UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

TESIS

MATHEMATICAL FORMULATIONS AND OPTIMIZATION ALGORITHMS FOR SOLVING RICH VEHICLE ROUTING PROBLEMS

PRESENTADA POR

PAMELA JOCELYN PALOMO MARTÍNEZ

COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE DOCTOR EN INGENIERÍA CON ESPECIALIDAD EN INGENIERÍA DE SISTEMAS

AGOSTO, 2018

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO

TESIS

MATHEMATICAL FORMULATIONS AND OPTIMIZATION ALGORITHMS FOR SOLVING RICH VEHICLE ROUTING PROBLEMS

PRESENTADA POR

PAMELA JOCELYN PALOMO MARTÍNEZ

COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE DOCTOR EN INGENIERÍA CON ESPECIALIDAD EN INGENIERÍA DE SISTEMAS

AGOSTO, 2018

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirección de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis "Mathematical formulations and optimization algorithms for solving rich vehicle routing problems", realizada por la alumna Pamela Jocelyn Palomo Martínez, con número de matrícula 1502428, sea aceptada para su defensa como opción al grado de Doctora en Ingeniería con Especialidad en Ingeniería de Sistemas.

El Comité de Tesis

Dra. María Angélica Salazar Aguilar
Directora

Dr. Vincent André Lionel Boyer
Revisor

Dr. Roger Z. Ríos Mercado
Revisor

Dr. José Luis González Velarde
Revisor

Dra. Irma Delia García Calvillo
Revisora

Vo. Bo.

Dr. Simón Martínez Martínez
Subdirector de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, Agosto 2018

To my beloved husband
and our beautiful children.

Contents

List of Figures

List of Tables

List of Algorithms

Acknowledgments

I am deeply grateful to Professor Angélica Salazar for giving me the opportunity to work under her wise and skillful yet kindly guidance over the last five years. Certainly, I am taking with me many words of wisdom and pleasant memories.

My thanks also go to Professor Vincent Boyer, Professor Roger Ríos, Professor José Luis González, and Professor Irma García for taking some of their time to read this thesis and make suggestions that certainly helped to enhance its quality.

I would also want to express my gratitude to Professor Gilbert Laporte for giving me the opportunity to work under his supervision during a three-month research stay at CIRRELT in Montreal, Canada. This was an unforgettable experience from which I took with me many things, both academically and culturally speaking.

My warmest gratitude goes to my beloved husband, Oscar Pérez. He has been a bedrock of love, confidence, and support over the last seven years. I thank him for being by my side while I was writing my Bachelor, Master, and PhD theses. I think that is a real proof of love.

My gratitude also goes to my mother, Pedro D., Tanya, Estefanía, and Marco for always believing that I am the most talented and intelligent person in the world. I can assure them that I am not, but the fact that they believe it, inspires me to be as good as I could possibly be.

Thanks to my dear friends, especially to Pedro L., Eduardo, and Norberto. We started this together and now we take our own way, but all the good times we have shared will remain in my heart.

I also wish to take this opportunity to extend my deepest acknowledgment to the CONACYT, the Sofía Kovalevskaya Foundation, the Mexican Mathematical Society, and the FIME-UANL for the financial support and scholarships they have granted me during my PhD studies.

Acronyms

GVNS General Variable Neighborhood Search . . . 26

ILS Iterated Local Search . . . 69

OP Orienteering Problem . . . 123

OPMVC Orienteering Problem with Madatory Visits and Conflicts . . . 123

OPVP Orienteering Problem with Variable Profits . . . 68

MCTOPMTW Multi-Constraint Team Orienteering Problem with Multiple Time Windows . . . 69

MCTOPTW Multi-Constraint Team Orienteering Problem with Time Windows 68

PCTSP Prize Collecting Traveling Salesman Problem . . . 125

TPPMSD Traveling Purchaser Problem with Multiple Stacks and Deliveries . . . 13

PTP Profitable Tour Problem . . . 2

rTOP rich Team Orienteering Problem . . . 123

RVNS Relinked Variable Neighborhood Search . . . 124

SA Simulated Annealing 69

SDCTOP Split Delivery Capacitated Team Orienteering Problem 67

SDCTOP-IS Split Delivery Capacitated Team Orienteering Problem with Incomplete Services 68

SDCTOP-MDA Split Delivery Capacitated Team Orienteering Problem with Minimum Delivery Amounts 68

SPDP Selective Pickup and Delivery Problem 8

TOP Team Orienteering Problem 124

TPP Traveling Purchaser Problem 123

TS Tabu Search 67

TSP Traveling Salesman Problem 124

VND Variable Neighborhood Descent 26

VNS Variable Neighborhood Search 124

VRP Vehicle Routing Problem 123

Summary

Pamela Jocelyn Palomo Martínez.

Ph.D. candidate in Engineering with a Specialization in Systems Engineering.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Title of the study: Mathematical formulations and optimization algorithms for solving rich vehicle routing problems.

Number of pages: 221.

Objectives and methods of study: The main objective of this work is to analyze and solve three different rich selective Vehicle Routing Problems (VRPs).

The first problem is a bi-objective variant of the well-known Traveling Purchaser Problem (TPP) in which the purchased products are delivered to customers. This variant aims to find a route for which the total cost (transportation plus purchasing costs) and the sum of the customers's waiting time are simultaneously minimized. A mixed integer bi-objective programming formulation of the problem is presented and tested with CPLEX 12.6 within an ϵ-constraint framework which fails to find non-dominated solutions for instances containing more than 10 nodes. Therefore, a heuristic based on relinked local search and Variable Neighborhood Search (VNS) is proposed to approximate the Pareto front for large instances. The proposed heuristic was tested over a large set of artificial instances of the problem. Computational results over small-sized instances show that the heuristic is competitive with the ϵ-constraint method. Also, computational tests over large-sized instances were carried out in order to study how the characteristics of the instances impact the algorithm performance.

The second problem consists of planning a selective delivery schedule of multiple products. The problem is modeled as a multi-product split delivery capacitated team orienteering problem with incomplete services, and soft time windows. The problem is modeled through a mixed integer linear programming formulation and approximated by means of a multi-start Adaptive Large Neighborhood Search (ALNS) metaheuristic. Computational results show that the multi-start metaheuristic reaches better results than its classical implementation in which a single solution is build and then improved.

Finally, an Orienteering Problem (OP) with mandatory visits and conflicts, is formulated through five mixed integer linear programming models. The main difference among them lies in the way they handle the subtour elimination constraints. The models were tested over a large set of instances of the problem. Computational experiments reveal that the model which subtour elimination constraints are based on a single-commodity flow formulation allows CPLEX 12.6 to obtain the optimal solution for more instances than the other formulations within a given computation time limit.

CONTRIBUTIONS: The main contributions of this thesis are:

- The introduction of the bi-objective TPP with deliveries since few bi-objective versions of the TPP have been studied in the literature. Furthermore, to the best of our knowledge, there is only one more work that takes into account deliveries in a TPP.

- The design and implementation of a hybrid heuristic based on relinked local search and VNS to solve the bi-objective TPP with deliveries. Additionally, we provide guidelines for the application of the heuristic when different characteristics of the instances are observed.

- The design and implementation of a multi-start adaptive large neighborhood search to solve a selective delivery schedule problem.

- The experimental comparison among different formulations for an OP with mandatory nodes and conflicts.

Signature of the faculty adviser: ______________________________

Professor María Angélica Salazar Aguilar

Chapter 1

Introduction

The transportation of products plays a fundamental role in supply chains since designing appropriate delivery schedules avoids delays and reduces costs, thus increasing customer satisfaction which translates to an increase in the company income.

Due to the importance of transportation, the Vehicle Routing Problem (VRP) has become a classical problem in the operations research literature. Several advanced algorithms have been developed for solving it and its variants (see Toth and Vigo (2014)). Nevertheless, some real-life applications do not enforce to visit all customers due to resource limitations or because it is possible to satisfy the requirements by visiting only a subset of customers. Therefore, in this kind of problems both selecting and routing decisions must be made. A VRP in which is not mandatory to visit all customers is known as a selective vehicle routing problem.

1.1 Selective vehicle routing problems

Despite the practical importance of selective VRPs , they have not been as widely studied as the classical VRPs. Nonetheless, they have increasingly gained attention from operational researchers; thus, several works regarding these problems can be found in the literature.

A class of selective VRPs is known as VRPs with profits. In these problems, a non-negative profit is associated with each customer and it is collected only if the customer is visited. Archetti et al. (2014c) present a survey on the most widely studied VRPs with profits. If there is only one vehicle available, the following problems arise:

- **The Orienteering Problem (OP):** The OP was was introduced by Tsiligirides (1984). The objective is to maximize the total collected profit while the duration of the route does not exceed a threshold.
- **Prize Collecting Traveling Salesman Problem (PCTSP):** This problem was introduced by Balas (1989), and the objective is to minimize the duration of the route by ensuring that the total collected profit is, at least, as large as a given limit.
- **Profitable Tour Problem (PTP):** Introduced by Dell'Amico et al. (1995), the PTP aims at minimizing the route duration minus the collected profit.

The Team Orienteering Problem (TOP) is the most extensively studied multi-vehicle VRP with profits. The TOP is an extension of the OP, introduced by Butt and Cavalier (1994) under the name of Multiple Tour Maximum Collection Problem. The name TOP was later coined by Chao et al. (1996).

Apart from the VRPs with profits, there are other selective VRPs in which no profits are associated with the customers but only a subset of customers is visited because it is possible to satisfy all requirements in this way. Below, some of these problems are described.

The Traveling Purchaser Problem (TPP) was introduced in the scheduling context by Burstall (1966) and in the routing context by Ramesh (1981). In the TPP, there is a demand of products to be satisfied. The products are available for sale in different markets but the offer and price vary from one market to another.

The objective of the problem is to design a route that visits a subset of markets to satisfy a given demand, while the sum of the traveling and the purchasing costs is minimized.

The Covering Tour Problem (CTP) was first introduced by Current (1982) and later studied by Gendreau et al. (1997). In the CTP the customer set is divided into two subsets: V and W. The problem consists of designing a route that visits some customers of V in such way that all customers in W are within a given distance from the route, while the length of the route is minimized.

Finally, the Selective Pickup and Delivery Problem (SPDP) consists of finding a minimum-length route over a set of customers for which pickups and delivery demands exist. All demands must be satisfied while only a subset of pickups have to be performed. Some of the studied variants assume that a profit is associated with each pickup and thus the objective is to minimize the length of the route minus the collected score. Single-vehicle versions of the SPDP have been studied by Süral and Bookbinder (2003); Gribkovskaia et al. (2008); Gutiérrez-Jarpa et al. (2009); Falcon et al. (2010) and Ting and Liao (2013), while multi-vehicle variants have been addressed by Gutiérrez-Jarpa et al. (2010) and Ting et al. (2017).

In this thesis, three different selective VRPs are studied. Two of them belong to the family of the VRPs with profits, while the remaining one does not consider the existence of profits.

The first problem is a bi-objective variant of the TPP, the so-called the bi-objective Traveling Purchaser Problem with Deliveries (2-TPPD). In the 2-TPPD, there is a set of customers that are geographically distributed in the same geographical area than the markets. The objective is to design a route to satisfy the demand of all customers by minimizing the total cost (traveling plus purchasing costs) and the sum of the customers's waiting time, simultaneously.

The second problem discussed in this thesis has to do with designing a selective delivery schedule of products with multiple side-constraints. This problem is modeled as a rich Team Orienteering Problem (rTOP) considering the following features: (i) delivery of multiple products, (ii) split deliveries, (iii) an heterogeneous fleet of vehicles, (iv) incomplete services, and (v) soft time windows. The objective is to design a set of routes in such a way that the collected score is maximized while all constraints are satisfied.

Finally, the third problem is a variant of the OP, called the Orienteering Problem with Madatory Visits and Conflicts (OPMVC). In this problem, it is mandatory to visit some customers and there are conflicting visits, meaning that if a customer is in conflict with another one, at most one of them can be visited. The OPMVC consists of designing a route whose duration does not exceed a time threshold, including all mandatory and some optional nodes, without conflicts among them, while the collected score is maximized.

1.2 Motivation

The interest in studying selective VRPs arises from the relatively scarce literature regarding these problems, despite the facts that they are more general than classical VRPs and they capture many real-life problems features. As a matter of fact, the problems analyzed in this thesis arise from real-life situations as it will be discussed in depth in this thesis.

1.3 Methodology

The methodology followed in this thesis consists in the next steps:

- First, the OPMVC was addressed as follows:
  - Literature review on subtour elimination constraints for the Traveling Salesman Problem (TSP).

  - Problem modeling by adapting the subtour elimination constraints to the OPMVC.
  - Test instance generation.
  - Empirical assessment of models over the instances generated in the previous step and the ones used by Palomo-Martínez et al. (2017) through CPLEX 12.6.
  - Analysis of results.

- After that, the next steps were carried out to study the rTOP:
  - Literature review on the OP, the TOP, and some of their variants that lie at the heart of the rTOP.
  - Problem modeling.
  - Test instance generation.
  - Model testing over the generated instances.
  - Design and computational implementation of a metaheuristic based on Adaptive Large Neighborhood Search (ALNS) for tackling the problem.
  - Algorithm testing.
  - Analysis of results.

- Next, the 2-TPPD was studied through the following steps:
  - Literature review on the TPP and the minimum latency problem.
  - Problem modeling.
  - Test instance generation.
  - Model testing by optimizing the two different objectives independently to state the bi-objective nature of the problem.
  - Design and computational implementation of an ϵ-constraint method.
  - Assessment of the ϵ-constraint method.

- Design and computational implementation of a metaheuristic based on relinked local search and Variable Neighborhood Search (VNS).
- Metaheuristic testing.
- Analysis of results.

1.4 Thesis structure

The remainder of this thesis is structured as follows. The 2-TPPD is addressed in Chapter 2. Chapter 3 relates to the rTOP, while Chapter 4 addresses the OPMVC. Conclusions and further research are discussed in Chapter 5. Extensive computational results associated with the rTOP and the OPMVC can be found in Appendices A and B, respectively.

Chapter 2

The bi-objective traveling purchaser problem with deliveries

The bi-objective Traveling Purchaser Problem with Deliveries (2-TPPD) is introduced in this chapter. The 2-TPPD is a variant of the well-known Traveling Purchaser Problem (TPP) in which the purchased products are delivered to a set of customers. A mixed integer bi-objective programming formulation is proposed to model the problem. Computational experiments reveal that CPLEX 12.6 in combination with an ϵ-constraint method cannot solve instances containing more than 10 nodes. Then, a heuristic based on relinked local search and Variable Neighborhood Search (VNS) is proposed to approximate the Pareto front of large instances. Three different variants of the heuristic are tested over a large set of instances of the problem. Furthermore, a comprehensive analysis on how the characteristics of the tested instances affect the performance of the heuristic is presented.

2.1 Motivation

The 2-TPPD arises from a real life situation faced by a local company. The company has a team of technicians devoted to deliver office supplies and to perform informatics and telecommunications activities at the company's branch offices. The required materials to carry out each activity are known. Due to the lack of space at the depot, the stock is not large enough for the team to perform all the scheduled activities at the branch offices; therefore, the team must purchase materials and perform the activities on the same working day. There is a vehicle owned by the company that is available for visiting material suppliers and branch offices. The working day starts and ends at the depot. The company wishes to minimize the total cost, which consists of the transportation and the purchasing costs. Besides, the company also wishes the activities to be performed as soon as possible.

2.2 Problem description

The 2-TPPD shares characteristics with a many-to-many version of the Selective Pickup and Delivery Problem (SPDP) described in Section 1.1 since both of them consist of the distribution of commodities from some locations to others and, also, it is not enforced to visit all pickup locations (suppliers). Nonetheless, in the 2-TPPD, the pickup orders are not stated *a priori* but it is part of the decision process to choose how many units of products will be purchased at each supplier location.

Then, the problem described in Section 2.1 is modeled as a variant of the well-known TPP, described in Section 1.1. In the variant introduced in this chapter, the demand is given by a set of customers (branch offices). The demanded products are the office supplies and those that are required to perform the activities. There is a stock stored at the depot, but it is assumed that it is insufficient to satisfy the total demand. Then, some products are purchased in the markets and then delivered to the customers. The service time is known for all markets and customers. The problem consists of designing a route starting and ending at the depot in which all customers and some markets are visited in such a way that all demands are satisfied and the total cost as well as the sum of the customers waiting time are minimized simultaneously. The latter objective is known in the literature as latency.

2.3 Literature review

Here, a brief review on the TPP and its variants is presented. The interested reader is referred to Manerba et al. (2017) for a comprehensive survey.

The TPP was introduced in the scheduling context by Burstall (1966) and in the routing context by Ramesh (1981). In order to solve the TPP, several heuristics (Golden et al., 1981; Ong, 1982; Pearn and Chien, 1998; Boctor et al., 2003; Teeninga and Volgenant, 2004; Riera-Ledesma and Salazar-González, 2005b; Kang et al., 2006) and metaheuristics such as Tabu Search (TS) (Voß, 1996b), Greedy Randomized Adaptive Search Procedure (GRASP) and VNS (de Assumpção Drummond et al., 2002), Ant Colony Optimization (ACO) (Bontoux and Feillet, 2008), and evolutionary algorithms (Goldbarg et al., 2009; Bernardino and Paias, 2016), have been proposed in the literature. Also, some exact procedures have been developed for its solution, such as branch-and-cut (B&C) (Laporte et al., 2003; Riera-Ledesma and Salazar-González, 2006) and constraint programming (Cambazard and Penz, 2012).

2.3.1 Single-vehicle traveling purchaser problem variants

The following single-vehicle variants of the TPP can be found in the literature:

- **The dynamic traveling purchaser problem:** In this variant the available offer at each market is reduced as time advances. It is assumed that the decision maker has complete information about the current offer at each market and is informed about consumptions as they occur. This problem has been addressed by:

- Angelelli et al. (2008): It is assumed that there is not available information about future events. The problem was solved through several greedy heuristics.
- Angelelli et al. (2011): As before, there is not available information about the future. The authors designed some look-ahead heuristics that try to incorporate future prediction. These heuristics were shown to deal better with product scarcity than the ones proposed by Angelelli et al. (2008).
- Angelelli et al. (2016): In this version of the dynamic TPP, the available offer is reduced according to a Markov process. The authors solved the problem by means of three versions of a heuristic.
- Angelelli et al. (2017): It is assumed that the available offer is time-dependent and is reduced at a constant rate. The problem was solved by B&C.

- **The stochastic traveling purchaser problem:**
  - Beraldi et al. (2015): In this variant of the TPP, the offer and prices are uncertain. The authors modeled the problem trough a two-stage stochastic programming formulation, where the first stage relates to market selection and visiting order, and the second, to the purchases. The problem was solved through B&C and a heuristic used to find initial solutions.
  - Kang and Ouyang (2011): In this variant the prices are stochastic. The authors solved the problem by means of dynamic programming, an iterative approximation algorithm, and a greedy heuristic.

- **The traveling purchaser problem with additional side-constraints:** This problem, studied by Gouveia et al. (2011), arises from an application in machine scheduling. The problem is modeled as a TPP in which there is a limit on the maximum number of markets to be visited, there is a limit on the number of units to be purchased in each market, only one unit of each item is required, and the number of products is small in comparison to the number of markets. The authors solved the problem through dynamic programming.

- **The traveling purchaser problem with budget constraint:** This variant, introduced by Mansini and Tocchella (2009), seeks to minimize the traveling cost while the purchasing cost is constrained not to exceed a given limit. The problem was solved by enhanced local search and VNS.

2.3.2 Multi-vehicle traveling purchaser problem variants

Apart from the single-vehicle variants of the TPP, the following multi-vehicle variants have been addressed in the literature:

- **The multiple traveling purchaser problem for maximizing system's reliability with budget constraints:** It was introduced by Choi and Lee (2011) and modeled as an integer linear programming formulation.

- **The multiple vehicle traveling purchaser problem:** This problem, studied by Riera-Ledesma and Salazar-González (2012), was used to model the school bus routing problem. The problem was solved through B&C.

- **The multiple vehicle traveling purchaser problem with resource constraints:** Riera-Ledesma and Salazar-González (2013) extended the school bus routing problem proposed by Riera-Ledesma and Salazar-González (2012) by taking into account the resource constraints: an upper bound in the length of the route and an upper bound on the total distance walked by the students. The problem was solve by a column generation scheme.

- **The distance constrained multi vehicle traveling purchaser problem:** This problem was introduced by Bianchessi et al. (2014). The problem consists of minimizing the purchasing cost while the distance cannot exceed a threshold. The authors solved the problem by branch-and-price (B&P).

- **The multi-vehicle traveling purchaser problem with pairwise incompatibility constraints:** This problem was proposed by Manerba and Mansini (2015) to address the situation in which load compatibilities arise and thus some products cannot be transported together in the same vehicle. The authors solved the problem through B&C. The same problem with unitary demands was later solved by Gendreau et al. (2016) by means of a B&P scheme.

2.3.3 Bi-objective traveling purchaser problem variants

Sometimes, it is difficult to measure the traveling cost and the purchasing cost in the same units. Therefore, the following authors have been approached the TPP as a bi-objective problem in which the traveling cost and the purchasing cost are minimized simultaneously.

- Ravi and Salman (1999): The authors proposed an approximation algorithm with a poly-logarithmic worst-case ratio for the bi-objective TPP in which the triangle inequality holds. They also developed a constant-factor approximation algorithm for the bi-objective TPP that models the ring-star network design problem.

- Riera-Ledesma and Salazar-González (2005a): In this work, it is proposed a solution algorithm based on a dynamic weighting method in which the single-objective problems are solved through an adaptation of the B&C developed by Laporte et al. (2003).

- Almeida et al. (2012): Two solution approaches were proposed to solve the problem, namely non-dominated sorting transgenetic algorithm and multiobjective transgenetic algorithm/decomposition. The results show the latter outperforms the former when different performance metrics are considered.

The green traveling purchaser problem is another bi-objective TPP in which the objectives to be minimized are the total cost and the CO_2 emissions. This problem was introduced by Hamdan et al. (2017) and solved through B&C by transforming the bi-objective model into a single-objective one by means of the weighted comprehensive criterion method.

2.3.4 The traveling purchaser problem with multiple stacks and deliveries

To the best of our knowledge, the only TPP variant that considers deliveries is the Traveling Purchaser Problem with Multiple Stacks and Deliveries (TPPMSD) proposed by Batista-Galván et al. (2013). In this problem, there is a set of pickup nodes (markets) and a set of delivery nodes (customers). Each delivery node is associated with a single product and when a market offers a product, it is able to fully satisfy its demand. Since the pickup and delivery nodes are widely separated, all pickups must be performed before the deliveries. Besides, the load space in the vehicle is divided into stacks with a fixed height and the loading operations follow a last-in-first-out policy. Thus, both pickups and deliveries must be consistent with the container configuration. The authors solved instances with up to 24 products and 32 markets through B&C.

Even though both the 2-TPPD and the TPPMSD are TPP variants in which deliveries are taken into account, substantial differences exist between them, as described below.

- Objective function:
  - **2-TPPD:** Bi-objective optimization problem. The objectives to minimize are the total cost and the latency.
  - **TPPMSD:** Single-objective optimization problem. The objective to minimize is the total cost.
- Markets and customers distribution:
  - **2-TPPD:** Both markets and customers are located in the same geographical area. Therefore, one of the main difficulties of the problem is that it has to be ensured that, when a customer is visited, the vehicle load is enough to satisfy its demand.
  - **TPPMSD:** Markets and customers are widely separated, such that all purchases are performed before the deliveries. Then, every time that a customer is visited, its demand can be satisfied by the vehicle load.
- Load constraints:
  - **2-TPPD:** No load constraints are considered.
  - **TPPMSD:** The loading space is divided into stacks and the loading operations follow a last-in-first-out policy, thus the routing decisions must be consistent with the container configuration.
- Stock:
  - **2-TPPD:** There is stock available at the depot, but it is not large enough to satisfy the total demand.
  - **TPPMSD:** There is no stock available.

2.4 Bi-objective optimization overview

For a better understanding of the 2-TPPD, this section discusses some basic concepts on bi-objective optimization.

