The index i and j represent the number of the instance (1…127), while the possible combination of GIE due to the variation of the index y and z can be expressed in the set $GIE = \{GIE_{yz}: y = \overline{1, n-1}; z = \overline{\imath + 1, n}\}$, where n means the total of solution approaches (here $n = 3$). As a general remark, we have to consider the optimization criterion of the case study (minimization), which means that, while much negative is the GIE_{yz}, the solution approach y provide higher effectiveness in the decision-making.

In order to determine the GIE, various statistical comparisons should be carried out, as well as some arithmetic operations. Particularly, the significance values of statistical test are reported in Table 4.11. Finally, we evaluate the Equation 4.17 with all performance indicators values that have been reported in Appendix A.19, which gives as result the figures of Table 4.11.

Table 4.11: Results of the GIEs applied to the generated instances

Combinations of GIE	**Values**	**SC significance**	**CT significance**
12	-4953.17	0.047	0.001
13	-4821.93	0.000	0.000
23	134.24	0.000	0.001

Having a closer look at the results in the Table 4.11, one might summarize the following conclusions:

1) The proposed procedure, in particular the classification process, leads to the significant increase of effectiveness in decision-making defined in our case study (see $GIE_{12} - 4953.17$). As can be seem, when the results of classification process are changed (exact ↔ approximate) the effectiveness is considerably decreased, which indicates that the prediction of the optimization categories (exact or approximate) is decisive in the decision-making.
2) The current solution approach of the dispatching center is considerable ineffective as a solution approach (due to the $GIE_{13} = -4821.93$). Given essentially, by the ignorance of the relevant optimization category that must considered depending of predictive variables values, and the poor efficacy of its decision rules for decision-making. Therefore, the route planning to repair the electrical breakdowns (our case study) should be assisted by the algorithmic proposals discussed in this thesis.

4.4 Summary

Several real-life case studies of VRPs can be solved using the algorithmic approaches described in this thesis. However, in this chapter we have analyzed the “route planning to repair the electrical breakdowns that occur in the power networks”, specifically in the city of Santa Clara. In this sense, we have given the fundamental reasons for choosing such case study, which are basically sustained in the Cuban particularities, such as limited resources and weather conditions.

In the largest part of the present chapter we have applied the procedure described in Chapter 3. Based on five real-life instances of the case study, we have analyzed and discussed the main result of the procedure stages. Initially, the classification process indicated the relevant optimization categories for all instances, the exact category for the instance I-32 and approximate optimization category for the remaining instances. Impressively, the prediction of optimization categories has been developed with accuracy above 90%.

Additionally, we have presented some algorithmic approaches according to the optimization category. Specifically, the Branch and Bound algorithms to solve exactly the case study instances and two approximate algorithms based on ACO. Within the approximate category, we have introduced the M-ACS, which showed the best performance both, the benchmark problems (compared with the well-know local search algorithm of Lin- Kernighan) as well as the case study instances (compared with ACS).

The experimental results after applying the third procedure stage have shown the relevance of algorithmic approaches, which are appropriate in all instances considering the performance indicator as well as the total available time to carry out the rote planning. In addition, we analyzed the parameter sensitivity in the approximate algorithms, determining the parameter influence in the solution quality for both approximate algorithms. Finally, we could show that, when the proposed procedure is applied to several datasets, the effectiveness is considerably increased.

Chapter 5

Conclusions

In this chapter we highlight the main contributions of our research and summarize the main results. Then, we present some potential avenues for future research.

5.1 Contributions

The researches in the field of optimization theory are primarily focused on developing new or improved algorithms for selected combinatorial problems. Not much effort has been made in gaining insight into the proper selection of algorithmic approaches in order to increase the effectiveness in the decision-making. Therefore, in this thesis we have shown how one can predict the relevant optimization methods, either as exact or approximate, according to the characteristics of the optimization problems.

The studies in this thesis focused on one of the most studied combinatorial problems: the Vehicle Routing Problems. Our contributions to this field cover the development of a new conception for the optimization process in the VRPs, the analysis and improvement of recently proposed approximate algorithms (specifically in metaheuristics), and the solution of real-life VRP considering the hard conditions of decision-making in Cuba. The main contributions can be summarized as follows.

Novel conception of the optimization process in VRP (Chapter 3)

- We have developed and described a novel conception for the VRP optimization process in order to increase the effectiveness in decision-making based on an integrative and proactive approach.

- We have proposed a conceptual model and procedure to illustrate the new conception of VRP optimization process which considers three integrated stages: previous, during, and after optimization.

- Based on the conceptual model we could show the main external and internal conditions that must be considered when some VRP is solved. Additionally, we have shown the specific inputs, processes and outputs involved in the relevant solution of the VRPs.

Knowledge Discovery (Chapter 3)

- Based on the revised literature and the work-team analysis we have presented a set of predictive variable that define the complexity of VRP instances in a multivariate context.

- With our research we have provided a Knowledge Base (KB) with significant size of VRP solutions, which made possible to train two classification algorithms: Discriminat Analysis and the C4.5.

- We have determined the minimum training-set size for the used classifiers. The size of KB is large enough for achieve a low error rate using both classifiers in the classification process.

Ant Colony Optimization (Chapter 4)

- We have introduced a new ant colony optimization (ACO) algorithm, called Multi-type Ant Colony System (M-ACS), which significantly improves the performance of other efficient algorithms. Comparisons of our algorithm to classical ACO algorithm (ACS) and the well-know local search heuristic of Lin-Kernighan have shown that, the M-ACS is currently one the best performing variant for the Multiple Traveling Salesman Problem (mTSP).

- We have analyzed the computational complexity of M-ACS, which proved to be the same computational complexity of ACO algorithms. In addition, we have verified that the worst-time complexity of M-ACS proves to be competitive in term of computational time compared with the efficient Lin-Kernighan heuristic.

- We have also applied the M-ACS to the case study described in this thesis. The formidable results prove the generality of the improvements introduced by M-ACS.

Computational implementations (Chapter 3 and Chapter 4)

- We have developed the software VRP solution classifier for predicting relevant solutions in the VRPs. Setting the predictive variables of a given VRP instance is possible to predict the optimization category with high accuracy and trivial computation time.

- We have implemented useful software, called ANTRO version 2.0. With this computational implementation we have provided a great support to the route planning to repair electrical breakdowns in Cuban power networks, conceiving all realistic complexities (weather conditions, unexpected breakdown, priority level of the breakdowns, and the probabilistic time for repair) of decision-making in the Cuban context.

5.2 Future work

The primary goals of this thesis have been the algorithmic assistance to the optimization process in the VRPs. Our contributions and the observations made in our work also pose a number of interesting open questions for the specific research issues attacked in this thesis and for the research in the area of optimization theory. More specifically, some future research can be developed in the following topics.

Ant Colony Optimization

- We have shown that, when applying the M-ACS to benchmark problems and real case study of VRPs the effectiveness is increased. However, we strongly believe that a proper hybridization with local search algorithm will provide better performance.

- Another issue deals with the setting of parameters in ACO algorithms. In our experience, the parameters given here for the case study performed very well over a wide range of instances. However, in other applications adaptive versions which dynamically tune the parameters during algorithm execution may increase algorithm robustness.

Classification process

- Our research has shown that the KB training-set size is large enough for the prediction of optimization categories. Despite this fact, the new VRP solutions should be followed in order to update the experiences of the relevant solutions in real application context of VRP.

- Current classification results are rather suitable to our case study instances, which have been modeling as mTSP. Yet, other classical extensions of the VRP can be used to model some real-life case study. Therefore, an important area for research is the analysis of classification results when the realistic case studies imply other VRP variants.

References

[1] Aarts, E. H. L. & Lenstra, J. K. [2003]. Local search in combinatorial optimization, *Princeton University Press.*

[2] Abid, M. [2008]. Operation Research for Management, *Global India Publications Pvt. Ltd*.

[3] Abreu Ledón, R. [2004]. Modelo y procedimiento para la toma de decisiones de inversión sobre el equipamiento en empresas manufactureras cubanas. Tesis presentada en opción al grado científico de Doctor en Ciencias Técnicas. Universidad Central "Marta Abreu" de Las Villas. Santa Clara, Cuba.

[4] Ai, T. J. & Kachitvichyanukul, V. [2009a]. A Particle Swarm Optimization for Vehicle Routing Problem with Time Windows. *International Journal of Operational Research,* 6, 519-537.

[5] Ai, T. J. & Kachitvichyanukul, V. [2009b]. A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 36, 1693-1702.

[6] Al-Ani, A. [2005]. Feature Subset Selection Using Ant Colony Optimization. *International Journal of Computational Intelligence,* 2, 53-58.

[7] Aljanaby, A., Ku-Mahamud, K. & Norwawi, N. M. [2010]. Interacted Multiple Ant Colonies Optimization Approach to Enhance the Performance of Ant Colony Optimization Algorithms. *Computer and Information Science,* 3, 1-29.

[8] Alvarenga, G. B., Mateus, G. R. & De Tomi, G. [2007]. A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. *Computers & Operations Research*, 34, 1561-1584.

[9] Andersen, J., Christiansen, M., Crainic, T. G. & Grønhaug, R. [2011]. Branch and Price for Service Network Design with Asset Management Constraints. *Transportation Science,* 45, 33-49.

[10] Antoniou, A. & Lu, W. [2007]. Practical Optimization: Algorithms and Engineering Applications, *Springer.*

[11] Archetti, C., Bertazzi, L., Laporte, G. & Grazia Speranza, M. [2007]. A Branch-and-Cut Algorithm for a Vendor-Managed Inventory-Routing Problem. *Transportation Science*, 41, 382-391.

[12] Ausiello, G., Bonifaci, V. & Laura, L. [2005]. The On-line Asymmetric Traveling Salesman Problem Algorithms and Data Structures. *In:* Dehne, F., López-Ortiz, A. & Sack, J.-R. (eds.). *Springer Berlin / Heidelberg*.

[13] Azi, N., Gendreau, M. & Potvin, J.-Y. [2007]. An exact algorithm for a single-vehicle routing problem with time windows and multiple routes. *European Journal of Operational Research*, 178, 755-766.

[14] Baker, K. R. [2011]. Optimization Modeling with Spreadsheets, *John Wiley & Sons*.

[15] Bakirli, G., Birant, D. & Kut, A. [2011]. An incremental genetic algorithm for classification and sensitivity analysis of its parameters. *Expert Systems with Applications*, 38, 2609-2620.

[16] Baldacci, R., Christofides, N. & Mingozzi, A. [2008]. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, 115, 351-385.

[17] Ballou, R. H. [2004]. Business logistics/supply chain management: planning, organizing, and controlling the supply chain, *Pearson Prentice Hall*.

[18] Bard, J. F., Huan, L., Dror, M. & Jaillet, P. [2002]. A branch-and-cut procedure for the vehicle routing problem with time windows. *Transportation Science*, 36, 250-269.

[19] Barnhart, C. & Laporte, G. [2007]. Transportation, *North Holland*.

[20] Barreto, S., Ferreira, C., Paixão, J. & Santos, B. S. [2007]. Using clustering analysis in a capacitated location-routing problem. *European Journal of Operational Research*, 179, 968-977.

[21] Bartholomew-Biggs, M. C. [2008]. Nonlinear optimization with engineering applications, *Springer*.

[22] Becher, M. [2008]. Integrated capacity and price control in revenue management: a fuzzy system approach, *Gabler*.

[23] Bektas, T., [2006]. The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, 34, 209-219.

[24] Belegundu, A. D. & Chandrupatla, T. R. [2011]. Optimization Concepts and Applications in Engineering, *Cambridge University Press*.

[25] Belfiore, P. C. & Yoshida Yoshizaki, H. T. [2009]. Scatter search for a real-life heterogeneous fleet vehicle routing problem with time windows and split deliveries in Brazil. *European Journal of Operational Research*, 199, 750-758.

[26] Belgacem, T. & Hifi, M. [2008]. Sensitivity analysis of the knapsack sharing problem: Perturbation of the weight of an item. *Computers & Operations Research*, 35, 295-308.

[27] Berbeglia, G., Cordeau, J.-F. & Laporte, G. [2010]. Dynamic pickup and delivery problems. *European Journal of Operational Research*, 202, 8-15.

[28] Bianchessi, N. & Righini, G. [2007]. Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. *Computers & Operations Research*, 34, 578-594.

[29] Black, K. [2011]. Business Statistics: For Contemporary Decision Making, *John Wiley & Sons*.

[30] Blumenfeld, D. [2009]. Operations Research Calculations Handbook, *CRC Press*.

[31] Böckenhauer, H.-J. & Komm, D. [2010]. Reoptimization of the metric deadline TSP. *Journal of Discrete Algorithms*, 8, 87-100.

[32] Bonabeau, E. [1999]. Swarm Intelligence: From natural to artificial systems, *Oxford University Press.*

[33] Borges, C. L. T. & Falcão, D. M. [2006]. Optimal distributed generation allocation for reliability, losses, and voltage improvement. *International Journal of Electrical Power & Energy Systems,* 28, 413-420.

[34] Bourgeois, N., Escoffier, B. & Paschos, V. [2009]. Efficient approximation of combinatorial problems by moderately exponential algorithms and data structures. *In:* Dehne, F., Gavrilova, M., Sack, J.-R. & Tóth, C. (eds.). *Springer Berlin / Heidelberg.*

[35] Bramel, J. & Simchi-Levi, D. [1995]. A location based heuristic for general routing problems. *Operation Research*, 43, 649-660.

[36] Brian, K. [2008]. Formulations and exact algorithms for the vehicle routing problem with time windows. *Computers & Operations Research,* 35, 2307-2330.

[37] Brito, J., Martínez, F., Moreno, J. & Verdegay, J. [2009]. A GRASP–VNS Hybrid for the Fuzzy Vehicle Routing Problem with Time Windows. Computer Aided Systems Theory - EUROCAST 2009. *In:* Moreno-Díaz, R., Pichler, F. & Quesada-Arencibia, A. (eds.). *Springer Berlin / Heidelberg.*

[38] Bullnheimer, B., Hartl, R. F. & Strauss, C. [1999]. A new rank-based version of the Ant System: A computational study. *Central European Journal for Operations Research and Economics,* 7, 25-38.

[39] Cao, B. Y., Zhang, C. & Li, T. [2008]. Fuzzy Information and Engineering, *Springer*.

[40] Chang, R.-S., Chang, J.-S. & Lin, P.-S. [2009]. An ant algorithm for balanced job scheduling in grids. *Future Generation Computer Systems*, 25, 20-27.

[41] Chaovalitwongse, W., Furman, K. C. & Pardalos, P. M. [2009]. Optimization and Logistics Challenges in the Enterprise, *Springer-Verlag*.

[42] Chen, B. [2007a]. An Ant Colony System Based Heuristic Algorithm for the Vehicle Routing Problem with Simultaneous Delivery and Pickup. *In:* 2nd *IEEE Conference on Industrial Electronics and Applications,* 136-141.

[43] Chen, B. [2007b]. A Multi-Ant Colony System for Vehicle Routing Problem with Time-Dependent Travel Times. *In: IEEE International Conference on Automation and Logistics,* 446-449.

[44] Chen, P., Huang, H.-K. & Dong, X.-Y. [2010]. Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem. *Expert Systems with Applications*, 37, 1620-1627.

[45] Chen, Y.-W., Lu, Y.-Z. & Yang, G.-K. [2008]. Hybrid evolutionary algorithm with marriage of genetic algorithm and extremal optimization for production scheduling. *International Journal of Advanced Manufacturing Technology,* 36, 959-968.

[46] Chetouane, F., Barker, K. & Viacaba Oropeza, A. S. [2012]. Sensitivity analysis for simulation-based decision making: Application to a hospital emergency service design. *Simulation Modelling Practice and Theory*, 20, 99-111.

[47] Cheung, B. K. S., Choy, K. L., Li, C.-L., Shi, W. & Tang, J. [2008]. Dynamic routing model and solution methods for fleet management with mobile technologies. *International Journal of Production Economics*, 113, 694-705.

[48] Choi, E. & Tcha, D.-W. [2007]. A column generation approach to the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*, 34, 2080-2095.

[49] Christiansen, C. H. & Lysgaard, J. [2007]. A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research Letters*, 35, 773-781.

[50] Christofides, N., Mingozzi, A. & Toth, P. [1979]. The Vehicle Routing Problem. In: Combinatorial Optimizations, N. Christofides, R. Mingozzi, P. Toth, and C. Sandi (eds.). John *Wiley & Sons, New York.*

[51] Clarke, G. & Wright, W. [1964]. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12, 568-561.

[52] Colorti, D. & Roizzoli, A. E. [2007]. Ant colony optimization for real-world vehicle routing problems. *SIGEVOlution*, 2, 2-9.

[53] Contreras, I. & Fernández, E. [2012]. General network design: A unified view of combined location and network design problems. *European Journal of Operational Research*. [In press].

[54] Cordeau, J.-F. & Laporte, G. [2003]. The Dial-a-Ride Problem (DARP): Variants, modeling issues and algorithms. *4OR: A Quarterly Journal of Operations Research*, 1, 89-101.

[55] Cordeau, J.-F. & Laporte, G. [2007]. The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, 153, 29-46.

[56] Cordeau, J.-F., Gendreau, M., Hertz, A., Laporte, G. & Sormany, J.-S. [2005]. New Heuristics for the Vehicle Routing Problem Logistics Systems: Design and Optimization. *In:* Langevin, A. & Riopel, D. (eds.). *Springer US.*

[57] Cordón, O., Viana, I. F. D., Herrera, F. & Moreno, L. [2000]. New ACO model integrating evolutionary computation concepts: The Best-Worst Ant System. *In:* Dorigo, M., Middendorf, M. & Stützle, T., (eds). ANTS 2000, Université Libre de Bruxelles, Belgium. *Springer*.

[58] Costa Salas, Y. J., Abreu Ledón, R., Machado Oses, C. & Coello Machado, N. I. [2010]. Asistencia decisional en el proceso de optimización para el ruteo de vehículos. *Revista Ingeniería Industrial*, 31, 1-7.

[59] Costa Salas, Y. J., Abreu Ledón, R., Nowé, A. & Coello Machado, N. I. [2012]. A new approach to solve the Multiple Traveling Salesman Problem based on Ant Colony Optimization. *Computer & Industrial Engineering*. [Paper in press 2012].

[60] Costa Salas, Y. J., Coello Machado, N. I., Glistau, E. & Machado Oses, C. [2011]. A new algorithm for Facility Location Problem based on Dynamic Mesh Optimization. *ACTA TECHNICA CORVINIENSIS*, 4, 19-22.

[61] Costanza, J., Cutello, V. & Pavone, M. [2011]. A Memetic Immunological Algorithm for Resource Allocation Problem Artificial Immune Systems. *In:* Liò, P., Nicosia, G. & Stibor, T. (eds.). *Springer Berlin / Heidelberg*.

[62] Cumbre de Johannesburgo. [2002]. Conferencia de las Naciones Unidas sobre el Medio Ambiente y el Desarrollo (CNUMAD). Johannesburgo, Suráfrica.

[63] Daganzo, C. F. [1984]. The distance traveled to visit N-points with a maximum of C-stops per vehicle - an analytic model and an application. *Transportation Science*, 18, 331-350.

[64] Dantzig, G. B. & Ramser, J. H. [1959]. The Truck Dispatching Problem. *Management Science*, 6, 80-91.

[65] Dantzig, G. B., Fulkerson, R. & Johnson, M. [1954]. "Solution of large-scale traveling salesman problem". *Operation Research*, 2, 393-410.

[66] Dazhi, W. & Dingwei, W. [2007]. An Approach to Solve Large Scale Multiple Traveling Salesman Problem with Balanced Assignment. *The 6th International Conference on Management*, Wuhan, China.

[67] Deb, K. [2004]. Optimization for engineering design: algorithms and examples, *Prentice-Hall of India.*

[68] Del Valle, A. E. [2006]. Sistema de Posicionamiento Global (GPS). En Red. Suplemento Científico Técnico Juventud Rebelde, 111, 2-3.

[69] Demazeau, Y., Pavón, J. & Corchado, J. M. [2009]. Seventh International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009), *Springer.*

[70] Desaulniers, G., Lessard, F. & Hadjar, A. [2008]. Tabu Search, Partial Elementarity, and Generalized k-Path Inequalities for the Vehicle Routing Problem with Time Windows. *Transportation Science*, 42, 387-404.

[71] Diwekar, U. M. [2008]. Introduction to Applied Optimization, *Springer*.

[72] Dondo, R. & Cerdá, J. [2007]. A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European Journal of Operational Research*, 176, 1478-1507.

[73] Dorigo, M. & Gambardella, L. M. [1997]. Ant Colony System: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1, 53-66.

[74] Dorigo, M. & Stützle, T. [2003]. The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances. Handbook of Metaheuristics. *In:* Glover, F. & Kochenberger, G. (eds.). *Springer New York.*

[75] Dorigo, M. & Stützle, T. [2004]. Ant Colony Optimization. *MIT Press, Cambridge, MA.*

[76] Dorigo, M. & Stützle, T. [2010]. Ant Colony Optimization: Overview and Recent Advances. Handbook of Metaheuristics. *In:* Gendreau, M. & Potvin, J.-Y. (eds.). *Springer US.*

[77] Dorigo, M., Birattari, M. & Stützle, T. [2006]. Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1, 28-29.

[78] Dorigo, M., Caro, G. D. & Gambardella, L. M. [1999]. Ant algorithms for discrete optimization. *Artificial Life*, 5, 137-172.

[79] Dorigo, M., Maniezzo, V. & Colorni, A. [1996]. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems Man, and Cybernetics*, 26, 29-41.

[80] Dorigo, M. & Stützle, T. [1999]. ACO Algorithms for the Traveling Salesman Problem. Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming, Genetic Programming and Industrial Applications, *John Wiley & Sons*.

[81] Dréo, J. [2006]. Metaheuristics for hard optimization: methods and case studies, *Springer*.

[82] Dressler, F. & Akan, O. B. [2010]. A survey on bio-inspired networking. *Computer Networks*, 54, 881-900.

[83] Du, T., Wang, F. K. & Lu, P.-Y. [2007]. A real-time vehicle-dispatching system for consolidating milk runs. *Transportation Research Part E: Logistics and Transportation Review*, 43, 565-577.

[84] Dušan, T. [2008]. Swarm intelligence systems for transportation engineering: Principles and applications. *Transportation Research Part C: Emerging Technologies*, 16, 651-667.

[85] Eksioglu, B., Vural, A. V. & Reisman, A. [2009]. The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57, 1472-1483.

[86] Engle, K. M., & Gangopadhyay, A. [2010]. An Efficient Method for Discretizing Continuous Attributes. *International Journal of Data Warehousing and Mining (IJDWM)*, 6, 1-21.

[87] Eriksson, L. [2008]. Design of Experiments: Principles and Applications, *Umetrics Academy*.

[88] Faedo, D. [1999]. El transporte de carga y pasajeros, Ed. Pueblo y Educación. La Habana.

[89] Falaghi, H., Haghifam, M.-R. & Singh, C. [2009]. Ant Colony Optimization-Based Method for Placement of Sectionalizing Switches in Distribution Networks Using a Fuzzy Multiobjective Approach. *IEEE Transactions on Power Delivery*, 24, 268-276.

[90] Farahani, R. Z., Rezapour, S. & Kardar, L. [2011]. Logistics Operations and Management: Concepts and Models, *Elsevier*.

[91] Fernández, E. [2006]. Itinerarios de Vehículos. ALFA-Loglny: Logística Inversa y gestión respetuosa con el Medio Ambiente. *Programa ALFA II de la Comunidad Europea*.