DEFINITION 2.1 (BI-OBJECTIVE OPTIMIZATION PROBLEM) *A bi-objective optimization problem is defined as follows:*

$$\begin{aligned} \textit{minimize } F(x) &= (F_1(x), F_2(x)) && (2.1)\\ \textit{subject to:} & \\ x &\in X && (2.2) \end{aligned}$$

where $F : X \to \mathbb{R}^2$.

$X \subset \mathbb{R}^n$ *is known as the feasible solution (or decision) space and* $\mathbb{R}^n$ *is known as the solution (or decision) space. On the other hand,* $Y = \{(F_1(x), F_2(x)) \in \mathbb{R}^2 : x \in X\}$ *is known as the feasible objective space and* $\mathbb{R}^2$ *is the objective space.*

It is assumed that there is no solution that optimizes $F_1 : X \to \mathbb{R}$ and $F_2 : X \to \mathbb{R}$, simultaneously. Then, we say that the objectives are in conflict and we are looking for compromise solutions rather than optimal ones. With this purpose, we describe the concepts of Pareto optimality and weak Pareto optimality.

DEFINITION 2.2 (PARETO OPTIMALITY) *Let* $x_1 \in X$ *and* $x_2 \in X$ *be two different solutions. We say that* x_1 *(Pareto) dominates* x_2 *if and only if* $F_1(x_1) \leq F_1(x_2)$, $F_2(x_1) \leq F_2(x_2)$, *and at least one of the inequalities is strict.*

A solution $x^* \in X$ *is known as Pareto optimal or Pareto efficient if there is no other solution that dominates it.*

DEFINITION 2.3 (WEAK PARETO OPTIMALITY) *A solution* $x^* \in X$ *is known as* *weak Pareto optimal if there is no other solution* $x \in X$ *such that* $F_1(x) < F_1(x^*)$, *and* $F_2(x) < F_2(x^*)$.

Notice that many Pareto optimal solutions may exist for the same problem. Thus, a solution algorithm for a bi-objective optimization problem must report a set of Pareto optimal solutions. Then, we define the Pareto set and the Pareto front concepts.

DEFINITION 2.4 (PARETO SET) *The Pareto set* PS *is defined as follows:*

$$PS = \{x \in X : x \text{ is Pareto optimal}\}. \tag{2.3}$$

DEFINITION 2.5 (PARETO FRONT) *The Pareto front* PF *is defined as the image of the Pareto set* PS *in the objective space, i.e.,*

$$PF = \left\{(F_1(x), F_2(x)) \in \mathbb{R}^2 : x \in PS\right\}. \tag{2.4}$$

In practice, it can be quite difficult to calculate the whole Pareto set and Pareto front. Then, many solution approaches return approximations of these sets, which are defined as follows.

DEFINITION 2.6 (PARETO SET APPROXIMATION) *Let* $\tilde{PS} \in X$ *be a set of feasible solutions.* $\tilde{PS}$ *is a Pareto set approximation if for all* $x^1 \in \tilde{PS}$ *does not exists any other solution* $x^2 \in \tilde{PS}$ *such that* x^2 *dominates* x^1.

DEFINITION 2.7 (PARETO FRONT APPROXIMATION) *Let* $\tilde{PS}$ *be a Pareto set approximation. Then, the Pareto front approximation* $\tilde{PF}$ *associated with* $\tilde{PS}$ *is defined as its image in the objective space, i.e.,*

$$\tilde{PF} = \left\{(F_1(x), F_2(x)) \in \mathbb{R}^2 : x \in \tilde{PS}\right\}. \tag{2.5}$$

To evaluate the quality of a Pareto front approximation, three major criteria have been considered in the literature: capacity, convergence, and diversity. The capacity refers to the number of solutions in the Pareto front approximation that meet some requirements. The convergence relates to the proximity of the Pareto front approximation to the Pareto front. The diversity refers to how evenly dispersed are the points in the Pareto front approximation. (Jiang et al., 2014)

Taking into account these criteria, several performance metrics have been proposed in the literature to measure the quality of Pareto front approximations. Table 2.1 describes the four metrics that will be used in this chapter to evaluate the algorithms proposed to solve the 2-TPPD. Detailed information about the performance metrics used to assess the quality of multi-objective optimization algorithms can be found in Jiang et al. (2014).

Table 2.1: Performance metrics to evaluate bi-objective optimization algorithms

Performance metric	**Criteria**			**Description**
	Capacity	**Convergence**	**Diversity**	
Overall Nondominated Vector Generation (ONVG)	✓			Proposed by Veldhuizen and Lamont (2000), it measures the number of points in the Pareto front approximation

Continues on next page

Continued from previous page

Performance metric	Criteria			Description
	Capacity	Convergence	Diversity	
k-distance (kD)			✓	This metric was introduced by Zitzler et al. (2001). For each point in the objectives space, it measures the Euclidean distance to the k-th nearest point. The average k-distance is considered in this thesis to evaluate the whole Pareto front approximation.
Size of space covered or hypervolume (Hv)		✓	✓	Introduced by Zitzler and Thiele (1999). In the bi-dimensional case, it measures the area of the dominated portion of the rectangle $(0, 0)$ and a reference point chosen in such a way that all points in the Pareto front approximation dominate it. A larger hypervolume corresponds to a better Pareto front approximation.

Continues on next page

Continued from previous page

Performance metric	Criteria: Capacity	Criteria: Convergence	Criteria: Diversity	Description
Two set coverage	✓			Proposed by Zitzler (1999). Let A and B be two Pareto front approximations, C(A,B) measures the proportion of points in B that are dominated by at least one in A: $$C(A,B) = \frac{\left\lvert \left\{ \begin{array}{c} x \in B : x \text{ is} \\ \text{dominated by} \\ \text{at least one} \\ \text{solution in } A \end{array} \right\} \right\rvert}{\lvert B \rvert}. \quad (2.6)$$

2.5 Mathematical model

In this section, the problem is formally described and modeled through a mixed integer bi-objective programming formulation.

2.5.1 Notation

Let C be the set of customers and P_i the set of products required by customer $i \in C$. The number of units of product $p \in P_i$ that are demanded by customer i is denoted as d_{pi}. The set $P = \underset{i \in C}{\cup} P_i$ is the product set. The number of units of product p stored at the depot is denoted as s_p. For every product p, there is a set of markets M_p in which it can be purchased. Each market $i \in M_p$ makes q_{pi} units of product p available for sale at unitary cost c_{pi}. The set $M = \underset{p \in P}{\cup} M_p$ is the market set. In the 2-TPPD a complete graph $G = (N, A)$ is given, where $N = \{0\} \cup C \cup M \cup \{n+1\}$ is the node set, A is the arc set, and nodes 0 and $n+1$ are the same depot, where $n = |C| + |M|$. The travel cost and the travel time for arc (i, j) are denoted as e_{ij} and t_{ij}, respectively. The service time for node $i \in N$ is denoted as a_i.

The objective is to design a route that minimizes the total cost and the latency, simultaneously, subject to the following constraints:

- the route starts at 0 and ends at $n+1$;
- all customer demands are satisfied;
- the quantity of product p delivered to a customer i cannot exceed its demand d_{ip};
- some markets are visited; and
- when a market i is visited, the purchased units w_{pi} must not exceed the offer q_{pi}.

2.5.2 Mixed integer bi-objective programming formulation

The following decision variables are used to model the 2-TPPD:

$$x_{ij} = \begin{cases} 1 & \text{if arc } (i,j) \in A \text{ is traversed} \\ 0 & \text{otherwise;} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{if node } i \in N \text{ is visited} \\ 0 & \text{otherwise;} \end{cases}$$

$$u_{ij} = \begin{cases} 1 & \text{if node } i \in N \text{ is visited before customer } j \in C; i \neq j \\ 0 & \text{otherwise;} \end{cases}$$

v_i arrival time at node $i \in N$;

w_{pi} quantity of product $p \in P$ purchased in market $i \in M_p$.

Then, the TPP is modeled as follows:

$$\text{minimize } z_1 = \sum_{(i,j)\in A} e_{ij}x_{ij} + \sum_{p\in P}\sum_{i\in M_p} c_{pi}w_{pi} \tag{2.7}$$

$$\text{minimize } z_2 = \sum_{i\in C} v_i \tag{2.8}$$

subject to:

$$\sum_{i\in N:(0,i)\in A} x_{0i} = 1 \tag{2.9}$$

$$\sum_{i\in N:(i,n+1)\in A} x_{in+1} = 1 \tag{2.10}$$

$$\sum_{j\in N:(j,i)\in A} x_{ji} = y_i \qquad i \in N\backslash\{0\} \tag{2.11}$$

$$\sum_{j\in N:(i,j)\in A} x_{ij} = y_i \qquad i \in N\backslash\{n+1\} \tag{2.12}$$

$$y_i = 1 \qquad i \in C \tag{2.13}$$

$$v_i + a_i + t_{ij} \leq v_j + T(1 - x_{ij}) \qquad (i,j) \in A \tag{2.14}$$

$$u_{ij} \leq y_i \qquad i \in N, j \in C \tag{2.15}$$

$$T(u_{ij}-1) \leq v_j - v_i \leq T u_{ij} \qquad i \in N \backslash \{n+1\}, j \in C \tag{2.16}$$

$$w_{pi} \leq q_{pi} y_i \qquad p \in P, i \in M_p \tag{2.17}$$

$$s_p + \sum_{j \in M_p} w_{pj} u_{ji} - \sum_{j \in C \backslash \{i\}} d_{pj} u_{ji} \geq d_{pi} \qquad i \in C, p \in P_i \tag{2.18}$$

$$v_i \geq 0 \qquad i \in N \tag{2.19}$$

$$w_{pi} \in \{0\} \cup \mathbb{Z}^+ \qquad p \in P, i \in M_p \tag{2.20}$$

$$x_{ij} \in \{0,1\} \qquad (i,j) \in A \tag{2.21}$$

$$y_i \in \{0,1\} \qquad i \in N \tag{2.22}$$

$$u_{ij} \in \{0,1\} \qquad j \in C, i \in N \backslash \{j\} \tag{2.23}$$

Objective function (2.7) seeks to minimize the sum of the traveling cost and the purchasing cost, while objective function (2.8) seeks to minimize the latency. Notice that the latency is defined as the sum of the customers' waiting time and it does not take into account the markets' waiting time.

Constraints (2.9) and (2.10) ensure that the route starts and ends at the depot, respectively. Constraints (2.11) and (2.12) assure flow conservation. Constraints (2.13) impose that all customers must be served. Constraints (2.14) ensure time consistency and avoid subtours, where T is a sufficiently large constant. Constraints (2.15) and (2.16) assure that a node is visited before a customer if its arrival time is smaller than the arrival time at the customer. Constraints (2.17) impose that the purchased units at a market cannot exceed its offer. Constraints (2.18) ensure that the vehicle load is large enough to satisfy the demand when a customer is visited. Finally, constraints (2.19)–(2.23) define the domain of the decision variables.

To solve the model, two solution approaches are proposed: an exact one based on the ϵ-constraint method, and a heuristic one based on relinked local search and VNS, the so-called Relinked Variable Neighborhood Search (RVNS).

2.6 The ϵ-constraint method

The ϵ-constraint is one of the most popular methods to find non-dominated solutions for bi-objective optimization problems. It was introduced by Haimes et al. (1971) and works by optimizing one objective function, while the other becomes a constraint whose upper bound systematically changes, thus obtaining points in the Pareto front.

Model (2.7)–(2.23) was reformulated to be tested under an ϵ-constraint scheme as follows:

$$\text{minimize } z_2 = \sum_{i \in C} v_i \tag{2.24}$$

subject to:

$$z_1 = \sum_{(i,j) \in A} e_{ij} x_{ij} + \sum_{p \in P} \sum_{i \in M_p} c_{pi} w_{pi} \leq \epsilon \tag{2.25}$$

$$(2.9) - (2.23).$$

It is worth mentioning that model (2.7)–(2.23) can also be reformulated as the minimization of z_1 considering z_2 as a constraint. Nonetheless, the given reformulation was chosen since it has a straightforward interpretation. Notice that different values of ϵ capture different levels on the available budget.

On the other hand, it is noteworthy that constraints (2.18) are non-linear. Then, in order to find non-dominated solutions by solving formulation (2.9)–(2.25) with an exact algorithm such as branch and bound, constraints (2.18) were linearized.

Let $\bar{w}_{pji}$ be an integer variable defined by (2.26). This variable can be interpreted as the quantity of product p that is purchased in market j before visiting customer i.

$$\bar{w}_{pji} = w_{pj} u_{ji}, \quad i \in C, p \in P_i, j \in M_p \tag{2.26}$$

Then, constraints (2.18) can be re-written as follows:

$$s_p + \sum_{j \in M_p} \bar{w}_{pji} - \sum_{j \in C \setminus \{i\}} d_{pj} u_{ji} \geq d_{pi} \quad i \in C, p \in P_i. \tag{2.27}$$

Notice that w_{pi} is bounded by q_{pi}. Then, it can be defined as

$$w_{pi} = \sum_{j=0}^{I_{pi}-1} 2^j \hat{w}_{pij} \quad p \in P, i \in M_p \tag{2.28}$$

where $\hat{w}_{pij}$ are binary variables and I_{pi} is an integer number such that $2^{I_{pi}-1} \leq q_{pi} \leq 2^{I_{pi}}$. From (2.26) and (2.28), we have that

$$\bar{w}_{pji} = \sum_{k=0}^{I_{pj}-1} 2^k \hat{w}_{pjk} u_{ji} \quad i \in C, p \in P_i, j \in M_p. \tag{2.29}$$

Note that (2.29) is nonlinear, then we define a binary variable $\underline{w}_{pjik}$ as follows:

$$\underline{w}_{pjik} = \hat{w}_{pjk} u_{ji} \quad i \in C, p \in P_i, j \in M_p, k : 0 \leq k < I_{pj}. \tag{2.30}$$

Thus, constraints (2.29) become

$$\bar{w}_{pji} = \sum_{k=0}^{I_{pj}-1} 2^k \underline{w}_{pjik} \quad i \in C, p \in P_i, j \in M_p \tag{2.31}$$

Finally, we add the following constraints:

$$u_{ji} + \hat{w}_{pjk} \leq 1 + \underline{w}_{pjik} \quad i \in C, p \in P_i, j \in M_p, k : 0 \leq k < I_{pj} \tag{2.32}$$

$$u_{ji} + \hat{w}_{pjk} \geq 2\underline{w}_{pjik} \quad i \in C, p \in P_i, j \in M_p, k : 0 \leq k < I_{pj}. \tag{2.33}$$

Thus, constraints (2.27), (2.28), (2.31), (2.32), and (2.33) replace constraints (2.18).

In order to find non-dominated solutions, the linear model was solved using CPLEX 12.6 and considering 10 different values of ϵ.

2.7 Relinked variable neighborhood search

In this section, it is introduced an algorithm based on relinked local search and VNS. This section describes the main features of these approaches and how they are combined to solve the 2-TPPD.

2.7.1 Relinked local search

Different heuristics have been proposed in the literature to solve multi-objective optimization problems. The most popular of them are genetic algorithms, such as the Nondominated Sorting Genetic Algorithm and its improved version (NSGA and NSGA-II, respectively) (Agarwal and Gupta, 2008; Basu, 2008; dos Santos Coelho and Alotto, 2008; Kanagarajan et al., 2008; Zahraie and Tavakolan, 2009; Yang and Natarajan, 2010; Basu, 2011; Cao et al., 2011; Panda, 2011; Wang et al., 2011; Chitra and Subbaraj, 2012; Basu, 2013; Bensmaine et al., 2013; Ghoddousi et al., 2013; Kalaivani et al., 2013; Panda and Yegireddy, 2013; Chen et al., 2014; Carlucci et al., 2015; Sheng et al., 2015), the Pareto Archived Evolution Strategy (PAES) (Nabeta et al., 2008; Alcalá et al., 2009; Montoya et al., 2010; Rostami and Neri, 2016), the Niched Pareto Genetic Algorithm (NPGA) (Dridi et al., 2008; Baraldi et al., 2009; Zhang et al., 2009), and the Strength Pareto Evolutionary Algorithm and its improved version (SPEA and SPEA2, respectively) (Wang et al., 2008; Dufo-López et al., 2011; Sheng et al., 2012). Nevertheless, it can be difficult to find an appropriate solution representation when many decisions must be taken.

Relinked local search is based on the initial phase of the Scatter Tabu Search Procedure for Non-Linear Multiobjective Optimization proposed by Molina et al. (2007). The relinked local search has the advantage of using single-objective local search algorithms to approximate the Pareto front; thus, the solutions do not have to represented in a particular manner.

The method consists of relinking $p+1$ local search algorithms, where p is the number of objectives. The relinking is carried out as follows: the local search algorithm dedicated to optimize the first objective is applied starting from an initial solution X_0. The resulting solution is called X_1. After that, the local search algorithm focused on optimize the second objective is applied using X_1 as initial solution, obtaining solution X_2, and so on. When solution X_p has been reached, a local search approach devoted to optimize the first objective function is applied starting from X_p, in order to complete a cycle around the Pareto set.

In the work of Molina et al. (2007), the authors used tabu search algorithms to carry out the relinked local search. In this thesis, the relinked local search is executed by relinking two different VNS schemes, one dedicated to minimize the total cost and another focused on minimizing the latency.

2.7.2 Variable neighborhood search

Variable neighborhood search (VNS) is a metaheuristic proposed by Mladenović and Hansen (1997) in which several neighborhoods are systematically explored seeking to both intensify and diversify the search. This framework has been successfully applied in recent years to solve Vehicle Routing Problems (VRPs) (see Paraskevopoulos et al. (2008); Fleszar et al. (2009); Hemmelmayr et al. (2009); Imran et al. (2009); Pirkwieser and Raidl (2009); Bruglieri et al. (2015); Polat et al. (2015)), VRPs with profits (see Labadie et al. (2012); Palomo-Martínez et al. (2017)), scheduling problems (see Gao et al. (2008); Adibi et al. (2010); Yazdani et al. (2010)), network design problems (see Eskandarpour et al. (2013, 2014)), and facility layout problems (see Abedzadeh et al. (2013); Hosseini et al. (2014)).

Even though there are several variants of VNS, this section describes the General Variable Neighborhood Search (GVNS) since this version of VNS is used within RVNS. Further information about other variants of VNS can be found in

Hansen and Mladenović (2014).

GVNS comprises two main schemes: the shaking phase and the Variable Neighborhood Descent (VND) phase. The former is devoted to help the algorithm to escape from local optima (diversification), while the latter seeks to descent to local optima (intensification).

VND consists of exploring several neighborhood structures within a local search scheme (see Algorithm 1). Let $f(x)$ be an objective function to be minimized. Given a solution x, and a set of neighborhood structures $N_1, N_2, \ldots, N_d$, VND searches the local optimal in N_i starting from x, as shown in line 3. Lines 4 to 9 show how the search moves from one neighborhood to another: if the solution obtained from the local search improves the incumbent one, then the incumbent solution is updated and the search returns to the first neighborhood; otherwise, the search moves to the next neighborhood. The algorithm stops when all neighborhood structures have been explored and the incumbent has not been updated.

It is noteworthy that step 3 can be computationally expensive so, usually, the neighborhoods are not fully explored to find the local optima but the search moves to the first improving solution. Also, a common practice to select the order in which the neighborhoods are applied is to rank them by the complexity of their application.

On the other hand, the shaking step is a simple operator that disturbs a given solution by returning a random neighbor of it. Algorithm 2 shows how the shaking step and VND are coupled into the GVNS scheme. As shown in line 3, GVNS makes use of several neighborhood structures in the shaking step. After the shaking has been carried out, a VND algorithm is applied as shown in line 4. The criterion to move from one neighborhood to another for the shaking step is similar to the one followed in VND, as shown in lines 5 to 10.

Algorithm 1 Variable neighborhood descent

Require:

x ▷ Initial solution

$N_1, N_2, \ldots, N_d$ ▷ Neighborhood structures

$i \leftarrow 1$
repeat
  $x^* \leftarrow \arg\min_{x' \in N_i(x)} \{f(x')\}$
  if $f(x^*) < f(x)$ **then**
    $x \leftarrow x^*$
    $i \leftarrow 1$
  else
    $i \leftarrow i + 1$
  end if
until $i = d$
return x

2.7.3 Relinked variable neighborhood search

The RVNS requires an initial set of feasible solutions $\mathscr{P}$. An iteration of the algorithm consists of selecting one solution $x \in \mathscr{P}$ and then applying a relinked local search that starts from x and in which the local search algorithms are GVNS schemes. Since, the 2-TPPD is a bi-objective optimization problem, two GVNSs are used to relink three searches. One GVNS is dedicated to minimize the cost and the other one is focused on minimizing the latency.

It is not straightforward to fix the order in which the objectives are minimized within the relinked local search. Thus, in each iteration of the GVNS, this order is randomly set.

Set $\mathscr{P}$ is updated every time that a local optimal is found within the GVNSs. If the local optimal is not dominated by any solution belonging to $\mathscr{P}$, then it is

Algorithm 2 General variable neighborhood search

Require:

x ▷ Initial solution

$N_1, N_2, \ldots, N_d$ ▷ Neighborhood structures for the VND

$\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_s$ ▷ Neighborhood structures for the shaking

```
i ← 1
repeat
    Choose x' ∈ 𝒩_i(x) at random
    Let x* be the solution obtained by applying VND starting from x' and using
    neighborhood structures N_1, N_2, ..., N_d
    if f(x*) < f(x) then
        x ← x*
        i ← 1
    else
        i ← i + 1
    end if
until i = s
```

included in $\mathscr{P}$ and the solutions dominated by the local optimal, if there are any, are removed from $\mathscr{P}$.

The RVNS stops iterating when it reaches 10 consecutive iterations without updating $\mathscr{P}$ or when an iteration ends and a limit computation time of 1800 s has been reached. Finally, dominated solutions are removed from $\mathscr{P}$, so this solution set becomes a Pareto set approximation.

The following subsections describe the construction method used to find the initial solutions, the criteria used to select the initial solution at each iteration of the RVNS, and the GVNSs methods used to minimize the cost and the latency.

2.7.3.1 Construction method

A solution is generated by creating a route starting and ending at the depot and containing all customers in a random order. After that, for each customer it is checked whether it is possible to satisfy its demand considering the nodes included in the route. If not, it is randomly selected a non-routed market that offers the products that are not possible to satisfy. The market is routed in a random position before the customer. The process stops when the demand of all customers can be satisfied. Once the route is constructed, the purchasing decisions are made in an optimal manner.

This procedure is replicated $|N| + |P|$ times in order to obtain the initial set of solutions.

2.7.3.2 Selection criteria to choose the initial solution

Three different criteria were explored to select the initial solution at each iteration of the RVNS. These criteria are described below:

1. Select the solution that has been part of $\mathscr{P}$ for the largest number of iterations.

2. Select the most disperse solution in the objectives space. The most disperse solution x is defined by Equation (2.34), where ED is the Euclidean distance.
$$x = \arg\max_{x' \in \mathscr{P}} \left\{ \min_{\bar{x} \in \mathscr{P} \setminus \{x'\}} \{ED(x', \bar{x})\} \right\}. \qquad (2.34)$$

3. Select a random solution from $\mathscr{P}$.

2.7.3.3 General variable neighborhood search to minimize the latency

In the GVNS devoted to minimize the latency, only one neighborhood structure is used for the shaking step: given a solution, a small percentage of the visited nodes is randomly chosen and then relocated at any random position. The shaken solution is kept if this permutation allows to satisfy the demand; otherwise, the perturbation is discarded.

For a better understanding of the local search operators used in the GVNS, the concept of *block* is here introduced:

DEFINITION 2.8 (BLOCK) *Given a route that starts at the depot, visits some markets and all customers, and ends at the depot, a block is defined as a sequence of two or more visited nodes that meet the following conditions:*

1. *the nodes are visited consecutively;*

2. *all nodes are either markets or customers; and*

3. *if the nodes are markets, the node visited before the first node belonging to the sequence and the node visited after the last node belonging to the sequence are not markets; otherwise, if the nodes are customers, the node visited before the first node belonging to the sequence and the node visited after the last node belonging to the sequence are not customers.*

For example, consider the route $d \rightarrow m_4 \rightarrow c_{10} \rightarrow m_1 \rightarrow m_6 \rightarrow c_6 \rightarrow c_4 \rightarrow c_2 \rightarrow m_8 \rightarrow c_9 \rightarrow c_8 \rightarrow c_7 \rightarrow m_{10} \rightarrow c_1 \rightarrow c_5 \rightarrow c_3 \rightarrow d$, where d is the depot, m_1, m_4, m_6, m_8, and m_{10} are markets, and $c_1, c_2, \ldots, c_{10}$ are customers. This route contains one block of markets: $m_1 \rightarrow m_6$; and three blocks of customers: $c_6 \rightarrow c_4 \rightarrow c_2$, $c_9 \rightarrow c_8 \rightarrow c_7$, and $c_1 \rightarrow c_5 \rightarrow c_3$.

The following local search operators are applied in order of appearance:

- *Intra-block relocate (IntraR):* For every block, each node belonging to it, is relocated at a random position of the same block.
- *Intra-block swap (IntraS):* For every block and for every node belonging to it, a different random node belonging to the same block is selected and their positions are exchanged.
- *Intra-block 2-opt (Intra2):* For every block and for every node belonging to it, another random node in the same block is selected to perform the classical 2-opt move (see Croes (1958)).
- *Inter-block relocate (InterR):* Every node in the current route is relocated. Customers are relocated at a random position of a posterior random block. Markets are relocated at a random position of a previous random block.
- *Market remove (MR):* A market is removed from the route when the demand can be satisfied by the remaining markets.

It is worth mentioning that the GVNS does not explore the entire neighborhood, instead the search moves to the first improving solution. Furthermore, every time that a move is performed, the purchasing decisions are made in an optimal manner.

2.7.3.4 General variable neighborhood search to minimize the cost

The GVNS dedicated to minimize the cost uses the same shaking operator used by the GVNS focused on minimizing the latency. Moreover, the same local search operators are applied, only MR slightly changes: a market is removed if the demand can be satisfied by the remaining markets and the decrease in the travel cost offsets the increase in the purchasing cost. This version of MR is based on the DROP procedure proposed by Voß (1996a).

The local search operators are applied in the same order and, at the end, the following operator is also carried out:

- *Market insert (MI):* A market that does not belong to the route is routed if the decrement in the purchasing cost offsets the increment in the travel cost. After that, the markets in which no products are purchased are removed, if there are any. This operator is based on the ADD and simplification procedures proposed by Voß (1996a) and Bontoux and Feillet (2008), respectively.

2.8 Computational experiments

This section is divided into three subsections. The first one is devoted to describe the test instances, the second one describes the experimental environment, and the third one relates to the computational tests results. In turn, the third subsection is divided into three groups of experiments. The first group is dedicated to compare the results obtained by the RVNS with the ones obtained by the ϵ-constraint method. The second group of experiments focuses on the evaluation of the selection criteria introduced in Section 2.7.3.2. Finally, the third group of experiments is focused on test how the characteristics of the tested instances affect the performance of the RVNS.

2.8.1 Instances

To the best of our knowledge, the 2-TPPD has not been studied before thus no existing benchmark instances are available. Therefore, test instances were generated to evaluate the efficiency of the proposed solution approaches. Capacitated instances are those in which if a market makes a product available for sale, it is able to fully satisfy its demand. Otherwise, the instances are called uncapacitated. Both capacitated and uncapacitated instances were generated.

It is noteworthy that even though the real-life situation from which the 2-TPPD arises considers carrying out activities at the customers's locations, the proposed model is flexible enough to consider cases in which only deliveries are performed at the customers's locations. Then, instances with high and low customers's service time were also generated. Customers with high service time are those in which activities must be performed, while customers with low service time are those in which only deliveries must be performed.

The instances are divided into five classes, namely, S, LCH, LCL, LUH, and LUL, according to their characteristics. Table 4.2 summarizes the characteristics of each instance class.

In all classes, the nodes are located in a $[0, 1000] \times [0, 1000]$ square and the travel cost between two nodes is set by the Euclidean distance between them, rounded to the nearest integer. It is assumed that one unit of currency is paid per unit of time, then $e_{ij} = t_{ij}$ for all $(i, j) \in A$. The number of customers was randomly set to $0.1(|N| - 2), 0.5(|N| - 2)$ or $0.9(|N| - 2)$, rounded to the nearest integer. For each instance, a random subset of products has an initial stock equals to zero, the remaining ones have a stock randomly set between one and five units.

In the capacitated instances, if a market offers a product p, the offer is set to a quantity between one and 15 units, and the total demand of such product (sum of the demanded quantities by all customers) was randomly set to $0.1 \times \sum_{i \in M_p} q_{pi}$,

Table 2.2: Characteristics of the instance classes

Class	**Size**	$\|\boldsymbol{N}\|$	$\|\boldsymbol{P}\|$	**Markets capacity**	**Customers service time**
S	12	10	10, 15, 20	Capacitated, uncapacitated	High, low
LCH	16	50,100,150,200	50,100,150,200	Capacitated	High
LCL	16	50,100,150,200	50,100,150,200	Capacitated	Low
LUH	16	50,100,150,200	50,100,150,200	Uncapacitated	High
LUL	16	50,100,150,200	50,100,150,200	Uncapacitated	Low

$0.5 \times \sum_{i \in M_p} q_{pi}$, or $0.9 \times \sum_{i \in M_p} q_{pi}$. On the uncapacitated instances, if a market offers a product, the number of units available for sale is set to the total demand. In both cases, the prices of the products go from 1 to 500 units.

For each instance, let $\bar{t}$ be the average travel time between two nodes. Low customer service times were set at random between 0 and $\bar{t}$, while high customer services times were set at random between $\bar{t}$ and $2 \times \bar{t}$.

The name of the instances follows the format $|N|\_|M|\_|C|\_|P|\_tc\_tr$. Parameter tc, indicates whether the instance is capacitated (c) or uncapacitated (nc). Parameter tr, indicates whether the customers service time is high, i.e., repairs are required (r); or the customers service time is low, i.e., no repairs are required (nr). The capacitated instances have an extra parameter d at the end of the name ($|N|\_|M|\_|C|\_|P|\_tc\_tr\_d$). Parameter d indicates the demand level: l if the total demand equals $0.1 \sum_{i \in M_p} q_{pi}$, m if the total demand equals $0.5 \sum_{i \in M_p} q_{pi}$, or h if the total demand equals $0.9 \sum_{i \in M_p} q_{pi}$.

2.8.2 Experimental environment

Three different variants of RVNS were tested according to the criteria described in Section 2.7.3.2. Version RVNS-LNI uses criterion 1, version RVNS-MD uses criterion 2, and RVNS-R uses criterion 3.

The ϵ-constraint (hereafter, EC) and all RVNS versions were coded in C++ and compiled in GNU on a 2.1 GHz Intel Xeon(R) CPU E5-2620 v2 under Ubuntu 16.04 operating system. The mixed integer linear programming models associated with each value of ϵ in the ϵ-constraint method were solved through CPLEX 12.6.

2.8.3 Experimental results

This section describes and analyzes the experimental results through three groups of experiments: one devoted to compare RVNS with ϵ-constraint, another one devoted to compare the criteria introduced in Section 2.7.3.2, and another one devoted to state the effect of the characteristics of the instances in the RVNS performance.

2.8.3.1 Comparison between ϵ-constraint and RVNS

The mixed integer linear programming model (2.9)–(2.17), (2.19)–(2.25), (2.27), (2.28), and (2.31)–(2.33) was solved with CPLEX 12.6 considering 10 different values of ϵ per instance. The limit computation time to solve each single-objective model was set to 7200 CPU seconds, using 10 threads. Within this computation time, CPLEX was not able to solve instances containing more than 10 nodes; then, the experiments reported in this section were carried out over the instances belonging to class S.

The ϵ-constraint method and the three versions of RVNS were executed once per instance. It is worth noticing that the ϵ-constraint may return weakly Pareto optimal solutions. Thus, seeking to make fair comparisons, the weakly Pareto optimal solutions are removed after the ϵ-constraint execution finishes.

The Pareto front approximations obtained in this group of experiments were evaluated using the overall nondominated vector generation, the hypervolume, and the two set coverage metrics, as well as the computation time. As in Knowles et al. (2006), the objectives space was normalized to avoid results distortion when the hypervolume is calculated. Equation (2.35) was used for this purpose.

$$z_i'(x) = \frac{z_i(x) - z_i^{min}}{z_i^{max} - z_i^{min}}, \tag{2.35}$$

where z_i^{min} and z_i^{max} are the minimum and the maximum values of the objective function z_i $(i = 1, 2)$ obtained from all the experiments performed in this group, respectively. Besides, the chosen reference point is $(1, 1)$.

Table 2.3 displays the overall nondominated vector generation, the hypervolume, and execution time in seconds per instance.

With respect to the computation time, it is quite evident that EC requires by far more time than any version of RVNS to approximate the Pareto front. EC needed 1762.2 s (29.37 min), in average, to find a Pareto front approximation. While the slowest version of the heuristic, RVNS-R, needed only 0.05 s, in average. Besides, to find Pareto front approximations for all the instances, EC required 21,146.37 s (5.87 hrs, approximately). On the other hand, RVNS-R required only 0.43 s.

In order to establish whether there are significant differences in the performance of the algorithms when the overall nondominated vector generation and the hypervolume are considered, Quade tests were performed. The null and alternative hypothesis are stated as follows:

H_0: All of the algorithms effects are identical,

H_1: At least one of the algorithms effects is different than the others.

Table 2.3: Overall nondominated vector generation, hypervolume, and execution time in seconds for instances of Class S

Instance	**ONVG**				**Hv**				**Execution time (s)**			
	EC	**RVNS-LNI**	**RVNS-MD**	**RVNS-R**	**EC**	**RVNS-LNI**	**RVNS-MD**	**RVNS-R**	**EC**	**RVNS-LNI**	**RVNS-MD**	**RVNS-R**
10_1_8_10_c_r_m	5	3	2	3	0.002	0.002	0.002	0.002	1878.16	0.03	0.02	0.02
10_4_5_20_c_r_l	3	4	2	4	0.409	0.409	0.386	0.409	4490.82	0.03	0.01	0.05
10_8_1_15_c_r_l	3	3	3	3	1.000	0.956	0.956	0.956	405.65	0.06	0.03	0.06
10_1_8_10_c_nr_m	6	2	2	6	0.100	0.098	0.098	0.035	1855.58	0.02	0.02	0.05
10_4_5_20_c_nr_l	4	3	3	3	0.583	0.548	0.582	0.583	2341.99	0.01	0.02	0.02
10_8_1_15_c_nr_l	3	3	3	3	0.956	0.956	0.956	0.956	367.61	0.05	0.03	0.05
10_1_8_10_nc_r	3	3	2	3	0.002	0.002	0.002	0.002	1674.91	0.03	0.01	0.03
10_4_5_20_nc_r	3	4	2	4	0.409	0.409	0.386	0.409	3887.42	0.02	0.01	0.05
10_8_1_15_nc_r	3	3	3	3	0.956	0.956	0.956	0.956	284.21	0.05	0.05	0.05
10_1_8_10_nc_nr	5	7	2	6	0.100	0.051	0.098	0.035	1778.44	0.03	0.02	0.03
10_4_5_20_nc_nr	4	2	3	3	0.583	0.550	0.582	0.583	1849.98	0.02	0.02	0.02
10_8_1_15_nc_nr	3	3	3	3	0.956	0.956	0.956	0.956	331.60	0.02	0.03	0.05
Average	3.75	3.33	2.50	3.67	0.504	0.490	0.497	0.490	1762.20	0.03	0.02	0.05

Table 2.4: Adjusted p-values to evaluate differences among the overall nondominated vector generation values reported by the algorithms over class S

	EC	RVNS-LNI	RVNS-MD
RVNS-LNI	1.0000	-	-
RVNS-MD	**0.0300**	0.0940	-
RVNS-R	1.0000	1.0000	**0.0300**

Considering the overall nondominated vector generation, the null hypothesis is rejected with a p-value of 0.01519. Then, at least one of the algorithms returns Pareto front approximations with different overall nondominated vector generation value.

In order to state which algorithm has a different performance, a post-hoc test using a Holm adjustment of the p-values was carried out. The adjusted p-values are displayed in Table 2.4. The values in bold are those that allow to state significant differences between the algorithms corresponding to the respective column and row. From Tables 2.3 and 2.4, we can conclude that RVNS-MD finds Pareto front approximation with a smaller overall nondominated vector generation value than EC and RVNS-R. Nonetheless, there are not significant differences between EC and RVNS-LNI, and EC and RVNS-R.

Finally, considering the hypervolume, the null hypothesis cannot be rejected with a p-value of 0.1392. Then, all algorithms return Pareto front approximation with similar hypervolume.

Considering that all versions of the heuristic require by far less computation time than EC, RVNS-LNI and RVNS-R find Pareto front approximations with similar overall nondominated vector generation value than EC, and all algorithms return Pareto front approximations with similar hypervolume, we can conclude that RVNS-LNI and RVNS-R are efficient to approximate the Pareto front of the instances of class S compared with the ϵ-constraint method.

2.8.3.2 Comparison among selection criteria to choose the initial solution at each iteration of RVNS

Each version of RVNS was executed once for each instance belonging to classes LCH, LCL, LUH, and LUL. The overall nondominated vector generation, the k-distance (k=2), and the hypervolume were calculated for each Pareto front approximation reported by the algorithms. Also, for each combination of two Pareto front approximations of the same instance, the two set coverage was calculated. In addition, the computation time required to approximate the Pareto fronts was stored.

As in the previous group of experiments, the objectives space was normalized using Equation (2.35), where z_i^{min} and z_i^{max} are the minimum and the maximum values of the objective function z_i $(i = 1, 2)$ obtained from all the experiments performed for this section, respectively.

Tables 2.5, 2.7, 2.9, and 2.11 display the overall nondominated vector generation, the k-distance, the hypervolume, and execution time in seconds for each instance belonging to Classes LCH, LCL, LUH, and LUL, respectively. On the other hand, Tables 2.6, 2.8, 2.10, and 2.12 display the two set coverage for each possible combination of two algorithms evaluated over each instance belonging to Classes LCH, LCL, LUH, and LUL, respectively.

Seeking to know whether there are significant differences in the performance of the algorithms when the overall nondominated vector generation, the k-distance, the hypervolume, and the execution time are considered, some Quade tests were performed for each class and for each performance metric.

Besides, Wilcoxon signed ranks tests were performed for each combination of two algorithms and for each class to state the existence of differences in the performance of the algorithms considering the two set coverage. The null and alternative hypotheses are stated as follows:

Table 2.5: Overall nondominated vector generation, k-distance, hypervolume, and execution time for instances of Class LCH

Instance	**ONVG**			**kD**			**Hv**			**Execution time (s)**		
	RVNS-LNI	**RVNS-MD**	**RVNS-R**	**RVNS-LNI**	**RVNS-MD**	**RVNS-R**	**RVNS-LNI**	**RVNS-MD**	**RVNS-R**	**RVNS-LNI**	**RVNS-MD**	**RVNS-R**
100_10_89_200_c_r_h	14	12	29	0.0009	0.0009	0.0003	0.752	0.754	0.752	59.17	56.20	189.18
100_89_10_100_c_r_h	26	45	24	0.0003	0.0002	0.0004	0.676	0.676	0.676	1724.57	1805.84	1805.40
100_89_10_150_c_r_m	48	52	45	0.0008	0.0008	0.0011	0.803	0.803	0.803	1804.21	1817.54	1820.31
100_89_10_50_c_r_m	68	63	50	0.0004	0.0004	0.0004	0.937	0.937	0.937	1804.83	1803.75	1805.09
150_134_15_100_c_r_l	111	92	96	0.0004	0.0004	0.0004	0.972	0.972	0.972	1800.46	1800.81	1802.01
150_134_15_150_c_r_l	90	91	85	0.0005	0.0005	0.0005	0.959	0.959	0.960	1805.82	1809.67	1810.51
150_134_15_50_c_r_l	70	81	85	0.0004	0.0004	0.0003	0.989	0.989	0.989	818.74	890.68	1802.16
150_15_134_200_c_r_m	25	34	36	0.0011	0.0011	0.0009	0.516	0.514	0.522	1024.21	399.34	1802.79
200_179_20_100_c_r_m	50	36	37	0.0012	0.0011	0.0019	0.713	0.712	0.711	1870.53	1839.89	1808.26
200_179_20_150_c_r_h	28	14	16	0.0011	0.0033	0.0017	0.076	0.076	0.076	1834.95	1819.48	1919.25
200_179_20_50_c_r_m	46	56	36	0.0017	0.0013	0.0019	0.834	0.833	0.832	1818.42	1815.93	1820.23
200_20_179_200_c_r_m	37	30	32	0.0019	0.0019	0.0027	0.113	0.119	0.121	1810.54	1841.27	1825.11
50_24_25_100_c_r_h	21	45	45	0.0002	0.0001	0.0001	0.921	0.921	0.922	59.27	98.25	404.56
50_44_5_150_c_r_h	7	4	7	0.0008	0.0012	0.0002	0.775	0.774	0.775	83.38	37.28	206.22
50_44_5_200_c_r_l	73	73	79	0.0002	0.0001	0.0001	0.999	0.999	0.999	190.34	165.51	223.02
50_44_5_50_c_r_m	37	33	56	0.0003	0.0003	0.0002	0.979	0.979	0.979	110.78	143.68	463.23
Average	46.94	47.56	47.38	0.0008	0.0009	0.0008	0.751	0.751	0.752	1163.76	1134.07	1344.21

Table 2.6: Two set coverage for instances of Class LCH

Instance	**RVNS-LNI vs RVNS-MD**		**RVNS-LNI vs RVNS-R**		**RVNS-MD vs RVNS-R**	
	C(RVNS-LNI, RVNS-MD)	**C(RVNS-MD, RVNS-LNI)**	**C(RVNS-LNI, RVNS-R)**	**C(RVNS-R, RVNS-LNI)**	**C(RVNS-MD, RVNS-R)**	**C(RVNS-R, RVNS-MD)**
100_10_89_200_c_r_h	0.00	0.71	0.00	0.79	0.34	0.67
100_89_10_100_c_r_h	0.31	0.42	0.46	0.54	0.50	0.38
100_89_10_150_c_r_m	0.35	0.58	0.58	0.31	0.71	0.31
100_89_10_50_c_r_m	0.37	0.65	0.22	0.56	0.32	0.52
150_134_15_100_c_r_l	0.26	0.74	0.17	0.70	0.39	0.48
150_134_15_150_c_r_l	0.52	0.47	0.59	0.39	0.61	0.41
150_134_15_50_c_r_l	0.53	0.33	0.54	0.46	0.44	0.59
150_15_134_200_c_r_m	0.53	0.00	0.00	0.80	0.00	0.94
200_179_20_100_c_r_m	0.56	0.16	0.54	0.36	0.49	0.33
200_179_20_150_c_r_h	0.50	0.36	0.13	0.46	0.38	0.43
200_179_20_50_c_r_m	0.48	0.41	0.39	0.43	0.22	0.61
200_20_179_200_c_r_m	0.07	0.38	0.19	0.32	0.50	0.17
50_24_25_100_c_r_h	0.09	0.81	0.00	1.00	0.00	1.00
50_44_5_150_c_r_h	0.50	0.14	0.14	0.43	0.00	1.00
50_44_5_200_c_r_l	0.53	0.29	0.75	0.11	0.66	0.25
50_44_5_50_c_r_m	0.06	0.78	0.00	1.00	0.05	0.85
Average	0.35	0.45	0.29	0.54	0.35	0.56

Table 2.7: Overall nondominated vector generation, k-distance, hypervolume, and execution time for instances of Class LCL

Instance	**ONVG**			**kD**			**Hv**			**Execution time (s)**		
	RVNS-LNI	**RVNS-MD**	**RVNS-R**	**RVNS-LNI**	**RVNS-MD**	**RVNS-R**	**RVNS-LNI**	**RVNS-MD**	**RVNS-R**	**RVNS-LNI**	**RVNS-MD**	**RVNS-R**
100_10_89_200_c_nr_h	15	13	24	0.0012	0.0009	0.0002	0.883	0.879	0.882	68.89	44.88	357.94
100_89_10_100_c_nr_h	28	18	17	0.0003	0.0007	0.0004	0.677	0.677	0.677	1196.35	1283.71	1801.02
100_89_10_150_c_nr_m	50	65	43	0.0007	0.0006	0.0009	0.804	0.804	0.804	1806.37	1807.04	1820.01
100_89_10_50_c_nr_m	55	58	58	0.0004	0.0005	0.0005	0.938	0.939	0.938	1805.00	1803.52	1801.70
150_134_15_100_c_nr_l	107	94	89	0.0004	0.0004	0.0005	0.976	0.976	0.976	1802.69	1804.35	1800.02
150_134_15_150_c_nr_l	86	98	82	0.0005	0.0005	0.0006	0.963	0.963	0.963	1809.91	1810.52	1804.25
150_134_15_50_c_nr_l	89	75	79	0.0004	0.0004	0.0004	0.992	0.991	0.991	1414.84	885.51	1802.56
150_15_134_200_c_nr_m	50	34	56	0.0006	0.0013	0.0006	0.830	0.826	0.831	1472.27	972.96	1802.65
200_179_20_100_c_nr_m	32	40	43	0.0015	0.0017	0.0014	0.717	0.716	0.717	1825.92	1800.87	1965.83
200_179_20_150_c_nr_h	20	30	19	0.0018	0.0010	0.0011	0.077	0.077	0.077	1840.37	1879.75	1997.17
200_179_20_50_c_nr_m	61	35	44	0.0014	0.0022	0.0015	0.839	0.839	0.839	1816.31	1868.94	1865.68
200_20_179_200_c_nr_m	45	36	47	0.0023	0.0016	0.0017	0.653	0.661	0.671	1812.73	1804.43	1814.32
50_24_25_100_c_nr_h	33	26	48	0.0002	0.0001	0.0001	0.930	0.929	0.930	158.29	53.08	420.77
50_44_5_150_c_nr_h	7	5	11	0.0009	0.0013	0.0003	0.775	0.775	0.775	77.28	50.04	223.16
50_44_5_200_c_nr_l	65	64	75	0.0001	0.0002	0.0001	0.999	0.999	0.999	188.14	257.10	290.46
50_44_5_50_c_nr_m	36	31	49	0.0002	0.0003	0.0002	0.980	0.980	0.980	145.55	128.55	494.11
Average	48.69	45.13	49.00	0.0008	0.0009	0.0007	0.815	0.814	0.816	1202.56	1140.95	1378.85

Table 2.8: Two set coverage for instances of Class LCL

Instance	**RVNS-LNI vs RVNS-MD**		**RVNS-LNI vs RVNS-R**		**RVNS-MD vs RVNS-R**	
	C(RVNS-LNI, RVNS-MD)	**C(RVNS-MD, RVNS-LNI)**	**C(RVNS-LNI, RVNS-R)**	**C(RVNS-R, RVNS-LNI)**	**C(RVNS-MD, RVNS-R)**	**C(RVNS-R, RVNS-MD)**
100_10_89_200_c_nr_h	1.00	0.00	0.42	0.07	0.00	1.00
100_89_10_100_c_nr_h	0.00	0.93	0.00	0.82	0.24	0.28
100_89_10_150_c_nr_m	0.34	0.52	0.23	0.68	0.37	0.58
100_89_10_50_c_nr_m	0.26	0.65	0.62	0.22	0.74	0.21
150_134_15_100_c_nr_l	0.22	0.74	0.31	0.74	0.62	0.38
150_134_15_150_c_nr_l	0.28	0.70	0.50	0.49	0.68	0.20
150_134_15_50_c_nr_l	0.40	0.49	0.37	0.67	0.42	0.56
150_15_134_200_c_nr_m	0.56	0.34	0.57	0.40	0.59	0.50
200_179_20_100_c_nr_m	0.60	0.22	0.35	0.47	0.26	0.60
200_179_20_150_c_nr_h	0.083	0.75	0.26	0.60	0.47	0.20
200_179_20_50_c_nr_m	0.37	0.48	0.55	0.31	0.55	0.43
200_20_179_200_c_nr_m	0.00	0.91	0.00	0.93	0.57	0.084
50_24_25_100_c_nr_h	0.92	0.09	0.00	0.97	0.00	1.00
50_44_5_150_c_nr_h	0.00	0.43	0.09	0.71	0.27	0.40
50_44_5_200_c_nr_l	0.20	0.71	0.28	0.66	0.67	0.25
50_44_5_50_c_nr_m	0.58	0.42	0.084	0.81	0.04	0.87
Average	0.37	0.52	0.29	0.60	0.41	0.48