[92] Festa, P. [2010]. Experimental Algorithms: 9th International Symposium, SEA 2010, Ischia Island, Naples, Italy, May 20-22, 2010. Proceedings, *Springer*.

[93] Fink, A. & Rothlauf, F. [2008]. Advances in Computational Intelligence in Transport, Logistics, and Supply Chain Management, *Springer.*

[94] Fisher, M. L. & Jaikumar, R. [1981]. A generalized assignment heuristic for vehicle routing. *Networks*, 11, 109-124.

[95] Flisberg, P. [2009]. A hybrid method based on linear programming and tabu search for routing of logging trucks. *Computers and Operations Research*, 36, 1122-1244.

[96] Forestiero, A., Mastroianni, C. & Spezzano, G. [2008]. So-Grid: A self-organizing Grid featuring bio-inspired algorithms. *ACM Transactions on Autonomous and Adaptive System*, 3, 1-37.

[97] Fountas, C. [2010]. Swarm Intelligence: The Ant Paradigm Multimedia Services in Intelligent Environments. *In:* Tsihrintzis, G. A., Virvou, M. & Jain, L. C. (eds.). *Springer Berlin Heidelberg.*

[98] Francis, P. & Smilowitz, K. [2006]. Modeling techniques for periodic vehicle routing problems. *Transportation Research Part B: Methodological*, 40, 872-884.

[99] Francis, P., Smilowitz, K. & Tzur, M. [2006]. The Period Vehicle Routing Problem with Service Choice. *Transportation Science*, 40, 439-454.

[100] Gál, T. & Greenberg, H. J. [1997]. Advances in sensitivity analysis and parametric programming, *Kluwer Academic Publishers*.

[101] Gambardella, L. M., Taillard, E. D. & Agazzi, G. [1999]. MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows. *In: Mcgraw-Hill* (ed.) *New Ideas in Optimization.*

[102] Garrido, P. & Castro, C. [2009]. Stable solving of CVRPs using hyperheuristics. Proceedings of the 11th Annual conference on Genetic and evolutionary computation. Montreal, Canada: *ACM*.

[103] Garrido, P. & Riff, M. [2010]. DVRP: a hard dynamic combinatorial optimisation problem tackled by an evolutionary hyper-heuristic. *Journal of Heuristics*, 16, 795-834.

[104] Garza Ríos, R. & Gonzáles Sánchez, C. [2004]. DRsoft: Un soporte computacional para el diseño de rutas de distribución. *Revista Investigación Operacional*, 25, 3-8.

[105] Garza Ríos, R. [2001]. Procedimiento muticriterio para la planificación de las rutas de distribución. Tesis presentada en opción al grado científico de Doctor en Ciencias Técnicas. Instituto Superior Politécnico José Antonio Echevarría. Ciudad de la Habana, Cuba.

[106] Geiger, M. J., Habenicht, W., Sevaux, M. & Sörensen, K. [2009]. Metaheuristics in the Service Industry, *Springer.*

[107] Gendreau, M. & Potvin, J. Y. [2010]. Handbook of Metaheuristics, *Springer.*

[108] Ghosh, D., Chakravarti, N. & Sierksma, G. [2006]. Sensitivity analysis of a greedy heuristic for knapsack problems. *European Journal of Operational Research*, 169, 340-350.

[109] Giorgi, G., Guerraggio, A. & Thierfelder, J. [2004]. Mathematics of optimization: smooth and non-smooth case, *Elsevier*.

[110] Godinho, M. T., Gouveia, L. & Magnanti, T. L. [2008]. Combined route capacity and route length models for unit demand vehicle routing problems. *Discrete Optimization*, 5, 350-372.

[111] Goetschalckx, M. [2011]. Supply Chain Engineering, *Springer.*

[112] Goh, C. K. & Tan, K. C. [2009]. Evolutionary Multi-objective Optimization in Uncertain Environments: Issues and Algorithms, *Springer.*

[113] Golden, B. L., Raghavan, S. & Wasil, E. A. [2008]. The vehicle routing problem: latest advances and new challenges, *Springer.*

[114] Gómez Díaz, Y. R. [2010]. Algoritmos que combinan conjuntos aproximados y optimización basada en colonias de hormigas para la selección de rasgos. Extensión a múltiples fuentes de datos. Tesis presentada en opción al grado científico de Doctor en Ciencias Técnicas. Universidad Central "Marta Abreu" de Las Villas. Santa Clara, Cuba.

[115] Gomez, Y., Bello, R., Puris, A., García, M. M. & Nowé, A. [2008]. Two Step Swarm Intelligence to Solve the Feature Selection Problem. *Journal of Universal Computer Science*, 14, 2582-2596.

[116] Griva, I., Nash, S. G. & Sofer, A. [2009]. Linear and nonlinear optimization, *Society for Industrial and Applied Mathematics.*

[117] Gudehus, T. & Kotzab, H. [2009]. Comprehensive logistics, *Springer*.

[118] Gutin, G. & Punnen, A. P. [2002]. The traveling salesman problem and its variants. *Kluwer*.

[119] Hanshar, F. & Ombuki-Berman, B. [2007]. Dynamic vehicle routing using genetic algorithms. *Applied Intelligence,* 27, 89-99.

[120] Hashimoto, H., Ibaraki, T., Imahori, S. & Yagiura, M. [2006]. The vehicle routing problem with flexible time windows and traveling times. *Discrete Applied Mathematics*, 154, 2271-2290.

[121] Helbing, D. [2007]. Managing complexity: insights, concepts, applications, *Springer*.

[122] Helsgaun, K. [2009]. General *k*-opt submoves for the Lin-Kernighan TSP heuristic. *Mathematical Programming Computation*. 1, 119-163.

[123] Hemmelmayr, V. C., Doerner, K. F. & Hartl, R. F. [2009]. A variable neighborhood search heuristic for periodic routing problems. *European Journal of Operational Research*, 195, 791-802.

[124] Hillier, F. S. & Lieberman, G. J. [2010]. Introducción a la investigación de operaciones, *McGraw-Hill.*

[125] Ho, W., Ho, G. T. S., Ji, P. & Lau, H. C. W. [2008]. A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence*, 21, 548-557.

[126] Ho, W., Ho, G. T. S., Ji, P. & Lau, H. C. W. [2008]. A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence*, 21, 548-557.

[127] Ho, Y. C., Zhao, Q. C. & Jia, Q. S. [2007]. Ordinal optimization: soft optimization for hard problems, *Springer*.

[128] Hoff, A., Andersson, H., Christiansen, M., Hasle, G. & Løkketangen, A. [2010]. Industrial aspects and literature survey: Fleet composition and routing. *Computers & Operations Research*, 37, 2041-2061.

[129] Hsu, C.-I., Hung, S.-F. & Li, H.-C. [2007]. Vehicle routing problem with time-windows for perishable food delivery. *Journal of Food Engineering*, 80, 465-475.

[130] Hu, Z., Ding, Y. & Shao, Q. [2009]. Immune co-evolutionary algorithm based partition balancing optimization for tobacco distribution system. *Expert Systems with Applications*, 36, 5248-5255.

[131] Hunt, E. B., Marin, J. & Stone, P. J. [1966]. Experiments in induction. *Academic Press, New York*.

[132] Ibaraki, T., Imahori, S., Nonobe, K., Sobue, K., Uno, T. & Yagiura, M. [2008]. An iterated local search algorithm for the vehicle routing problem with convex time penalty functions. *Discrete Applied Mathematics*, 156, 2050-2069.

[133] Imran, A., Salhi, S. & Wassan, N. A. [2009]. A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research*, 197, 509-518.

[134] Iori, M., Salazar-González, J. J. & Vigo, D. [2007]. An Exact Approach for the Vehicle Routing Problem with Two-Dimensional Loading Constraints. *Transportation Science*, 41, 253-264.

[135] Jan, S. [2007]. Post-optimality analysis of the optimal solution of a degenerate linear program using a pivoting algorithm. *Computers & Operations Research*, 34, 1871-1884.

[136] Jeżowski, J. & Thullie, J. [2009]. 19th European Symposium on Computer Aided Process Engineering: ESCAPE-19: June 14-17, 2009, Cracow, Poland, *Elsevier Science & Technology*.

[137] Jiang, Y., Shen, X., Chen, J. & Tripathi, R. [2008]. Analysis and approximation of optimal co-scheduling on chip multiprocessors. Proceedings of the 17th international conference on Parallel architectures and compilation techniques. Toronto, Ontario, Canada: *ACM*.

[138] Jing-Quan, L. [2009]. Real-time vehicle rerouting problems with time windows. *European Journal of Operational Research*, 194, 711-727.

[139] José, B. [2009]. A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem. *European Journal of Operational Research*, 195, 716-728.

[140] Joshi, M. C. & Moudgalya, K. M. [2004]. Optimization: theory and practice, *Alpha Science International Ltd.*

[141] Jotshi, A., Gong, Q. & Batta, R. [2009]. Dispatching and routing of emergency vehicles in disaster mitigation using data fusion. *Socio-Economic Planning Sciences*, 43, 1-24.

[142] Jozefowiez, N., Semet, F. & Talbi, E.-G. [2007]. Target aiming Pareto search and its application to the vehicle routing problem with route balancing. *Journal of Heuristics*, 13, 455-469.

[143] Kalcsics, J. & Nickel, S. [2008]. Operations research proceedings 2007: selected papers of the Annual International Conference of the German Operations Research Society (GOR), Saarbrücken, September 5-7, 2007, *Springer.*

[144] Kallehauge, B. [2008]. Formulations and exact algorithms for the vehicle routing problem with time windows. *Computers and Operations Research*, 35, 2307-2330.

[145] Kandiller, L. [2007]. Principles of mathematics in operations research, *Springer.*

[146] Kara, I. [2011]. Arc based integer programming formulations for the Distance Constrained Vehicle Routing problem. *In:* Logistics and Industrial Informatics (LINDI), *3rd IEEE International Symposium*, 33-38.

[147] Karaboga, D. & Akay, B. [2009]. A survey: algorithms simulating bee swarm intelligence. *Artificial Intelligence Review*, 31, 61-85.

[148] Karlaftis, M. G., Kepaptsoglou, K. & Sambracos, E. [2009]. Containership routing with time deadlines and simultaneous deliveries and pick-ups. *Transportation Research Part E: Logistics and Transportation Review*, 45, 210-221.

[149] Kennedy, J. & Eberhart, R. C. [1995]. Particle swarm optimization. *In: IEEE International Conference on Neural Networks*, Piscataway, New York, USA.

[150] Khereddine, B. & Gzara, M. [2011]. FDClust: A New Bio-Inspired Divisive Clustering Algorithm Advances in Swarm Intelligence. *In:* Tan, Y., Shi, Y., Chai, Y. & Wang, G. (eds.). *Springer Berlin / Heidelberg.*

[151] Kim, H., Yang, J. & Lee, K.-D. [2009]. Vehicle routing in reverse logistics for recycling end-of-life consumer electronic goods in South Korea. *Transportation Research Part D: Transport and Environment*, 14, 291-299.

[152] Klemeš, J., Friedler, F., Bulatov, I. & Varbanov, P. [2010]. Sustainability in the Process Industry: Integration and Optimization, *McGraw-Hill*.

[153] Kreowski, H. J., Scholz-Reiter, B. & Thoben, K. D. [2010]. Dynamics in Logistics: Second International Conference, LDIC 2009, Bremen, Germany, August 2009. Proceedings, *Springer.*

[154] Kytöjoki, J., Nuortio, T., Bräysy, O. & Gendreau, M. [2007]. An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. *Computers & Operations Research*, 34, 2743-2757.

[155] Laporte, G. & Martín, I. R. [2007]. Locating a cycle in a transportation or a telecommunications network. *Networks*, 50, 92-108.

[156] Laporte, G. [1992]. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59, 345-358.

[157] Laporte, G., Gendreau, M., Potvin, J. Y. & Semet, F. [2000]. Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research*, 7, 285-300.

[158] Larsen, A., Madsen, O. B. G. & Solomon, M. M. [2007]. Classification of Dynamic Vehicle Routing Systems Dynamic Fleet Management. *In:* Zeimpekis, V., Tarantilis, C. D., Giaglis, G. M. & Minis, I. (eds.). *Springer US.*

[159] Larsen, A., Madsen, O. B. G. & Solomon, M. M. [2008]. Recent Developments in Dynamic Vehicle Routing Systems. The Vehicle Routing Problem: Latest Advances and New Challenges. *In:* Golden, B., Raghavan, S. & Wasil, E. (eds.). *Springer US*.

[160] Larson-Hall, J. [2009]. *A Guide to Doing Statistics in Second Language Research Using SPSS*, Routledge.

[161] Lau, H. C. W., Chan, T. M., Tsui, W. T. & Pang, W. K. [2010]. Application of Genetic Algorithms to Solve the Multidepot Vehicle Routing Problem. *IEEE Transactions on Automation Science and Engineering*, 7, 383-392.

[162] Li, F., Golden, B. & Wasil, E. [2007]. The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. *Computers & Operations Research*, 34, 2918-2930.

[163] Lin, S.-W., Yu, V. F. & Chou, S.-Y. [2009]. Solving the truck and trailer routing problem based on a simulated annealing heuristic. *Computers & Operations Research*, 36, 1683-1692.

[164] Löhne, A. [2011]. Vector Optimization with Infimum and Supremum, *Springer*.

[165] López Milán, E., Miguel Fernández, S. & Plá Argonés, L. M. [2004]. El problema del transporte de la caña de azúcar en Cuba. *Revista Investigación Operacional*, 25, 148-157.

[166] Lucas, M. T. & Chhajed, D. [2004]. Applications of Location Analysis in Agriculture: A Survey. *Journal of the Operational Research Society*, 55, 561-578.

[167] Ma, X. [2010]. Application of adaptive hybrid sequences niche artificial fish swarm algorithm in vehicle routing problem. *International Conference on Future Computer and Communication* (ICFCC), 1, 654-658.

[168] Mahalanobis, P.C. [1936]. On the generalized distance in statistics. *Proc. Natl. Inst. Sci. India*, 2, 49-55.

[169] Marinakis, Y. & Marinaki, M. [2010]. A hybrid genetic – Particle Swarm Optimization Algorithm for the vehicle routing problem. *Expert System with Applications*, 37, 1446-1455.

[170] Marinakis, Y. & Migdalas, A. [2007]. Annotated bibliography in vehicle routing. *Operational Research*, 7, 27-46.

[171] Marinakis, Y., Marinaki, M. & Dounias, G. [2010]. A hybrid particle swarm optimization algorithm for the vehicle routing problem. *Engineering Applications of Artificial Intelligence*, 23, 463-472.

[172] Matis, P. [2008]. Decision support system for solving the street routing problem. *Transport*, 23, 230-235.

[173] Meihua, W., Xuhong, T., Shan, C. & Shumin, W. [2011]. Hybrid Ant Colony Optimization Algorithm for Two Echelon Vehicle Routing Problem. *Procedia Engineering*, 15, 3361-3365.

[174] Meisel, S. [2011]. Dynamic Vehicle Routing: Anticipatory Optimization for Dynamic Decision Making. *Springer New York*.

[175] Merz, P. [2000]. Memetic Algorithms for Combinatorial Optimization Problems: Finest Landscape and Effective Search Strategies. *PhD thesis*. Universität-Gesamthochschule Siegen, Germany.

[176] Mester, D. & Bräysy, O. [2007]. Active-guided evolution strategies for large-scale capacitated vehicle routing problems. *Computers & Operations Research*, 34, 2964-2975.

[177] Michalewicz, Z. & Fogel, D. [2004]. How to solve it. Modern heuristics, *Springer*.

[178] Miguel Andres, F. [2007]. Analysis of the efficiency of urban commercial vehicle tours: Data collection, methodology, and policy implications. *Transportation Research Part B: Methodological*, 41**,** 1014-1032.

[179] Miguel Andres, F. [2010]. An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. *Transportation Research Part C: Emerging Technologies*, 18, 668-679.

[180] Mihelič, J. & Robič, B. [2005]. Solving the k-center Problem Efficiently with a Dominating Set Algorithm. *Journal of Computing and Information Technology*, 3, 225-233.

[181] Minis, I., Zeimpekis, V., Dounias, G. & Ampazis, N. [2010]. Supply Chain Optimization, Design, and Management: Advances and Intelligent Methods, *IGI Global*.

[182] MITRANS [2008]. Guía básica para el Reordenamiento del Transporte. Ciudad de La Habana, Cuba.

[183] MITRANS [1984]. NRMT083 “Control de las transportaciones de carga”. Ciudad Habana, Cuba.

[184] MITRANS [2005]. Procedimiento de trabajo para la explotación de los medios de transporte con el Sistema de Gestión y Control de Flotas (GPS). Ciudad Habana, Cuba.

[185] Moghaddam, K. S. & Usher, J. S. [2011]. Sensitivity analysis and comparison of algorithms in preventive maintenance and replacement scheduling optimization models. *Computers & Industrial Engineering*, 61, 64-75.

[186] Mole, R. H. & Jameson, S. R. [1976]. A sequential route-building algorithm employing a generalised savings criterion. *Operational Research Quarterly*, 27, 503-511.

[187] Mukherjee, S., Tamayo, P., Rogers, S., Rifkin, R., Engle, A., Campbell, C., Golub, T. R. & Mesirov, J. P. [2003]. Estimating Dataset Size Requirements for Classifying DNA Microarray Data. *Journal of Computational Biology*, 10, 119-142.

[188] Mullen, R. J., Monekosso, D., Barman, S. & Remagnino, P. [2009]. A review of ant algorithms. *Expert Systems with Applications*, 36, 9608-9617.

[189] Murty, K. G. [2009]. Optimization for Decision Making: Linear and Quadratic Models, *Springer*.

[190] Mustafi, C. K. [2007]. Operations Research Methods and Practice, *New Age International (P) Ltd*.

[191] Na, B., Jun, Y. & Kim, B.-I. [2011]. Some extensions to the sweep algorithm. *International Journal of Advanced Manufacturing Technology*, 56, 1057-1067.

[192] Nemhauser, G. & Wolsey, L. [1998]. Integer and Combinatorial Optimization, *John Wiley & Sons*, *New York.*

[193] Novoa, C. & Storer, R. [2009]. An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, 196, 509-515.

[194] Öncan, T., Altinel, İ. K. & Laporte, G. [2009]. A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers & Operations Research*, 36, 637-654.

[195] Ordóñez, F., Sungur, I. & Dessouky, M. [2007]. A Priori Performance Measures for Arc-Based Formulations of Vehicle Routing Problem. *Transportation Research Record: Journal of the Transportation Research Board*, 2032, 53-62.

[196] Ortega, P., Oliva, C., J. & Ferland Cepeda, M. [2009]. Multiple Ant Colony System for a VRP with Time Windows and Scheduled Loading. *Revista chilena de ingeniería*, 17, 393-403.

[197] Osman, I. H. & Laporte, G. [1996]. Metaheuristics: A bibliography. *Annals of Operations Research*, 63, 513-623.

[198] Padberg, M. & Rinaldi, G. A. [1991]. Brach and Cut algorithm for the resolution of large-scale symmetric Traveling Salesman Problem. *SIAM*, 33, 60-100.

[199] Papadakos, N., Tzallas-Regas, G., Rustem, B. & Thoms, J. [2011]. Risky traveling salesman problem. *European Journal of Operational Research*, 212, 69-73.

[200] Paraskevopoulos, D., Repoussis, P., Tarantilis, C., Ioannou, G. & Prastacos, G. [2008]. A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows. *Journal of Heuristics*, 14, 425-455.

[201] Pereira, F. B. & Tavares, J. [2008]. Bio-inspired algorithms for the vehicle routing problem, *Springer*.

[202] PERFECCINAMIENTO EMPRESARIAL [2007]. Bases del Perfeccionamiento Empresarial en Cuba. La Habana, Cuba. Revisado en 3 de Febrero de 2012 en: http://www.gacetaoficial.cu/.

[203] Prescott-Gagnon, E., Desaulniers, G. & Rousseau, L.-M. [2009]. A branch-and-price-based large neighborhood search algorithm for the vehicle routing problem with time windows. *Networks*, 54, 190-204.

[204] Puris, A., Bello, R. & Herrera, F. [2010]. Analysis of the efficacy of a Two-Stage methodology for ant colony optimization: Case of study with TSP and QAP. *Expert Systems with Applications*, 37, 5443-5453.

[205] Quadrifoglio, L. & Dessouky, M. M. [2008]. Sensitivity Analyses over the Service Area for Mobility Allowance Shuttle Transit (MAST) Services Computer-aided Systems in Public Transport. *In:* Hickman, M., Mirchandani, P. & Voß, S. (eds.). *Springer Berlin Heidelberg*.

[206] Quinlan, J. R. [1979]. Discovering rules by induction from large collections of examples. *In:* Michie D (eds.), *Expert systems in the micro electronic age*. Edinburgh University Press, Edinburgh.

[207] Quinlan, J. R. [1993]. *C4.5 Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann.

[208] Rajan, C. C. A. & Mohan, M. R. [2004]. An evolutionary programming-based tabu search method for solving the unit commitment problem. *Power Systems IEEE Transactions*, 19, 577-585.

[209] Rao, S. S. [2009]. Engineering optimization: theory and practice, *John Wiley & Sons*.

[210] Ravindran, A. & Ravindran, A. R. [2008]. Operations research methodologies, *CRC Press*.

[211] Ravindran, A. R. [2008]. Operations Research Applications, *CRC Press*.

[212] Reimann, M., Doerner, K. & Hartl, R. [2003]. Analyzing a Unified Ant System for the VRP and Some of Its Variants Applications of Evolutionary Computing. *In:* Cagnoni, S., Johnson, C., Cardalda, J., Marchiori, E., Corne, D., Meyer, J.-A., Gottlieb, J., Middendorf, M., Guillot, A., Raidl, G. & Hart, E. (eds.). *Springer Berlin / Heidelberg*.

[213] Resolución 249 de la UDECAM [2005]. “Indicaciones generales para el análisis de la carta de porte y hoja de ruta”. Ciudad Habana, Cuba.

[214] Righini, G. & Salanil, M. [2009]. Decremental state space relaxation strategies and initialization heuristics for solving the rienteering Problem with Time Windows with dynamic programming. *Computers & Operations Research*, 36, 1191-1203.

[215] Ripley, B. D. [2008]. Pattern recognition and neural networks, *Cambridge University Press.*

[216] Ropke, S., & Cordeau, J.-F. [2009]. Branch and Cut and Price for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, 43, 267-286.

[217] Rottkemper, B., Fischer, K. & Blecken, A. [2012]. A transshipment model for distribution and inventory relocation under uncertainty in humanitarian operations. *Socio-Economic Planning Sciences*, 46, 98-109.

[218] Runka, A. [2009]. Ant Colony Optimization with Local Search for the Dynamic Vehicle Routing Problem. *Brock University*.

[219] Savelsbergh, M. & Song, J.-H. [2007]. Inventory routing with continuous moves. *Computers & Operations Research*, 34, 1744-1763.

[220] Scholz, J. [2011]. Real-Time Spatial Optimization: Based on the Application in Wood Supply Chain Management, *GRIN Verlag GmbH*.

[221] Sendas [2009]. La Revista Cubana del Transporte. Ciudad de La habana, Cuba. En http://www.sendasweb.cu/Sendas40/index.htm

[222] Shanmugam, G., Ganesan, P. & Vanathi, P. T. [2010]. A Hybrid Particle Swarm Optimization with Genetic Operator for Vehicle Routing Problem. *Journal of Advances in Information Technology*, 1, 181-188.

[223] Sim, H. P., Day, J., George, A. P., Gifford, T., Nienow, J. & Powell, W. B. [2009]. An Approximate Dynamic Programming Algorithm for Large-Scale Fleet Management: A Case Application. *Transportation Science*, 43, 178-197.

[224] Smale, S. [1998]. Mathematical problems for the next century. *The Mathematical Intelligencer*, 20, 7-15.

[225] Sodhi, M. M. S. & Tang, C. S. [2010]. A Long View of Research and Practice in Operations Research and Management Science: The Past and the Future, *Springer*.

[226] Stephan, S. [2006]. A tabu search heuristic for the truck and trailer routing problem. *Computers & Operations Research*, 33, 894-909.

[227] Stützle, T. & Hoos, H. [2000]. MAX-MIN Ant System. *Future Generation Computer Systems*, 16, 889-914.

[228] Stützle, T. G. [1998]. Local Algorithms for Combinatorial Problems. Analysis, Improvements and New Applications. *PhD thesis*. Technichen Universität Darmstadt, Germany.

[229] Stützle, T., López-Ibáñez, M., Dorigo, M., Cochran, J. J., Cox, L. A., Keskinocak, P., Kharoufeh, J. P. & Smith, J. C. [2010]. A Concise Overview of Applications of Ant Colony Optimization. *Wiley Encyclopedia of Operations Research and Management Science. John Wiley & Sons, Inc.*

[230] Sungur, I., Ordóñez, F. & Dessouky, M. [2008]. A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Transactions*, 40, 509-523.

[231] Tajnsek, V., Pihler, J. & Roser, M. [2011]. Advanced Logistical Systems for the Maintenance of Overhead Distribution Lines Through DCC With the Use of Laser Monitoring. *IEEE Transactions on Power Delivery*, 26, 1337-1343.

[232] Talbi, E.-G. [2009]. Frontmatter. Metaheuristics. *John Wiley & Sons, Inc*.

[233] Tang, L., Liu, J., Rong, A. & Yang, Z. [2000]. A multiple traveling salesman problem model for hot rolling scheduling in Shangai Baoshan Iron & Steel Complex. *European Journal of Operational Research*, 124, 267-282.

[234] Tang, L., Liu, J., Rong, A., Yang, Z. [2000]. A multiple traveling salesman problem model for hot rolling scheduling in Shangai Baoshan Iron & Steel Complex. *European Journal of Operational Research*. 124, 267-282.

[235] Tatarakis, A. & Minis, I. [2009]. Stochastic single vehicle routing with a predefined customer sequence and multiple depot returns. *European Journal of Operational Research*, 197, 557-571.

[236] The symmetric and asymmetric TSPLIB, Traveling Salesman Problem Instances: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/.

[237] Torres Gemeil, M., Mederos Cabrera, B., Daduna, R. & Comas Pullés, R. [2003]. Generalidades sobre logística. Logística: temas seleccionados, Tomo I. Editorial Universitaria.

[238] Torresani, L., Kolmogorov, V. & Rother, C. [2008]. Feature Correspondence Via Graph Matching: Models and Global Optimization Computer Vision – ECCV 2008. In: Forsyth, D., Torr, P. & Zisserman, A. (eds.). Springer Berlin / Heidelberg.

[239] Toth, P. & Vigo, D. [2002]. The vehicle routing problem, *Society for Industrial and Applied Mathematics.*

[240] Tseng, Y., Yue, W. L. & Taylor, M. [2005]. The role of transportation in logistics chain. *Proceedings of the Eastern Asia Society for Transportation Studies*, 5, 1657-1672.

[241] Tseng, Y., Yue, W. L. & Taylor, M. [2005]. The role of transportation in logistics chain. *Proceedings of the Eastern Asia Society for Transportation Studies*, 5, 1657-1672.

[242] Uyar, E. & Technology, G. I. O. [2008]. Routing in stochastic environments, *Georgia Institute of Technology.*

[243] Vanderbeck, F. & Wolsey, L. A. [1996]. An exact algorithm for IP column generation. *Operations Research Letters*, 19, 151-159.

[244] Vapnik, V. N. [1998]. Statistical Learning Theory, *John Wiley, New York*.

[245] Venkataraman, P. [2009]. Applied optimization with MATLAB programming, *John Wiley & Sons*.

[246] Voudouris, C., Owusu, G. & Dorne, R. [2008]. Service chain management: technology innovation for the service business, *Springer.*

[247] Wang, C. & Cheng, H.-Z. [2008]. Optimization of Network Configuration in Large Distribution Systems Using Plant Growth Simulation Algorithm. *IEEE Transactions on Power System*, 23, 119-126.

[248] Wang, C.-H. & Lu, J.-Z. [2009]. A hybrid genetic algorithm that optimizes capacitated vehicle routing problems. *Expert Systems with Applications*, 36, 2921-2936.

[249] Wenning, B. L. [2010]. Context-based Routing in Dynamic Networks, *Vieweg+Teubner Verlag*.

[250] Wilhelm, R., Engblom, J., Ermedahl, A., Holsti, N., Thesing, S., Whalley, D., Bernat, G., Ferdinand, C., Heckmann, R., Mitra, T., Mueller, F., Puaut, I., Puschner, P., Staschulat, J. & Stenström, P. [2008]. The worst-case execution-time problem-overview of methods and survey of tools. *ACM Trans. Embed. Comput. Syst*., 7, 1-53.

[251] Wolpert, D. H. & Macready, W. G. [1997]. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1, 67-82.

[252] Wren, A. [1971]. Computers in Transport Planning and Operation. *Ian Allan, London.*

[253] Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou Z.-H., Steinbach, M., Hand, D. J. & Steinberg, D. [2008]. Top 10 algorithms in data mining. *Knowledge Information System*, 14, 1-37.

[254] Xie, F. & Levinson, D. M. [2011]. Evolving Transportation Networks, *Springer.*

[255] Yang, X. S. [2010]. Engineering Optimization: An Introduction with Metaheuristic Applications, *John Wiley & Sons*.

[256] Yazgi Tütüncü, G., Carreto, C. A. C. & Baker, B. M. [2009]. A visual interactive approach to classical and mixed vehicle routing problems with backhauls. *Omega*, 37, 138-154.

[257] Yi, W. & Kumar, A. [2007]. Ant colony optimization for disaster relief operations. *Transportation Research Part E: Logistics and Transportation Review*, 43, 660-672.

[258] Young, M. F., Murata, K. K., Romero, V. J., Gauntt, R. O. & Rochau, G. E. [2004]. Advanced Nuclear Energy Analysis Technology. *Sandia National Laboratories*. New Mexico, USA.

[259] Zabala A., C. A. [2005]. Implementación del Sistema de Colonia de Hormigas con Búsqueda Local del Problema de Ruteo de Vehículos con Capacidad y Ventanas de Tiempo (CVRPTW). Departamento de Ingeniería Industrial, Universidad de los Andes, Bogotá, Colombia.

[260] Zäpfel, G. & Bögl, M. [2008]. Multi-period vehicle routing and crew scheduling with outsourcing options. *International Journal of Production Economics*, 113, 980-996.

[261] Zapfel, G., Zäpfel, G., Braune, R. & Bögl, M. [2010]. Metaheuristic Search Concepts: A Tutorial with Applications to Production and Logistics, *Springer.*

[262] Zeng, Z. & Wang, J. [2010]. Advances in Neural Network Research and Applications, *Springer.*

[263] Zhang, H.-G. & Wu, Y.-X. [2010]. Study on the Model and Improved Immune Algorithm for Vehicle Routing Problem with Backhauls. *International Conference on Computing, Control and Industrial Engineering*, 371-374.

[264] Zhao, F., Li, S., Sun, J. & Mei, D. [2009]. Genetic algorithm for the one-commodity pickup-and-delivery traveling salesman problem. *Computers & Industrial Engineering*, 56, 1642-1648.

[265] Zhong, Y. & Cole, M. H. [2005]. A vehicle routing problem with backhauls and time windows: a guided local search solution. *Transportation Research Part E: Logistics and Transportation Review*, 41**,** 131-144.

Appendix A.1: Complex variants of the TSP

VARIANTS	PARTICULAR CHARACTERISTICS	AUTHORS
Online Traveling Salesman Problem (OLTSP)	The requests for visits to cities arrive online while the salesman is traveling	Ausiello et al.[2005]
Deadline Traveling Salesman Problem (DLTSP)	A subset of the vertices is given which have deadlines imposed on them	Böckenhauer & Komm [2010]
k-delivery TSP	The problem is to find a shortest tour for the vehicles in which all *pegs* can be transported to their *slots* without exceeding the capacity of the vehicle	Zhao et al. [2009]
Multiple Traveling Salesman Problem (mTSP)	Can be used more than one salesman in the Hamiltonian cycle	Bektas [2006]

Appendix A.2: Exact *vs* approximate methods

	Advantages	Disadvantages
Exact methods	▪ Guarantee the optimal solution ▪ Sometimes are easy to implement	▪ The computational complexity is expressed in a non polynomial function ▪ Considerable computation time in the large-scale problems ▪ Mostly are not capable to work with dynamic variables ▪ Low flexibility
Approximate methods	▪ They often find high quality solution ▪ Are able to successfully attack large instances ▪ Mostly the computation time is small	▪ They cannot guarantee to find optimal ▪ solutions in finite time ▪ Require of high computational knowledge for be implemented ▪ Sometime cannot escape of local optima

Appendix A.3: List of approximate algorithms

	Algorithms
Heuristics	▪ Nearest Neighbor (NN) ▪ Insertion ▪ Saving ▪ Tour improvement ▪ Sweep ▪ Two-phase ▪ Local Search Algorithms (LSAs)
Metaheuristics	▪ Ant Colony Optimization (ACO) ▪ Evolutionary programming ▪ Genetic Algorithms (GAs) ▪ Particle Swarm Optimization (PSO) ▪ Bee Algorithms (BAs) ▪ Fish Algorithms (FAs) ▪ Tabu Search (TS) ▪ Simulated Annealing (SA) ▪ Artificial Immune System (AIS) ▪ Artificial Neural Networks (ANN)
Approximation	▪ Continuous approximation [Daganzo, 1984] ▪ Metric Steiner Tree ▪ MST-based algorithm ▪ A simple factor 2 algorithm ▪ Metric TSP – Factor 3/2
Trial and error	▪ Simulation

Appendix A.4: Formal structure of the Knowledge Base

SAMPLES	Dependent variable		Predictive variables									
	Method (Y)		X_1	Time windows (X_2)		X_3	Fleet type (X_4)		Number of objectives (X_5)		Customer demand (X_6)	
	Approximate	Exact	Number of nodes	With time windows	Without time windows	Fleet size	Homogeneous	Heterogeneous	One-objective	Multi-objective	Deterministic	Stochastic
Rizzoli et al. [2007] (ACO)	x		300	x		100		x	x		x	
	x		600	x		100		x	x		x	
	x		500	x		10		x	x		x	
Anbuudayasankar et al. [2012] (GA)	x		100		x	5	x			x	x	
Belenguer et al. [2005] (TS)	x		94	x		7		x		x		x
	x		114	x		7		x		x		x
	x		122	x		7		x		x		x
	x		124	x		7		x		x		x
	x		148	x		7		x		x		x
	x		116	x		7		x		x		x
	x		123	x		7		x		x		x
Jung & Karney [2006] (PSO)	x		123		x	1	x			x	x	
Bräysy [2002] (LS)	x		100		x	1	x		x		x	
	x		200		x	1	x		x		x	
	x		400		x	1	x		x		x	

SAMPLES	**Dependent variable**		**Predictive variables**									
	Method (Y)		**X_1**	**Time windows (X_2)**		**X_3**	**Fleet type (X_4)**		**Number of objectives (X_5)**		**Customer demand (X_6)**	
	Approximate	**Exact**	**Number of nodes**	**With time windows**	**Without time windows**	**Fleet size**	**Homogeneous**	**Heterogeneous**	**One-objective**	**Multi-objective**	**Deterministic**	**Stochastic**
Bredstrãm et al. [2004] (BB)		x	55	x		1	x			x	x	
Caballero et al. [2007] (TS)	x		93		x	5	x			x	x	
Chen et al. [2009] (CP)		x	75		x	6	x			x	x	
Cordeau & Laporte [2003] (TS)	x		144	x		13	x		x			x
Cornillier et al. [2007] (BB)		x	44		x	24	x		x		x	
Di Pierro et al. [2009] (ET)	x		535		x	50	x		x		x	
Donati et al. [2008] (ACO)	x		60		x	10		x		x	x	
Duman et al. [2007] (LS)	x		44		x	4	x		x		x	
Faulin [2003] (SH)	x		100		x	7	x		x		x	
Faulin [2011] (LIP)		x	34		x	7	x		x		x	
Ioannou [2001] (LS)	x		1943	x		72	x		x		x	
Garcia-Najera [2009] (GA)	x		400		x	15	x			x	x	
Goetschalckx et al. [2002] (LIP)		x	80		x	10	x			x	x	
Hu et al. [2009] (BB)		x	207		x	50	x		x		x	
Pacheco & Delgado [1999] (TS)	x		40	x		1		x	x		x	

SAMPLES	**Dependent variable**		**Predictive variables**									
	Method (Y)		**X_1**	**Time windows (X_2)**		**X_3**	**Fleet type (X_4)**		**Number of objectives (X_5)**		**Customer demand (X_6)**	
	Approximate	**Exact**	**Number of nodes**	**With time windows**	**Without time windows**	**Fleet size**	**Homogeneous**	**Heterogeneous**	**One-objective**	**Multi-objective**	**Deterministic**	**Stochastic**
Tomalá Pincay [2010] (ACO)	x		66	x		4	x		x		x	
Martin el at. [2010] (ACO) and (BB)	x		106	x		10		x	x		x	
			412	x		25		x	x		x	
		x	20		x	1		x	x		x	
López Pérez & Badii (2005) (GA) and (LIP)		x	70	x		1	x		x		x	
	x		120	x		1	x		x		x	
Yepes & Medina [2004]	x		100	x		20		x		x	x	
Jozefowiez et al. [2008] (SA), (TS), (GA) and (GP)	x		150	x		1		x		x	x	
	x		200	x		20		x		x	x	
	x		140	x		15		x		x	x	
	x		400		x	50		x		x	x	
		x	50		x	1		x		x	x	
	x		150	x		15		x		x	x	
Kaveh & Nasr [2011] (LS)	x		163		x	10	x		x		x	
Kim et al. [2006] (IH)	x		102	x		5		x		x		x

SAMPLES	Dependent variable		Predictive variables									
	Method (Y)		X_1	Time windows (X_2)		X_3	Fleet type (X_4)		Number of objectives (X_5)		Customer demand (X_6)	
	Approximate	Exact	Number of nodes	With time windows	Without time windows	Fleet size	Homogeneous	Heterogeneous	One-objective	Multi-objective	Deterministic	Stochastic
Kim et al. [2006] (IH)	x		277	x		5		x		x		x
	x		335	x		5		x		x		x
	x		444	x		5		x		x		x
	x		804	x		5		x		x		x
	x		1051	x		5		x		x		x
	x		1351	x		5		x		x		x
	x		1599	x		5		x		x		x
	x		1932	x		5		x		x		x
	x		2100	x		5		x		x		x
Laporte & Norbert [1980] (CP)		x	20		x	4	x		x		x	
Rathinam et al. [2007] (BB)		x	15		x	3	x		x		x	
Mester et al. [2007] (ET)	x		50		x	54	x		x		x	
	x		75		x	55	x		x		x	
	x		100		x	55	x		x		x	
	x		150		x	55	x		x		x	

SAMPLES	Dependent variable		Predictive variables									
	Method (Y)		X_1	Time windows (X_2)		X_3	Fleet type (X_4)		Number of objectives (X_5)		Customer demand (X_6)	
	Approximate	**Exact**	**Number of nodes**	**With time windows**	**Without time windows**	**Fleet size**	**Homogeneous**	**Heterogeneous**	**One-objective**	**Multi-objective**	**Deterministic**	**Stochastic**
Mester et al. [2007] (ET)	x		385	x		55	x		x		x	
	x		417	x		54	x		x		x	
Montalvo et al. [2010] (PSO)	x		240	x		-	-		x		x	
Norback [1991] (IH)	x		63		x	7	x		x		x	
	x		308		x	24	x		x		x	
Nuortio et al. [2006] (LS)	x		82		x	1	x		x		x	
Rodríguez & Zamakola [2000] (LS)	x		170	x		25	x		x		x	
Ombuki et al. [2006] (GA)	x		100	x		10		x		x	x	
Ozfirat et al. [2010] (LIP)		x	41		x	9		x		x	x	
Penna et al [2007] (LS)	x		20		x	1		x	x		x	
	x		50		x	1		x	x		x	
	x		75		x	1		x	x		x	
	x		100		x	1		x	x		x	
Polacek et al. [2007] (LS)	x		293		x	23	x			x	x	
	x		173		x	20	x			x	x	

SAMPLES	Dependent variable		Predictive variables									
	Method (Y)		X_1	Time windows (X_2)		X_3	Fleet type (X_4)		Number of objectives (X_5)		Customer demand (X_6)	
	Approximate	**Exact**	**Number of nodes**	**With time windows**	**Without time windows**	**Fleet size**	**Homogeneous**	**Heterogeneous**	**One-objective**	**Multi-objective**	**Deterministic**	**Stochastic**
Polacek et al. [2007] (LS)	x		287		x	21	x			x	x	
	x		175		x	22	x			x	x	
	x		283		x	22	x			x	x	
	x		175		x	21	x			x	x	
	x		238		x	23	x			x	x	
	x		136		x	21	x			x	x	
	x		279		x	22	x			x	x	
	x		174		x	23	x			x	x	
	x		278		x	20	x			x	x	
	x		168		x	23	x			x	x	
Saadatseresht et al. [2009] (ET)	x		118	x		-		x		x	x	
Semet [1994] (TS)	x		5000	x		14		x	x		x	
Seyedhosseini [2010] (PSO)	x		82		x	100	x		x		x	
Shahrzad [2011] (PSO)	x		60	x		15	x		x			x
Suthikarnnarunai [2008] (BB)		x	75		x	12		x	x		x	

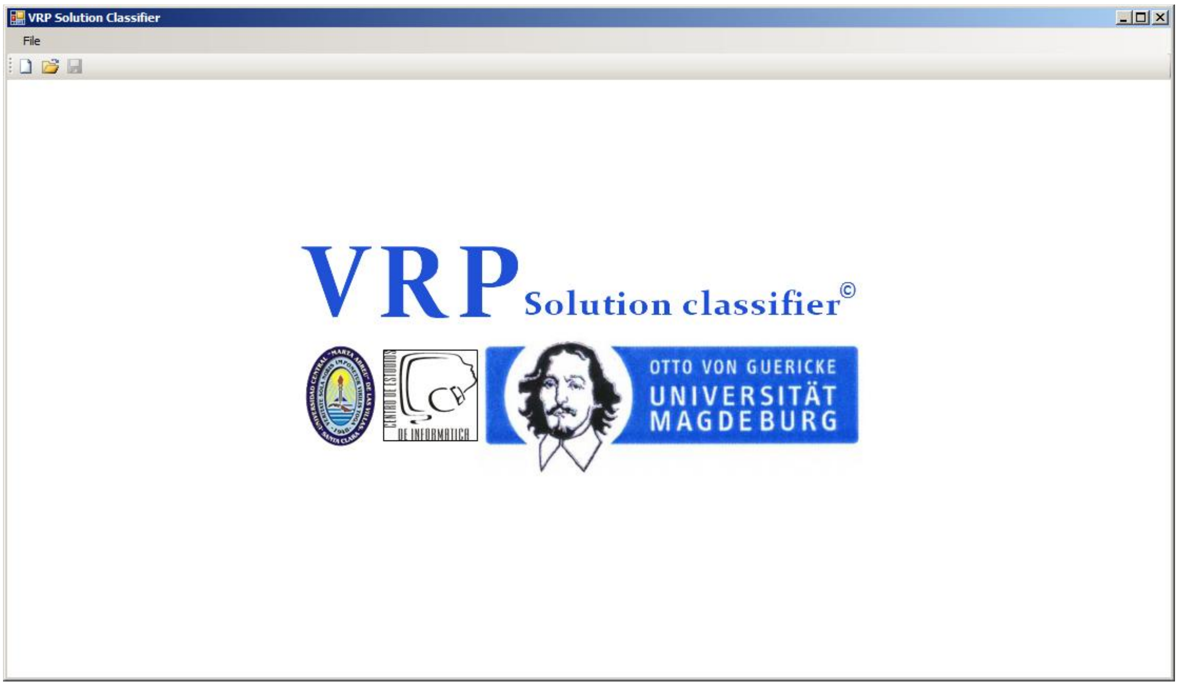
SAMPLES	**Dependent variable**		**Predictive variables**									
	Method (Y)		**X_1**	**Time windows (X_2)**		**X_3**	**Fleet type (X_4)**		**Number of objectives (X_5)**		**Customer demand (X_6)**	
	Approximate	**Exact**	**Number of nodes**	**With time windows**	**Without time windows**	**Fleet size**	**Homogeneous**	**Heterogeneous**	**One-objective**	**Multi-objective**	**Deterministic**	**Stochastic**
Tarantilis & Kiranoudis [2001] (LS)	x		299		x	23		x	x		x	
Tarantilis & Kiranoudis [2002a)b)] (LS)	x		174		x	12	x		x		x	
	x		100		x	13		x	x		x	
Tarantilis & Kiranoudis [2007] (LS)	x		300		x	27		x	x		x	
Tzeng et al. [2007] (BB)		x	48		x	3		x		x	x	
Cardoen et al. [2009] (BP)		x	20	x		1	x			x	x	
		x	25	x		1	x			x	x	
		x	30	x		1	x			x	x	
		x	35	x		1	x			x	x	
		x	40	x		1	x			x	x	
		x	45	x		1	x			x	x	
		x	50	x		1	x			x	x	
		x	55	x		1	x			x	x	
		x	60	x		1	x			x	x	
		x	65	x		1	x			x	x	

SAMPLES	Dependent variable		Predictive variables									
	Method (Y)		X_1	Time windows (X_2)		X_3	Fleet type (X_4)		Number of objectives (X_5)		Customer demand (X_6)	
	Approximate	Exact	Number of nodes	With time windows	Without time windows	Fleet size	Homogeneous	Heterogeneous	One-objective	Multi-objective	Deterministic	Stochastic
Cardoen et al. [2009] (BP)		x	70	x		1	x			x	x	
Cardoen et al. [2009] (BP)		x	75	x		1	x			x	x	
Cardoen et al. [2009] (BP)		x	80	x		1	x			x	x	
Charfeddine & Montreuil [2010] (ACO)	x		100	x		15		x		x		x
Charfeddine & Montreuil [2010] (ACO)	x		200	x		20		x		x		x
Charfeddine & Montreuil [2010] (ACO)	x		400	x		35		x		x		x
Charfeddine & Montreuil [2010] (ACO)	x		600	x		40		x		x		x
Charfeddine & Montreuil [2010] (ACO)	x		1000	x		50		x		x		x
Chevrier et al. [2009] (ET)	x		100	x		10		x		x		x
Chevrier et al. [2009] (ET)	x		1000	x		100		x		x		x
Hsu et al. [2007] (IH)	x		71	x		9		x	x			x
Montemanni et al. [2005]	x		50		x	50	x		x			x
Montemanni et al. [2005]	x		199		x	50	x		x			x
Rasmussen et al. [2012] (BP) and (IH)	x		150	x		15	x		x		x	
Rasmussen et al. [2012] (BP) and (IH)	x		107	x		8	x		x		x	