Table 2.9: Overall nondominated vector generation, k-distance, hypervolume, and execution time for instances of Class LUH

Instance	**ONVG**			**kD**			**Hv**			**Execution time (s)**		
	RVNS-LNI	**RVNS-MD**	**RVNS-R**	**RVNS-LNI**	**RVNS-MD**	**RVNS-R**	**RVNS-LNI**	**RVNS-MD**	**RVNS-R**	**RVNS-LNI**	**RVNS-MD**	**RVNS-R**
100_10_89_200_nc_r	31	29	39	0.0012	0.0019	0.0009	0.774	0.775	0.776	214.68	178.33	801.69
100_89_10_100_nc_r	89	94	88	0.0014	0.0007	0.0016	0.816	0.816	0.816	334.57	471.07	830.84
100_89_10_150_nc_r	92	86	118	0.0010	0.0011	0.0006	0.862	0.862	0.862	632.42	492.84	698.21
100_89_10_50_nc_r	96	89	92	0.0004	0.0004	0.0004	0.962	0.962	0.962	139.25	92.48	157.92
150_134_15_100_nc_r	87	107	99	0.0003	0.0004	0.0003	0.977	0.977	0.977	568.41	565.81	827.96
150_134_15_150_nc_r	103	87	121	0.0004	0.0004	0.0003	0.965	0.965	0.965	1207.84	810.23	1801.42
150_134_15_50_nc_r	57	56	61	0.0003	0.0003	0.0003	0.992	0.992	0.992	138.41	151.29	376.92
150_15_134_200_nc_r	26	28	45	0.0024	0.0022	0.0021	0.538	0.536	0.541	619.03	340.41	1804.46
200_179_20_100_nc_r	100	125	98	0.0012	0.0009	0.0015	0.811	0.811	0.811	1800.56	1760.51	1806.12
200_179_20_150_nc_r	96	89	87	0.0032	0.0041	0.0030	0.471	0.470	0.470	1806.94	1817.37	1816.66
200_179_20_50_nc_r	76	85	72	0.0013	0.0012	0.0012	0.901	0.901	0.901	646.95	508.53	402.10
200_20_179_200_nc_r	47	52	44	0.0029	0.0024	0.0051	0.0856	0.0867	0.0861	1806.71	1803.69	1804.39
50_24_25_100_nc_r	69	61	83	0.0003	0.0004	0.0003	0.955	0.955	0.955	131.37	120.66	260.32
50_44_5_150_nc_r	89	89	119	0.0012	0.0017	0.0010	0.869	0.869	0.869	173.14	174.38	310.46
50_44_5_200_nc_r	55	58	59	0.0002	0.0002	0.0001	0.999	0.999	0.999	66.58	129.30	153.63
50_44_5_50_nc_r	77	52	73	0.0003	0.0003	0.0003	0.990	0.990	0.990	38.88	17.18	57.87
Average	74.38	74.19	81.13	0.0011	0.0012	0.0012	0.815	0.815	0.816	645.36	589.63	869.44

Table 2.10: Two set coverage for instances of Class LUH

Instance	RVNS-LNI vs RVNS-MD		RVNS-LNI vs RVNS-R		RVNS-MD vs RVNS-R	
	C(RVNS-LNI, RVNS-MD)	C(RVNS-MD, RVNS-LNI)	C(RVNS-LNI, RVNS-R)	C(RVNS-R, RVNS-LNI)	C(RVNS-MD, RVNS-R)	C(RVNS-R, RVNS-MD)
100_10_89_200_nc_r	0.48	0.083	0.05	0.61	0.05	0.76
100_89_10_100_nc_r	0.36	0.52	0.24	0.72	0.28	0.56
100_89_10_150_nc_r	0.37	0.57	0.66	0.27	0.71	0.24
100_89_10_50_nc_r	0.63	0.34	0.58	0.46	0.32	0.57
150_134_15_100_nc_r	0.50	0.37	0.60	0.28	0.52	0.36
150_134_15_150_nc_r	0.34	0.67	0.45	0.55	0.65	0.29
150_134_15_50_nc_r	0.46	0.32	0.083	0.72	0.086	0.71
150_15_134_200_nc_r	0.43	0.42	0.088	0.62	0.086	0.64
200_179_20_100_nc_r	0.52	0.45	0.51	0.35	0.49	0.34
200_179_20_150_nc_r	0.53	0.42	0.33	0.55	0.24	0.64
200_179_20_50_nc_r	0.66	0.30	0.72	0.25	0.51	0.47
200_20_179_200_nc_r	0.35	0.43	0.66	0.081	0.80	0.080
50_24_25_100_nc_r	0.085	0.65	0.28	0.74	0.47	0.49
50_44_5_150_nc_r	0.088	0.60	0.76	0.29	0.79	0.22
50_44_5_200_nc_r	0.24	0.76	0.05	0.93	0.34	0.60
50_44_5_50_nc_r	0.48	0.35	0.04	0.97	0.08	0.92
Average	0.42	0.46	0.39	0.53	0.41	0.50

Table 2.11: Overall nondominated vector generation, k-distance, hypervolume, and execution time for instances of Class LUL

Instance	**ONVG**			**kD**			**Hv**			**Execution time (s)**		
	RVNS-LNI	**RVNS-MD**	**RVNS-R**	**RVNS-LNI**	**RVNS-MD**	**RVNS-R**	**RVNS-LNI**	**RVNS-MD**	**RVNS-R**	**RVNS-LNI**	**RVNS-MD**	**RVNS-R**
100_10_89_200_nc_r	31	29	39	0.0012	0.0019	0.0009	0.774	0.775	0.776	214.68	178.33	801.69
100_89_10_100_nc_r	89	94	88	0.0014	0.0007	0.0016	0.816	0.816	0.816	334.57	471.07	830.84
100_89_10_150_nc_r	92	86	118	0.0010	0.0011	0.0006	0.862	0.862	0.862	632.42	492.84	698.21
100_89_10_50_nc_r	96	89	92	0.0004	0.0004	0.0004	0.962	0.962	0.962	139.25	92.48	157.92
150_134_15_100_nc_r	87	107	99	0.0003	0.0004	0.0003	0.977	0.977	0.977	568.41	565.81	827.96
150_134_15_150_nc_r	103	87	121	0.0004	0.0004	0.0003	0.965	0.965	0.965	1207.84	810.23	1801.42
150_134_15_50_nc_r	57	56	61	0.0003	0.0003	0.0003	0.992	0.992	0.992	138.41	151.29	376.92
150_15_134_200_nc_r	26	28	45	0.0024	0.0022	0.0021	0.538	0.536	0.541	619.03	340.41	1804.46
200_179_20_100_nc_r	100	125	98	0.0012	0.0009	0.0015	0.811	0.811	0.811	1800.56	1760.51	1806.12
200_179_20_150_nc_r	96	89	87	0.0032	0.0041	0.0030	0.471	0.470	0.470	1806.94	1817.37	1816.66
200_179_20_50_nc_r	76	85	72	0.0013	0.0012	0.0012	0.901	0.901	0.901	646.95	508.53	402.10
200_20_179_200_nc_r	47	52	44	0.0029	0.0024	0.0051	0.0856	0.0867	0.0861	1806.71	1803.69	1804.39
50_24_25_100_nc_r	69	61	83	0.0003	0.0004	0.0003	0.955	0.955	0.955	131.37	120.66	260.32
50_44_5_150_nc_r	89	89	119	0.0012	0.0017	0.0010	0.869	0.869	0.869	173.14	174.38	310.46
50_44_5_200_nc_r	55	58	59	0.0002	0.0002	0.0001	0.999	0.999	0.999	66.58	129.30	153.63
50_44_5_50_nc_r	77	52	73	0.0003	0.0003	0.0003	0.990	0.990	0.990	38.88	17.18	57.87
Average	74.38	74.19	81.13	0.0011	0.0012	0.0012	0.815	0.815	0.816	645.36	589.63	869.44

Table 2.12: Two set coverage for instances of Class LUL

Instance	**RVNS-LNI vs RVNS-MD**		**RVNS-LNI vs RVNS-R**		**RVNS-MD vs RVNS-R**	
	C(RVNS-LNI, RVNS-MD)	**C(RVNS-MD, RVNS-LNI)**	**C(RVNS-LNI, RVNS-R)**	**C(RVNS-R, RVNS-LNI)**	**C(RVNS-MD, RVNS-R)**	**C(RVNS-R, RVNS-MD)**
100_10_89_200_nc_nr	0.97	0.00	0.63	0.27	0.00	1.00
100_89_10_100_nc_nr	0.36	0.47	0.68	0.22	0.74	0.20
100_89_10_150_nc_nr	0.32	0.51	0.086	0.78	0.20	0.76
100_89_10_50_nc_nr	0.60	0.33	0.53	0.33	0.32	0.56
150_134_15_100_nc_nr	0.29	0.50	0.32	0.58	0.47	0.39
150_134_15_150_nc_nr	0.44	0.46	0.40	0.63	0.40	0.57
150_134_15_50_nc_nr	0.54	0.40	0.05	0.89	0.080	0.80
150_15_134_200_nc_nr	0.61	0.086	0.23	0.52	0.27	0.81
200_179_20_100_nc_nr	0.61	0.29	0.50	0.41	0.28	0.61
200_179_20_150_nc_nr	0.37	0.70	0.22	0.69	0.41	0.52
200_179_20_50_nc_nr	0.49	0.44	0.72	0.21	0.61	0.42
200_20_179_200_nc_nr	0.00	0.94	0.084	0.28	0.67	0.08
50_24_25_100_nc_nr	0.47	0.40	0.00	0.77	0.05	0.83
50_44_5_150_nc_nr	0.81	0.23	0.43	0.64	0.09	0.95
50_44_5_200_nc_nr	0.41	0.53	0.43	0.45	0.55	0.36
50_44_5_50_nc_nr	0.53	0.43	0.40	0.66	0.20	0.88
Average	0.49	0.43	0.37	0.52	0.33	0.61

Table 2.13: p-values obtained from the Quade tests and the Wilcoxon signed ranks tests executed to state differences among RVNS versions

Class	**Quade tests**				**Wilcoxon signed ranks tests**		
	ONVG	**kD**	**H**	**Execution time (s)**	**Two set coverage**		
					RVNS-LNI vs RVNS-MD	**RVNS-LNI vs RVNS-R**	**RVNS-MD vs RVNS-R**
LCH	0.8672	0.9886	0.6444	**0.0016**	0.464	0.067	0.258
LCL	0.2326	0.4610	0.2165	**0.0001**	0.211	**0.021**	0.928
LUH	0.3856	0.4309	0.6130	**0.0003**	0.696	0.298	0.495
LUL	**0.0470**	0.5333	0.2968	**0.0003**	0.562	0.159	**0.046**

H_0: C(A,B) and C(B,A) belong to identical populations,

H_1: C(A,B) and C(B,A) do not belong to identical populations,

where A and B are distinct RVNS versions.

The p-values obtained from the Quade tests and the Wilcoxon signed ranks tests are displayed in Table 2.13. The values in bold are those that allow to reject the respective null hypothesis.

With respect to the overall nondominated vector generation, the k-distance, and the hypervolume, the Quade tests point to the conclusion that the null hypotheses cannot be rejected so all algorithms perform similar for all instance classes.

On the other hand, taking into account the execution time, the p-values lead us to conclude that the null hypotheses are rejected and so at least one of the algorithms requires a different execution time to approximate the Pareto front.

With the purpose of determine which algorithm performs different for each

Table 2.14: Adjusted p-values to evaluate differences among the overall nondominated vector generation reported by the algorithms for instances of Class LUL

	RVNS-LNI	**RVNS-MD**
RVNS-MD	0.46	-
RVNS-R	**0.05**	0.17

instance class, post-hoc tests with a Holm adjustment were carried out. The adjusted p-values are shown in Table 2.15. Values in bold are those that allow to state significant differences between the algorithms corresponding to the respective column and row. As shown in Table 2.15, significant differences were found when the performance of the algorithms is evaluated with the overall nondominated vector generation in instances belonging to Class LUL and also, at least one of the algorithms, requires different computation time to approximate the Pareto front for all instance classes.

Seeking to know which algorithms perform different, post-hoc tests using a Holm adjustment were carried out. The adjusted p-values obtained to assess differences in the overall nondominated vector generation for instances of Class LUL are displayed in Table 2.14 and the ones obtained to evaluate differences in the execution time are shown in Table 2.15. The values in bold are those that allow to reject the null hypothesis that states that the algorithms corresponding to the respective row and column do not have a significant difference in their performance. From Tables 2.11 and 2.14 we can conclude that RVNS-R reports Pareto front approximations with a larger overall nondominated vector generation value for instances of Class LUL than the ones reported by RVNS-LNI. On the other hand, from Tables 2.5, 2.7, 2.9, 2.11, and 2.15, we can conclude that RVNS-R needs more computation time than the other versions of RVNS to find Pareto front approximations for all instance classes.

Finally, the Wilcoxon signed ranks tests lead us to conclude that none of the al-

Table 2.15: Adjusted p-values to evaluate differences among the execution time reported by the algorithms for each instance class

	Class LCH		**Class LCL**		**Class LUH**		**Class LUL**	
	RVNS-LNI	**RVNS-MD**	**RVNS-LNI**	**RVNS-MD**	**RVNS-LNI**	**RVNS-MD**	**RVNS-LNI**	**RVNS-MD**
RVNS-MD	0.7446	-	0.3460	-	0.08796	-	0.5043	-
RVNS-R	**0.0032**	**0.0052**	**0.0015**	**0.0002**	**0.0066**	**0.0002**	**0.0018**	**0.0004**

gorithms reports Pareto front approximations that dominate significantly the Pareto front approximations reported by the others in Classes LCH and LUL, considering the two set coverage.

With respect to Class LCL, significant differences were found when RVNS-LNI is compared with RVNS-R. This result, along with the information displayed in Table 2.8, suggests that C(RVNS-R,RVNS-LNI) is larger than C(RVNS-LNI,RVNS-R) and therefore, the number of points in the Pareto front approximation reported by RVNS-LNI covered by at least one point in the Pareto front approximation reported by RVNS-R is larger than the number of points in the Pareto front approximation reported by RVNS-R covered by at least one point in the Pareto front approximation reported by RVNS-LNI.

Finally, with respect to Class LUL, significant differences between C(RVNS-MD,RVNS-R) and C(RVNS-R,RVNS-MD) were found when RVNS-MD is compared with RVNS-R. This result, along with the information displayed in Table 2.12, suggests that C(RVNS-R,RVNS-MD) is larger than C(RVNS-MD,RVNS-R) and then the number of points in the Pareto front approximation reported by RVNS-MD covered by at least one point in the Pareto front approximation reported by RVNS-R is larger than the number of points in the Pareto front approximation reported by RVNS-R covered by at least one point in the Pareto front approximation reported by RVNS-MD.

2.8.3.3 Performance assessment under instances variations

In the previous section it was shown that RVNS-R is the only algorithm that performs different than the others when it comes to the execution time and the two set coverage. Then, this version of RVNS was used to carry out an statistical analysis whose goal is to state how the variations of the instances impact the algorithm performance. With this purpose, the overall nondominated vector generation, the k-distance, the hypervolume, and the execution time variations among instance classes were analyzed.

The statistical tests used to contrast differences in the performance of RVNS-R under different characteristics of the instances were Kruskal-Wallis tests, which null and alternative hypotheses are stated as follows:

H_0: The performance of RVNS-R is similar for all instance classes considering metric i,

H_1: The performance of RVNS-R is different for at least one instance class considering metric i,

where i is the overall nondominated vector generation, the k-distance, the hypervolume, or the execution time.

For a better understanding of the results reported in this section, notice that a Pareto set approximation may contain solutions with different number of markets. For each instance solved through RVNS-R, the minimum and maximum percentage of visited markets was stored. The average of these values per class are displayed in Table 2.16. It is should be noted that the number of visited markets seems to be larger in capacitated instances than in uncapacitated instances.

Considering the k-distance and the hypervolume, the Kruskal-Wallis tests lead

Table 2.16: Average minimum and average maximum percentage of visited markets

Percentage of markets	**Class LCH**	**Class LCL**	**Class LUH**	**Class LUL**
Minimum	48.20	48.11	6.40	6.16
Maximum	93.93	94.70	59.54	60.20

us to accept the respective null hypotheses (p-values of 0.7319 and 0.5392, respectively). Then, we conclude that RVNS-R performs similar for all instance classes.

Despite these results, it can be observed that the Pareto front approximations reported by RVNS-R for Classes LCL and LUL tend to present lower values of latency than those reported for Classes LCH and LUH, as shown in Figure 2.1. This is an expected outcome due to the fact that in the former cases the service time to the customers is lower than in the latter ones, thus the customers have to spend less time waiting for service. Figure 2.1 also shows that the Pareto front approximations reported for Classes LUH and LUL tend to present lower values of cost than those reported for Classes LCH and LCL. As mentioned before, the former cases require to visit a lower quantity of markets to satisfy the demand; therefore, the transportation cost is smaller than the one observed for the latter instance classes.

If the performance of RVNS-R is measured using the overall nondominated vector generation, the Kruskal-Wallis test allows us to reject the null hypothesis with a p-value of 0.00006654. Therefore, the overall nondominated vector generation is different for at least one instance class.

In order to know which class has the Pareto front approximation with different cardinality, a Dunn's test with a Holm adjustment was carried out. The adjusted p-values obtained from the tests are displayed in Table 2.17. The values in bold are those that allow to reject the null hypothesis that the instance classes corresponding to the respective row and column have similar overall nondominated vector generation value.

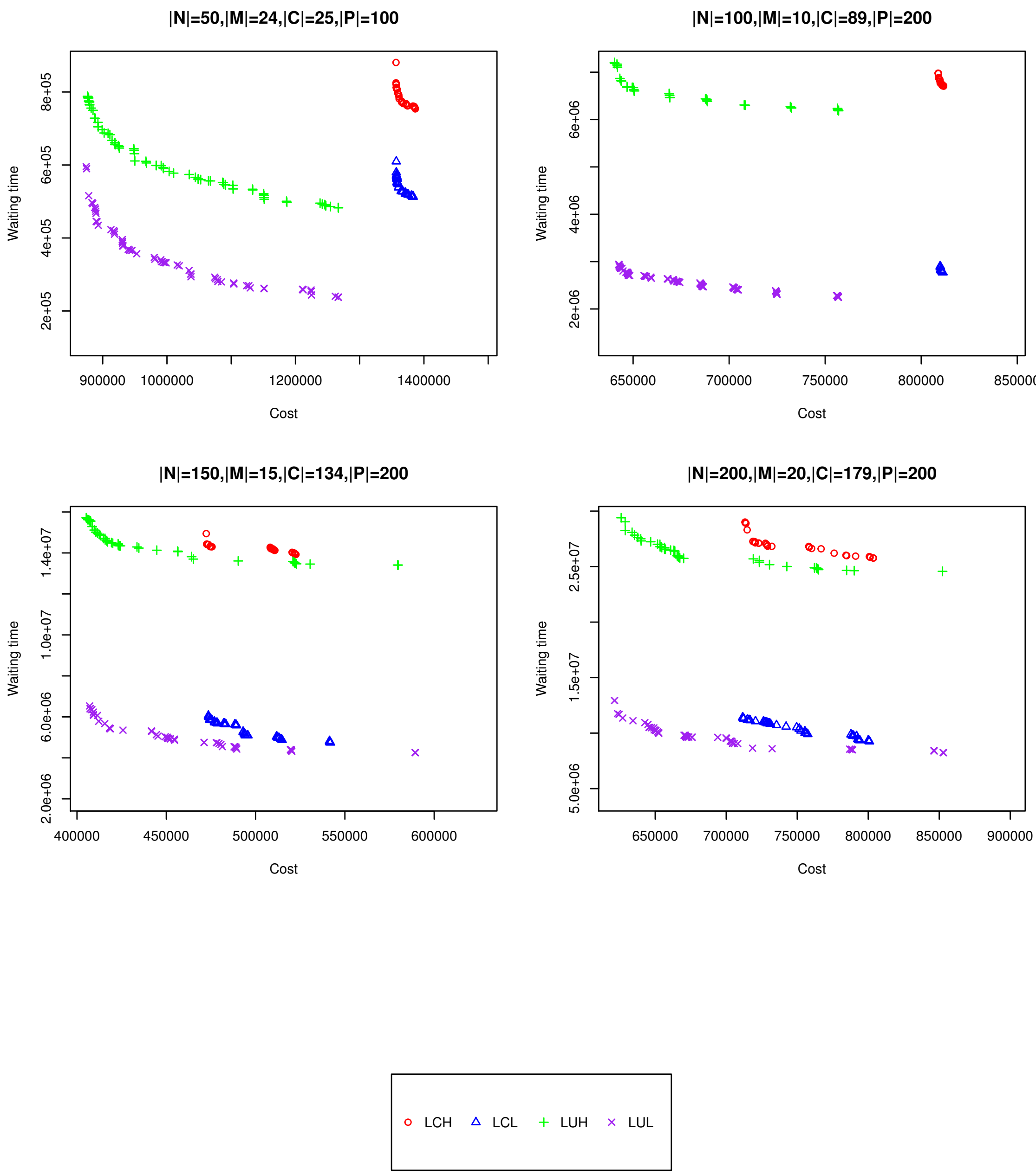

Figure 2.1: Pareto front approximations reported by RVNS-R for instances belonging to different classes

Table 2.17: p-values to evaluate differences among the overall nondominated vector generation reported by RVNS-R for each instance class

	LCH	LCL	LUH
LCL	0.408	-	-
LUH	**0.0032**	**0.0052**	-
LUL	**0.0008**	**0.0015**	0.6181

The results shown in Table 2.17 and the information displayed in Tables 2.5, 2.7, 2.9, and 2.11 lead us to conclude that the Pareto front approximations corresponding to capacitated instances have a smaller overall nondominated vector generation value than the Pareto front approximations associated with the uncapacitated instances. This can be explained by noticing that more markets have to be visited to satisfy the demand in the capacitated instances; then, it is possible to find more permutations of nodes (routes) for uncapacitated cases than for capacitated ones, thus allowing the algorithm to find more potential non-dominated solutions in the former cases.

On the other hand, if the performance of RVNS-R is measured by the execution time, a p-value of 0.03618 leads us to reject the null hypothesis of the Kruskal-Wallis test. Then, RVNS-R requires different execution time for at least one instance class.

Seeking to determine for which class RVNS-R performs different with respect to the execution time, a Dunn's test was carried out; nevertheless, the test was inclusive. Though, some conclusions can be stated from the boxplot displayed in Figure 2.2. Notice that for classes LCH and LCL, the median values (1803.94 s and 1802.13 s, respectively) are higher than those calculated for classes LUH and LUL (749.95 s and 516.38 s, respectively). Also, the median values corresponding to capacitated classes are similar to the third quartile values corresponding to uncapacitated classes (1802.16 s and 1801.14 s, respectively). We can conclude that RVNS-R finished its execution because it ran out of time for more than half of the capacitated instances

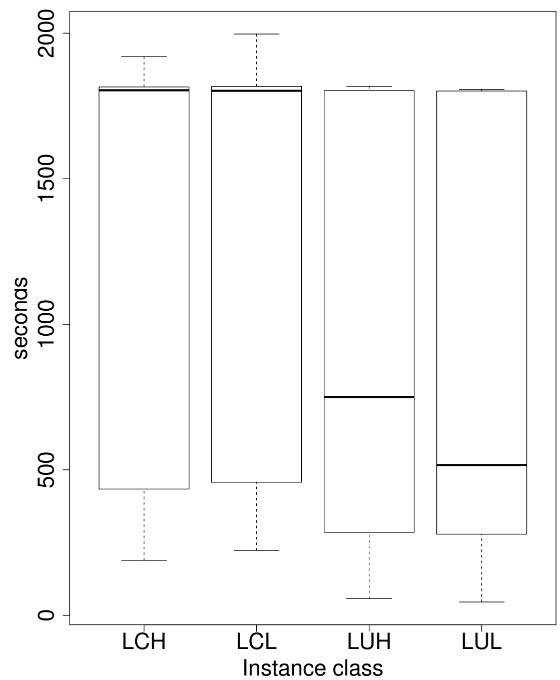

Figure 2.2: RVNS-R execution time

and for more than 25% of the uncapacitated ones, since the RVNS stopping criterion is to reach 1800 seconds of execution time or 10 consecutive iterations without updating the pool of solutions.