SAMPLES	Dependent variable		Predictive variables									
	Method (Y)		X_1	Time windows (X_2)		X_3	Fleet type (X_4)		Number of objectives (X_5)		Customer demand (X_6)	
	Approximate	Exact	Number of nodes	With time windows	Without time windows	Fleet size	Homogeneous	Heterogeneous	One-objective	Multi-objective	Deterministic	Stochastic
Rasmussen et al. [2012] (BP) and (IH)		x	60	x		7	x		x		x	
		x	61	x		6	x		x		x	
		x	20	x		4	x		x		x	
		x	50	x		10	x		x		x	
	x		80	x		16	x		x		x	
Sol [1998] (BP)		x	50	x		5	x		x		x	
Solnon et al. [2008] (ACO) and (GA)	x		704		x	84	x		x		x	
	x		1260		x	202	x		x		x	
	x		1319		x	275	x		x		x	
	x		996		x	235	x		x		x	
	x		325		x	150	x		x		x	
	x		65		x	10	x		x		x	
	x		780		x	73	x		x		x	
	x		931		x	24	x		x		x	
	x		231		x	64	x		x		x	

SAMPLES	Dependent variable		Predictive variables									
	Method (Y)		X_1	Time windows (X_2)		X_3	Fleet type (X_4)		Number of objectives (X_5)		Customer demand (X_6)	
	Approximate	**Exact**	**Number of nodes**	**With time windows**	**Without time windows**	**Fleet size**	**Homogeneous**	**Heterogeneous**	**One-objective**	**Multi-objective**	**Deterministic**	**Stochastic**
Solnon et al. [2008] (ACO) and (GA)	x		90		x	11	x		x		x	
	x		376		x	19	x		x		x	
	x		1247		x	328	x		x		x	
	x		1037		x	156	x		x		x	
	x		519		x	209	x		x		x	
	x		459		x	141	x		x		x	
	x		875		x	156	x		x		x	
	x		273		x	42	x		x		x	
	x		264		x	19	x		x		x	
	x		219		x	18	x		x		x	
Oppen et al. [2010] (CG)		x	100	x		10	x		x			x
Dumas et al. [1991] (BP)		x	100		x	5		x	x		x	
Bélanger et al. [2006] (BB) and (BP)		x	250	x		25	x		x			x
Calvete et al. [2007] (GP)		x	75	x		1	x			x	x	
Ropke & Cordeau [2009] (BP)		x	100	x		13	x		x		x	

Appendix A.5: The software VRP solution classifier

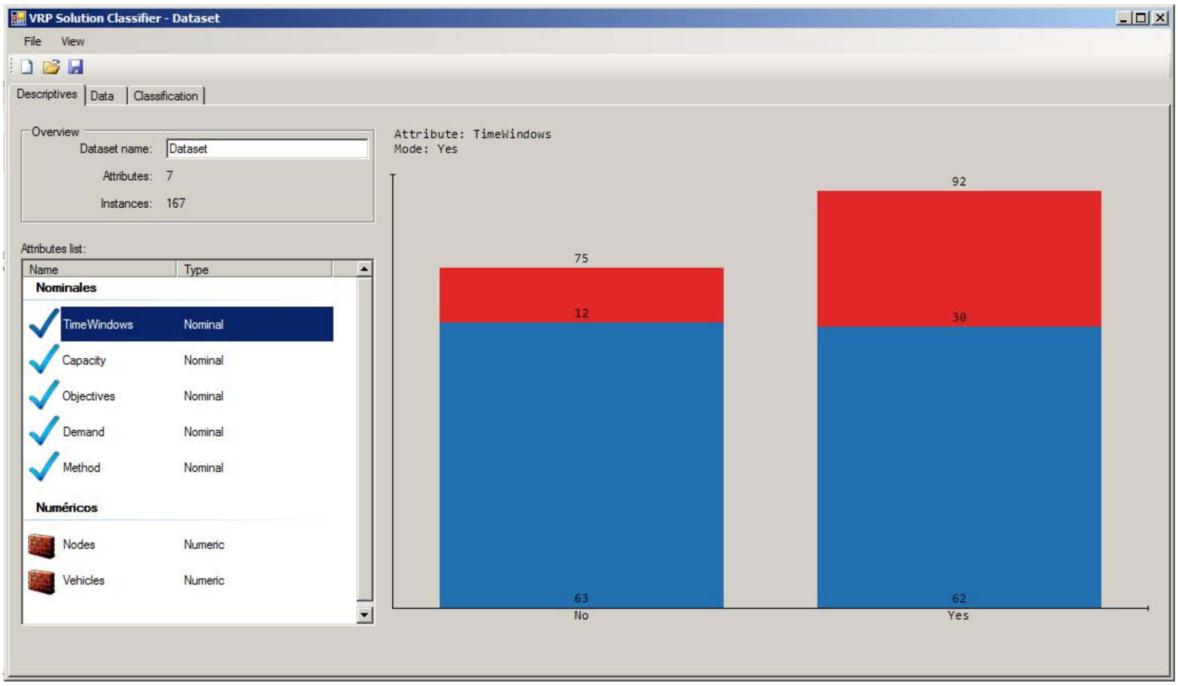


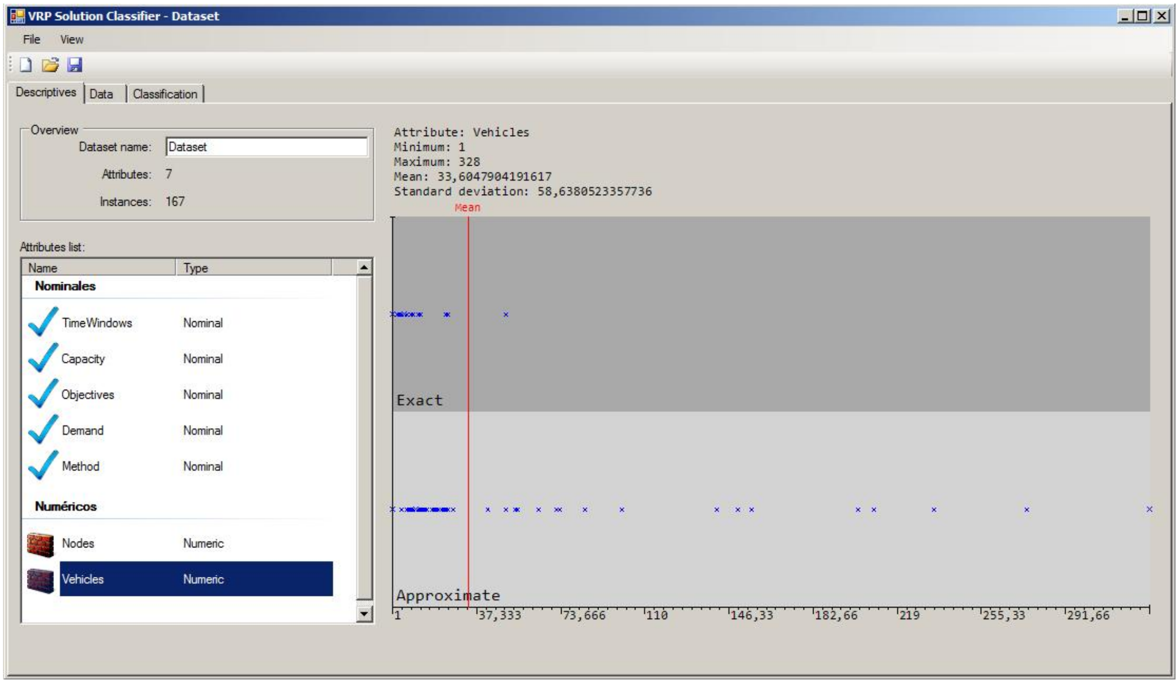
Motivation and description: The VRP solution classifier is a computational implementation, which provides the reliable prediction of the optimization categories (exact or approximate), and its relative specific algorithms for a given VRP instance. The software has encoded two accurate classification algorithms: the Discriminat Analysis and C4.5. For the training of the classifiers is defined a Knowledge Base in which are included valuable solutions of the real-life VRPs. The application has the following utilities:

1) Add and delete the samples in the KB.
2) Obtain descriptive statistics of all variables.
3) Visualization of analytical classification results.
4) Visualization of classification tree.
5) Obtain the classification quality using *cross-validation*

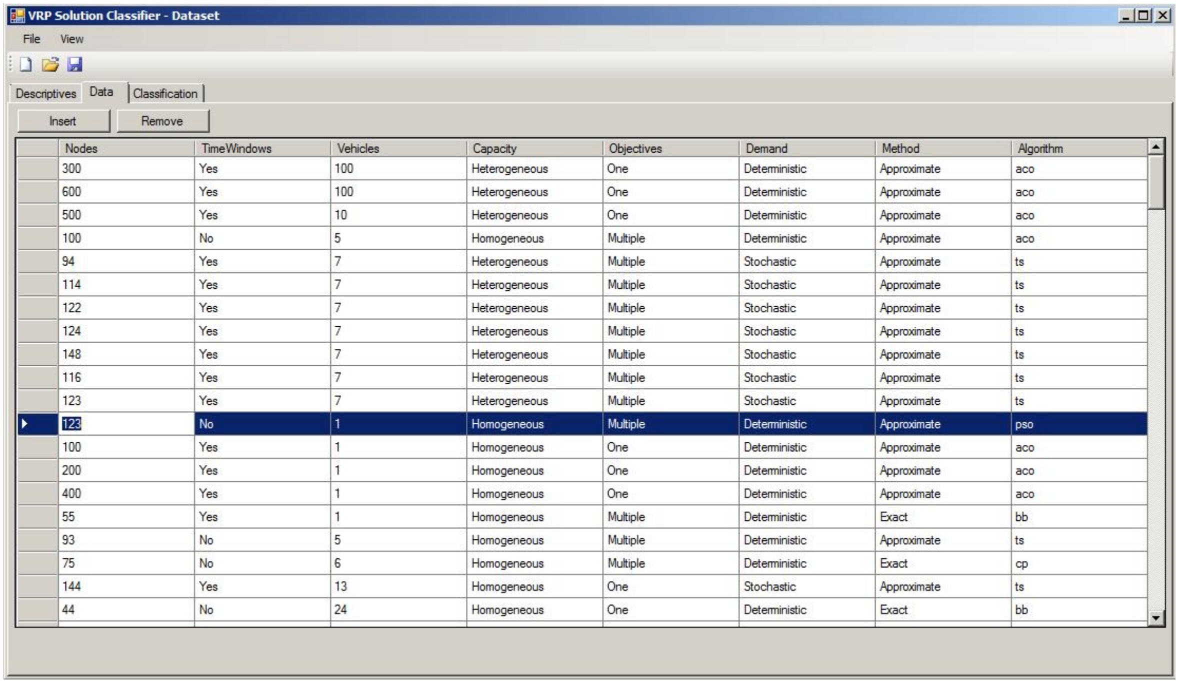
The software consists of two modules: the Descriptives-Data and the Classification module. Both modules with its utilities are described below.

Descriptives-Data module: This module presents the descriptive statistics associated to any variables (predictive or classification variables) of the classification process. It is able to show the descriptive analysis both discrete and continuous variables.



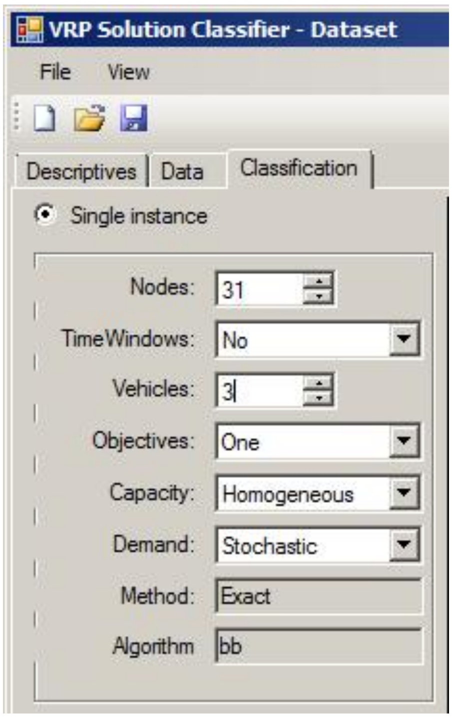


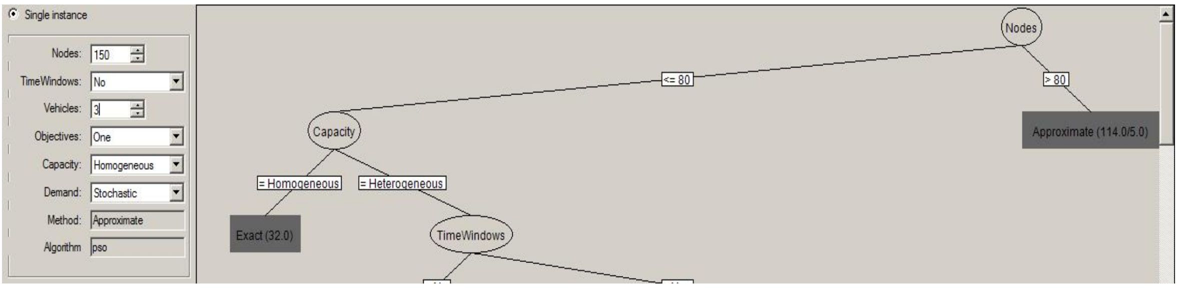
Command Insert-Remove: Both commands operate directly over the samples in the KB, inserting the new case, in which should be defined the values of predictive and dependent variables. In case of errors in the data inputs, this command provides the way to eliminate any sample.



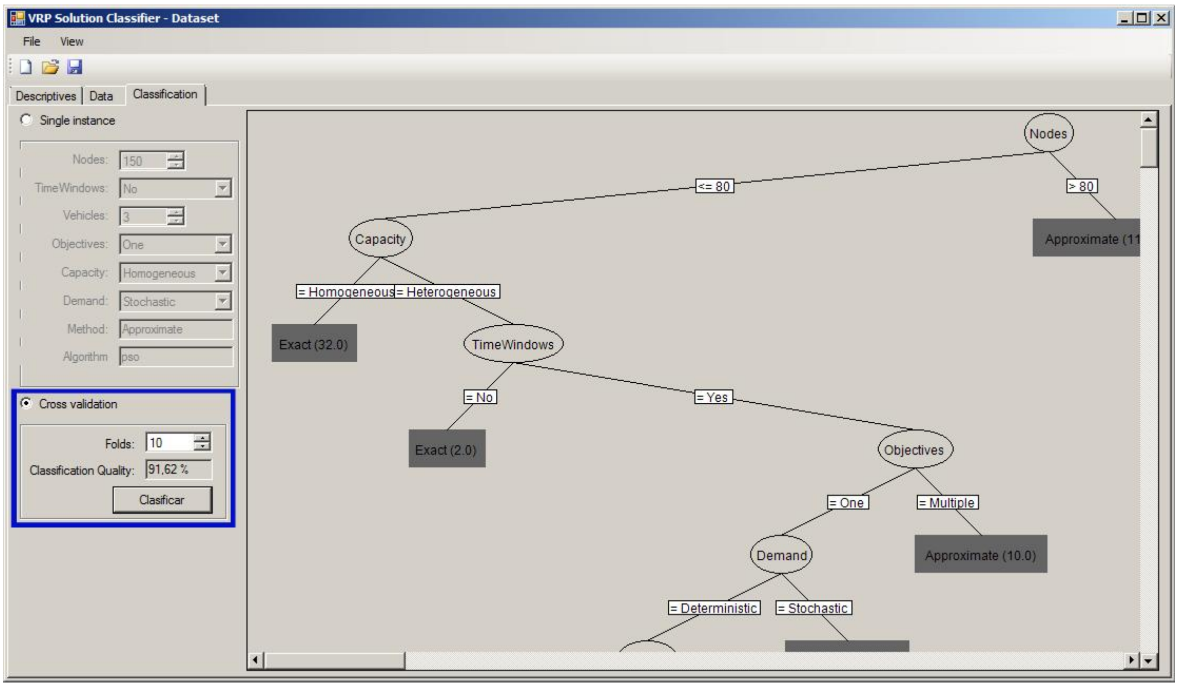
Nodes	TimeWindows	Vehicles	Capacity	Objectives	Demand	Method	Algorithm
300	Yes	100	Heterogeneous	One	Deterministic	Approximate	aco
600	Yes	100	Heterogeneous	One	Deterministic	Approximate	aco
500	Yes	10	Heterogeneous	One	Deterministic	Approximate	aco
100	No	5	Homogeneous	Multiple	Deterministic	Approximate	aco
94	Yes	7	Heterogeneous	Multiple	Stochastic	Approximate	ts
114	Yes	7	Heterogeneous	Multiple	Stochastic	Approximate	ts
122	Yes	7	Heterogeneous	Multiple	Stochastic	Approximate	ts
124	Yes	7	Heterogeneous	Multiple	Stochastic	Approximate	ts
148	Yes	7	Heterogeneous	Multiple	Stochastic	Approximate	ts
116	Yes	7	Heterogeneous	Multiple	Stochastic	Approximate	ts
123	Yes	7	Heterogeneous	Multiple	Stochastic	Approximate	ts
123	No	1	Homogeneous	Multiple	Deterministic	Approximate	pso
100	Yes	1	Homogeneous	One	Deterministic	Approximate	aco
200	Yes	1	Homogeneous	One	Deterministic	Approximate	aco
400	Yes	1	Homogeneous	One	Deterministic	Approximate	aco
55	Yes	1	Homogeneous	Multiple	Deterministic	Exact	bb
93	No	5	Homogeneous	Multiple	Deterministic	Approximate	ts
75	No	6	Homogeneous	Multiple	Deterministic	Exact	cp
144	Yes	13	Homogeneous	One	Stochastic	Approximate	ts
44	No	24	Homogeneous	One	Deterministic	Exact	bb

Classification module: In this module is possible to carry out the classification process, showing the main results according two visualization forms. The first, displays the classification results in an interface text box, while the second presents the classification results displaying a decision tree.





Command Cross-validation: The estimation of the classification quality is based on the cross-validation method. For the proper application of cross-validation is required defining the folds size, which is possible to perform in a text box of the module interface. Thus, a percentage value of the classification quality is obtained.



Appendix A.6: Methodology for minimum training-set size

1) Subsampling and significance permutation test
   A. Subsampling procedure
      i. Given l_{C1} samples from class 1(exact category) and l_{C2} samples from class 2 (approximate category), the total number of samples is $l = l_{C1} + l_{C2}$, where $1 \geq 10$.
      ii. Select 10 training-set sizes $n_1, \dots, n_j, \dots, n_{10}$) over the interval $[10, l - 10]$.
         1. For each training-set size n_j run the following subsampling procedure $T_1 = 50$ times, indexed by $i = 1, \dots, T_1$.
            a) Randomly split the dataset into training-set with n_j samples and a validation-set with $l - n_j$ samples subject to the requirement that $\frac{n_{c2}}{n_{c1}} \approx \frac{l_{c2}}{l_{c1}}$ where n_{c2} and n_{c1} are the number of samples from the class 2 and 1 in the training-set. Call the two dataset generated $S_{n,i}$.
            b) Train the classifiers on each of the training-sets and measure its error rate on its corresponding validation-set; call each of these error rates $e_{n,i}$.
   B. Permutation test
      i. For each subsampled train/validation split $S_{n,i}$, run the following permutation procedure $T_2 > 50$ times indexed by $j = 1, \dots, T_2$.
         1. Randomly permute the labels of the samples in the training-set (leave the validation-set alone); call the dataset generated $S_{n,i,j}^{ran}$.
         2. Train the classifiers on the training-set and measure its error on the validation-set, call this error rate $e_{n,i,j}^{ran}$.
   C. Significance calculation
      1. For each training-set size n, construct an empirical distibution function from the error rates of the permuted dataset $P_n^{ran}(x) = \frac{1}{T_1 \times T_2} \sum_{i=1}^{T_1} \sum_{j=1}^{T_2} \theta\left(x - e_{n,i,j}^{ran}\right)$, where $\theta(z) = 1$ if $z \geq 0$ and 0 otherwise.
      2. Given the above empirical distribution function, compute for each $\bar{e}_n$ the value $t_n = P_n^{ran}(\bar{e}_n)$; statistical significance with respect to an α-value of p is achieved for n_0, the smallest n for which $t_n < p$.
2) Learning curve and training-set estimation
   A. Assume the subsampling procedure was run for M different simple sizes n, indexed by $l - 1, \dots, M$; take th sequence of error rates and compute the following quantities for each training-set size $n > n_0$ for which the classifier passed the significance test ($t_n < p$): the mean error rate $\bar{e}_n = \frac{1}{T_1} \sum_{i=1}^{T_1} e_{n,i}$, the 25th and 75th quantiles of the vector of error rates $\{e_{n,1}, \dots e_{n,T_1}\}$
   B. Use the above quantities to fit the following learning curve:
      i. Given training-set sizes n_l and mean error rates $\bar{e}_{n_l}$, compute α, a, b via the following minimization procedure: $min_{\alpha,a,b} \sum_{l=1}^{M} \left(a n_l^{-\alpha} + b - \bar{e}_{n_l}\right)^2$ subject to

$\alpha, a, b \geq 0$; designate the values as α, a, b as α_m, a_m, b_m. The resulting curve estimates the error rate as a function of a training-set size

$$L_m: e(n) = a_m n^{-\alpha_m} + b_m.$$

ii. Repeat the above procedure for the 25th and the 75th quantiles of the vector of error rates $\{e_{n,1}, \ldots e_{n,T_1}\}$.

Appendix A.7: Estimation results of minimum training-set size

Table A. Estimation results for the minimum training-set size classifying with Discriminant Analysis

Size of KB	Training-set sizes	Training-set which passed the significance test (p-value < 0.05)	Learning curve function	Decision
215	15 30 45 60 75 90 105 120 150 180	For $n \geq 60$	$error(n) = 1.43n^{-0.66} + 0.011$	Dataset is large enough for achieve a low error rate ($< 10\%$), training the classifier with at least 105 samples as the training-set

Table B. Estimation results for the minimum training-set size classifying with C4.5 algorithm

Size of KB	Training-set sizes	Training-set which passed the significance test (*p-value* < 0.05)	Learning curve function	Decision
215	15 30 45 60 75 90 105 120 150 180	For $n \geq 45$	$error(n) = 1.31n^{-0.69} + 0.009$	Dataset is large enough for achieve a low error rate ($< 10\%$), training the classifier with at least 60 samples as the training-set

Appendix A.8: Parametric statistical test

t-student (2 samples)
$X \sim N(\mu_x; \sigma^2_x)$ $Y \sim N(\mu_y; \sigma^2_y)$ ▪ Hypothesis: $H_0: \mu_x = \mu_y$ $H_1: \mu_x \neq \mu_y$ ▪ Statistician: $t = \dfrac{\bar{x} - \bar{y}}{S_p \cdot \sqrt{\dfrac{1}{n_x} + \dfrac{1}{n_y}}}$ ▪ Critical region of acceptance: $\lvert t \rvert > t_{\frac{\alpha}{2}; n_1 + n_2 - 2}$
$\bar{x}, \bar{y}$: Mean of sample "*x*" and "*y*" (mean of the performance in the algorithms "*x*" and "y"). S_p: Standard deviation of both samples. n_x, n_y: Sizes of the samples "*x*" and "*y*".

Design of experiment (2, 3 and *k* samples)		
Experiment	**Commercial software**	**Reference**
One-Way Analysis of Variance (ANOVA)	SPSS	Black [2011]
Two-Way ANOVA	SPSS, MINITAB	Black [2011] and Larson-Hall [2009]
Factorial Analysis	STATGRAPHICS	Eriksson [2008]

Appendix A.9: Nonparametric statistical test

Mann-Whitney U test (two independent samples)
▪ Hypothesis: H_0: $\mu_1 = \mu_2$ H_1: $\mu_1 \neq \mu_2$ ▪ Statistician: $U = n_1 \cdot n_2 + \frac{n_1 \cdot (n_1 + 1)}{2} - R_1$ ▪ Critical region of acceptance: $U \leq -Z_{1-\alpha}$
R_1: Rank of the group 1. n_1: Size of the group 1. n_2: Size of the group 2.

Wilcoxon Signed-Rank test (two related samples)
▪ Hypothesis: H_0: Population (algorithm performance) are identical H_1: Population are not identical ▪ Statistician: $R_+ + R_- = \frac{n \cdot (n + 1)}{2}$ ▪ Critical region of acceptance: $R^+ \leq T_{\alpha,n}$
n: Number of pairs. R_+*:* The sum of the ranks with positive difference. R_- : The sum of the ranks with positive difference.

Friedman test (k related samples)
▪ Hypothesis: H_0: $\mu_1 = \mu_2 = \ldots \mu_k$ H_1: Al least one μ_k differs ▪ Statistician: $$F = \frac{12}{n \cdot k \cdot (k+1)} \sum_{j=1}^{k} R_j^2 - 3 \cdot n \cdot (k+1)$$ ▪ Critical region of acceptance: $$F \geq \chi^2_{\alpha;k-1}$$
n: Size of the groups. k: Number of group. R_j : Total of the rank for the group “j”.

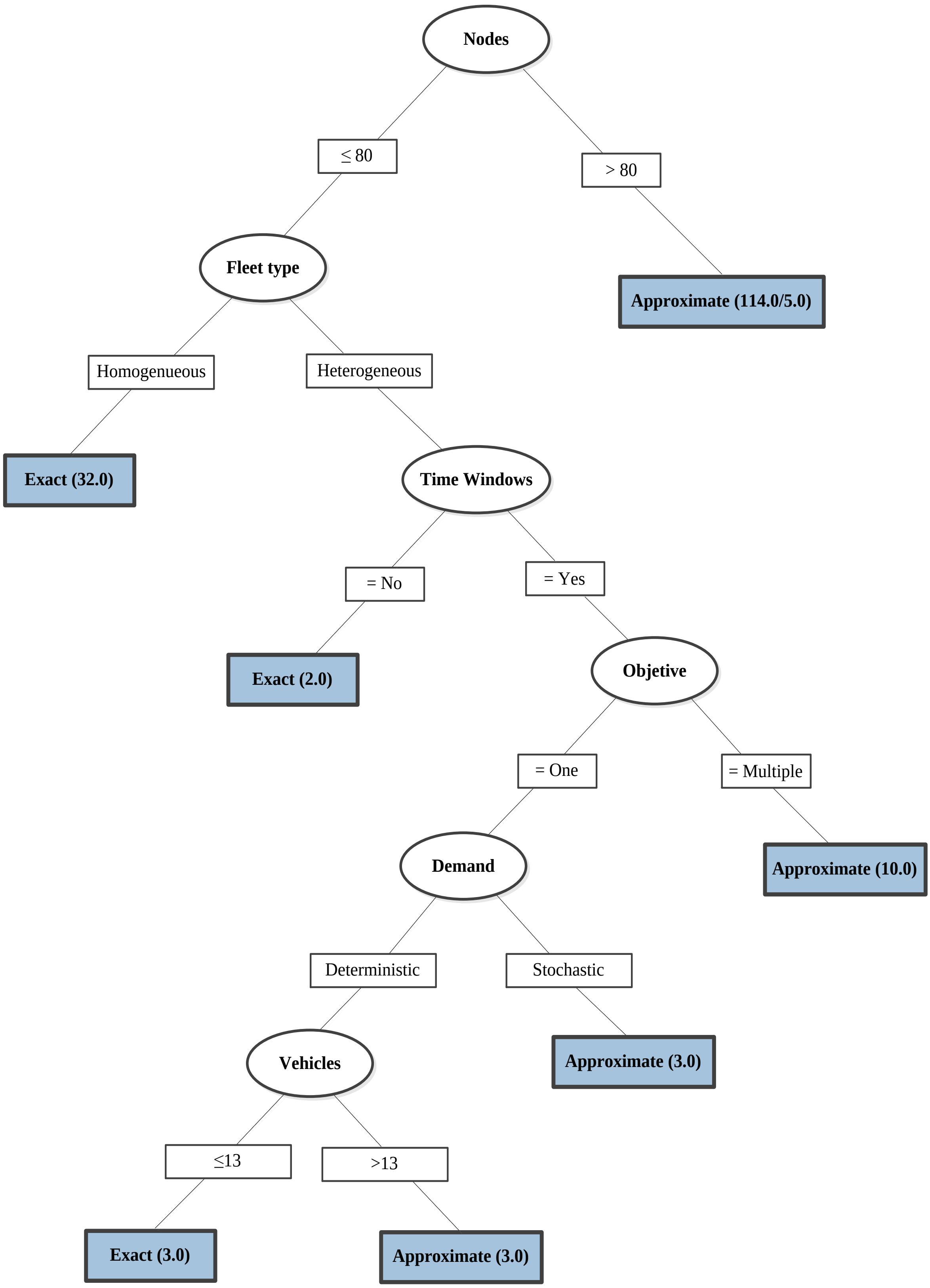
Appendix A.10: Decision variants in the post-optimization

Decision variants	Solution quality (SQ)	Computation time (CT)	Total available time	Relevant decision
Classification = Exact, two exact algorithms [Vart1]	Equal	Significant differences (SD)	Significantly higher than CT	Select the exact algorithm with less CT
Classification = approximate, two approximate algorithms [Vart2]	(SD)	No-significant differences (NSD)	NSD respect CT	Choose the algorithm with the best SQ
Same results as above [Vart3]	NSD	SD	NSD respect CT	Choose the algorithm with the best CT
Same results as above [Vart4]	SD	SD	NSD respect CT	Choose using the multi-criterion technique
Classification = Exact, one exact and one approximate [Vart5]	NSD	SD	NSD respect CT	Choose any algorithm
Same results as above [Vart6]	SD	NSD	NSD respect CT	Choose the exact algorithm
Same results as above [Vart7]	SD	NSD	SD respect CT	Choose the exact algorithm
Same results as above [Vart8]	SD	SD	NSD respect CT	Choose the exact algorithm
Same results as above [Vart9]	SD	SD	SD respect CT	Choose the approximate algorithm
Classification = approximate, one exact, one approximate [Vart10]	NSD	SD	NSD respect CT	Choose the approximate algorithm
Same results as above [Vart11]	NSD	SD	SD respect CT	Choose the approximate algorithm
Same results as above [Vart12]	SD	NSD	NSD respect CT	Choose the exact algorithm
Same results as above [Vart13]	SD	SD	SD respect CT	Choose using the multi-criterion technique

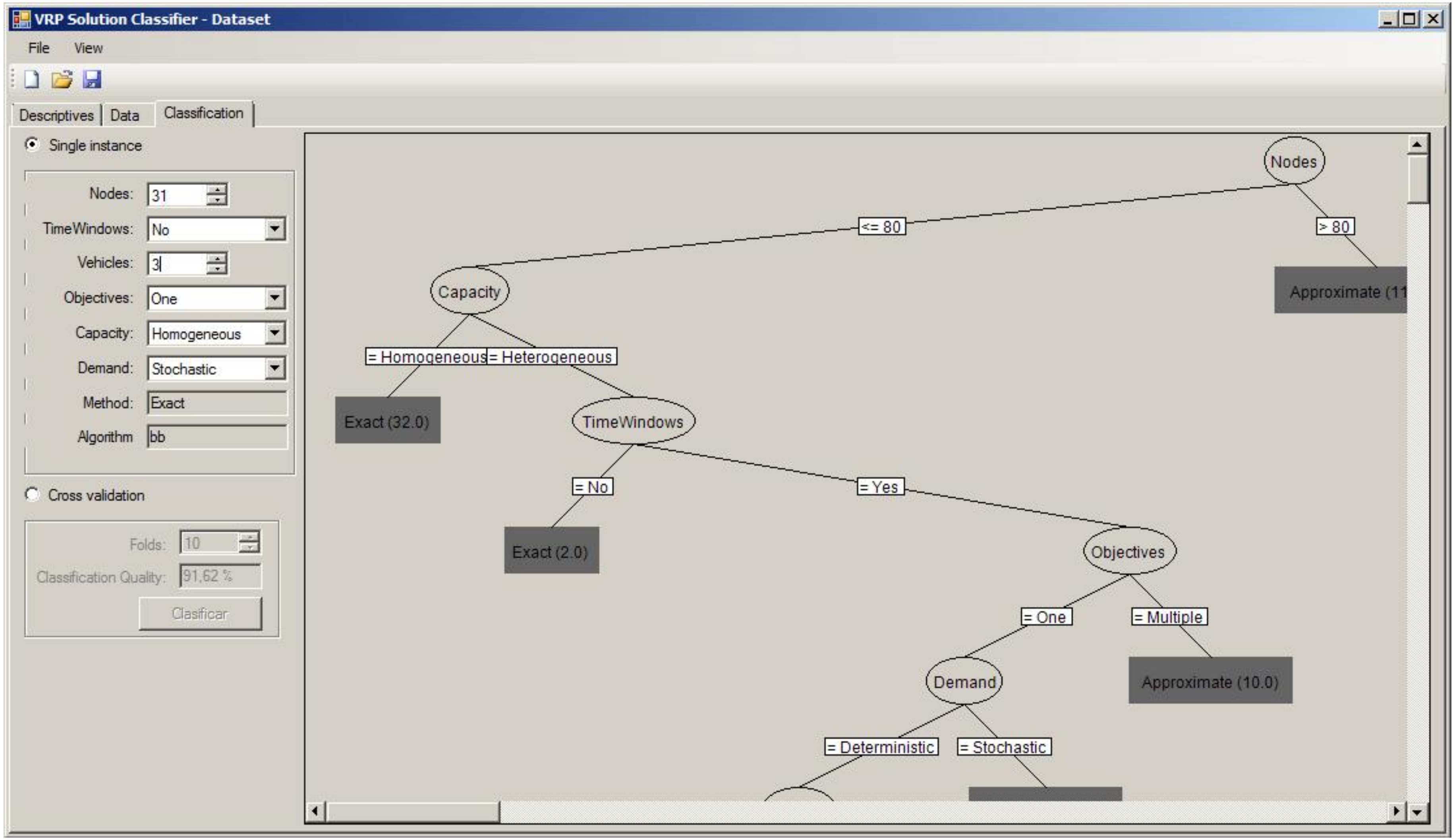
Appendix A.11: Distance matrix between breakdowns

Voltaje	km	P	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
	P	…..	1.174	5.502	3.701	7.774	20.41	3.421	3.881	12.43	1.802	7	3.621	1.822	4.362	4.25	18.98	2.805	0.958	1.949
2	1		……	4.46	2.8	6.85	21	2.4	2.84	12.25	1.62	6.98	2.65	0.73	3.41	3.2	17.85	1.78	1.56	0.94
2	2			….	1.773	2.302	18.89	2.107	3.851	14.01	6.412	11.845	2.218	4.617	5.687	3.807	12.34	5.476	4.523	4.163
3	3				……	4.06	18.2	0.35	2.08	13.13	4.22	9.69	0.46	2.86	4	2.05	12.95	3.67	2.81	2.6
2	4					…..	21.192	4.409	6.153	16.31	8.714	14.147	4.52	6.919	7.989	6.109	11.02	7.778	6.825	6.465
3	5						…….	18.55	20.28	15.6	21.77	26.76	18.66	20.46	21.97	20.25	18.34	21.4	19.44	20.16
3	6							….	1.728	12.76	3.929	9.363	0.338	2.49	3.624	1.67	14.43	3.4	2.748	2.052
2	7								….	13.73	4.45	9.79	1.79	2.26	2.14	0.37	13.33	1.74	2.88	2.69
3	8									…..	13.19	17.971	13.098	13.505	15.11	13.932	26.349	14.408	12.142	13.056
3	9										….	5.68	4.13	2.32	4.88	4.83	14.69	3.27	2.35	2.47
3	10											…	9.647	7.83	9.959	10.456	23.75	9.1	7.78	8.304
3	11												…	3.22	3.76	1.56	13.41	3.39	2.7	3.06
3	12													…	2.684	2.762	16.878	1.038	1.771	0.432
3	13														….	2.1	16.95	1.61	3.43	2.47
3	14															….	16.434	2.053	3.246	2.395
2	15																….	16.62	15.76	15.55
3	16																	…	2.997	1.313
3	17																		…..	1.42
3	18																			….
3	19																			
3	20																			
3	21																			
3	22																			
3	23																			
3	24																			
3	25																			
2	26																			
1	27																			

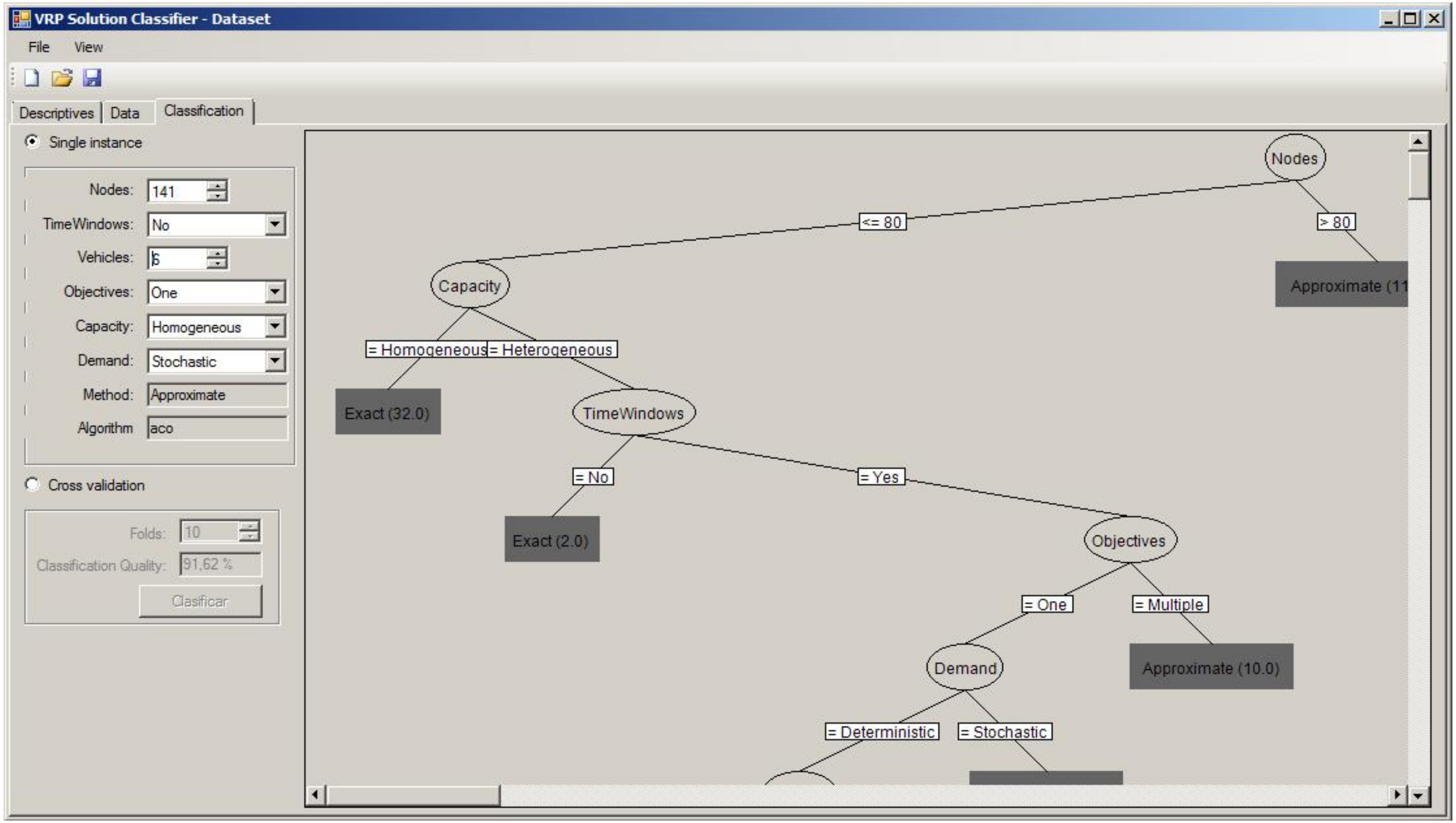
Appendix A.12: Decision tree for classifying the VRPs



Appendix A.13: Exact algorithm prediction to the I-32 case study instance



Appendix A.14: Approximate algorithm prediction to the case study instances (e.g. I-142)



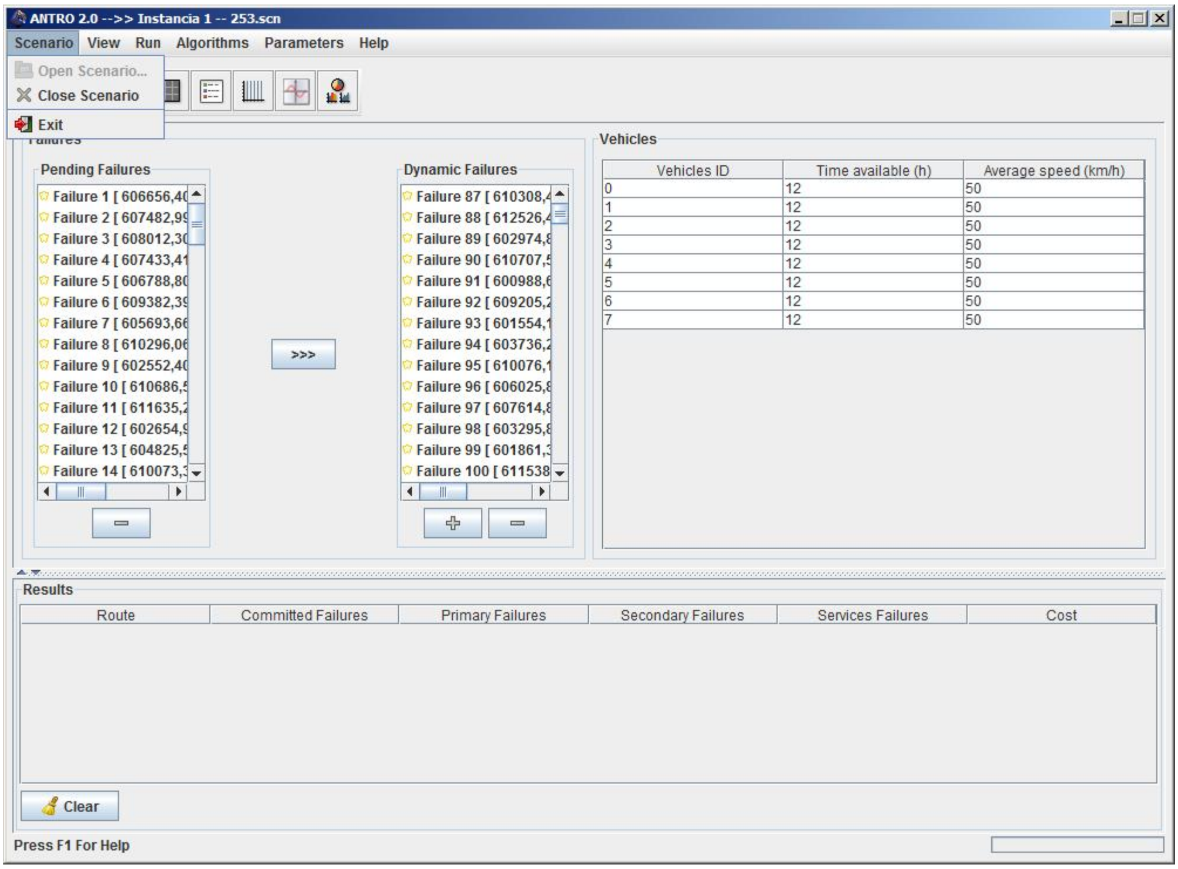
Appendix A.15: The software ANTRO version 2.0

Motivation and description: The ANTRO version 2.0 is a computational implantation, which allows the algorithmic assistance for route planning in the repair of electrical breakdown (or failure). This software presents a friendly interface that can be used by the dispatchers of Cuban Electric Company. The software provides dispatchers the following utilities:

1) Load and execute different scenarios (case study instances) of decision-making related with the case study.
2) Allocate the vehicles of the fleet to the set of breakdowns, considering the real-life conditions described in the case study (e.g. the priority level, stochastic repair time and the occurrence of the unexpected breakdowns).
3) Develop the route planning considering two approximate algorithms, the ACS and M-ACS.
4) Display the results in analytical and graphical format.

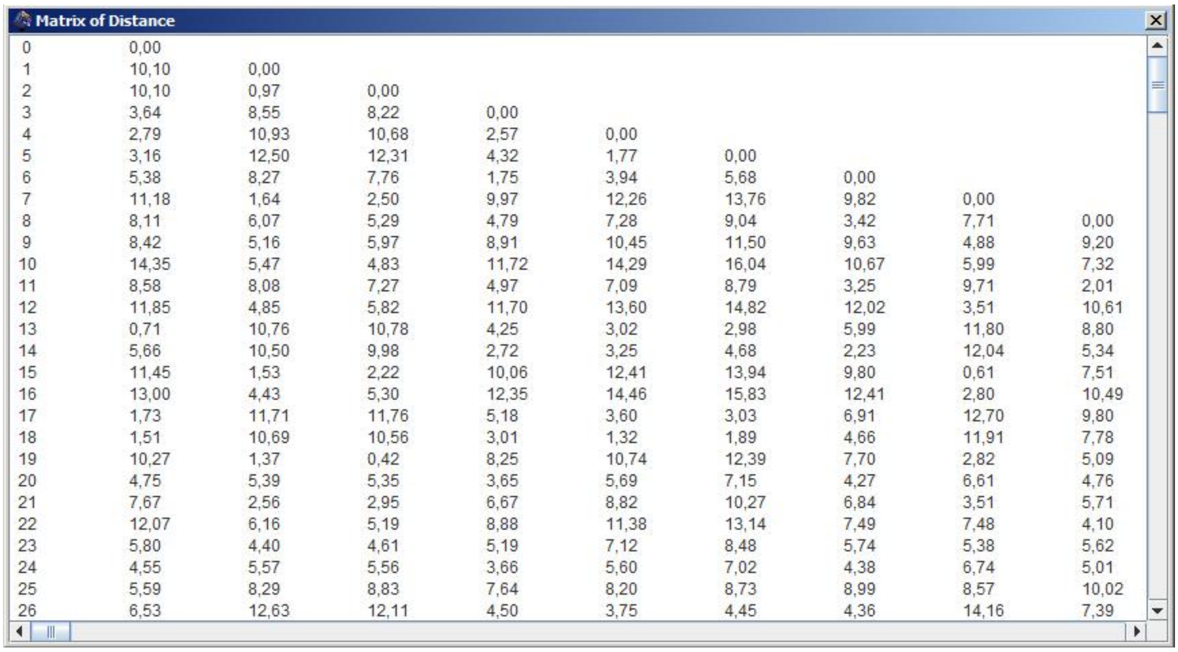
The software utilities and the main interfaces are described below.

Scenarios menu: In this menu option are selected and then loaded the case study instances. Furthermore, the breakdown list, either the pending or dynamic breakdowns can be added, deleted or exchanged between both lists.



View menu: Here, some options can be deployed to visualize the algorithm results, *either from analytical or graphical perspective*. Both results are summarized and printed in a pdf document.