Tables 2.5, 2.7, 2.9, and 2.11 confirm these observations: RVNS-R reached the time limit for 11 instances of class LCH, 11 instances of class LCL, five instances of class LUH, and six instances of class LUL.

In order to explain these results, consider that more markets are visited in the solutions corresponding to capacitated instances and thus it is more likely to observe blocks of markets. All local search operators, except MR and MI, operate over blocks. Then, more neighbor solutions are evaluated when solving capacitated instances than when solving uncapacitated ones, thus increasing the execution time.

Finally, it was carried out another set of statistical tests whose aim is to analyze how the variations of the characteristics of the instances impact the local search operators performance. With this purpose, the efficiency of the operators was measured as follows:

$$\text{efficiency}(LSO_i) = \frac{\text{number of times } LSO_i \text{ improved the solution}}{\text{number of times } LSO_i \text{ was executed}} \times 100 \qquad (2.36)$$

where $LSO_i \in \{$IntraR (cost), IntraR (latency), IntraS (cost), IntraS (latency), Intra2 (cost), Intra2 (latency), InterR (cost), InterR (latency), MR (cost), MR (latency), MI $\}$.

The efficiency of the local search operators is displayed in Tables 2.18–2.21.

In order to state whether the performance of RVNS-R is affected by the characteristics of the instances, Kruskal-Wallis tests were performed. The null and alternative hypotheses are stated as follows:

H_0: All of the instance classes effects are identical for local search operator LSO_i efficiency,

H_1: At least one of the instance classes effects is different than the others for local search operator LSO_i efficiency.

For operators InterR (cost), MR (cost), MI, IntraR (latency), IntraS (latency), Intra2 (latency), InterR (latency), and MR (latency), the null hypothesis is not rejected with respective p-values of 0.06325, 0.914, 0.05794, 0.06434, 0.768, 0.08812, 0.2996, and 0.2644. Then, these algorithms have similar efficiency despite the instance class.

On the other hand, the null hypothesis is rejected for IntraR (cost), IntraS (cost), and Intra2 (cost) with respective p-values of 3.62×10^{-5}, 0.004602, and 0.004272. Then, these algorithms have different efficiency for at least one instance class. In order to determine for which instance class the efficiency of the local search operators is different, Dunn's tests with Holm adjustment were carried out.

Table 2.18: Efficiency of local search operators for class LCH

	Cost						Latency				
Instance	**IntraR**	**IntraS**	**Intra2**	**InterR**	**MR**	**MI**	**IntraR**	**IntraS**	**Intra2**	**InterR**	**MR**
100_10_89_200_c_r_h	43.40	29.99	53.51	5.18	0.00	0.00	55.06	34.60	41.97	3.95	0.00
100_89_10_100_c_r_h	41.30	28.20	40.05	37.86	1.91	57.72	35.10	21.80	36.12	12.89	43.97
100_89_10_150_c_r_m	43.88	29.22	40.83	40.40	13.68	72.44	42.63	26.26	46.36	23.73	54.07
100_89_10_50_c_r_m	39.68	25.40	39.28	41.07	15.48	64.21	43.14	26.42	43.20	24.20	53.83
150_134_15_100_c_r_l	36.48	21.56	29.93	39.79	12.92	65.64	46.06	31.33	39.07	35.87	55.66
150_134_15_150_c_r_l	38.16	21.54	29.19	42.27	11.95	67.70	46.98	29.80	41.78	33.55	56.07
150_134_15_50_c_r_l	35.29	20.78	26.12	38.76	10.83	62.75	43.00	29.94	34.82	36.61	55.27
150_15_134_200_c_r_m	40.36	27.11	42.39	27.67	0.77	37.78	41.84	26.89	34.07	21.38	22.07
200_179_20_100_c_r_m	40.60	30.63	51.46	51.88	17.19	73.58	50.72	29.88	46.15	43.96	62.75
200_179_20_150_c_r_h	49.30	30.93	48.26	53.85	2.08	65.96	50.63	44.44	38.46	27.50	51.72
200_179_20_50_c_r_m	39.32	27.44	41.79	50.00	10.26	66.67	45.73	29.36	43.97	51.67	60.40
200_20_179_200_c_r_m	41.86	27.82	45.32	30.27	0.00	36.30	42.49	30.40	36.43	31.29	22.98
50_24_25_100_c_r_h	32.13	23.96	27.51	35.84	0.94	40.76	32.16	21.12	22.14	18.79	35.64
50_44_5_150_c_r_h	34.60	16.99	23.10	6.08	0.81	35.92	21.55	12.35	19.37	0.44	26.75
50_44_5_200_c_r_l	33.54	20.61	26.86	21.42	9.98	62.91	37.31	25.88	30.45	13.43	53.01
50_44_5_50_c_r_m	39.64	24.53	34.38	27.19	4.73	59.93	37.66	22.47	34.21	7.81	51.34
Average	39.35	25.42	37.50	34.35	7.10	54.39	42.00	27.68	36.79	24.19	44.10

Table 2.19: Efficiency of local search operators for class LCL

	Cost						Latency				
Instance	**IntraR**	**IntraS**	**Intra2**	**InterR**	**MR**	**MI**	**IntraR**	**IntraS**	**Intra2**	**InterR**	**MR**
100_10_89_200_c_nr	36.0185	26.2228	44.9969	8.25893	0	0	48.0309	32.5409	33.3333	6.4978	0
100_89_10_100_c_nr	40.0993	25.7601	37.1556	29.2654	1.67504	55.0256	34.9895	22.8202	30.6834	12.8773	43.1871
100_89_10_150_c_nr	41.8063	29.8194	43.7413	41.1619	12.395	68.8249	44.2559	26.5808	44.0191	24.7863	53.7879
100_89_10_50_c_nr	40.3386	24.8601	38.8903	40.269	13.5116	65.1749	42.8771	26.725	42.6386	23.5833	54.4166
150_134_15_100_c_nr	36.6987	22.4684	29.7551	39.8605	12.0773	65.1648	45.0463	32.2096	40.2703	33.3032	55.9023
150_134_15_150_c_nr	35.7991	20.8245	30.4406	40.08152	9.9359	69.395	47.9868	31.6624	40.8542	35.0078	56.0386
150_134_15_50_c_nr	34.2219	19.6052	25.372	37.4377	8.80478	61.599	42.4712	30.5265	36.3186	33.9419	53.7871
150_15_134_200_c_nr	34.6504	22.5528	37.7943	22.6096	0.08287	22.4227	36.6462	22.2642	27.7393	21.4971	22.1271
200_179_20_100_c_nr	38.959	31.5245	43.0189	54.9669	16.1765	68.4211	49.7496	29.5681	47.1698	52.6786	60.3774
200_179_20_150_c_nr	47.4398	35.2436	55.7522	54	0	65.2174	44.3182	21.4286	41.5584	33.3333	53.3333
200_179_20_50_c_nr	38.9297	24.7423	42.2374	54.1502	11.2069	66.0194	47.5277	24.8077	46.0358	51.6588	54.902
200_20_179_200_c_nr	41.0121	28.7757	44.6675	40.5896	0	39.313	43.2039	32.4786	37.9747	38.2653	30.5785
50_24_25_100_c_nr	31.2334	20.5806	30.08381	31.4535	0.339271	36.5106	28.6405	20.3218	26.1955	16.2707	37.2313
50_44_5_150_c_nr	39.7536	23.2342	33.414	8.36364	1.19048	36.9478	26.6827	17.7049	21.1155	1.0101	25.5102
50_44_5_200_c_nr	33.7416	19.5522	26.5049	21.7966	8.8964	65.0185	38.4633	26.8877	28.4131	14.708	53.1353
50_44_5_50_c_nr	40.5997	24.496	35.2339	25.8215	3.84455	63.4941	36.9565	20.8265	34.8773	8.75796	51.1344
Average	38.21	25.02	37.44	34.38	6.26	53.03	41.12	26.21	36.20	25.51	44.09

Table 2.20: Efficiency of local search operators for class LUH

	Cost						Latency				
Instance	**IntraR**	**IntraS**	**Intra2**	**InterR**	**MR**	**MI**	**IntraR**	**IntraS**	**Intra2**	**InterR**	**MR**
100_10_89_200_nc_r	41.7166	29.1748	49.8627	15.8842	0.55814	16.1833	48.7385	28.2756	33.1283	15.8528	30.0252
100_89_10_100_nc_r	28.82	20.3612	24.7643	31.1378	7.08139	74.8611	45.7557	31.926	37.9293	17.3823	51.5242
100_89_10_150_nc_r	31.4645	21.2577	24.7114	31.7584	7.65704	72.7905	45.0814	29.7333	37.3008	16.2182	52.0071
100_89_10_50_nc_r	30.08989	21.2057	25.7073	27.1192	7.08769	71.2958	44.1543	29.2041	35.8705	19.7817	50.6803
150_134_15_100_nc_r	32.9958	21.0066	28.0632	26.6154	10.7817	72.3397	46.5947	31.9026	35.0303	21.0821	50.2955
150_134_15_150_nc_r	32.0601	20.6647	26.0472	31.1542	8.28951	71.9393	48.5068	32.5369	38.8483	23.6516	51.9398
150_134_15_50_nc_r	30.7768	19.1865	24.7272	21.0662	6.48697	67.184	44.0686	25.1365	29.5091	18.3048	49.9269
150_15_134_200_nc_r	37.9518	22.6178	48.513	22.0217	0.578704	26.1932	49.6625	22.2346	27.8736	20.4183	24.6558
200_179_20_100_nc_r	29.5577	19.686	24.2472	40.8415	7.70138	76.0749	49.1401	32.8791	42.1805	24.1873	50.9434
200_179_20_150_nc_r	29.9605	20.5686	26.3463	44.7461	7.68621	77.8541	51.1562	33.9492	43.6189	19.6899	51.3514
200_179_20_50_nc_r	30.4302	20.3135	25.1913	35.6099	7.31707	76.2546	46.7502	30.08656	34.5194	21.6397	49.3917
200_20_179_200_nc_r	40.0866	26.2651	49.6732	22.7273	0.840336	28.2486	51.2141	24.095	32.3398	22.2467	28.3286
50_24_25_100_nc_r	38.0816	21.8872	34.4956	23.4316	3.4761	49.9678	45.9019	22.3822	28.8364	15.1066	44.5918
50_44_5_150_nc_r	33.9784	22.6904	24.306	21.2209	7.83423	66.369	41.1072	25.5106	27.4762	14.0046	50.4711
50_44_5_200_nc_r	31.9501	19.027	22.3045	19.1208	10.0196	64.6288	39.8792	26.5327	27.0862	12.4765	50.08608
50_44_5_50_nc_r	30.6084	18.1549	21.0319	15.6055	11.5385	67.8372	37.0883	23.7225	24.7821	10.3548	50.08616
Average	33.16	21.50	30.00	26.88	6.56	61.25	45.92	28.14	33.52	18.27	46.03

Table 2.21: Efficiency of local search operators for class LUL

	Cost						Latency				
Instance	**IntraR**	**IntraS**	**Intra2**	**InterR**	**MR**	**MI**	**IntraR**	**IntraS**	**Intra2**	**InterR**	**MR**
100_10_89_200_nc_nr	39.2287	26.8689	46.7372	30.9547	0.660689	13.4917	46.0154	25.1074	30.08973	15.7462	27.0776
100_89_10_100_nc_nr	29.7112	21.5311	25.1472	32.9775	8.59179	74.3237	46.7888	28.8101	35.8238	17.3881	51.1292
100_89_10_150_nc_nr	29.1994	20.0264	25.669	30.7333	6.76933	73.8128	46.0401	31.3081	37.6691	16.1937	50.7304
100_89_10_50_nc_nr	29.343	20.0097	24.6196	26.0799	7.26379	71.9081	44.0493	29.723	37.9485	18.8925	51.2048
150_134_15_100_nc_nr	31.9975	20.417	27.3866	25.9492	10.08695	72.5	46.9973	31.0598	38.2257	20.5824	50.3333
150_134_15_150_nc_nr	30.08366	18.8985	24.2787	30.5393	7.98875	73.0358	47.6692	32.1753	37.5624	22.6336	51.5498
150_134_15_50_nc_nr	27.5391	18.0303	23.8606	20.8014	7.62024	63.6579	43.6806	25.2218	31.3291	15.8986	49.9087
150_15_134_200_nc_nr	37.7352	23.8448	49.1637	19.2808	0.758294	23.2092	48.355	21.9523	26.835	17.496	23.5409
200_179_20_100_nc_nr	29.1557	20.3219	26.4027	39.7953	7.67004	76.058	50.0306	32.9409	38.9173	25.5689	50.6838
200_179_20_150_nc_nr	33.2172	22.2995	28.0281	47.1137	9.23594	77.8908	50.2383	34.4789	40.0688	25.9684	51.5504
200_179_20_50_nc_nr	29.9851	21.1708	24.6249	35.171	6.37795	75.778	47.8691	31.8365	34.1216	25.1282	50.08712
200_20_179_200_nc_nr	42.2291	32.0551	53.1746	20.5273	0.473934	38.8095	50.7129	26.0366	32.4641	26.4479	32.021
50_24_25_100_nc_nr	36.8531	24.1925	34.6377	25.9423	3.53293	53.1968	45.85	23.2558	27.64	15.5457	42.8249
50_44_5_150_nc_nr	32.9383	21.3512	24.6104	22.0673	7.77298	66.1538	39.8557	26.8172	27.9331	12.8011	48.9514
50_44_5_200_nc_nr	34.7206	22.0036	23.7272	20.6369	9.25926	61.2245	38.6472	24.3789	29.0212	12.2469	50.8791
50_44_5_50_nc_nr	27.48	16.564	18.9588	10.8821	7.69231	65.4429	36.2488	19.0338	23.8663	9.40439	49.5675
Average	32.59	21.85	30.06	27.47	6.36	61.28	45.57	27.76	33.10	18.62	45.76

Table 2.22: p-values to evaluate differences among the efficiency of IntraR (cost), IntraS (cost), and Intra2 (cost) reported by RVNS-R for each instance class

	IntraR (cost)			**IntraS (cost)**			**Intra2 (cost)**		
	LCH	**LCL**	**LUH**	**LCH**	**LCL**	**LUH**	**LCH**	**LCL**	**LUH**
LCL	0.5497	-	-	0.6486	-	-	0.8643	-	-
LUH	**0.0013**	**0.006**	-	**0.0124**	0.0318	-	**0.0215**	**0.022**	-
LUL	**0.0003**	**0.0022**	0.3485	**0.0208**	0.0435	0.4099	**0.0244**	**0.0223**	0.4773

The adjusted p-values of the Dunn's test corresponding to IntraR (cost), IntraS (cost), and Intra2 (cost) are displayed in Table 2.22. The values in bold are those that allow to reject the null hypothesis that the operator has similar efficiency in the instance classes corresponding to the respective row and column.

The information displayed in Tables 2.18–2.22 leads us to conclude that IntraR (cost), IntraS (cost), and Intra2 (cost) are more efficient in capacitated instances than in uncapacitated ones. Since it is more likely to observe blocks of markets in the former cases, more neighbor solutions are explored within these local search operators, thus increasing the odds to improve the current solution which translates in a larger efficiency.

2.8.4 Chapter conclusions

A variant of the well-known TPP was introduced, the so called the bi-objective Traveling Purchaser Problem with Deliveries (2-TPPD). The importance of the introduction of the 2-TPPD comes from the relatively scarce bi-objective contributions in the TPP literature and the fact that, to the best of our knowledge, there is only one work in the literature considering deliveries in the TPP context.

An ϵ-constraint method whose single-objective problems were solved through CPLEX 12.6 was shown to be unable to solve instances containing more than 10 nodes. This was an expected outcome due to the large number of variables and constraints required to linearize the model and to the subtour elimination constraints, which are known to make difficult to solve VRPs to optimality.

Three versions of a RVNS were proposed in order to approximate the Pareto front of larger instances. Considering the k-distance and the hypervolume, the three versions of RVNS perform similarly. RVNS-R reports Pareto front approximations with a larger overall nondominated vector generation value than RVNS-LNI. Even though RVNS-R requires a larger execution time than RVNS-LNI and RVNS-MD, it reports Pareto front approximations with a better two set coverage than RVNS-LNI for instances of class LCL and than RVNS-MD for instances of class LUL.

RVNS-R seems to report Pareto front approximations with a larger cardinality in a shorter running time for uncapacitated instances than for capacitated ones. Also, the efficiency of some local search algorithms varies depending on the instance class. Hence the importance of designing metaheuristics based on multiple local search operators for solving difficult problems.

Chapter 3

A rich team orienteering problem

In this chapter, a rich Team Orienteering Problem (rTOP) is used to model a real-life problem faced by a local perishable products supplier. The rich Team Orienteering Problem (rTOP) takes into account several additional features such as the delivery of multiple products, split deliveries, capacitated vehicles, incomplete services, and soft time windows. The problem is modeled through a mixed linear integer programming formulation and solved by a multi-start variant of an Adaptive Large Neighborhood Search (ALNS) scheme. Computational experiments were carried out over a large set of instances of the problem. The results reveal that the multi-start ALNS produces better results than the classical implementation of the metaheuristic in which a single solution is built and the improved. Besides, the proposed heuristic outperformed CPLEX in 186 out of 195 instances.

3.1 Motivation

The problem studied in this chapter arises from a real-life problem faced by a local perishable goods supplier, it consists of planning the daily delivery schedule to local customers.

At the beginning of the working day, the stock and the customer requests become available. Sometimes, the stock of regular products is not large enough to satisfy the total demand, so it may be enlarged by adding units of lower quality products to it. Even so, the stock may remain insufficient. Then, for each customer three options are available: the request is ignored, it is partially satisfied, or it is fully satisfied. When it is decided to partially or fully satisfy a request with both regular and lower quality products, the amount of delivered regular products must be larger than the amount of lower quality products.

The company owns a heterogeneous fleet of vehicles to perform the scheduled deliveries within the time windows imposed by the customers and within the drivers working hours. There are some customers that allow the drivers to arrive after the closing of their time window but before a given time threshold; in such case, the drivers can deliver the products but they must wait a certain time imposed by the customer. Besides, each customer can be served by more than one vehicle.

Even though this problem arises from the specific necessities of a products supplier, it takes into account several features that may be faced by other suppliers of different kind of products. Therefore, the design of an efficient solution algorithm for the problem can be highly useful to solve similar problematics for different companies.

3.2 Problem description

In order to decide which customer requests will be satisfied, we propose to associate each customer with a score that reflects its importance. For example, long-term customers are associated with a larger score than new ones. The score corresponding to each customer is collected according to the proportion of the satisfied demand. Furthermore, a smaller score is collected for delivering lower quality products than for delivering regular ones. The objective is then to collect as much score as possible.

Taking into account the fact that the duration of the vehicle routes must not exceed the duration of the working day and the maximization of the score, the problem described in the previous section can be modeled as the Team Orienteering Problem (TOP) that is described in Section 1.1, with multiple additional features:

- *delivery of multiple products:* several products are offered by the supplier and delivered to the customers;
- *split deliveries:* customers can be served by more than one vehicle;
- *vehicles capacity:* there is an available heterogeneous fleet of vehicles to perform the deliveries and the number of units of a product delivered by a vehicle cannot exceed its capacity;
- *incomplete services:* it is possible to visit a customer and not satisfy its total demand, but the collected score will be proportional to the satisfied demand; besides, a lower score is collected for delivering lower quality products than for delivering regular products;
- *soft time windows:* if a vehicle arrives at a customer location before the closing of its time window, the service starts within the time window; otherwise, if the vehicle arrives after the closing of the time window but before a time threshold, the service will start at the time threshold; and
- *service level:* if units of a lower quality product are delivered to a customer, it most receive at least that number of units of regular product.

3.3 Literature review

Several variants of the Orienteering Problem (OP) have been studied in the literature. A brief literature review on OPs related to the rTOP is here described.

Further information about the OP, its variants, and applications can be found in Vansteenwegen et al. (2011a) and Gunawan et al. (2016).

- **Capacitated Team Orienteering Problem (CTOP):** It is an extension of the TOP in which a fleet of identical capacitated vehicles is available. This problem has been studied by:
  - **Archetti et al. (2009):** The authors introduced the CTOP and solved it through branch-and-price (B&P), Variable Neighborhood Search (VNS), and Tabu Search (TS).
  - **Archetti et al. (2013b):** The authors proposed an improved B&P to solve the problem.
  - **Luo et al. (2013):** The authors solved the problem through an adaptive ejection pool with toggle-rule diversification algorithm. Their results outperformed the ones reported by Archetti et al. (2009).
  - **Tarantilis et al. (2013):** The authors proposed a bilevel filter-and-fan method to solve the CTOP. Their results also outperformed the ones reported by Archetti et al. (2009).
- **Capacitated Team Orienteering Problem with Incomplete Services (CTOP-IS):** This problem is an extension of the CTOP in which it is not necessary to fully satisfy the demand of a customer when it is visited. The collected score in each visit depends on the percentage of the satisfied demand. The CTOP-IS was introduced by Archetti et al. (2013a). The authors carried out a worst-case analysis and solved the problem through a B&P scheme. It was shown that the collected score may double by allowing incomplete services.
- **Split Delivery Capacitated Team Orienteering Problem (SDCTOP):** This problem is another extension of the CTOP in which a customer can be visited by more than one vehicle. The SDCTOP was introduced by Archetti et al. (2014b), who carried out a worst-case analysis that revealed that the

collected score may double by allowing split deliveries. Also, a B&P algorithm was proposed to solve the problem. The results showed that the increase in the collected score is instance-dependent.

- **Split Delivery Capacitated Team Orienteering Problem with Minimum Delivery Amounts (SDCTOP-MDA):** This problem, proposed by Wang et al. (2014), is an extension of the SDCTOP that arises from noticing that even that split deliveries can cause an increment in the collected score, they can cause inconveniences to the customers. In this problem, a minimum amount of demand must be delivered in each visit. The authors carried out a worst-case analysis that reveals that the collected score can double if the minimum delivery amount is less than half the demand. On the other hand, the collected score can increase by up to 50% if the minimum delivery amount is half the demand.

- **Split Delivery Capacitated Team Orienteering Problem with Incomplete Services (SDCTOP-IS):** This problem, introduced by Archetti et al. (2014a), is an extension of the SDCTOP and the CTOP-IS in which both incomplete services and split deliveries are taken into account. The authors solved the problem through B&P, VNS, and TS. The two latter heuristics were adapted from Archetti et al. (2009).

- **Orienteering Problem with Variable Profits (OPVP):** This problem, introduced by Erdoğan and Laporte (2013), is a variant of the OP in which the score is partially collected. Contrary to the CTOP-IS, in the OPVP, the percentage of the score collected in each visit depends on the time spent in the visit. The authors solved the problem through branch-and-cut (B&C).

- **Multi-Constraint Team Orienteering Problem with Time Windows (MCTOPTW):** The MCTOPTW was proposed to design routes for tourists. Each customer is seen as a point of interest in a city and associated with several attributes. Each attribute has an available budget and the visited points of interest cannot exceed the budget attributes. Thus, these are knapsack constraints. Besides, hard time windows are imposed to the points of interest. The problem has been addressed by:

- **García et al. (2010):** The authors introduced the MCTOPTW and solved by Iterated Local Search (ILS).
- **Sylejmani et al. (2012):** The authors solved the MCTOPTW through TS.

- **Multi-Constraint Team Orienteering Problem with Multiple Time Windows (MCTOPMTW):** This problem is an extension of the MCTOPTW in which it is taken into account that the points of interests can have more than one time window per day. This problem has been addressed by:
  - **Souffriau et al. (2013):** The authors introduced the MCTOPMTW and hybridized the ILS proposed by García et al. (2010) with a Greedy Randomized Adaptive Search Procedure (GRASP) to solve it.
  - **Lin and Yu (2015):** The authors solved the MCTOPMTW by means of a Simulated Annealing (SA) that outperformed the algorithm proposed by Souffriau et al. (2013).

It is to note that the rTOP is an extension of the SDCTOP-IS, since it takes into account vehicles capacity, incomplete services, and split deliveries. In addition, the rTOP considers a heterogeneous fleet of vehicles instead of a homogeneous one, soft time windows, and the distribution of multiple products.