<u>Visualization of the distance matrix:</u>



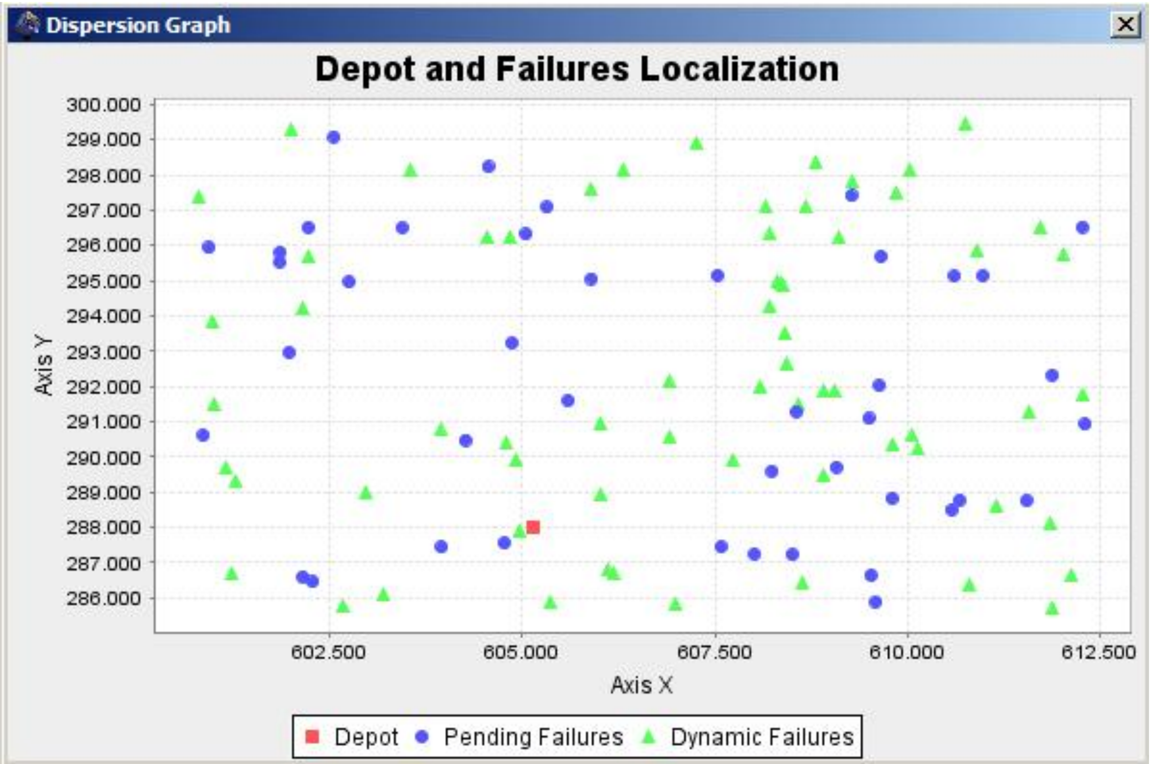
0	0,00								
1	10,10	0,00							
2	10,10	0,97	0,00						
3	3,64	8,55	8,22	0,00					
4	2,79	10,93	10,68	2,57	0,00				
5	3,16	12,50	12,31	4,32	1,77	0,00			
6	5,38	8,27	7,76	1,75	3,94	5,68	0,00		
7	11,18	1,64	2,50	9,97	12,26	13,76	9,82	0,00	
8	8,11	6,07	5,29	4,79	7,28	9,04	3,42	7,71	0,00
9	8,42	5,16	5,97	8,91	10,45	11,50	9,63	4,88	9,20
10	14,35	5,47	4,83	11,72	14,29	16,04	10,67	5,99	7,32
11	8,58	8,08	7,27	4,97	7,09	8,79	3,25	9,71	2,01
12	11,85	4,85	5,82	11,70	13,60	14,82	12,02	3,51	10,61
13	0,71	10,76	10,78	4,25	3,02	2,98	5,99	11,80	8,80
14	5,66	10,50	9,98	2,72	3,25	4,68	2,23	12,04	5,34
15	11,45	1,53	2,22	10,06	12,41	13,94	9,80	0,61	7,51
16	13,00	4,43	5,30	12,35	14,46	15,83	12,41	2,80	10,49
17	1,73	11,71	11,76	5,18	3,60	3,03	6,91	12,70	9,80
18	1,51	10,69	10,56	3,01	1,32	1,89	4,66	11,91	7,78
19	10,27	1,37	0,42	8,25	10,74	12,39	7,70	2,82	5,09
20	4,75	5,39	5,35	3,65	5,69	7,15	4,27	6,61	4,76
21	7,67	2,56	2,95	6,67	8,82	10,27	6,84	3,51	5,71
22	12,07	6,16	5,19	8,88	11,38	13,14	7,49	7,48	4,10
23	5,80	4,40	4,61	5,19	7,12	8,48	5,74	5,38	5,62
24	4,55	5,57	5,56	3,66	5,60	7,02	4,38	6,74	5,01
25	5,59	8,29	8,83	7,64	8,20	8,73	8,99	8,57	10,02
26	6,53	12,63	12,11	4,50	3,75	4,45	4,36	14,16	7,39

General information of one case study instance:

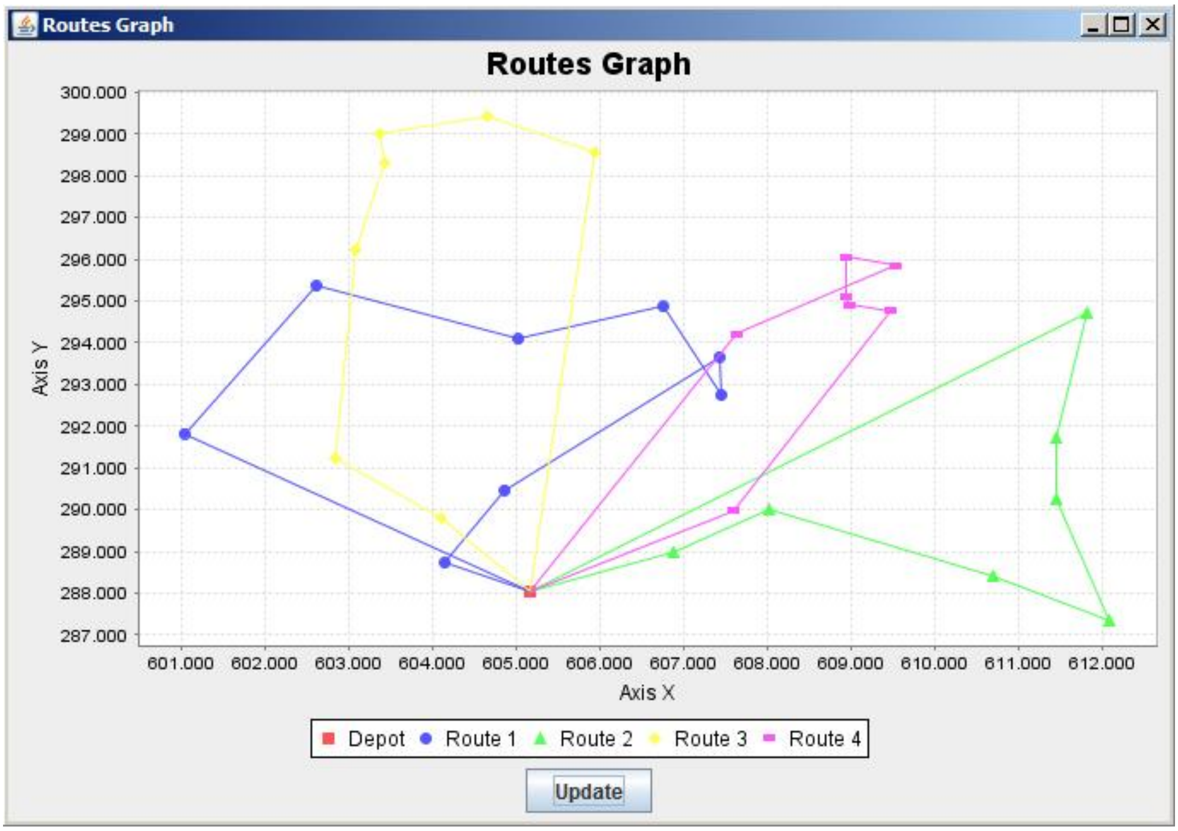
Scenario Description

Number of Pending Failures	86
Number of Dynamic Failures	167
Number of Total Failures	253
Depot Location	[X:605151,31; Y:288022,94]
Number Vehicles	8

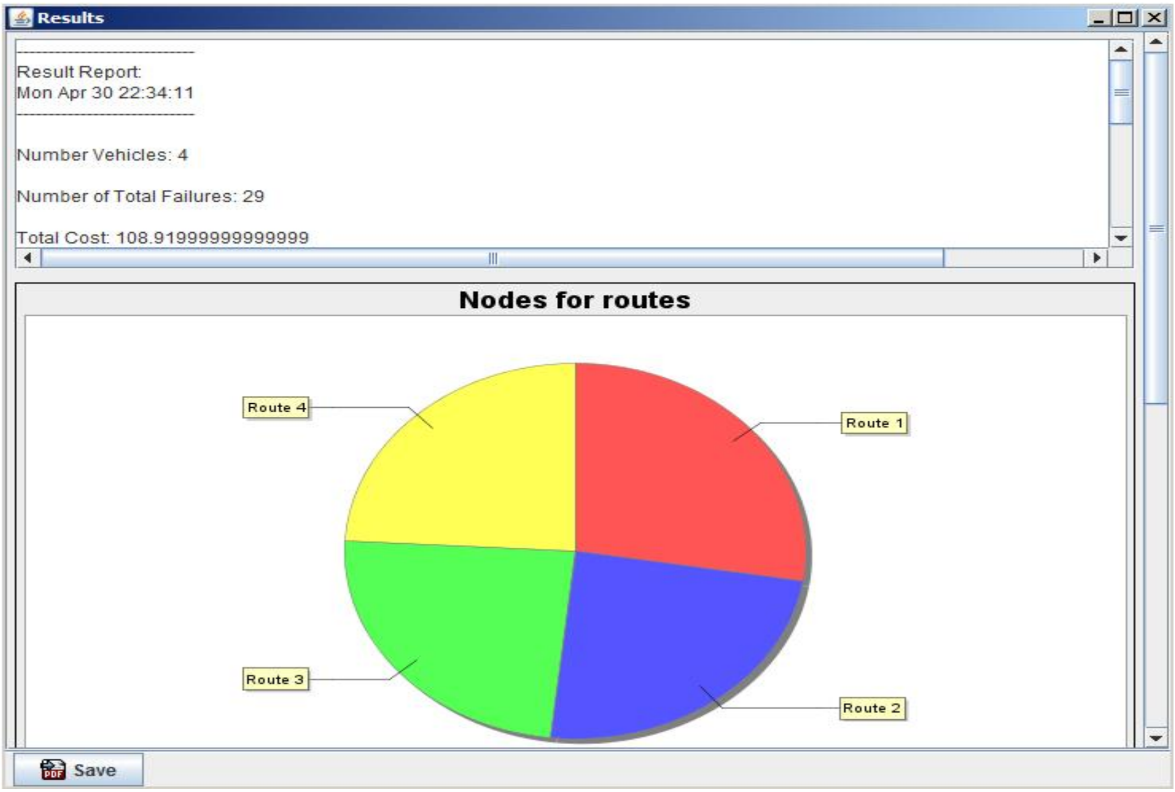
Graphical representation of the breakdowns:



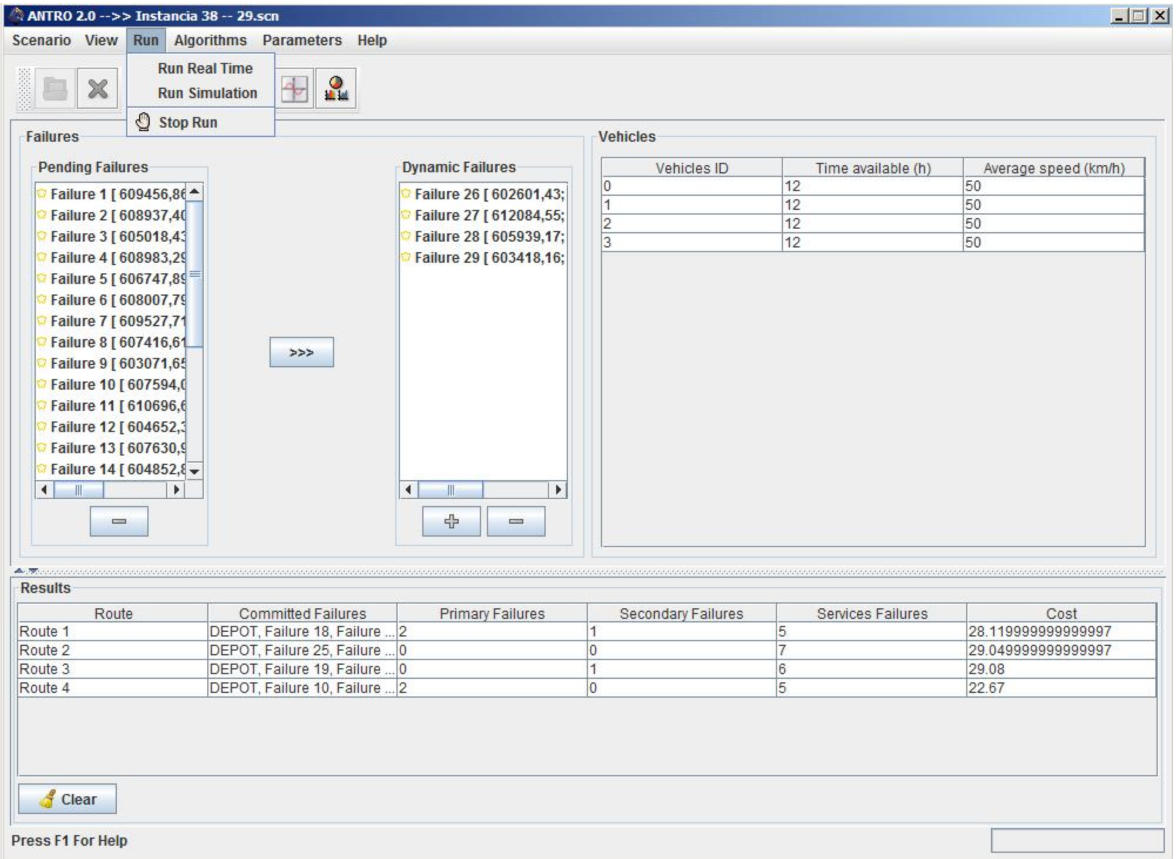
Graphical representation of the route planning:



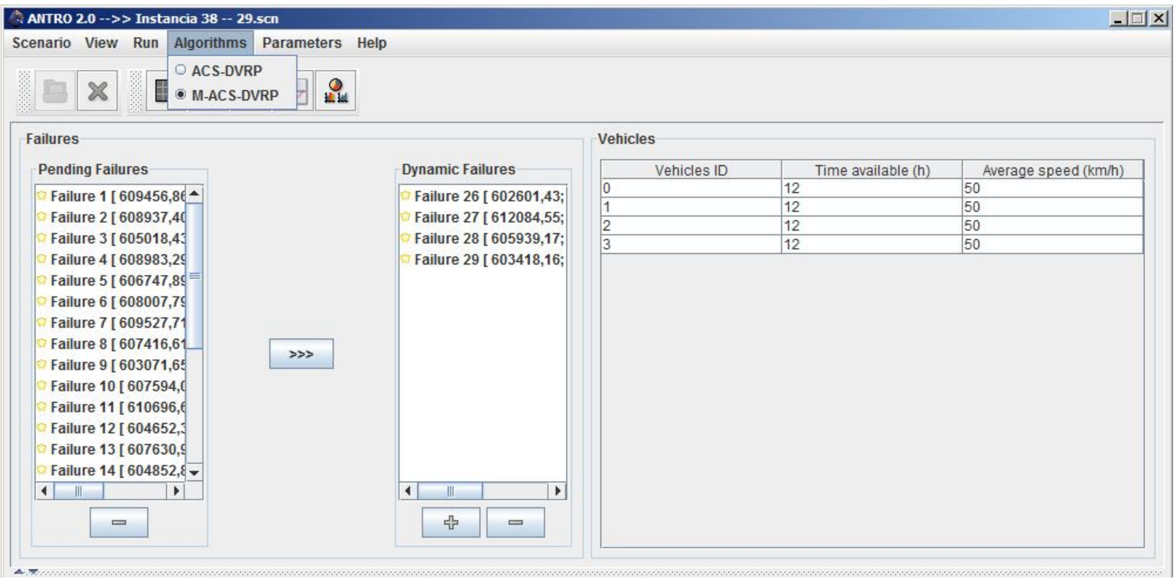
Report of the route planning results:



Run menu: In this menu can be defined two types of algorithm runs: the "Run Simulation", which is used when the dynamic breakdown exits in any lists (pending or dynamic), and the "Run Real Time", executed when the dynamic breakdowns are defined introducing its geographical coordinates.

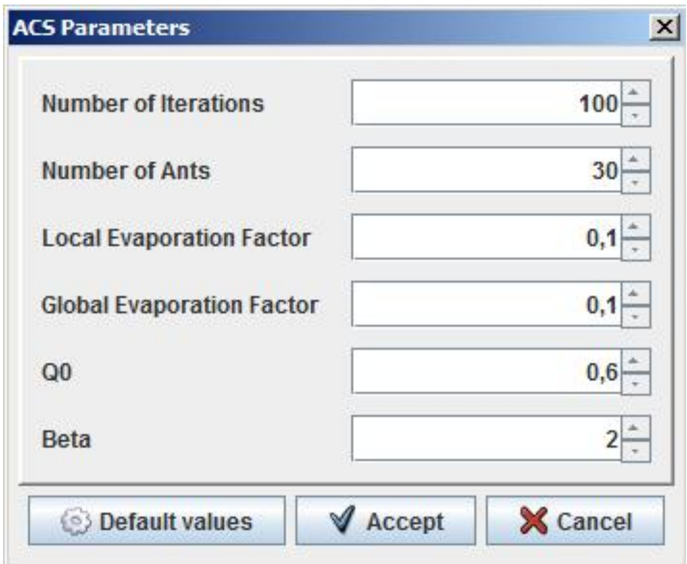


Algorithms menu: The users can define which approximate algorithm, either ACS or M-ACS, they want to apply for a given case study instance.

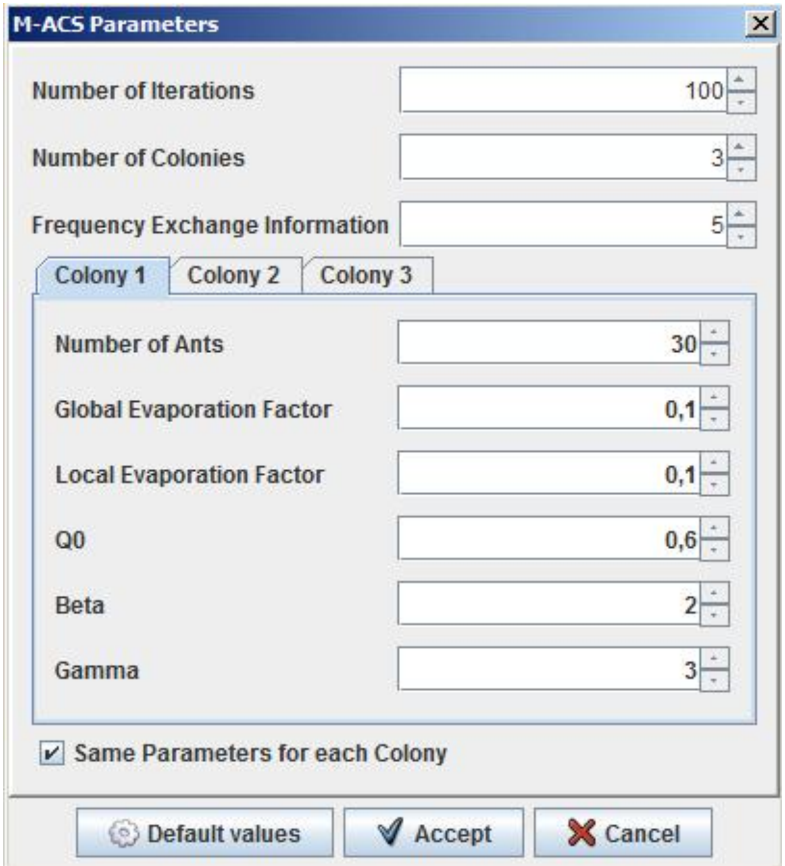


Parameters menu: In this menu are introduced the parameters of both approximate algorithms, depending of which has been selected in previous menu (Algorithms menu). Furthermore, the user can define the input data related with case study instance, for instance, the breakdown priority level and repair time.