On the other hand, note that each product can be associated to a knapsack constraint in which the right-hand side is the stock and the weights are the demands. Also, it is evident that the vehicle capacities are also knapsack constraints. Then, both the distribution of multiple products and the vehicles capacity can be modeled by the knapsack constraints considered in the MCTOPTW and in the MCTOPMTW. Nonetheless, in these two problems, the nodes can be visited at most once, the score is fully obtained when the node is visited, and hard time windows are enforced, instead of soft time windows.

To the best of our knowledge, the rTOP takes into account several features that have not been considered in the TOP literature, such as a heterogeneous fleet, multiple products, and soft time windows with a penalty scheme reflected in a waiting time rather than in a cost.

Table 3.1 summarizes the literature review here described and highlights the similarities and differences among the rTOP and the previously discussed problems. Each column displays the following information:

- **Authors:** The authors who studied the problem.
- **MP:** Indicates whether the delivery of multiple products is taken into account.
- **TW:** Indicates whether hard time windows (**H**) or soft time windows (**S**) are considered.
- **C:** Indicates whether the vehicles are homogeneous (**Ho**) or heterogeneous (**He**).
- **IS:** Indicates whether incomplete services are considered.
- **SD:** Indicates whether split deliveries are taken into account.
- **Solution method:** The algorithm used to solve the problem.

3.4 Mathematical model

In this section, the rTOP is formally described and modeled through a mixed integer linear programming formulation.

Table 3.1: Problems related to the rich Team Orienteering Problem

Authors	**MP**	**TW**		**C**		**IS**	**SD**	**Solution method**
		H	**S**	**Ho**	**He**			
Archetti et al. (2009)				✓				B&P, VNS, and TS
Archetti et al. (2013b)				✓				B&P
Luo et al. (2013)				✓				Adaptive ejection pool with toggle-rule diversification algorithm
Tarantilis et al. (2013)				✓				Bilevel filter-and-fan method
Archetti et al. (2013a)				✓		✓		B&P
Archetti et al. (2014b)				✓			✓	B&P
Wang et al. (2014)				✓			✓[a]	
Archetti et al. (2014a)				✓		✓	✓	B&P, VNS, and TS
Erdoğan and Laporte (2013)						✓		B&C
García et al. (2010)	✓[b]	✓			✓[b]			ILS
Souffriau et al. (2013)	✓[b]	✓[c]			✓[b]			ILS with GRASP
Lin and Yu (2015)	✓[b]	✓[c]			✓[b]			SA
This chapter	✓		✓		✓	✓	✓	ALNS

[a] A minimum amount of the customer demand must be delivered in each visit

[b] This feature can be modeled through a knapsack constraint

[c] Multiple time windows

3.4.1 Notation

Let $G = (N_0, A)$ be a complete directed graph where $N_0 = \{0, \dots, n, n+1\}$ is the node set, and $A = \{(i,j) : i, j \in N_0, i \neq j\}$ is the arc set, 0 and $n+1$ are two copies of the depot, and $N = \{1, \dots, n\}$ is the customer set. The product set is denoted by P, and the vehicle set by V. The number of units of product $p \in P$ demanded by customer $i \in N$ is denoted by q_{ip}. Each customer i has an associated score β_i that is fully obtained if the demand of i is completely satisfied with regular products (or partially obtained, if the demand is partially satisfied with regular products), and a lower score γ_i that is partially collected if the demand is partially satisfied with lower quality products. The time window within which the vehicles can serve i without a penalty is denoted by $[a_i, b_i]$, while B_i denotes the maximum arrival time at $i, (a_i < b_i \leq B_i)$. If a vehicle arrives at i between b_i and B_i, service must start at B_i. It is worth noticing that if $b_i = B_i$, the customer has a hard time window. The service time of i is denoted by e_i. The depots have a service time equal to zero and a hard time window $[0, t_{max}]$, where t_{max} is the maximum route duration. The total stock of regular and lower quality product p are denoted by f_p and g_p, respectively. The capacity of vehicle $k \in V$ is denoted by c_k, and the travel time of arc $(i,j) \in A$ is denoted by d_{ij}.

3.4.2 Mixed integer linear programming formulation

The following decision variables are also needed:

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \in V \text{ travels from } i \text{ to } j; (i,j) \in A, k \in V \\ 0 & \text{otherwise;} \end{cases}$$

$$y_{ik} = \begin{cases} 1 & \text{if customer } i \text{ is visited by vehicle } k; i \in N, k \in V \\ 0 & \text{otherwise;} \end{cases}$$

$$v_{ik} = \begin{cases} 1 & \text{if vehicle } k \text{ arrives at customer } i \text{ before } b_i \\ 0 & \text{otherwise;} \end{cases}$$

t_{ik} arrival time at customer i by vehicle k;

s_{ik} starting time of service at customer i by vehicle k;

r_{ikp} number of units of regular product p delivered by vehicle k to customer i;

u_{ikp} number of units of lower quality product p delivered by vehicle k to customer i.

The rTOP is then formulated as follows:

$$\text{maximize } z = \sum_{i\in N} \beta_i \left(\frac{\sum_{k\in V}\sum_{p\in P} r_{ikp}}{\sum_{p\in P} q_{ip}} \right) + \sum_{i\in N} \gamma_i \left(\frac{\sum_{k\in V}\sum_{p\in P} u_{ikp}}{\sum_{p\in P} q_{ip}} \right) \tag{3.1}$$

subject to:

$$\sum_{i\in N_0\setminus\{0\}} x_{0ik} = 1 \qquad k \in V \tag{3.2}$$

$$\sum_{i\in N_0\setminus\{n+1\}} x_{i,n+1,k} = 1 \qquad k \in V \tag{3.3}$$

$$\sum_{j\in N_0\setminus\{n+1\}} x_{jik} = y_{ik} \qquad i \in N, k \in V \tag{3.4}$$

$$\sum_{j\in N_0\setminus\{0\}} x_{ijk} = y_{ik} \qquad i \in N, k \in V \tag{3.5}$$

$$\sum_{k\in V} (r_{ikp} + u_{ikp}) \le q_{ip} \qquad i \in N, p \in P \tag{3.6}$$

$$\sum_{i\in N}\sum_{p\in P} (r_{ikp} + u_{ikp}) \le c_k \qquad v \in V \tag{3.7}$$

$$\sum_{i\in N}\sum_{k\in V} r_{ikp} \le f_p \qquad p \in P \tag{3.8}$$

$$\sum_{i\in N}\sum_{k\in V} u_{ikp} \le g_p \qquad p \in P \tag{3.9}$$

$$\sum_{k\in V} r_{ikp} \ge \sum_{k\in V} u_{ikp} \qquad i \in N, p \in P \tag{3.10}$$

$$t_{n+1,k} \le t_{max} \qquad k \in V \tag{3.11}$$

$$s_{ik} + e_i + d_{ij} - t_{jk} \le t_{max}(1 - x_{ijk}) \qquad (i,j) \in A, k \in V \tag{3.12}$$

$$t_{max}(v_{ik} - 1) \le b_i - t_{ik} \le t_{max} v_{ik} \qquad i \in N, k \in V \tag{3.13}$$

$$t_{max}(y_{ik} - 1) \le B_i - t_{ik} \le t_{max} y_{ik} \qquad i \in N, k \in V \tag{3.14}$$

$$a_i v_{ik} + B_i(y_{ik} - v_{ik}) \le s_{ik} \qquad i \in N, k \in V \tag{3.15}$$

$$b_i v_{ik} + B_i(y_{ik} - v_{ik}) + t_{max}(1 - y_{ik}) \geq s_{ik} \qquad i \in N, k \in V \tag{3.16}$$

$$x_{ijk} \in \{0,1\} \qquad (i,j) \in A, k \in V \tag{3.17}$$

$$y_{ik} \in \{0,1\} \qquad i \in N, k \in V \tag{3.18}$$

$$r_{ikp} \in \{0\} \cup \mathbb{Z}^+ \qquad i \in N, k \in V, p \in P \tag{3.19}$$

$$u_{ikp} \in \{0\} \cup \mathbb{Z}^+ \qquad i \in N, k \in V, p \in P \tag{3.20}$$

$$t_{ik} \geq 0 \qquad i \in N_0, k \in V \tag{3.21}$$

$$s_{ik} \geq 0 \qquad i \in N, k \in V \tag{3.22}$$

$$v_{ik} \in \{0,1\} \qquad i \in N, k \in V. \tag{3.23}$$

Objective function (3.1) seeks to maximize the sum of the proportional collected score for delivering both regular and lower quality products. Equations (3.2) and (3.3) ensure that all routes start and end at the depot, while equations (3.4) and (3.5) assure the flow conservation. Constraints (3.6) guarantee that the total delivered amount of regular and lower quality units of a product required by a customer does not exceed the demand. Constraints (3.7) ensure that the vehicles capacity is not exceeded. Constraints (3.8) guarantee that the total delivered amount of a regular product is not larger than its available stock. Similarly, Constraints (3.9) assure that the total delivered amount of a lower quality product does not exceed its stock. Constraints (3.10) ensure that when a customer request is fully or partially satisfied with both regular and lower quality products, the amount of regular product is larger than the amount of lower quality product. Constraints (3.11) guarantee that the duration of the routes does not exceed the time limit. Constraints (3.12) ensure time consistency and avoid subtours. Constraints (3.13) and (3.14) ensure that if a customer is visited by a vehicle, it either arrives before the closing of the time window, or after that time, but before the maximum arrival time. Constraints (3.15) and (3.16) guarantee that if the arrival time takes place before the closing of the time window, the service starts within the time window, and if the arrival occurs after the closing of the time window but before the maximum arrival time, then the service starts at B_i. Finally, Constraints (3.17)–(3.23) define the domain of the decision variables.

3.5 Multi-start adaptive large neighborhood search

In this section, the ALNS scheme is introduced as well as the proposed multi-start variant to solve the rTOP.

3.5.1 Adaptive large neighborhood search

ALNS is a metaheuristic proposed by Ropke and Pisinger (2006). This framework has been widely applied in recent years to solve Vehicle Routing Problems (VRPs) (see Salazar-Aguilar et al. (2011); Demir et al. (2012); Hemmelmayr et al. (2012); Ribeiro and Laporte (2012); Masson et al. (2013); Adulyasak et al. (2014); Azi et al. (2014); Salazar-Aguilar et al. (2014); Emeç et al. (2016); Luo et al. (2016); Mancini (2016), and Schiffer and Walther (2018)).

The ALNS applies several destroy and repair operators in order to generate large neighborhoods through which the search space is explored to improve an initial solution. All destroy and repair operators have a weight that is dynamically adjusted according to the quality of the solutions that have been obtained by using them. At each iteration of the ALNS, one destroy operator and one repair operator are randomly chosen according to a probability distribution that depends on the operators weights. Let Ω^+ and Ω^- be the sets of repair and destroy operators, respectively; and let ρ^+ and ρ^- be the weights vector of the destroy and repair operators, respectively. Then, the ALNS for the minimization case is outlined in Algorithm 3.

Algorithm 3 Adaptive large neighborhood search

Require: x ▷ A feasible solution

$x^* \leftarrow x$
$\rho^- \leftarrow (1, \dots, 1), \rho^+ \leftarrow (1, \dots, 1)$
repeat
    Select operators $\omega^- \in \Omega^-$ and $\omega^+ \in \Omega^+$ using ρ^- and ρ^+
    $x^t \leftarrow \omega^-(x)$
    $x^t \leftarrow \omega^+(x^t)$
    if x^t is accepted **then**
        $x \leftarrow x^t$
    end if
    if $z(x^t) < z(x^*)$ **then**
        $x^* \leftarrow x^t$
    end if
    Update ρ^- and ρ^+
until stop criterion is met
return x^*

3.5.2 Multi-start adaptive large neighborhood search

As shown in Algorithm 3, the ALNS starts from a single solution which is then improved. Even though the destroy operators diversify the search and allow ALNS to escape from local optima, it is possible that the initial solution is far from an optimal one so the algorithm will require a large number of iterations to reach it. The multi-start ALNS seeks to compensate this weakness by combining the ALNS scheme with a multi-start procedure.

Multi-start algorithms generate and then improve a pool solutions. The best of them is reported as an approximation to the global optimum. Enlarging the pool size increases the number of local optimal solutions, thus increasing the odds of finding a better solution.

The multi-start ALNS is a two-phase algorithm which first phase consists of building a certain number of solutions and then applying ALNS to each of them for a limited number of iterations. In the second phase, only the ALNS that found the best solution from among all the ALNSs executed in the first phase continues its execution for a certain number of iterations. The first phase of the algorithm is devoted to provide diversification to the search since the increment on the number of initial solutions increments the odds to obtain better solutions. On the other hand, the second phase seeks to intensify the search, since it tries to improve the current solution.

The following subsections describe the construction method used to find the initial solutions, the destroy operators, the repair operators, the acceptance criterion, and the weights update mechanism used in the multi-start ALNS.

3.5.2.1 CONCEPTS AND NOTATION

Due to the presence of the soft time windows, many calculations must be carried out when disturbing a solution to evaluate its feasibility. Seeking to overcome this issue, for each customer i visited in a solution by a vehicle k, two values are stored, the minimum and the maximum times at which the arrival at i can take place without impacting the arrival times at the other customers, $minShift_{ik}$ and $maxShift_{ik}$, respectively.

Let R be a solution containing $m = |V|$ routes, each of which associated with a vehicle. From now on, the terms route and vehicle will be used indistinctly. Then, $minShift_{ik}$ and $maxShift_{ik}$ are defined as follows.

DEFINITION 3.1 ($minShift$) *Let $N(k)$ be the set of customers that are visited in route k. Then, for each route $k \in R$ and for every node $i \in N(k) \cup N_0$, $minShift_{ik}$ is defined through the following equations:*

$$minShift_{ik} = s_{\underline{i}r} + e_{\underline{i}} + d_{\underline{i}i} \qquad i \in N(k), k \in R \tag{3.24}$$

$$minShift_{0k} = 0 \qquad k \in R, \tag{3.25}$$

where $\underline{i}$ is the predecessor of i in r.

DEFINITION 3.2 ($maxShift$) *Let $N(k)$ be the set of customers that are visited in route k. Then, for each route $k \in R$ and for every node $i \in N(k) \cup N_0$, $maxShift_{ik}$ is defined through the following equations:*

$$maxShift_{ik} = \begin{cases} B_i & \textit{if } B_i \leq t_{\bar{i}r} - d_{i\bar{i}} - e_i \\ b_i & \textit{if } b_i \leq t_{\bar{i}r} - d_{i\bar{i}} - e_i < B_i \\ t_{\bar{i}r} - d_{i\bar{i}} - e_i & \textit{if } t_{\bar{i}r} - d_{i\bar{i}} - e_i < b_i \end{cases} \qquad i \in N(k), k \in R \tag{3.26}$$

$$maxShift_{n+1,k} = t_{max} \qquad k \in R, \quad (3.27)$$

where $\bar{i}$ is the successor of i in r.

Also, when evaluating the possibility of inserting a non-visited customer l between two visited customers i and j, the feasibility of this move is evaluated through $minArrival_l(i,j,k)$ which is defined as follows:

Definition 3.3 ($minArrival$) *Given an arc (i,j) traversed by vehicle k, the minimal arrival time at customer l to be inserted between i and j is defined as*

$$minArrival_l(i,j,k) = s_{ir}^{min} + e_i + d_{il}, \qquad (3.28)$$

where s_{ir}^{min} is the starting time of service at i, assuming that the arrival time takes place at $minShift_{ik}$.

Then, it is feasible to insert the customer l between i and j if and only if $minArrival_l(i,j,k) \leq B_l$ and $s_{lr} + e_l + d_{lj} \leq maxShift_{jk}$.

Finally, Δt_{ik} is defined as the increment in the duration of route k after performing the cheapest feasible insertion of customer i.

3.5.2.2 Construction method

Initially, m routes containing only nodes 0 and $n+1$ are built, and t_{0k} and $t_{n+1,k}$ are set to 0 for all routes. Then, iteratively, some customers are added to these routes as follows. The route k' with the smallest duration is selected and it is built a candidate list (CL) that contains the customers whose demand have not been completely satisfied and that can be inserted in k' without loosing feasibility. Each customer $i \in CL$ is then evaluated according to (3.29):

$$\frac{\beta_i\left(\frac{\sum_{p\in P}\left(q_{ip}-\sum_{k\in V}\left(r_{ikp}+u_{ikp}\right)\right)}{\sum_{p\in P} q_{ip}}\right)}{\left(B_i - t_{ik'}\right)\Delta t_{ik'}}. \tag{3.29}$$

In (3.29), it is assumed that i will be inserted in the cheapest feasible position. Besides, $t_{ik'}$ is set equal to the sum of the end of the service at the predecessor and the travel time from it, if it is feasible; otherwise, it is set to the minimum arrival time, calculated as in (3.28).

The evaluation function (3.29) provides a trade-off among the potential increase of the collected score, the feasibility of the arrival time at the customer, and the increment in the route duration.

Thereafter, the following equation is used to build a restricted candidate list (RCL):

$$RCL = \{i \in CL : f(i) \in [\alpha f_{min} + (1-\alpha) f_{max}, f_{max}]\}, \tag{3.30}$$

where $\alpha \in [0, 1]$.

A customer from the RCL is randomly selected and then inserted into k'. The amount of products to be delivered to this customer is set according to Algorithm 4. Besides, it could be necessary to update the arrival times at the predecessor and/or successor. If that is the case, this procedure is performed as in Algorithm 5.

The procedure is repeated until all routes have empty candidate lists.

It should be noted that the parameter α in the RCL controls the level of randomness used to select the candidate customers to be included in the solution, so different values of α are expected to produce different solutions. Then, the construction operator consists on repeating the above-mentioned procedure eleven times using $\alpha = 0, 0.1, ..., 1$. After that, the best solution among all of them becomes the outcome of the construction algorithm.

Algorithm 4 Set deliveries to a new visit

Require:

i ▷ Customer to be added

k' ▷ Route in which a new visit to i will be added

$r_{ik'p} = 0$, for all $p \in P$

$u_{ik'p} = 0$, for all $p \in P$

$slackVehicle = c_{k'} - \sum_{j \in N(k')} \sum_{p \in P} (r_{jk'p} + u_{jk'p})$

for $p \in P$ such that $q_{ip} > 0$ **do**

  if $slackVehicle = 0$ **then**

    break

  end if

  $slackCustomer = q_{ip} - \sum_{k \in V} (r_{ikp} + u_{ikp})$

  $slackRegular = f_p - \sum_{k \in V} \sum_{j \in N(k)} r_{jkp}$

  $r_{ik'p} = \min\{slackCustomer, slackRegular, slackVehicle\}$

  $slackVehicle = slackVehicle - r_{ik'p}$

end for

if $slackVehicle > 0$ **then**

  for $p \in P$ such that $q_{ip} > 0$ **do**

    if $slackVehicle = 0$ **then**

      break

    end if

    $slackCustomer = q_{ip} - \sum_{k \in V} (r_{ikp} + u_{ikp})$

    $slackLower = g_p - \sum_{k \in V} \sum_{j \in N(k)} u_{jkp}$

    $slackRegularLower = \sum_{k \in V} \sum_{j \in N(k)} (r_{ikp} - u_{ikp})$

    $u_{ik'p} = \min\{slackCustomer, slackLower, slackRegularLower, slackVehicle\}$

    $slackVehicle = slackVehicle - u_{ik'p}$

  end for

end if

return $r_{ik'p}, u_{ik'p}$ ▷ Deliveries to customer i in route k' for all $p \in P$

Algorithm 5 Rules for updating the arrival times

Require:

l ▷ Customer to be inserted

i, j ▷ Nodes between which l will be inserted

k ▷ Route in which l will be inserted

1: $t_{lk} = s_{ik} + e_i + d_{il}$

2: **if** $t_{lk} > B_l$ or $s_{lk} + e_l + d_{lj} > maxShift_{jk}$ **then**

3: $t_{ik} = minShift_{ik}$

4: $t_{lk} = s_{ik} + e_i + d_{il}$

5: **end if**

6: $t'_{jk} = s_{lk} + e_l + d_{lj}$

7: **if** $t'_{jk} > t_{jk}$ **then**

8: $t_{jk} = t'_{jk}$

9: **end if**

return t_{ik}, t_{jk}, t_{lk} ▷ Arrival times at l, i, and j

3.5.2.3 Destroy operators

Hereunder are described the destroy operators used in the multi-start ALNS. Some of these operators require to shift the arrival times at the customers to their earliest arrival time without losing solution feasibility. In such cases, the earliest arrival time at a customer i by a vehicle k is set to $minShift_{ik}$.

- *Elimination by average score (EAS)*: This operator is based on the elimination operator used by Hu and Lim (2014). Let $\bar{s}$ be the average score obtained by the visits included in the current solution. A visit is eliminated with probability $prob$ if its collected score is smaller than $\bar{s}$, and with probability $1 - prob$, otherwise. When the multi-start ALNS execution starts, $prob$ is set to 0.1. The probability $prop$ is updated at the end of each iteration in which EAS is carried out as follows: it is set to $\min\{prob + 0.1, 1\}$ if the solution is not accepted, and to 0.1, otherwise.

- *Random elimination (RE)*: A random number of random visits are removed from the current solution.

- *Elimination of a sequence-1 (ES-1)*: This operator is a modification of the shake step used by Vansteenwegen et al. (2009). Let *start* and *length* be two integers. For every route, it is removed a sequence of *length* consecutive visits starting from *start*. If the end of a route is reached before removing *length* visits, customers visited after depot 0 are removed. The arrival times at the customers remaining in the solution is set to their earliest arrival time. At the beginning of the multi-start ALNS, both *start* and *length* are set to 1 and they are updated every time ES-1 is executed. Parameter *start* is set to 1 if the solution is accepted, and to $start + length$, otherwise. On the other hand, *length* is set to 1 if the solution is accepted, and to $length + 1$, otherwise. If either *start* or *length* is larger than the minimum number of customers included in a route, then the respective parameter is set to this number.

- *Elimination of a sequence-2 (ES-2)*: This operator is similar to ES-1, except that the arrival times at the remaining visits do not change.

- *Elimination based on history (EH)*: Let *update* be the number of times in which the incumbent solution has been updated (line 11 of Algorithm 3), and let *i_incumbent* be the number of times in which customer i has been included at least once in the incumbent solution. Each visit to customer i is removed with probability $1 - p_i$, where $p_i = \frac{i\_incumbent+1}{update+1}$.

- *Intra-route exchange (IntraE)*: For every route, and for every customer included in it, another random customer visited in the same route is selected. Operator IntraE exchanges the visits positions, if feasible. If the exchange is performed, the arrival time at all visits in the route are set to the minimum possible. The exchange is accepted only if the route duration is decreased.

- *Inter-route exchange (InterE)*: For every route k_i, and for every customer i visited in it, a customer j visited in route k_j is randomly chosen, considering that $i \neq j$ and $k_i \neq k_j$. Operator InterE exchanges i and j positions, if feasible. It is to note that it is possible that a visit to j is already included in k_i and/or i is already visited in k_j. If it is the case, the visits are merged according to Algorithm 6; otherwise, new visits are created by following Algorithms 5 and 4. The arrival time at each customer visited in the selected routes is set to the minimum possible. The exchange is accepted only if the duration of both routes is decreased.

- *Intra-route relocate (IntraR)*: For every route, and for every customer included in it, a random position of the same route is selected. Operator IntraR relocates the customer in the selected random position, if feasible. The arrival time at each customer included in the route is set to the minimum possible. The move is accepted if the route duration decreases.

- *Inter-route relocate (InterR)*: For every route k_1, and for every customer i visited in it, a random route k_0 is chosen, considering that $k_1 \neq k_0$. Notice that it is possible that customer i is already visited in k_0. If that is the case, Algorithm 6 is executed to update the deliveries. Otherwise, the visit is relocated to k_0 in a random position according to Algorithms 5 and 4. In either cases, the arrival times at all customers in k_1 are shifted to the minimum possible. The relocation is accepted if the route duration decreases.

Notice that when the destroy operators are executed, some visits are either removed or their deliveries are modified (except for IntraE and IntraR). As a consequence, for the customer whose visit was removed or modified, it is possible to obtain solutions in which the total delivered amount of lower quality product is larger than the total delivered quantity of regular product. For a better understanding of this observation, consider the following example.