Interface of the ACS parameters:

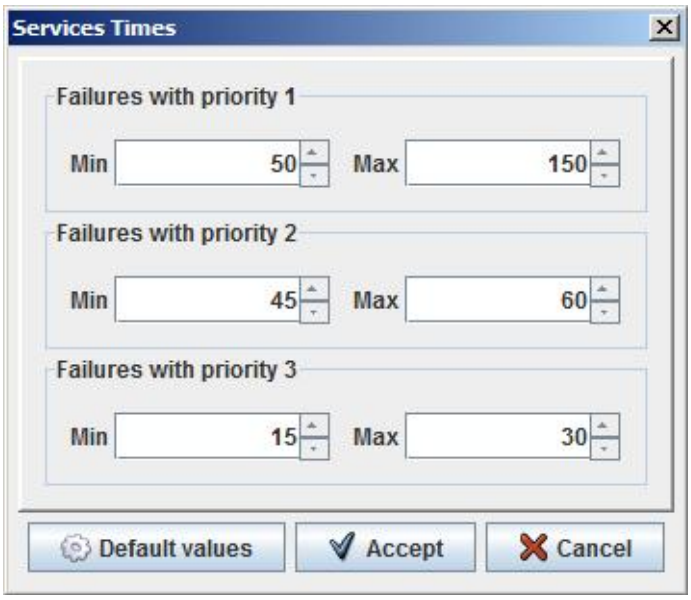


Interface of the ACS parameters:



Priority level interface:

Repair time interface:



Appendix A.16: *Java* source code of the ANTRO version 2.0

```
public Main(int numberAverias, File averiasFile,
        File weightFile) {
    this.averias = new Averia[numberAverias + 1];
    this.readAverias(averiasFile.getAbsolutePath());
    this.matrixWeight = new int[numberAverias + 1][numberAverias + 1];
    this.readDataFile(weightFile.getAbsolutePath());
  }

private void readAverias(String filePath) {
    try {
      BufferedReader in = new BufferedReader(new FileReader(filePath));
      String[] tok = null;
      String line = null;
      int numberLine = 1;
      Averia a = new Averia(0, 0, 0, 0);
      this.averias[0] = a;
      while ((line = in.readLine()) != null) {
        tok = line.split("[ \t]+");
        int x = Integer.parseInt(tok[0]);
        int y = Integer.parseInt(tok[1]);
        int t = Integer.parseInt(tok[2]);
        a = new Averia(x, y, t, numberLine);
        this.averias[numberLine++] = a;
      }
      in.close();
    } catch (IOException ex) {
      System.err.println(ex.getMessage());
    }
  }

public int[][] subMatrixWeight(int[] subTour) {
    int[][] subMatrixWeight =
        new int[subTour.length][subTour.length];
    Arrays.sort(subTour);
    for (int i = 0; i < subTour.length; i++) {
      for (int j = i + 1; j < subTour.length; j++) {
        subMatrixWeight[j][i] = subMatrixWeight[i][j] =
            this.matrixWeight[subTour[i]][subTour[j]];
      }
    }
    return subMatrixWeight;
  }
```

```
public void addNode(int n, int t) {
        int[] temp = new int[this.array.length + 1];
        System.arraycopy(this.array, 0, temp, 0, this.array.length);
        temp[temp.length - 1] = n;
        this.array = temp.clone();
        temp = new int[this.times.length + 1];
        System.arraycopy(this.times, 0, temp, 0, this.times.length);
        temp[temp.length - 1] = t;
        this.times = temp.clone();
    }
}

private int generateTimeRaparation() {
        int time = 0;
        switch (getType()) {
            case 1: {
                time = 45 + Math.abs(new Random().nextInt() % 15);  // entre 45 y 60 min
                break;
            }
            case 2: {
                time = 20 + Math.abs(new Random().nextInt() % 15);  // entre 20 y 35 min
                break;
            }
            case 3: {
                time = 10 + Math.abs(new Random().nextInt() % 10);  // entre 10 y 20 min
                break;
            }
        }
        return time;
}

public int[] tipoAveria(int[] subTour) {
        Arrays.sort(subTour);
        int[] types = new int[subTour.length];
        for (int i = 0; i < types.length; i++) {
            types[i] = this.averias[subTour[i]].getType();
        }
        return types;
    }
```

```
public static void main(String[] args) {
        Main m = new Main(Integer.parseInt(args[0]), new File(args[1]),
                new File(args[2]));
        Cluster[] c = m.asignar(Integer.parseInt(args[3]), Integer.parseInt(args[4]));
        Environment env;
        AcsTsp a;
        int totalCoste = 0;
        int totalTime = 0;
        for (int i = 0; i < c.length; i++) {
            totalTime = 0;
            System.out.println("VehÃ-culo: " + (i + 1));
            System.out.println("Cantidad de averÃ-as a reparar: "
                    + (c[i].array.length - 1));
            System.out.print("AverÃ-as a reparar: ");
            for (int j = 1; j < c[i].array.length; j++) {
                System.out.print(c[i].array[j] + " ");
            }
            System.out.println("");
            env = new Environment(c[i].array, m.tipoAveria(c[i].array),
                    m.subMatrixWeight(c[i].array), Double.parseDouble(args[5]));
            a = new AcsTsp(env, Integer.parseInt(args[6]), Long.parseLong(args[7]),
                    Double.parseDouble(args[8]), Integer.parseInt(args[9]));
            a.printGlobalSolution();
            System.out.print("Taza del tiempo (min): ");
            for (int j = 0; j < c[i].times.length; j++) {
                totalTime += c[i].times[j];
                System.out.print(c[i].times[j] + " ");
            }
            System.out.println("");
            System.out.println("Tiempo total de las reparaciones (min): " + totalTime);
            totalCoste += a.getBestGlobalSolution().coste;
            System.out.println("");
//          Object[] obj = a.executeAcs();
//          a.printGlobalSolution();
//          System.out.println("Number Estages: " + ((Integer) obj[1]).intValue());
//          System.out.println("Time (sec): " + ((Long) obj[0]).longValue() / 1000);
//          System.out.println("");
//          totalCoste += ((Double) obj[2]).doubleValue();
        }
        System.out.println("Costo total: " + totalCoste);
        System.out.println("");
//      System.out.println("Average coste: " + (int) totalCoste / c.length);
    }
```

Appendix A.17: ACS solution report to case study instances

Result report: I-32
Number vehicles: 3 Number orders: 31 Total cost: 44.90593478261439 Cost for route: Route 1: 11.870060551282766 Route 2: 21.30090415900707 Route 3: 11.734970072324549 Best configuration of routes: Route 1: DEPOT -> Order 26 -> Order 24 -> Order 20 -> Order 31 -> Order 23 -> Order 29 -> Order 30 -> Order 6 -> Order 22 -> Order 2 -> Order 5 -> DEPOT Route 2: DEPOT -> Order 4 -> Order 7 -> Order 12 -> Order 9 -> Order 13 -> Order 16 -> Order 27 -> Order 28 -> Order 18 -> Order 17 -> DEPOT Route 3: DEPOT -> Order 14 -> Order 1 -> Order 25 -> Order 3 -> Order 15 -> Order 8 -> Order 21 -> Order 19 -> Order 11 -> Order 10 -> DEPOT Best iteration: 12

Result report: I-94
Number vehicles: 5 Number orders: 93 Total cost: 97.83174556362388 Cost for route: Route 1: 14.56697576733946 Route 2: 14.954278651363264 Route 3: 11.265440048769786 Route 4: 45.360074124048154 Route 5: 11.684976972103195 Best configuration of routes: Route 1: DEPOT -> Order 40 -> Order 46 -> Order 36 -> Order 41 -> Order 52 -> Order 24 -> Order 30 -> Order 31 -> Order 83 -> Order 1 -> Order 21 -> Order 67 -> Order 5 -> Order 49 -> Order 39 -> Order 66 -> Order 64 -> Order 86 -> Order 14 -> DEPOT Route 2: DEPOT -> Order 25 -> Order 54 -> Order 23 -> Order 32 -> Order 51 -> Order 11 -> Order 6 -> Order 87 -> Order 72 -> Order 8 -> Order 63 -> Order 10 -> Order 94 -> Order 18 -> Order 53 -> Order 73 -> Order 44 -> Order 58 -> Order 85 -> DEPOT Route 3: DEPOT -> Order 93 -> Order 75 -> Order 81 -> Order 13 -> Order 38 -> Order 68 -> Order 19 -> Order 34 -> Order 3 -> Order 60 -> Order 79 -> Order 2 -> Order 4 -> Order 42 -> Order 65 -> Order 77 -> Order 37 -> Order 28 -> Order 57 -> DEPOT Route 4: DEPOT -> Order 82 -> Order 91 -> Order 33 -> Order 90 -> Order 89 -> Order 88 -> Order 80 -> Order 92 -> Order 16 ->Order 17 -> Order 27 -> Order 26 -> Order 78 -> Order 15 -> Order 56 -> Order 45 -> Order 9 -> Order 12 -> Order 69 -> DEPOT Route 5: DEPOT -> Order 76 -> Order 47 -> Order 70 -> Order 48 -> Order 71 -> Order 35 -> Order 62 -> Order 22 -> Order 29 -> Order 74 -> Order 59 -> Order 61 -> Order 55 -> Order 43 -> Order 20 -> Order 50 -> Order 7 -> Order 84 -> DEPOT Best iteration: 35

Result report: I-142
Number vehicles: 5
Number orders: 141
Total cost: 88.31436514005219
Cost for route:
Route 1: 17.002615263116102
Route 2: 25.403389271203395
Route 3: 14.508517911315632
Route 4: 16.06196862961225
Route 5: 15.337874064804812
Best configuration of routes:
Route 1: DEPOT -> Order 26 -> Order 68 -> Order 46 -> Order 24 -> Order 82 -> Order 106 -> Order 137 -> Order 20 -> Order 122 -> Order 25 -> Order 44 -> Order 90 -> Order 45 -> Order 49 -> Order 74 -> Order 131 -> Order 23 -> Order 139 -> Order 134 ->Order 29 -> Order 72 -> Order 51 -> Order 79 -> Order 57 -> Order 83 -> Order 19 -> Order 67 -> Order 70 -> Order 109 -> DEPOT
Route 2: DEPOT -> Order 116 -> Order 65 -> Order 12 -> Order 89 -> Order 21 -> Order 111 -> Order 84 -> Order 123 -> Order 98 -> Order 53 -> Order 63 -> Order 121 -> Order 97 -> Order 133 -> Order 135 -> Order 42 -> Order 102 -> Order 120 -> Order 114 ->Order 80 -> Order 16 -> Order 28 -> Order 27 -> Order 37 -> Order 35 -> Order 18 -> Order 17 -> Order 92 -> DEPOT
Route 3: DEPOT -> Order 138 -> Order 85 -> Order 107 -> Order 47 -> Order 95 -> Order 104 -> Order 33 -> Order 32 -> Order 141-> Order 41 -> Order 52 -> Order 1 -> Order 14 -> Order 87 -> Order 127 -> Order 43 -> Order 36 -> Order 75 -> Order 62 -> Order 48 -> Order 129 -> Order 4 -> Order 71 -> Order 9 -> Order 34 -> Order 96 -> Order 100 -> Order 88 -> DEPOT
Route 4: DEPOT -> Order 66 -> Order 55 -> Order 99 -> Order 61 -> Order 140 -> Order 76 -> Order 60 -> Order 54 -> Order 50 -> Order 30 -> Order 118 -> Order 38 -> Order 73 -> Order 125 -> Order 6 -> Order 136 -> Order 22 -> Order 2 -> Order 101 -> Order 105 -> Order 69 -> Order 126 -> Order 103 -> Order 128 -> Order 113 -> Order 40 -> Order 3 -> Order 31 -> DEPOT
Route 5: DEPOT -> Order 117 -> Order 7 -> Order 93 -> Order 39 -> Order 11 -> Order 94 -> Order 59 -> Order 86 -> Order 81 -> Order 10 -> Order 77 -> Order 91 -> Order 13 -> Order 132 -> Order 78 -> Order 112 -> Order 115 -> Order 110 -> Order 15 -> Order 8 -> Order 64 -> Order 124 -> Order 58 -> Order 130 -> Order 56 -> Order 119 -> Order 5 -> Order 108 -> DEPOT
Best iteration: 40

Result report: I-170
Number vehicles: 6 Number orders: 169 Total cost: 136.7595665804056 Cost for route: Route 1: 16.86195563800663 Route 2: 13.936000596951263 Route 3: 13.008844143597363 Route 4: 14.209577080091142 Route 5: 61.58067953647952 Route 6: 17.16250958527964 Best configuration of routes: Route 1: DEPOT -> Order 130 -> Order 25 -> Order 67 -> Order 64 -> Order 11 -> Order 3 -> Order 135 -> Order 121 -> Order 14 -> Order 97 -> Order 62 -> Order 120 -> Order 155 -> Order 125 -> Order 163 -> Order 127 -> Order 41 -> Order 101 -> Order 139 -> Order 18 -> Order 66 -> Order 69 -> Order 88 -> Order 28 -> Order 55 -> Order 118 -> Order 4 -> Order 39 -> Order 2 -> DEPOT Route 2: DEPOT -> Order 157 -> Order 23 -> Order 81 -> Order 105 -> Order 47 -> Order 131 -> Order 102 -> Order 68 -> Order 122 -> Order 56 -> Order 156 -> Order 20 -> Order 110 -> Order 82 -> Order 145 -> Order 83 -> Order 52 -> Order 96 -> Order 57 -> Order 78 -> Order 7 -> Order 63 -> Order 50 -> Order 71 -> Order 126 -> Order 149 -> Order 136 -> Order 112 -> DEPOT Route 3: DEPOT -> Order 151 -> Order 114 -> Order 152 -> Order 111 -> Order 79 -> Order 93 -> Order 58 -> Order 85 -> Order 167 -> Order 10 -> Order 38 -> Order 119 -> Order 113 -> Order 45 -> Order 53 -> Order 132 -> Order 75 -> Order 148 -> Order 59 - > Order 61 -> Order 74 -> Order 35 -> Order 42 -> Order 153 -> Order 143 -> Order 65 -> Order 54 -> Order 115 -> DEPOT Route 4: DEPOT -> Order 154 -> Order 86 -> Order 13 -> Order 51 -> Order 129 -> Order 19 -> Order 89 -> Order 43 -> Order 24 -> Order 30 -> Order 107 -> Order 44 -> Order 48 -> Order 73 -> Order 123 -> Order 22 -> Order 29 -> Order 117 -> Order 146 -> Order 37 -> Order 72 -> Order 169 -> Order 5 -> Order 128 -> Order 21 -> Order 1 -> Order 100 -> Order 104 -> DEPOT Route 5: DEPOT -> Order 140 -> Order 116 -> Order 6 -> Order 92 -> Order 144 -> Order 138 -> Order 109 -> Order 77 -> Order 90-> Order 76 -> Order 162 -> Order 9 -> Order 159 -> Order 80 -> Order 158 -> Order 15 -> Order 160 -> Order 17 -> Order 26 -> Order 27 -> Order 36 -> Order 34 -> Order 16 -> Order 161 -> Order 164 -> Order 166 -> Order 103 -> Order 49 -> DEPOT Route 6: DEPOT -> Order 150 -> Order 91 -> Order 141 -> Order 95 -> Order 134 -> Order 99 -> Order 137 -> Order 33 -> Order 108 -> Order 87 -> Order 12 -> Order 124 -> Order 8 -> Order 70 -> Order 106 -> Order 94 -> Order 46 -> Order 84 -> Order 32 -> Order 133 -> Order 165 -> Order 147 -> Order 40 -> Order 60 -> Order 142 -> Order 98 -> Order 31 -> Order 168 -> DEPOT Best iteration: 22

Result report: I-220
Number vehicles: 7 Number orders: 219 Total cost: 178.884106342438 Cost for route: Route 1: 78.5415306545839 Route 2: 26.600217082300006 Route 3: 15.326222044059806 Route 4: 11.637771377938 Route 5: 15.752672788617295 Route 6: 14.37225672003841 Route 7: 16.65343567490061 Best configuration of routes: Route 1: DEPOT -> Order 138 -> Order 161 -> Order 75 -> Order 36 -> Order 43 -> Order 66 -> Order 151 -> Order 184 -> Order 189 -> Order 99 -> Order 61 -> Order 150 -> Order 41 -> Order 141 -> Order 85 -> Order 33 -> Order 199 -> Order 104 -> Order 205 -> Order 212 -> Order 174 -> Order 196 -> Order 188 -> Order 172 -> Order 200 -> Order 169 -> Order 179 -> Order 17 -> Order 45 -> Order 25 -> Order 31 -> Order 137 -> DEPOT Route 2: DEPOT -> Order 181 -> Order 195 -> Order 159 -> Order 91 -> Order 10 -> Order 170 -> Order 175 -> Order 94 -> Order 185 -> Order 59 -> Order 86 -> Order 81 -> Order 167 -> Order 147 -> Order 19 -> Order 37 -> Order 35 -> Order 28 -> Order 27 -> Order 168 -> Order 18 -> Order 186 -> Order 16 -> Order 166 -> Order 67 -> Order 70 -> Order 202 -> Order 80 -> Order 11 -> Order 108 -> Order 176 -> Order 1 -> DEPOT Route 3: DEPOT -> Order 127 -> Order 162 -> Order 87 -> Order 62 -> Order 55 -> Order 140 -> Order 76 -> Order 60 -> Order 54 - > Order 156 -> Order 50 -> Order 155 -> Order 173 -> Order 32 -> Order 198 -> Order 47 -> Order 95 -> Order 107 -> Order 197 -> Order 190 -> Order 201 -> Order 209 -> Order 71 -> Order 152 -> Order 112 -> Order 114 -> Order 120 -> Order 39 -> Order 77 -> Order 48 -> Order 46 -> DEPOT Route 4: DEPOT -> Order 26 -> Order 68 -> Order 204 -> Order 65 -> Order 116 -> Order 12 -> Order 102 -> Order 97 -> Order 163 -> Order 133 -> Order 171 -> Order 53 -> Order 178 -> Order 121 -> Order 63 -> Order 98 -> Order 153 -> Order 83 -> Order 89 -> Order 111 -> Order 21 -> Order 57 -> Order 124 -> Order 72 -> Order 29 -> Order 56 -> Order 119 -> Order 5 -> Order 216 -> Order 219 -> Order 90 -> DEPOT Route 5: DEPOT -> Order 106 -> Order 165 -> Order 24 -> Order 82 -> Order 20 -> Order 208 -> Order 122 -> Order 215 -> Order 49 -> Order 74 -> Order 131 -> Order 203 -> Order 23 -> Order 213 -> Order 214 -> Order 38 -> Order 73 -> Order 177 -> Order 125 -> Order 6 -> Order 206 -> Order 136 -> Order 22 -> Order 2 -> Order 191 -> Order 139 -> Order 183 -> Order 180 -> Order 40 -> Order 3 -> Order 14 -> DEPOT Route 6: DEPOT -> Order 207 -> Order 145 -> Order 109 -> Order 88 -> Order 92 -> Order 149 -> Order 96 -> Order 142 -> Order 100 -> Order 158 -> Order 34 -> Order 9 -> Order 210 -> Order 132 -> Order 211 -> Order 13 -> Order 78 -> Order 146 -> Order 110 -> Order 115 -> Order 160 -> Order 15 -> Order 194 -> Order 64 -> Order 8 ->

Order 135 -> Order 42 -> Order 217 -> Order 218 -> Order 44 -> Order 52 -> DEPOT

Route 7: DEPOT -> Order 93 -> Order 7 -> Order 148 -> Order 117 -> Order 4 -> Order 129 -> Order 143 -> Order 164 -> Order 79 - > Order 58 -> Order 157 -> Order 144 -> Order 134 -> Order 154 -> Order 118 -> Order 30 -> Order 126 -> Order 103 -> Order 69 -> Order 130 -> Order 105 -> Order 128 -> Order 101 -> Order 84 -> Order 123 -> Order 192 -> Order 193 -> Order 113 -> Order 187 - > Order 182 -> Order 51 -> DEPOT

Best iteration: 77

Appendix A.18: M-ACS solution report

Result report: I-32
Number vehicles: 3 Number orders: 31 Total cost: 44.74051120458302 Cost for route: Route 1: 13.128259266422468 Route 2: 9.721328195331118 Route 3: 21.89092374282944 Best configuration of routes: Route 1: DEPOT -> Order 26 -> Order 14 -> Order 1 -> Order 25 -> Order 23 -> Order 22 -> Order 2 -> Order 6 -> Order 21 -> Order 19 -> Order 9 -> DEPOT Route 2: DEPOT -> Order 12 -> Order 24 -> Order 20 -> Order 31 -> Order 5 -> Order 3 -> Order 29 -> Order 30 -> Order 8 -> Order 15 -> DEPOT Route 3: DEPOT -> Order 4 -> Order 7 -> Order 13 -> Order 11 -> Order 10 -> Order 16 -> Order 27 -> Order 28 -> Order 18 -> Order 17 -> DEPOT Best colony: 3 Best iteration: 32

Result report: I-94
Number vehicles: 5
Number orders: 94
Total cost: 98.32460649703218
Cost for route:
Route 1: 45.933386715860266
Route 2: 11.878467246673036
Route 3: 10.7069226176238
Route 4: 14.725292287945845
Route 5: 15.080537628929218
Best configuration of routes:
Route 1: DEPOT -> Order 82 -> Order 93 -> Order 91 -> Order 89 -> Order 90 -> Order 33 -> Order 88 -> Order 80 -> Order 92 -> Order 16 -> Order 17 -> Order 26 -> Order 27 -> Order 78 -> Order 15 -> Order 53 -> Order 56 -> Order 18 -> Order 64 -> DEPOT
Route 2: DEPOT -> Order 75 -> Order 36 -> Order 76 -> Order 70 -> Order 41 -> Order 48 -> Order 19 -> Order 30 -> Order 31 -> Order 22 -> Order 42 -> Order 28 -> Order 58 -> Order 65 -> Order 77 -> Order 73 -> Order 49 -> Order 39 -> Order 86 -> DEPOT
Route 3: DEPOT -> Order 25 -> Order 54 -> Order 23 -> Order 32 -> Order 13 -> Order 38 -> Order 68 -> Order 79 -> Order 2 -> Order 4 -> Order 55 -> Order 61 -> Order 43 -> Order 20 -> Order 44 -> Order 37 -> Order 50 -> Order 7 -> Order 84 -> DEPOT
Route 4: DEPOT -> Order 40 -> Order 46 -> Order 47 -> Order 81 -> Order 52 -> Order 71 -> Order 24 -> Order 35 -> Order 62 -> Order 29 -> Order 74 -> Order 59 -> Order 5 -> Order 21 -> Order 67 -> Order 1 -> Order 83 -> Order 66 -> Order 14 -> DEPOT
Route 5: DEPOT -> Order 51 -> Order 34 -> Order 3 -> Order 87 -> Order 72 -> Order 6 -> Order 11 -> Order 57 -> Order 8 -> Order 63 -> Order 10 -> Order 94 -> Order 45 -> Order 9 -> Order 12 -> Order 69 -> Order 60 -> Order 85 -> DEPOT
Best colony: 1
Best iteration: 26

Result report: I-142

Number vehicles: 5

Number orders: 141

Total cost: 88.47567226745277

Cost for route:

Route 1: 13.079007043369128

Route 2: 28.371701921439282

Route 3: 17.255233900955773

Route 4: 14.6505224788158

Route 5: 15.119206922872806

Best configuration of routes:

Route 1: DEPOT -> Order 26 -> Order 68 -> Order 46 -> Order 127 -> Order 14 -> Order 52 -> Order 1 -> Order 137 -> Order 20 -> Order 122 -> Order 25 -> Order 45 -> Order 49 -> Order 44 -> Order 90 -> Order 31 -> Order 108 -> Order 139 -> Order 23 -> Order 131 -> Order 74 -> Order 119 -> Order 5 -> Order 40 -> Order 3 -> Order 113 -> Order 72 -> Order 134 -> Order 56 -> DEPOT

Route 2: DEPOT -> Order 116 -> Order 65 -> Order 12 -> Order 117 -> Order 7 -> Order 93 -> Order 39 -> Order 11 -> Order 94 -> Order 59 -> Order 86 -> Order 81 -> Order 10 -> Order 80 -> Order 16 -> Order 28 -> Order 27 -> Order 37 -> Order 35 -> Order 18 - > Order 17 -> Order 67 -> Order 70 -> Order 19 -> Order 121 -> Order 97 -> Order 133 -> Order 42 -> DEPOT

Route 3: DEPOT -> Order 138 -> Order 61 -> Order 140 -> Order 76 -> Order 60 -> Order 54 -> Order 50 -> Order 41 -> Order 32 -> Order 85 -> Order 33 -> Order 107 -> Order 104 -> Order 95 -> Order 47 -> Order 141 -> Order 129 -> Order 4 -> Order 48 -> Order 106 -> Order 82 -> Order 24 -> Order 71 -> Order 110 -> Order 115 -> Order 112 -> Order 102 -> Order 15 -> DEPOT

Route 4: DEPOT -> Order 66 -> Order 99 -> Order 62 -> Order 55 -> Order 75 -> Order 36 -> Order 43 -> Order 87 -> Order 30 -> Order 118 -> Order 38 -> Order 73 -> Order 125 -> Order 6 -> Order 136 -> Order 22 -> Order 2 -> Order 101 -> Order 105 -> Order 128 -> Order 126 -> Order 103 -> Order 69 -> Order 130 -> Order 79 -> Order 124 -> Order 51 -> Order 29 -> DEPOT

Route 5: DEPOT -> Order 34 -> Order 109 -> Order 88 -> Order 92 -> Order 100 -> Order 96 -> Order 9 -> Order 132 -> Order 78 -> Order 91 -> Order 77 -> Order 120 -> Order 114 -> Order 53 -> Order 63 -> Order 98 -> Order 84 -> Order 123 -> Order 89 -> Order 21 -> Order 111 -> Order 83 -> Order 57 -> Order 58 -> Order 64 -> Order 8 -> Order 135 -> Order 13 -> DEPOT

Best colony: 1

Best iteration: 41

Result report: I-170
Number vehicles: 6 Number orders: 169 Total cost: 141.63212823401994 Cost for route: Route 1: 11.929748042458288 Route 2: 14.001534878846725 Route 3: 12.14974979143773 Route 4: 66.01821913860176 Route 5: 12.110782744147258 Route 6: 25.422093638528196 Best configuration of routes: Route 1: DEPOT -> Order 157 -> Order 23 -> Order 81 -> Order 45 -> Order 53 -> Order 59 -> Order 75 -> Order 148 -> Order 60 -> Order 98 -> Order 142 -> Order 147 -> Order 165 -> Order 133 -> Order 106 -> Order 94 -> Order 46 -> Order 84 -> Order 32 -> Order 31 -> Order 103 -> Order 40 -> Order 49 -> Order 132 -> Order 54 -> Order 61 -> Order 65 -> Order 143 -> Order 35 -> DEPOT Route 2: DEPOT -> Order 154 -> Order 86 -> Order 13 -> Order 51 -> Order 89 -> Order 43 -> Order 24 -> Order 107 -> Order 131 - > Order 102 -> Order 68 -> Order 122 -> Order 97 -> Order 62 -> Order 120 -> Order 155 -> Order 125 -> Order 96 -> Order 145 -> Order 82 -> Order 88 -> Order 110 -> Order 78 -> Order 63 -> Order 7 -> Order 14 -> Order 112 -> Order 3 -> DEPOT Route 3: DEPOT -> Order 150 -> Order 134 -> Order 95 -> Order 87 -> Order 108 -> Order 137 -> Order 33 -> Order 12 -> Order 9 - > Order 162 -> Order 76 -> Order 113 -> Order 119 -> Order 101 -> Order 41 -> Order 127 -> Order 163 -> Order 52 -> Order 83 -> Order 20 -> Order 56 -> Order 156 -> Order 57 -> Order 50 -> Order 71 -> Order 28 -> Order 121 -> Order 19 -> DEPOT Route 4: DEPOT -> Order 140 -> Order 116 -> Order 6 -> Order 92 -> Order 151 -> Order 114 -> Order 152 -> Order 111 -> Order 138 -> Order 109 -> Order 144 -> Order 70 -> Order 8 -> Order 124 -> Order 77 -> Order 90 -> Order 17 -> Order 160 -> Order 26 -> Order 27 -> Order 36 -> Order 34 -> Order 16 -> Order 161 -> Order 164 -> Order 166 -> Order 105 -> Order 42 -> DEPOT Route 5: DEPOT -> Order 130 -> Order 25 -> Order 67 -> Order 64 -> Order 11 -> Order 115 -> Order 47 -> Order 168 -> Order 30 - > Order 44 -> Order 48 -> Order 73 -> Order 123 -> Order 22 -> Order 29 -> Order 146 -> Order 37 -> Order 117 -> Order 149 -> Order 136 -> Order 126 -> Order 55 -> Order 118 -> Order 4 -> Order 39 -> Order 2 -> Order 135 -> Order 74 -> DEPOT Route 6: DEPOT -> Order 91 -> Order 141 -> Order 99 -> Order 93 -> Order 79 -> Order 10 -> Order 38 -> Order 167 -> Order 58 -> Order 85 -> Order 80 -> Order 159 -> Order 15 -> Order 158 -> Order 139 -> Order 18 -> Order 66 -> Order 69 -> Order 21 -> Order 128 -> Order 1 -> Order 5 -> Order 100 -> Order 104 -> Order 169 -> Order 72 -> Order 129 -> Order 153 -> DEPOT Best colony: 1 Best iteration: 86

Result report: I-220
Number vehicles: 7 Number orders: 219 Total cost: 173.81348675379405 Cost for route: Route 1: 14.220729292899858 Route 2: 14.614079579376686 Route 3: 75.95005069171887 Route 4: 16.16749974062923 Route 5: 15.229121763906116 Route 6: 13.677867642857267 Route 7: 23.954138042406086 Best configuration of routes: Route 1: DEPOT -> Order 138 -> Order 161 -> Order 75 -> Order 36 -> Order 43 -> Order 66 -> Order 151 -> Order 184 -> Order 157 -> Order 134 -> Order 56 -> Order 29 -> Order 72 -> Order 79 -> Order 58 -> Order 164 -> Order 21 -> Order 111 -> Order 83 -> Order 153 -> Order 89 -> Order 57 -> Order 130 -> Order 69 -> Order 126 -> Order 103 -> Order 154 -> Order 118 -> Order 124 -> Order 51 -> Order 64 -> Order 135 -> DEPOT Route 2: DEPOT -> Order 181 -> Order 93 -> Order 195 -> Order 159 -> Order 115 -> Order 110 -> Order 146 -> Order 91 -> Order 10 -> Order 170 -> Order 175 -> Order 94 -> Order 185 -> Order 59 -> Order 86 -> Order 81 -> Order 167 -> Order 147 -> Order 19 - > Order 97 -> Order 102 -> Order 163 -> Order 133 -> Order 171 -> Order 53 -> Order 178 -> Order 121 -> Order 63 -> Order 123 -> Order 84 -> Order 98 -> Order 105 -> DEPOT Route 3: DEPOT -> Order 127 -> Order 162 -> Order 87 -> Order 62 -> Order 55 -> Order 189 -> Order 61 -> Order 41 -> Order 141 -> Order 173 -> Order 32 -> Order 198 -> Order 33 -> Order 85 -> Order 107 -> Order 197 -> Order 47 -> Order 95 -> Order 199 -> Order 104 -> Order 212 -> Order 205 -> Order 174 -> Order 196 -> Order 188 -> Order 172 -> Order 200 -> Order 169 -> Order 179 - > Order 17 -> Order 18 -> DEPOT Route 4: DEPOT -> Order 26 -> Order 68 -> Order 204 -> Order 65 -> Order 116 -> Order 12 -> Order 215 -> Order 49 -> Order 74 - > Order 131 -> Order 203 -> Order 23 -> Order 213 -> Order 214 -> Order 38 -> Order 73 -> Order 128 -> Order 125 -> Order 101 -> Order 6 -> Order 206 -> Order 136 -> Order 22 -> Order 2 -> Order 191 -> Order 177 -> Order 30 -> Order 144 -> Order 139 -> Order 183 -> Order 180 -> DEPOT Route 5: DEPOT -> Order 190 -> Order 201 -> Order 60 -> Order 76 -> Order 140 -> Order 99 -> Order 150 -> Order 155 -> Order 50 -> Order 156 -> Order 54 -> Order 46 -> Order 1 -> Order 52 -> Order 44 -> Order 90 -> Order 137 -> Order 182 -> Order 48 -> Order 4 -> Order 129 -> Order 143 -> Order 117 -> Order 148 -> Order 7 -> Order 152 -> Order 112 -> Order 114 -> Order 120 -> Order 39 -> Order 42 -> DEPOT Route 6: DEPOT -> Order 106 -> Order 165 -> Order 24 -> Order 82 -> Order 20 -> Order 208 -> Order 122 -> Order 176 -> Order 216 -> Order 108 -> Order 219 -> Order 31 -> Order 25 -> Order 218 -> Order 217 -> Order 45 -> Order 119 -> Order 5 -> Order 40 - > Order 3 -> Order 187 -> Order 113 -> Order 192 -> Order 193 -> Order 15

-> Order 194 -> Order 160 -> Order 78 -> Order 77 -> Order 11 -> Order 8 -> DEPOT

Route 7: DEPOT -> Order 207 -> Order 145 -> Order 109 -> Order 88 -> Order 92 -> Order 149 -> Order 96 -> Order 100 -> Order 142 -> Order 158 -> Order 34 -> Order 9 -> Order 71 -> Order 209 -> Order 210 -> Order 132 -> Order 211 -> Order 13 -> Order 37 - > Order 35 -> Order 28 -> Order 27 -> Order 168 -> Order 186 -> Order 16 -> Order 166 -> Order 70 -> Order 67 -> Order 80 -> Order 202 -> Order 14 -> DEPOT

Best colony: 2

Best iteration: 90

Appendix A.19: Results of the possible solution approaches in the effectiveness analysis

INSTANCES	SOLUTION APPROACH-I				SOLUTION APPROACH-II				SOLUTION APPROACH-III	
	Classification results ($\boldsymbol{CR}$)	Proposed algorithm ($\boldsymbol{A}$)	Solution quality ($\boldsymbol{SC}$)	Computation time ($\boldsymbol{CT}$)	$\boldsymbol{CR}$	$\boldsymbol{A}$	$\boldsymbol{SC}$	$\boldsymbol{CT}$	$\boldsymbol{SC}$	$\boldsymbol{CT}$
1	E	BB	144.21	0.52	A	ACO	156.24	0.10	171.80	0.40
2	E	BB	150.32	0.48	A	ACO	162.44	0.05	187.93	0.02
3	E	BB	119.67	0.45	A	ACO	146.65	0.03	156.06	0.01
4	E	BB	117.33	0.46	A	ACO	132.70	0.03	153.67	0.01
5	E	BB	121.15	0.51	A	ACO	148.58	0.08	157.95	0.03
6	E	BB	93.56	0.38	A	ACO	110.34	0.02	145.15	0.01
7	E	BB	124.90	0.50	A	ACO	139.43	0.05	154.59	0.03
8	E	BB	94.15	0.37	A	ACO	111.22	0.02	142.82	0.01
9	E	BB	151.19	0.48	A	ACO	163.22	0.07	180.21	0.04
10	E	BB	109.17	0.41	A	ACO	128.62	0.02	141.40	0.01
11	E	BB	293.83	1.02	A	ACO	214.33	0.17	222.96	0.05
12	A	ACO	255.43	0.35	E	BB	253.67	21.23	278.47	0.15
13	E	BB	191.23	6.73	A	ACO	204.48	0.12	220.91	0.08
14	E	BB	222.15	18.90	A	ACO	230.60	0.25	245.03	0.11
15	A	ACO	248.53	0.42	E	BB	245.32	28.14	256.90	0.26

INSTANCES	**SOLUTION APPROACH-I**				**SOLUTION APPROACH-II**				**SOLUTION APPROACH-III**	
	Classification results ($\boldsymbol{CR}$)	Proposed algorithm ($\boldsymbol{A}$)	Solution quality ($\boldsymbol{SC}$)	Computation time ($\boldsymbol{CT}$)	$\boldsymbol{CR}$	$\boldsymbol{A}$	$\boldsymbol{SC}$	$\boldsymbol{CT}$	$\boldsymbol{SC}$	$\boldsymbol{CT}$
16	E	BB	229.67	15.43	A	ACO	241.38	0.18	267.14	0.11
17	A	ACO	216.56	0.10	E	BB	212.11	12.34	220.23	0.18
18	E	BB	240.43	21.11	A	ACO	249.89	0.25	268.83	0.20
19	A	ACO	252.85	0.42	E	BB	248.18	28.45	271.34	0.41
20	E	BB	211.34	17.51	A	ACO	224.77	0.20	245.67	0.21
21	A	ACO	312.19	0.53	E	BB	308.01	34.21	336.91	0.51
22	A	ACO	302.83	0.68	E	BB	293.16	36.18	334.56	0.60
23	A	ACO	334.27	0.43	E	BB	229.01	33.49	357.60	0.43
24	A	ACO	335.44	0.82	E	BB	321.94	38.31	361.74	0.81
25	A	ACO	326.08	0.52	E	BB	315.13	34.18	351.58	0.50
26	A	ACO	303.01	0.45	E	BB	293.45	36.17	334.19	0.42
27	A	ACO	352.98	1.03	E	BB	350.03	40.41	372.57	0.54
28	A	ACO	318.05	0.53	E	BB	311.15	38.54	348.67	0.48
29	A	ACO	342.22	0.70	E	BB	333.76	35.37	367.18	0.62
30	A	ACO	344.40	0.73	E	BB	338.42	36.99	368.35	0.63

INSTANCES	SOLUTION APPROACH-I				SOLUTION APPROACH-II				SOLUTION APPROACH-III	
	Classification results ($\boldsymbol{CR}$)	Proposed algorithm ($\boldsymbol{A}$)	Solution quality ($\boldsymbol{SC}$)	Computation time ($\boldsymbol{CT}$)	$\boldsymbol{CR}$	$\boldsymbol{A}$	$\boldsymbol{SC}$	$\boldsymbol{CT}$	$\boldsymbol{SC}$	$\boldsymbol{CT}$
31	A	ACO	410.90	1.70	E	BB	402.65	41.95	461.30	1.56
32	A	ACO	424.13	1.92	E	BB	418.47	43.17	458.13	1.59
33	A	ACO	434.72	2.43	E	BB	425.96	44.89	467.06	1.90
34	A	ACO	428.26	4.17	E	BB	417.17	46.88	451.34	4.03
35	A	ACO	400.87	1.38	E	BB	392.70	38.98	423.11	1.30
36	A	ACO	420.07	1.18	E	BB	411.61	41.77	441.08	1.15
37	A	ACO	419.87	1.13	E	BB	409.24	40.93	448.84	1.10
38	A	ACO	414.19	1.27	E	BB	402.58	41.89	439.63	1.21
39	A	ACO	432.78	1.63	E	BB	420.92	44.35	465.26	1.50
40	A	ACO	429.56	1.82	E	BB	415.85	45.03	473.10	1.68
41	E	BB	192.32	17.01	A	ACO	201.23	0.12	210.98	0.08
42	E	BB	164.59	15.41	A	ACO	172.42	0.05	181.21	0.03
43	E	BB	170.31	16.57	A	ACO	178.70	0.07	189.81	0.04
44	A	ACO	199.62	0.13	E	BB	192.01	26.66	215.35	0.08
45	A	ACO	196.70	0.12	E	BB	191.68	26.13	221.61	0.08

INSTANCES	SOLUTION APPROACH-I				SOLUTION APPROACH-II				SOLUTION APPROACH-III	
	Classification results (CR)	Proposed algorithm (A)	Solution quality (SC)	Computation time (CT)	CR	A	SC	CT	SC	CT
46	A	ACO	220.61	0.17	E	BB	208.18	30.56	243.05	0.12
47	E	BB	102.64	0.45	A	ACO	116.11	0.02	143.26	0.01
48	A	ACO	205.67	0.12	E	BB	201.36	27.95	235.13	0.07
49	E	BB	141.48	16.54	A	ACO	153.18	0.03	171.50	0.01
50	E	BB	180.69	17.00	A	ACO	197.41	0.08	212.01	0.02
51	E	BB	158.32	1.74	A	ACO	164.20	0.03	187.22	0.01
52	E	BB	128.93	0.99	A	ACO	140.45	0.02	162.76	0.01
53	E	BB	128.17	1.20	A	ACO	132.08	0.02	148.37	0.01
54	A	ACO	212.60	0.20	E	BB	210.70	29.35	234.63	0.08
55	E	BB	136.16	1.23	A	ACO	138.90	0.03	151.52	0.02
56	E	BB	135.69	1.31	A	ACO	142.10	0.03	151.96	0.02
57	E	BB	109.00	1.22	A	ACO	121.00	0.02	127.47	0.01
58	E	BB	128.65	0.65	A	ACO	133.94	0.02	146.72	0.01
59	E	BB	108.32	0.89	A	ACO	122.81	0.02	134.19	0.01
60	E	BB	139.28	0.92	A	ACO	147.31	0.02	157.01	0.01

INSTANCES	**SOLUTION APPROACH-I**				**SOLUTION APPROACH-II**				**SOLUTION APPROACH-III**	
	Classification results ($\boldsymbol{CR}$)	Proposed algorithm ($\boldsymbol{A}$)	Solution quality ($\boldsymbol{SC}$)	Computation time ($\boldsymbol{CT}$)	$\boldsymbol{CR}$	$\boldsymbol{A}$	$\boldsymbol{SC}$	$\boldsymbol{CT}$	$\boldsymbol{SC}$	$\boldsymbol{CT}$
61	E	BB	188.04	4.56	A	ACO	196.70	0.08	201.69	0.01
62	E	BB	145.75	1.54	A	ACO	147.62	0.02	148.21	0.01
63	E	BB	118.34	0.95	A	ACO	133.00	0.02	161.52	0.01
64	E	BB	138.71	1.24	A	ACO	146.35	0.03	171.76	0.01
65	E	BB	152.62	3.25	A	ACO	167.18	0.05	183.50	0.01
66	A	ACO	212.23	0.17	E	BB	209.36	21.45	234.93	0.07
67	E	BB	145.32	0.82	A	ACO	157.62	0.02	165.13	0.01
68	E	BB	163.84	3.75	A	ACO	176.66	0.08	188.11	0.03
69	E	BB	137.65	1.34	A	ACO	151.12	0.02	153.23	0.01
70	E	BB	154.82	2.71	A	ACO	169.32	0.02	176.05	0.01
71	E	BB	98.63	0.82	A	ACO	117.07	0.02	142.60	0.01
72	E	BB	171.08	4.01	A	ACO	187.25	0.08	201.95	0.02
73	E	BB	148.68	2.59	A	ACO	158.02	0.03	181.59	0.01
74	E	BB	128.40	1.34	A	ACO	142.94	0.02	152.71	0.01
75	E	BB	122.77	1.21	A	ACO	135.53	0.02	135.58	0.01

INSTANCES	SOLUTION APPROACH-I				SOLUTION APPROACH-II				SOLUTION APPROACH-III	
	Classification results ($\boldsymbol{CR}$)	Proposed algorithm ($\boldsymbol{A}$)	Solution quality ($\boldsymbol{SC}$)	Computation time ($\boldsymbol{CT}$)	$\boldsymbol{CR}$	$\boldsymbol{A}$	$\boldsymbol{SC}$	$\boldsymbol{CT}$	$\boldsymbol{SC}$	$\boldsymbol{CT}$
76	E	BB	196.61	15.67	A	ACO	200.09	0.08	226.58	0.04
77	E	BB	155.35	7.31	A	ACO	171.36	0.05	188.52	0.02
78	E	BB	112.66	1.65	A	ACO	122.42	0.02	135.78	0.01
79	E	BB	158.52	4.87	A	ACO	171.61	0.05	190.97	0.02
80	E	BB	170.24	5.03	A	ACO	181.08	0.08	186.69	0.03
81	E	BB	173.05	5.46	A	ACO	188.04	0.07	217.09	0.02
82	E	BB	177.41	5.13	A	ACO	185.08	0.08	212.52	0.03
83	A	ACO	220.37	0.13	E	BB	218.14	26.78	232.22	0.04
84	E	BB	177.20	6.78	A	ACO	190.76	0.08	213.66	0.03
85	E	BB	183.11	7.02	A	ACO	183.97	0.10	210.54	0.03
86	E	BB	137.79	2.31	A	ACO	145.02	0.03	166.45	0.01
87	E	BB	178.73	6.45	A	ACO	184.55	0.10	202.29	0.04
88	E	BB	193.15	10.12	A	ACO	205.27	0.22	206.36	0.14
89	E	BB	195.61	8.34	A	ACO	197.51	0.20	200.72	0.12
90	E	BB	172.44	3.45	A	ACO	177.62	0.08	204.05	0.03

INSTANCES	**SOLUTION APPROACH-I**				**SOLUTION APPROACH-II**				**SOLUTION APPROACH-III**	
	Classification results ($\boldsymbol{CR}$)	Proposed algorithm ($\boldsymbol{A}$)	Solution quality ($\boldsymbol{SC}$)	Computation time ($\boldsymbol{CT}$)	$\boldsymbol{CR}$	$\boldsymbol{A}$	$\boldsymbol{SC}$	$\boldsymbol{CT}$	$\boldsymbol{SC}$	$\boldsymbol{CT}$
91	A	ACO	425.18	1.88	E	ACO	416.79	67.26	443.03	1.03
92	A	ACO	421.73	1.75	E	ACO	419.48	17.25	441.36	0.49
93	A	ACO	527.78	5.25	E	ACO	518.95	28.60	533.58	4.41
94	A	ACO	520.01	4.03	E	ACO	513.07	171.32	549.53	2.03
95	A	ACO	529.04	3.97	E	ACO	522.17	0.07	532.96	2.92
96	A	ACO	538.45	4.23	E	ACO	528.49	9.46	547.27	3.73
97	A	ACO	518.47	4.07	E	ACO	509.86	42.98	531.38	1.21
98	A	ACO	524.84	3.95	E	ACO	516.11	213.32	543.93	1.31
99	A	ACO	512.39	3.62	E	ACO	509.43	46.20	531.59	3.01
100	A	ACO	419.52	3.10	E	ACO	413.62	173.85	423.40	1.98
101	A	ACO	498.71	3.83	E	ACO	488.96	56.68	506.44	3.29
102	A	ACO	616.36	5.57	E	ACO	614.89	207.84	630.43	3.39
103	A	ACO	576.13	5.25	E	ACO	570.68	138.57	578.17	3.41
104	A	ACO	610.64	6.18	E	ACO	606.37	16.04	635.07	5.29
105	A	ACO	598.66	5.93	E	ACO	591.32	27.14	627.90	3.07

INSTANCES	SOLUTION APPROACH-I				SOLUTION APPROACH-II				SOLUTION APPROACH-III	
	Classification results ($\boldsymbol{CR}$)	Proposed algorithm ($\boldsymbol{A}$)	Solution quality ($\boldsymbol{SC}$)	Computation time ($\boldsymbol{CT}$)	$\boldsymbol{CR}$	$\boldsymbol{A}$	$\boldsymbol{SC}$	$\boldsymbol{CT}$	$\boldsymbol{SC}$	$\boldsymbol{CT}$
106	A	ACO	625.40	6.07	E	ACO	623.00	42.21	640.74	4.99
107	A	ACO	608.78	6.13	E	ACO	604.40	110.37	615.20	2.44
108	A	ACO	603.34	5.70	E	ACO	602.14	113.51	614.02	3.09
109	A	ACO	599.36	5.28	E	ACO	591.59	189.23	625.64	1.75
110	A	ACO	593.26	5.35	E	ACO	586.35	52.66	619.02	5.26
111	A	ACO	418.32	1.85	E	ACO	416.38	163.45	423.02	1.47
112	A	ACO	583.39	5.08	E	ACO	576.92	160.74	584.79	1.11
113	A	ACO	603.34	7.30	E	ACO	602.70	161.18	624.95	5.60
114	A	ACO	600.25	7.60	E	ACO	596.91	180.45	609.98	2.01
115	A	ACO	614.53	8.05	E	ACO	605.67	135.79	615.10	2.47
116	A	ACO	596.14	8.30	E	ACO	592.20	237.32	622.92	0.28
117	A	ACO	604.96	7.20	E	ACO	600.60	297.02	632.83	6.70
118	A	ACO	607.08	7.78	E	ACO	604.45	101.59	614.62	3.57
119	A	ACO	612.85	7.18	E	ACO	612.55	180.92	624.05	1.49
120	A	ACO	469.07	6.08	E	ACO	467.41	198.61	478.54	3.39

INSTANCES	**SOLUTION APPROACH-I**				**SOLUTION APPROACH-II**				**SOLUTION APPROACH-III**	
	Classification results ($\boldsymbol{CR}$)	Proposed algorithm ($\boldsymbol{A}$)	Solution quality ($\boldsymbol{SC}$)	Computation time ($\boldsymbol{CT}$)	$\boldsymbol{CR}$	$\boldsymbol{A}$	$\boldsymbol{SC}$	$\boldsymbol{CT}$	$\boldsymbol{SC}$	$\boldsymbol{CT}$
121	A	ACO	604.14	7.57	E	ACO	596.50	283.36	607.44	1.38
122	A	ACO	425.94	2.05	E	ACO	417.05	32.91	428.20	1.10
123	A	ACO	425.05	1.92	E	ACO	425.01	153.99	427.52	1.44
124	A	ACO	432.60	1.87	E	ACO	427.64	59.23	440.52	0.27
125	A	ACO	437.96	1.93	E	ACO	437.07	52.89	458.77	0.52
126	A	ACO	442.57	2.02	E	ACO	440.86	98.36	455.10	0.93
127	A	ACO	450.05	2.18	E	ACO	442.80	86.57	455.18	0.44