Example 3.4 (Infeasibility due to destroy mechanisms) *Consider an instance of the rTOP in which a single product is distributed. Also, consider a customer*

Algorithm 6 Merge visits

```
Require:
    i      ▷ Customer whose visits will be updated
    k_0    ▷ Route in which the visit to i will be kept
    k_1    ▷ Route from which the visit to i will be removed
slackVehicle = c_{k_0} − \sum_{j∈N(k_0)} \sum_{p∈P} (r_{jk_0p} + u_{jk_0p})
for p ∈ P such that q_{ip} > 0 do
    if slackVehicle = 0 then
        break
    end if
    increment_p = min{r_{ik_1p}, slackVehicle}
    r_{ik_0p} = r_{ik_0p} + increment_p
    slackVehicle = slackVehicle − increment_p
end for
if slackVehicle > 0 then
    for p ∈ P such that q_{ip} > 0 do
        if slackVehicle = 0 then
            break
        end if
        increment_p = min{u_{ik_1p}, slackVehicle}
        u_{ik_0p} = u_{ik_0p} + increment_p
        slackVehicle = slackVehicle − increment_p
    end for
end if
return r_{ik_0p}, u_{ik_0p}, for all p ∈ P
```

whose demand is equal to 10 and a feasible solution for the instance, in which two vehicles k_1 *and* k_2 *serve the customer. The customer receives four units of regular product from vehicle* k_1*, and two units of regular product and three units of lower quality product from vehicle* k_2*. In total, the customer receives nine units of the demanded product: six units of regular product and three units of lower quality product. Then, the deliveries are feasible.*

Now, suppose that a destroy operator removes the visit to the customer performed by vehicle k_1*. Now the customer only receives the units delivered by vehicle* k_2*. The solution becomes infeasible since three units of lower quality product are delivered to the customer while it only receives two units of regular product.*

If any constraint from the group (3.10) is violated after the execution of a destroy operator, a simple repair mechanism is applied as follows. For each customer i it is checked whether their deliveries are feasible. If there is a product p whose deliveries are infeasible, for each visit to i in the current solution, the delivered quantity of lower quality product p is set to the minimum between itself and the delivered amount of regular product p. Then, in Example 3.4, vehicle k_2 will now deliver two units of regular product and two units of lower quality product.

3.5.2.4 Repair operators

All repair operators are based on, iteratively, selecting a customer from a candidate list (CL) and then inserting a visit to it in a route of the current solution. The visit is inserted in the position of the route for which the duration increment is minimum. The process is repeated until it is not possible to insert more visits.

The customers belonging to the CL are those whose demand has not been fully satisfied. Every time a visit to a customer is inserted, Algorithms 5 and 4 are followed.

The repair operators only differ in how they select the customer to be visited as explained below:

- *Insertion based on evaluation function-1 (IEF-1)*: For each route, the customers belonging to the CL are evaluated according to (3.29). At each iteration, the customer with the smallest evaluation is inserted in the solution.
- *Insertion based on evaluation function-2 (IEF-2)*: As in IEF-1, the customers belonging to the CL are evaluated according to (3.29) for every route. At each iteration, a roulette wheel mechanism is followed to select the customer to be inserted in the solution.
- *Insertion based on score-1 (IS-1)*: The customers belonging to the CL are evaluated according to the numerator of (3.29), i.e., the potential score increment if the customer were included in the solution. At each iteration, the customer with the largest evaluation is inserted in the solution.
- *Insertion based on score-2 (IS-2)*: Similarly to IS-1, the customers in the CL are evaluated according to the numerator of (3.29). Iteratively, a roulette wheel mechanism is followed to select the customer to be inserted in the solution.
- *Insertion based on the route duration increment-1 (IRDI-1)*: For each route, the customers belonging to the CL are evaluated according to the minimum increment in the duration of that route if they were inserted. At each iteration, the customer with the smallest evaluation is inserted in the solution.
- *Insertion based on the route duration increment-2 (IRDI-2)*: As in IRDI-1, for every route, the customers belonging to the CL are evaluated according to the minimum increment in the duration of that route if they were inserted. At each iteration, a roulette wheel mechanism is used to select the customer to be inserted.
- *Insertion based on history-1 (IH-1)*: The customers belonging to the CL are evaluated according to the number of times that they have been visited in the

incumbent solution. At each iteration, the customer with the largest evaluation is inserted in the solution.

- *Insertion based on history-2 (IH-2)*: The customers belonging to the CL are evaluated according to the number of times that they have been visited in the incumbent solution, as in IH-1. At each iteration, a roulette wheel mechanism is followed to select the customer to be inserted in the solution.

3.5.2.5 Acceptance criterion

In the multi-start ALNS, a SA criterion is used to decide whether a solution will be accepted or not, as in Ropke and Pisinger (2006).

A solution R^t is accepted with a probability of $e^{-(z(R)-z(R^t))/T}$, where T is the temperature. The temperature starts at T_0 and decreases at each iteration. In the multi-start ALNS, the temperature decreases according to a linear function. In particular, at a certain iteration it, the temperature is calculated according to Equation (3.31).

$$T = T_0 - \frac{T_0 \times it}{\#total\ iterations + 1}, \tag{3.31}$$

As in Ropke and Pisinger (2006), the objective function value of the initial solution is calculated, and T_0 is set such that the probability of accepting a solution that is 5% worse than the initial one is equal to 50%.

3.5.2.6 Weights update

In order to select the destroy and repair operators to be executed in each ALNS iteration, a roulette wheel mechanism that takes into account the operators weights is followed. Every time that an operator w_i is used, its weight is adjusted according to (3.32):

$$\rho_i = \rho_i + \max\{\sigma_1, \sigma_2, \sigma_3\}, \tag{3.32}$$

where

$$\sigma_1 = \begin{cases} 3 & \text{if the solution is better than the incumbent} \\ 0 & \text{otherwise;} \end{cases}$$

$$\sigma_2 = \begin{cases} 2 & \text{if the solution is better than the current one} \\ 0 & \text{otherwise;} \end{cases}$$

$$\sigma_3 = \begin{cases} 1 & \text{if the solution is accepted} \\ 0 & \text{otherwise.} \end{cases}$$

3.6 Computational experiments

This section is divided into three subsections. The test instances are described in the first one, the second subsection describes the experimental environment, and the third one reports the computational tests results. In turn, the third subsection is divided into four groups of experiments. The first group is devoted to study how the number of iterations executed in the first phase of the multi-start ALNS impacts the quality of the reported solution. The second group of experiments was carried out to assess the contribution of the destroy and repair operators to the overall algorithm. The third group of experiments compares the results reported by the multi-start ALNS with those reported by CPLEX 12.6. Finally, the fourth group analyzes how the multi-start ALNS execution time is affected by variations on the stock level and on the lower score collected for delivering lower quality products.

Table 3.2: Characteristics of the instance classes

Class	Stock level	Lower score
1	$f_p = \sum_{i \in N} q_{ip}, \forall p \in P$	-
2	$f_p + g_p = \sum_{i \in N} q_{ip}, \forall p \in P$	$\gamma_i = 0.75\beta_i$
3	$f_p + g_p = \sum_{i \in N} q_{ip}, \forall p \in P$	$\gamma_i = 0.5\beta_i$
4	$f_p + g_p < \sum_{i \in N} q_{ip}, \forall p \in P$	$\gamma_i = 0.75\beta_i$
3	$f_p + g_p < \sum_{i \in N} q_{ip}, \forall p \in P$	$\gamma_i = 0.5\beta_i$

3.6.1 Instances

The instances used to assess the efficiency of the multi-start ALNS were adapted from those proposed by Vansteenwegen et al. (2009) for the TOP with time windows. The number of vehicles goes from three to 20; the number of customers, from 48 to 288; and the number of products, from five to 15.

In total, 195 instances were generated and then partitioned into five classes. In Class 1, the stock of regular product is sufficient to satisfy the whole demand. In Classes 2 and 3, the available quantity of regular product is insufficient to satisfy the demand but the demand can be fully satisfied by adding units of lower quality product; in Class 2, the lower score $\gamma_i = 0.75\beta_i$, and in Class 3, $\gamma_i = 0.5\beta_i$. Finally, in Classes 4 and 5, the total demand cannot be satisfied not even adding units of lower quality product to the stock of regular product; in Class 4, $\gamma_i = 0.75\beta_i$, and in Class 5, $\gamma_i = 0.5\beta_i$. Table 3.2 summarizes the characteristics of each instance class.

3.6.2 Experimental environment

Eight different versions of the multi-start ALNS were coded in C++ and tested in order to asses the efficiency of the algorithm, as it will be discussed further on. Besides, model (3.1)–(3.23) and its linear relaxation were also coded in C++ and solved through CPLEX 12.6. All algorithms and the model were compiled with GNU on a 2.1 GHz Intel Xeon(R) CPU E5-2620 v2 under Ubuntu 14.04 operating system.

3.6.3 Experimental results

This section reports the results obtained from the computational experiments, which are divided in four groups: one devoted to study the effect of the number of iterations executed in the first phase of the multi-start ALNS, another one seeking to determine the impact of the destroy and repair operators in the solution quality, another one devoted to analyze the quality of the reported solutions compared with the ones reported by CPLEX 12.6, and the last one devoted to analyze the differences in the execution time of the algorithm among instance classes.

3.6.3.1 Effect of the number of initial solutions

Eight different multi-start ALNS configurations were examined: mALNS (1, 100, 4900), mALNS (12, 100, 3800), mALNS (25, 100, 2500), mALNS (37, 100, 1300), mALNS (1, 100, 9900), mALNS (25, 100, 7500), mALNS (50, 100, 5000), and mALNS (75, 100, 2500). The name of the algorithm follows the format mALNS (a, b, c), where a is the number of initial solutions to be examined in the first phase, b is the number of iterations of the ALNSs executed in the first phase, and c is the number of iterations of the ALNS executed in the second phase.

It is worth noticing that all the first four configurations operate for 5000 ALNS iterations and the remaining ones, for 10000. Furthermore, mALNS (1, 100, 4900) and mALNS (1, 100, 9900) are the typical implementation of the method in which a single solution is built and then improved for 5000 and 10000 iterations, respectively. The rest of them use 25%, 50%, or 75% of the iterations in the first phase, and the remaining ones, in the second phase.

Table 3.3: Average relative gap in percentage with respect to the best found solution

Algorithm version	**Class 1**	**Class 2**	**Class 3**	**Class 4**	**Class 5**	**Complete set**
mALNS (1,100,4900)	3.18	3.30	2.82	3.24	2.65	3.04
mALNS (12,100,3800)	2.52	2.38	2.45	2.26	1.84	2.30
mALNS (25,100,2500)	3.11	2.11	2.57	2.12	1.58	2.30
mALNS (37,100,1300)	3.56	2.36	2.45	2.30	2.20	2.58
mALNS (1,100,9900)	2.86	2.73	2.46	3.60	2.33	2.80
mALNS (25,100,7500)	2.58	1.46	1.07	1.18	0.84	1.43
mALNS (50,100,5000)	2.13	1.57	1.59	1.15	1.09	1.50
mALNS (75,100,2500)	1.87	1.73	1.62	1.77	1.49	1.69

For every instance, all multi-start ALNS configurations were tested and the best reported objective function value among all of them was stored. Then, the relative gap with respect to the best solution R_b was calculated for each reported solution R, as in Equation (3.33). Table 3.3 displays the average relative gap in percentage per instance class and for the complete instance set. Detailed results are shown in Tables A.1–A.10.

$$gap = \frac{z(R_b) - z(R)}{z(R_b)} \times 100\% \tag{3.33}$$

It should be noted that, in average, the worst results were obtained by mALNS (1,100,4900), which is the classical ALNS implementation with 5000 iterations. Besides, it is remarkable that mALNS (12,100,3800), mALNS (25,100,2500), and mALNS (37,100,1300) achieve better results than those reported by mALNS (1,100,9900) despite the fact that the former algorithms only execute half iterations than the latter.

The fact that neither the algorithms using more iterations in the first phase nor the algorithms using more iterations in the second phase report the best results, reveals a trade-off between diversification and intensification. From the algorithms that execute 5000 iterations, the best results were reported by mALNS(12,100,3800) and from the algorithms that execute 10000 iterations, mALNS (25,100,7500) reports the best results. This suggests that, in the tested instances, a good compromise between diversification and intensification is using 25% of the iterations on the first phase of the multi-start ALNS and 75% on the second one.

For the remaining experiments the version that reported the best results will be used as reference, i.e. mALNS (25,100,7500).

3.6.3.2 Effect of the destroy and repair operators

In order to analyze the effect of the destroy and repair operators, further experiments were carried out by executing mALNS (25,100,7500) 17 times per instance, removing one of the operators each time. Table 3.4 displays the average percent gap of the objective function value obtained by removing each operator individually z_r, with respect to the objective function value reported by mALNS(25,100,7500), $z_{25,100,7500}$, for each instance class, and for the whole set of instances. The gap was calculated using Equation (3.34). Detailed results are shown in Tables A.11–A.20.

$$gap = \frac{z_{25,100,7500} - z_r}{z_{25,100,7500}} \times 100\%. \tag{3.34}$$

Notice that the larger the calculated gap for an operator, the worst the results obtained by removing it. Furthermore, a negative gap reveals that better results are obtained by removing the corresponding operator than by keeping it. EAS seems to be the best operator, since the solutions found by removing it are 2.8% worst, in average. On the other hand, it is remarkable that if some operators are removed, the quality of the solutions increases. This is because, when a bad operator is discarded, the algorithm has the opportunity of choosing better operators.

Table 3.4: Average relative gap in percentage with respect to ALNS(25,100,7500)

Removed operator	**Class 1**	**Class 2**	**Class 3**	**Class 4**	**Class 5**	**Complete set**
EAS	2.26	2.90	2.64	3.58	2.38	2.75
RE	-1.31	-0.58	0.19	-0.55	-0.59	-0.57
ES-1	-0.98	-0.17	-0.26	-0.65	-0.25	-0.46
ES-2	-0.25	0.43	0.56	-0.09	0.27	0.19
EH	-1.15	-0.16	-0.18	-0.27	-0.18	-0.39
IntraE	-0.64	-0.30	0.12	0.12	0.06	-0.13
InterE	-1.35	0.18	0.37	0.17	0.19	-0.09
IntraR	-0.63	0.07	0.69	0.31	0.10	0.11
InterR	-0.57	0.68	0.62	0.38	0.35	0.29
IEF-1	-0.95	0.20	0.28	0.30	0.03	-0.03
IEF-2	-0.83	0.36	0.05	0.56	0.15	0.06
IS-1	0.10	0.26	0.83	-0.02	0.11	0.25
IS-2	-0.61	0.66	0.62	-0.42	0.47	0.14
IRDI-1	-0.51	-0.30	0.18	0.11	0.31	-0.04
IRDI-2	0.00	-0.32	0.16	0.02	0.43	0.06
IH-1	-1.03	0.19	0.29	-0.46	-0.09	-0.22
IH-2	-0.64	-0.19	0.16	0.43	0.04	-0.04

Table 3.5: Average relative gap in percentage with respect to ALNS(25,100,7500)

Removed operators	**Class 1**	**Class 2**	**Class 3**	**Class 4**	**Class 5**	**Complete set**
RE, ES-1	-1.59	-1.13	-0.41	-1.06	-0.46	-0.93
RE, ES-1, EH	-1.80	-1.12	-0.83	-1.32	-1.16	-1.24
RE, ES-1, EH, IH-1	-1.81	-1.32	-1.07	-1.59	-0.96	-1.35
RE, ES-1, EH, IH-1, IntraE	-2.10	-1.36	-1.05	-1.71	-1.29	-1.50
RE, ES-1, EH, IH-1, IntraE, InterE	-2.53	-1.73	-0.88	-1.61	-0.95	-1.54
RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1	-2.29	-1.67	-0.79	-1.72	-1.14	-1.52
RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2	-2.34	-1.96	-1.18	-1.45	-1.10	-1.60
RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1	-2.43	-1.40	-1.08	-1.40	-1.19	-1.50

Further experiments were carried out in order to keep a good set of operators, as follows. First, the operators were ranked in ascending order of the average gap over the complete set of instances. Then, the two operators with the smallest gap were removed from mALNS (25,100,7500) simultaneously, i.e., RE and ES-1. Then, the three operators with the smallest gap were removed from mALNS (25,100,7500) simultaneously, i.e., RE, ES-1, and EH. The process continues until all operators with a negative gap over the complete set of instances are removed. The average percent gap of the obtained results with respect to ALNS(25,100,7500) is displayed in Table 3.5. The gap was calculated according to Equation (3.34). Detailed results are reported in Tables A.21–A.30.

The results shown in Table 3.5 reveal that the best results, in average, are obtained from removing RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, and IH-2, simultaneously.

Overall, the best results are achieved by the multi-start ALNS in which 25 initial solutions are built and improved though ALNS for 100 iterations each one. Then, the best solution is chosen and improved through ALNS for 7500 iterations. The ALNS schemes use EAS, ES-2, IntraR, InterR, IEF-1, IEF-2, IS-1, IS-2, and IRDI-2. This multi-start ALNS version will be hereafter called mALNS*.

3.6.3.3 Solutions quality

In order to assess the quality of the solutions reported by mALNS*, model (3.1)–(3.23) (hereafter called MrTOP) and its linear relaxation (hereafter called RMrTOP) were coded in C++ and solved by CPLEX 12.6. The computation time allowed to solve each instance was set to 7200 CPU seconds, using 10 threads.

CPLEX was not able to solve MrTOP for any instance to optimality but it reported a feasible solution for all cases, i.e. a lower bound for the optimal value of the objective function. On the other hand, CPLEX solved all instances with RMrTOP to optimality, thus providing upper bounds for the optimal value of the objective function in MrTOP. Note that it is possible to obtain better upper bounds than the ones provided by the linear relaxation from the nodes of the B&C tree of CPLEX when it is executed to solve MrTOP. Nonetheless, the upper bounds were not significantly improved after 7200 CPU seconds.

For each instance, we calculated the gap of the objective function value reported by mALNS* with respect to the lower bound and the upper bound, through Equations (3.35) and (3.36), respectively, where $z(\text{mALNS}^*)$ is the value of the objective function reported by mALNS*, $z(\text{MrTOP})$ is the lower bound, and $z(\text{RMrTOP})$ is the upper bound. A small number ϵ is added to the lower bound in the denom-

Table 3.6: Percent gap between the objective values reported by mALNS* and CPLEX for MRTOP

Class	**Minimum**	**Average**	**Maximum**
1	-2.49	60.30	183.59
2	3.78	50.42	163.58
3	-0.93	50.69	155.84
4	-6.58	26789523.07	1044790000
5	-4.73	89.04	2172.15
Complete set	-6.58	57.56[a]	1044790000

[a] Excluding the case in which the gap is equal to 1044790000.

inator of Equation (3.35) because one of the obtained bounds is equal to zero. In the reported calculations, ϵ was set to 0.0001. The calculated gaps are reported in Tables A.31–A.35.

$$gap_{LB} = \frac{z(\text{mALNS}^*) - z(\text{MCTOP})}{z(\text{MrTOP}) + \epsilon} \times 100\%. \tag{3.35}$$

$$gap_{UB} = \frac{z(\text{RMrTOP}) - z(\text{mALNS}^*)}{z(\text{RMrTOP})} \times 100\%. \tag{3.36}$$

Table 3.6 displays the smallest, the average, and the largest percent gap of the objective function value obtained by mALNS* with respect to the lower bound for each instance class. The fact that there are some negative gaps reveals that there are instances in which CPLEX outperforms mALNS*. In fact, Tables A.31–A.35 show that the lower bound reported by CPLEX is better than the result reported by mALNS* in only nine out of 195 instances: one instance from Class 1, one from Class 3, four from Class 4, and three from Class 5. Nevertheless, in average, mALNS* reports results 57.56% better than those reported by CPLEX for MrTOP.

Table 3.7 displays the smallest, the average, and the largest percent gap of the

Table 3.7: Percent gap between the objective values reported by mALNS* and CPLEX for RMrTOP

Class	**Minimum**	**Average**	**Maximum**
1	1.26	10.72	21.37
2	7.26	12.16	18.03
3	5.81	10.50	16.52
4	9.49	19.60	25.04
5	10.43	15.23	18.33
Complete set	1.26	13.66	25.04

Table 3.8: Number of instances with gap smaller than 5%, 10%, 20%, and 30% (mALNS* vs RMrTOP)

Class	**<5%**	**<10%**	**<20%**	**<30%**
1	7	19	38	39
2	0	8	39	39
3	0	22	39	39
4	0	1	22	39
5	0	0	39	39
Total	7	50	177	195

objective function value obtained by mALNS* with respect to the upper bound for each instance class. On the other hand, Table 3.8 displays the number of times that the gap with respect to the upper bound is smaller than 5%, 10%, 20%, and 30%.

The results reported in Table 3.7 reveal that, in the best case, the gap of the objective function reported by mALNS* is only 1.26% worse than the upper bound. This means that this solution is, at most, 1.26% worse than the optimum. On the other hand, in the worst case, the solution reported by the heuristic is, at most, 25.04% worse than the optimum. Furthermore, Table 3.8 shows that in seven out of 195 cases, the gap with respect to the upper bound is smaller than 5%, and in 50 out of 195 cases, the gap does not exceed 10%. Then, we can assure that in seven instances, the gap of the solution obtained by mALNS* with respect to the optimum is lower than 5%, and in 50 cases, it is lower than 10%.

3.6.3.4 Execution time

Table A.36 shows the running time in seconds required by mALNS* to approximate the optimal solution of the rTOP per instance. The total time required to solve each instance class is shown in Figure 3.1. This figure suggests that the computation time depends mostly on the stock level. In fact, the class that was solved in the shortest time is Class 1, in which the total stock is sufficient to satisfy the demand. On the other hand, the classes that required the largest time to be solved are Classes 4 and 5, in which the stock is not large enough to satisfy the demand. It is noteworthy that more decisions must be taken in the latter cases, thus increasing the solutions space size and requiring more computational effort to approximate the optimal solution.

Besides, note that the size of the neighborhoods that are explored through out the destroy and repair operators depends on the number of vehicles, customers, and products. Thus, the larger these parameters are, the longest the computation time required by mALNS* to solve the problem. Table 3.9 shows the average running time,

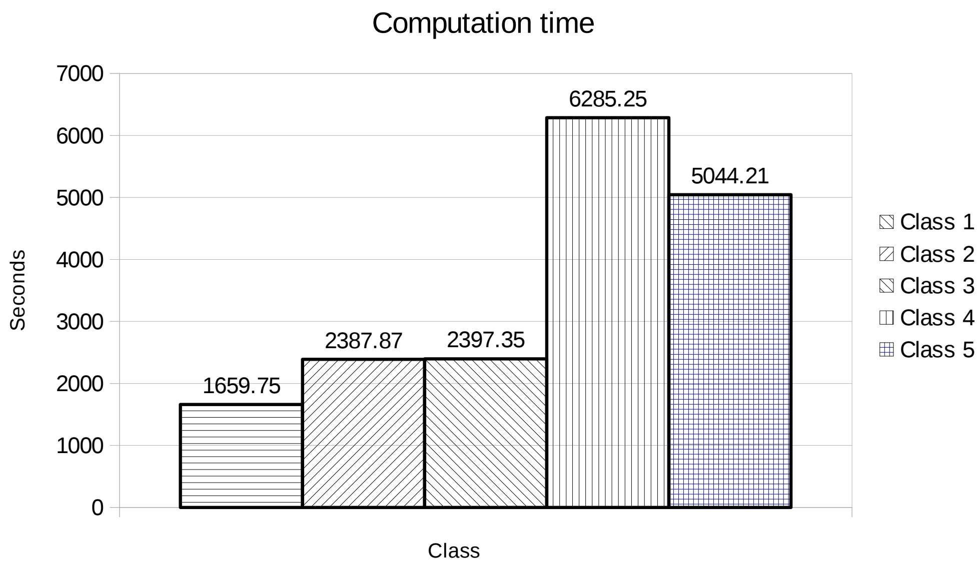

Figure 3.1: Computation time in seconds per instance class

the percent coefficient of variation (defined as the ratio of the standard deviation to the mean), and the minimum and maximum running time required to solve each case per instance class. The high coefficients of variation and the large differences between the maximum and the minimum execution time per instance class suggest that different levels of the above mentioned parameters affect the computation time. In fact, within each class, the instance that was solved faster has the following parameters: $|N| = 48, |V| = 3$, and $|P| = 5$. On the contrary, within classes 1, 3, and 5, the instance that required the largest computation time has parameters $|N| = 288, |V| = 20$, and, $|P| = 15$, while within classes 2 and 4, $|N| = 288, |V| = 18$, and, $|P| = 15$.

3.7 Chapter conclusions

In this chapter, a logistic problem arising from the daily delivery schedule of a perishable products supplier was modeled as a rich TOP in which the delivery of multiple products, split deliveries, vehicles capacity, incomplete services, and soft

Table 3.9: Analysis of the computation time per instance class

Class	**Average**	**Coefficient of variation**	**Minimum**	**Maximum**
1	42.56	75.49	6.1	160.62
2	61.23	108.78	6.53	300.47
3	61.47	109.67	7.1	339.19
4	161.16	192.85	7.72	1406.26
5	129.34	165.44	8.41	1118.18

time windows, are taken into account. The problem was modeled though a mixed integer linear programming formulation and solved by eight variants of a multi-start ALNS.

The computational results reveal that the multi-start ALNS produces better results than those found by the classical implementation of ALNS in which a single solution is built and then improved.

The proposed scheme has shown to produce better results than CPLEX 12.6 in 186 out of 195 instances. Furthermore, the computation of the linear relaxation of the proposed model allows to determine than in seven out of 195 instances, the solutions reported by the multi-start ALNS are, at most, 5% worse than the optimal solution, and in 50 out of 195 cases, the gap does not exceed 10%.

CHAPTER 4

THE ORIENTEERING PROBLEM WITH MANDATORY VISITS AND CONFLICTS

This chapter addresses a variant of the Orienteering Problem (OP) in which it is mandatory to visit some nodes and also incompatibilities among nodes arise. Five mixed integer linear programming formulations are proposed to model the problem. The main difference among the formulations lies in the way they tackle the subtour elimination constraints. The proposed formulations are tested over a large set of instances of the problem. Computational results reveal that the model in which the subtour elimination is addressed by a single-commodity flow formulation allows CPLEX 12.6 to find more optimal solutions within one hour of computation time than the other formulations.

4.1 MOTIVATION AND PROBLEM DESCRIPTION

The Orienteering Problem with Madatory Visits and Conflicts (OPMVC) is an extension of the OP, described in Section 1.1, in which there is a set of nodes that must be included in the route and besides, there are some nodes that have conflict with others, meaning that if a node has conflict with another one, at most one of them can be included in the route.

The problems consists then in designing a route that starts and ends at two fixed nodes, visits all mandatory nodes and some optional ones seeking to maximize the total collected score, while ensuring that the duration of the route does not exceed a threshold time.

The OPMVC has many potential practical applications. For example, it can be used to design personalized routes for tourists. Every point of interest in a city is seen as a node in the OP context and its score depends on the tourist interest in visiting it. The tourist has a limited time to visit the points of interest, so the route duration is constrained. There are some representative sites of a city that cannot be missed, then they are set as mandatory nodes. Conflicts among points of interest help to diversify the visited sites, for example, if the tourist wishes to visit just one church, they become incompatible among them.

Another potential application is the design of hazardous waste collection routes. Potential chemical reactions make it impossible to transport some products in the same vehicle. Then, nodes in which incompatible products have to be collected are incompatible among them. Furthermore, some nodes may require urgent pickups, thus becoming mandatory nodes. The collected score in each node location is proportional to the profit gained by the company for serving it and the route duration is constrained to be smaller than the driver working hours.

4.2 Literature review

As mentioned in Chapter 3, several variants, solution methods, and applications of the OP can be found in the literature. The interested reader is referred to Vansteenwegen et al. (2011b) and Gunawan et al. (2016) for extensive surveys on the OP.

The OPMVC was introduced by Palomo-Martínez (2015). In this work, the problem was modeled through two mixed integer linear programming formulations, and solved by column generation and by a hybrid heuristic scheme that combines

Greedy Randomized Adaptive Search Procedure (GRASP) with Variable Neighborhood Search (VNS). Afterwards, Palomo-Martínez et al. (2017) proposed an improved version of the GRASP-VNS which was later outperformed by a memetic algorithm proposed by Lu et al. (2018).

In Palomo-Martínez (2015), the OPMVC is modeled through two mixed integer linear programming formulations which main difference is the way they handle subtour elimination. Both formulations adapt subtour elimination constraints from the Traveling Salesman Problem (TSP) literature: the ones proposed by Miller et al. (1960), known as MTZ constraints, and those proposed by Wong (1980) and later by Claus (1984), which are based on a multi-commodity flow formulation.

In this work, five formulations for the OPMVC are proposed, tested, and compared to each other. One of the formulations is obtained from adapting the connectivity constraints proposed by Fischetti and Toth (1988) for the Prize Collecting Traveling Salesman Problem (PCTSP) to the OPMVC. The four remaining formulations are obtained by adapting subtour elimination constraints from the TSP literature. One of them is the same formulation proposed in Palomo-Martínez (2015), based on a multi-commodity flow formulation. The others adapt the subtour elimination constraints proposed by Gavish and Graves (1978), those proposed by Dantzig et al. (1954), and the ones proposed by Desrochers and Laporte (1991) to strengthen the MTZ constraints.

4.3 Mathematical model

In this section, the OPMVC is formally described and the five proposed formulations are introduced.

4.3.1 Notation

Let $G = (N, A)$ be a complete undirected graph in which triangular inequality holds. The node set N is divided into the depots set $N_1 = \{1, n\}$, the mandatory nodes set M, and the optional nodes set O; such that $M \cup O = N \backslash \{1, n\}$ and $M \cap O = \emptyset$. A score s_i is associated with each optional node $i \in O$. For each node $i \in N$, the set C_i contains the nodes that have conflict with it. The travel time of arc $(i, j) \in A$ is denoted by t_{ij} and the maximum allowed duration of the route is denoted by t_{max}.

The objective is to design a route that starts at 1 and ends at n, whose duration does not exceed t_{max}, and visits all mandatory nodes and some optional ones in order to maximize the total collected score.

4.3.2 Mixed integer linear programming formulations

The five proposed formulations make use of the following decision variables:

$$x_{ij} = \begin{cases} 1 & \text{if node } j \text{ is visited immediately after node } i, (i,j) \in A; \\ 0 & \text{otherwise.} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{if node } i \text{ is visited, } i \in N; \\ 0 & \text{otherwise.} \end{cases}$$

The OPMVC, without subtour elimination constraints, is formulated as follows:

$$\max z = \sum_{i \in O} s_i y_i \tag{4.1}$$

subject to:

$$\sum_{i \in N:(1,i) \in A} x_{1i} = 1 \tag{4.2}$$

$$\sum_{i \in N:(i,n) \in A} x_{in} = 1 \tag{4.3}$$

$$y_k = 1 \qquad k \in M \tag{4.4}$$

$$\sum_{j \in N:(j,i) \in A} x_{ji} = y_i \qquad i \in N \backslash \{1\} \tag{4.5}$$

$$\sum_{j \in N:(i,j) \in A} x_{ij} = y_i \qquad i \in N \backslash \{n\} \tag{4.6}$$

$$y_i + y_j \leq 1 \qquad i \in N, j \in C_i, C_i \neq \emptyset \tag{4.7}$$

$$\sum_{(i,j) \in A} t_{ij} x_{ij} \leq t_{max} \tag{4.8}$$

$$x_{ij} \in \{0,1\} \qquad (i,j) \in A \tag{4.9}$$

$$y_i \in \{0,1\} \qquad i \in N \tag{4.10}$$

Objective function (4.1) seeks to maximize the total collected score. Constraints (4.2) and (4.3) ensure that the route starts at node 1 and ends at node n, respectively. Constraints (4.4) assure that all mandatory nodes are included in the route. Constraints (4.5) and (4.6) guarantee flow conservation. Constraints (4.7) avoid visiting nodes in conflict with each other. Constraint (4.8) ensures that the duration of the route does not exceed the limit. Finally, constraints (4.9) and (4.10) are related to the domain of the decision variables.

Below, the subtour elimination constraints used in each model are described

4.3.2.1 Dantzig, Fulkerson, and Johnson's subtour elimination constraints

The subtour elimination constraints proposed by Dantzig et al. (1954), also known as clique constraints, provide the strongest known linear relaxation for the TSP but the exponential number of constraints makes their implementation impractical so, usually, they are combined with cut generation procedures.

The subtour elimination constraints proposed by Dantzig et al. (1954) are stated as follows:

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad S \subset \{2, \ldots, n\}, |S| \geq 2 \tag{4.11}$$

Constraints (4.11) are facet defining for the TSP and they can be used for solving the OPMVC. Nonetheless, they can be strengthen for the OPMVC as done by Feillet et al. (2005):

$$\sum_{i,j \in S : i \neq j} x_{ij} \leq \sum_{i \in S \setminus \{k\}} y_i \quad S \subset \{2, \ldots, n\}, |S| \geq 2, k \in S \tag{4.12}$$

Model (4.1)–(4.10), and (4.12) is hereafter called OPMVC-DFJ.

4.3.2.2 Fischetti and Toth's connectivity constraints

The following connectivity constraints that were proposed by Fischetti and Toth (1988) for the PCTSP, also prevent subtours for the OPMVC. Besides, they are equivalent to (4.12).

$$\sum_{i \in \bar{S}} \sum_{j \in S} x_{ij} \geq y_k \quad k \in S, S \subset \{2, 3, \ldots n\}, |S| \geq 2 \tag{4.13}$$

From now on, model defined by (4.1) – (4.10), and (4.13) is called OPMVC-FT.

4.3.2.3 Desrochers and Laporte's subtour elimination constraints

Due to their simplicity, MTZ subtour elimination constraints have been widely used to formulate free-loop solutions for the TSP. Nevertheless, they provide a very weak linear relaxation, so many efforts have been done to strengthen this formulation without compromising its simplicity. One of the most relevant contributions is due to Desrochers and Laporte (1991), who proposed the following facet defining constraints:

$$2 \leq u_i \leq n \qquad i \in N \backslash \{1\} \tag{4.14}$$

$$u_i - u_j + (n-2)x_{ij} + (n-4)x_{ji} \leq n-3 \quad (i,j) \in A. \tag{4.15}$$

Note that variable u_i can be interpreted as the position of node i in the route. Even though constraints (4.14) and (4.15) also avoid subtours in the OPMVC, it is to note that some nodes will not be visited, so variable u_i does not longer represent the position of node i in the route. It is possible to come back to this definition by bounding variables u_i, so constraints (4.14) are replaced by constraints (4.16) which ensure that the position in which node i is included in the route is not larger than the number of visited nodes.

$$2 \leq u_i \leq \sum_{j \in N} y_j \quad i \in N \backslash \{1\} \tag{4.16}$$

From now on, model (4.1) – (4.10), (4.15), and (4.16) will be called OPMVC-DL.

4.3.2.4 Gavish and Graves's subtour elimination constraints

The subtour elimination constraints proposed by Gavish and Graves (1978) for the TSP are based on a single-commodity flow formulation. Let g_{ij} be the flow of a single commodity traversing arc (i,j). Then, the subtour elimination constraints for the proposed by Gavish and Graves (1978) for the TSP are stated as follows:

$$\sum_{j=1}^{n} g_{ji} - \sum_{j=2}^{n} g_{ij} = 1 \qquad i \in N \backslash \{1\} \tag{4.17}$$

$$0 \leq g_{ij} \leq (n-1)x_{ij} \quad (i,j) \in A : j \neq 1 \tag{4.18}$$

Constraints (4.17) and (4.18) guarantee that $n-1$ units of flow leave the origin node and each node consumes one unit of flow. If variable g_{ij} is greater than zero, then its value is equal to the number of arcs from node j to the destination node in the optimal route.

It should be noted that constraints (4.17) assume that all nodes are visited. Therefore, in order to avoid subtours in the OPMVC, constraints (4.17) are replaced by constraints (4.19) which ensures that a node consumes one unit of flow only if it is included in the route.

$$\sum_{j=1}^{n-1} g_{ji} - \sum_{j=2}^{n-1} g_{ij} = y_i \quad i \in N \backslash \{1, n\} \tag{4.19}$$

The optimization model defined by (4.1) – (4.10), (4.18), and (4.19) is hereafter called OPMVC-GG.

4.3.2.5 Wong's subtour elimination constraints

The subtour elimination constraints proposed by Wong (1980) and later by Claus (1984) for the TSP are based on a multi-commodity flow formulation. This formulation has been proven to provide a linear relaxation as strong as the one provided by the subtour elimination constraints of Dantzig et al. (1954), and it is easier to implement in practice.

Consider $n-1$ commodities. Let node 1 be the origin node of one unit of each commodity and let node k be the destination node of commodity $k, k = 2, \ldots, n$. Let z_{ij}^k be the flow of commodity k traversing arc $(i, j) \in A$. Equations (4.20) – (4.25) are the subtour elimination constraints proposed by Wong (1980).

$$z_{ij}^k \leq x_{ij} \quad (i, j) \in A, k = 2, \ldots, n \tag{4.20}$$

$$\sum_{i \in N} z_{1i}^k = 1 \quad k = 2, 3, \ldots, n \tag{4.21}$$

$$\sum_{i \in N} z_{i1}^k = 0 \qquad k = 2, 3, \ldots, n \tag{4.22}$$

$$\sum_{i \in N} z_{ik}^k = 1 \qquad k = 2, 3, \ldots, n \tag{4.23}$$

$$\sum_{i \in N} z_{ki}^k = 0 \qquad k = 2, 3, \ldots, n \tag{4.24}$$

$$\sum_{i \in N} z_{ij}^k - \sum_{i \in N} z_{ji}^k = 0 \qquad k = 2, 3, \ldots, n, j \in N \backslash \{1\}, j \neq k \tag{4.25}$$

Constraints (4.20) guarantee that no flow will traverse arc (i, j) if it does not belong to the route. Constraints (4.21) mean that node 1 is the source node of one unit of each commodity, while Constraints (4.22) ensure that no flow will return to the origin node. Constraints (4.23) and (4.24) guarantee that one unit of commodity k enters to node k and does not leave it. Finally, Constraints (4.25) assure flow conservation.

Consider that a solution of the OPMVC is not a cycle, but a path starting at node 1 and ending at node n; also, it must be taken into account that not all nodes are visited. As a result of these considerations, Constraints (4.22) and (4.23) are reformulated as follows:

$$\sum_{i \in N} z_{in}^k = 1 - y_k \quad k = 2, 3, \ldots, n-1 \tag{4.26}$$

$$\sum_{i \in N} z_{ik}^k = y_k \qquad k = 2, 3, \ldots, n-1 \tag{4.27}$$

Constraints (4.26) ensure that only units of commodities associated with non-visited nodes enter to node n and Constraints (4.27) guarantee that one unit of commodity k enters to node k only if it is visited.

An additional modification to the original subtour elimination constraints is that there is not a commodity associated with node n. If so, Constraints (4.26) and (4.27) would be infeasible when $k = n$.

From now on, the formulation defined by (4.1) - (4.10), (4.20), (4.21), and (4.24) – (4.27) will be called OPMVC-W.

Table 4.1: Formulations for the Orienteering Problem with Mandatory Visits and Conflicts

Model	**Variables**	**Number of variables**	**Constraints**	**Number of constraints**
OPMVC-DFJ	-	0	(4.12)	$\sum_{i=2}^{n-1} iC(n-1, i)$
OPMVC-FT	-	0	(4.13)	$\sum_{i=2}^{n-1} iC(n-1, i)$
OPMVC-DL	$u_i, i \in N \backslash \{1\}$	$n-1$	(4.15) and (4.16)	$(n+1)(n-1)$
OPMVC-GG	$g_{ij}, (i,j) \in A$	$n(n-1)$	(4.18) and (4.19)	$n^2 - n - 1$
OPMVC-W	$z_{ij}^k, (i,j) \in A, k = 2, \ldots, n-1$	$n(n-1)(n-2)$	(4.20), (4.21), and (4.24)–(4.27)	$(n^2+2)(n-2)$

4.3.2.6 Summary

Table 4.1 summarizes the five proposed formulations. For each model, it is reported which and how many variables are added to model (4.1)–(4.10) to avoid subtours, as well as the subtour elimination constraints and its number.

4.4 Computational experiments

This section is divided in four subsections. The first one is devoted to describe the instances used to test the models, the second subsection relates to the experimental environment, the third one describes the methodology used to test the models, and the fourth one reports the computational tests results.

In turn, the fourth subsection is divided in two groups of results. The first group is devoted to compare the quality of the solutions reported by the solver by solving each model. The second group of results analyzes the computation time.

4.4.1 Instances

Nine instance classes were used to test the proposed models: Classes 1 to 6 were taken from Palomo-Martínez et al. (2017), while Classes 7 to 9 were generated for this research. All instances are based on those proposed by Fischetti et al. (1998) for the OP.

Each class contains the same set of graphs, whose size goes from 21 to 262 nodes. Every group has different percentage of mandatory nodes and each node can be free of conflicts or can be incompatible with 1, 2, or 3 nodes. Characteristics of each instance class are summarized in Table 4.2.

4.4.2 Experimental environment

Models OPMVC-DL, OPMVC-GG, OPMVC-W, OPMVC-DFJ, and OPMVC-FT were coded in C++ and solved through CPLEX 12.6 on a 2.10 GHz Intel Xeon(R) CPU E52620 v2 under Ubuntu 14.04 LTS operating system.

4.4.3 Methodology

Models OPMVC-DL, OPMVC-GG, and OPMVC-W were used to solve each instance through CPLEX. The solver stops when it reaches 3600 CPU seconds or its default

Table 4.2: Characteristics of the instance classes

Class	**Percentage of mandatory nodes**	**Percentage of free-conflict nodes**	**Number of instances**
Class 1	10%	<50%	62
Class 2	20%	<50%	55
Class 3	30%	<50%	53
Class 4	10%	>50% and <100%	62
Class 5	20%	>50% and <100%	55
Class 6	30%	>50% and <100%	53
Class 7	10%	100%	62
Class 8	20%	100%	55
Class 9	30%	100%	53

relative gap (1e-04).

On the other hand, due to the exponential number of constraints in OPMVC-DFJ and OPMVC-FT, the complete models were not implemented. Instead, violated constraints were systematically identified and added to the model as described below.

Model OPMVC-DFJ is solved by CPLEX 12.6 without the subtour elimination constraints and then, Algorithm 7 is executed to find subtours. Violated members of the subtour elimination constraints are identified if there are nodes with different labels. If that is the case, the corresponding subtour elimination constraints are added to the model which is solved again by CPLEX. This procedure is repeated until a solution without subtours is found or the algorithm reaches 3600 CPU seconds of execution.

As before, in order to solve OPMVC-FT, it is solved by CPLEX 12.6 without the connectivity constraints. Existing subtours are identified by solving a maximum flow problem from each node $i \in M$ to each node $j \in O$ on a graph whose arc

Algorithm 7 Identify violated members of the subtour elimination constraints

Require: : G' ▷ Directed graph given by the x^* and y^* values

Let N' be the set of nodes of G'
Set the nodes in N' as unlabeled
$l \leftarrow 1$
while there are unlabeled nodes in N' **do**
    Select an unlabeled node i from N'
    Find all the reachable unlabeled nodes from i in G' by means of a search algorithm and label them as l
    $l \leftarrow l + 1$
end while

capacities are given by the x^* values of the current solution, by means of the Edmonds Karp algorithm, as in Erdoğan et al. (2010). If the maximum flow is less than y_i and both i and j do not belong to the main tour, a violated constraint along the sets separated by the minimum cut has been identified. If any subtours are identified, their respective connectivity constraints are added to the model and it is solved again. The process is repeated until a solution without subtours is found or the time limit of 3600 CPU seconds is reached.

4.4.4 Experimental results

In this section, the computational tests results are reported. First, the quality of the obtained solutions is compared among the models. After that, it is presented a brief analysis of the computation time required by CPLEX 12.6 when using the proposed the models.

Detailed results obtained by CPLEX 12.6 using the models are displayed in Tables B.1–B.9.

4.4.4.1 Solutions quality

Table 4.3 displays the number and percentage of instances that were solve to optimality by CPLEX 12.6 per instance class.

Despite the fact that the subtour elimination constraints proposed by Wong (1980) and Dantzig et al. (1954) provide the strongest formulation for the TSP, models OPMVC-W and OPMVC-DFJ allowed CPLEX 12.6 to solve only 53.14% and 64.87% of the instances. In fact, the least number of instances were solved to optimality by using OPMVC-W. This is because, even though OPMVC-W has a polynomial number of constraints, its degree is equal to three, so the number of constraints increases rapidly with the number of nodes.

Nevertheless, it is remarkable that OPMVC-DFJ allowed the solver to find optimal solutions for almost all instances of Classes 2 and 3. Notice that the number of subtour elimination constraints in this model depends on the size of the cliques and that relatively few nodes will be visited in the optimal solutions of instances belonging to Classes 2 and 3 due to the high level of conflicts among nodes. Then, few violated members of the subtour elimination constraints are found when solving the model, thus allowing the algorithm to find optimal solutions within the time limit.

It is also important to highlight that despite the fact that OPMVC-DFJ and OPMVC-FT report a similar average number of optimal solutions, the coefficient of variation (defined as the ratio of the standard deviation to the mean) is evidently different. Since both formulations are equivalent, the difference is due to the algorithms used to identify the violated constraints.

Another relevant result is that even though the subtour elimination constraints proposed by Gavish and Graves (1978) are weaker than those proposed by Dantzig et al. (1954) and Wong (1980) for the TSP, they allowed the solver to find the largest number of optimal solutions for the OPMVC.

Table 4.3: Percentage of optimal solutions reported by CPLEX

Class	**OPMVC-DL**	**OPMVC-GG**	**OPMVC-W**	**OPMVC-DFJ**	**OPMVC-FT**
Class 1	62.90% (39/62)	70.97% (44/62)	58.06% (36/62)	69.35% (43/62)	62.90% (39/62)
Class 2	63.64% (35/55)	74.55% (41/55)	60.00% (33/55)	89.09% (49/55)	74.55% (41/55)
Class 3	71.70% (38/53)	75.47% (40/53)	58.49% (31/53)	98.11% (52/53)	88.68% (47/53)
Class 4	54.84% (34/62)	70.97% (44/62)	51.61% (32/62)	50.00% (31/62)	51.61% (32/62)
Class 5	52.73% (29/55)	76.36% (42/55)	50.91% (28/55)	61.82% (34/55)	61.82% (34/55)
Class 6	41.51% (22/53)	73.58% (39/53)	52.83% (28/53)	64.15% (34/53)	67.92% (36/53)
Class 7	53.23% (33/62)	64.52% (40/62)	50.00% (31/62)	41.94% (26/62)	46.77% (29/62)
Class 8	52.73% (29/55)	67.27% (37/55)	47.27% (26/55)	52.73% (29/55)	61.82% (34/55)
Class 9	43.40% (23/53)	67.92% (36/53)	49.06% (26/53)	56.60% (30/53)	56.60% (30/53)
Average percentage	55.18%	71.29%	53.14%	64.87%	63.63%
Coefficient of variation	17.50%	5.71%	8.63%	28.24%	19.64%
Total optimal solutions	282/510	363/510	271/510	328/510	322/510

Finally, the small coefficient of variation reported by OPMVC-GG suggests that CPLEX 12.6 is able to provide optimal solutions for instances of the OPMVC using this model despite the characteristics of the instances. Nevertheless, the percentage of solved instances belonging to Classes 7, 8, and 9 is slightly lower than the percentage of solved instances of the remaining classes. In fact, this behavior is similar for all the tested models. This suggests that, even that the increment of conflicts adds constraints to the models, they become easier to solve.

There are some instances for which CPLEX did not report the optimal solution, but it provided a feasible one. Then, for each instance, the best found solution by the five models was recorded. After that, the number of times in which each formulation allowed the solver to find the best solution was recorded, as reported in Table 4.4.

Note that the percentage of instances for which OPMVC-W, OPMVC-DFJ, and OPMVC-C reported the best feasible solutions is equal to the percentage of instances for which they obtained optimal solutions. This is evident for OPMVC-DFJ and OPMVC-C because the complete models were not solved; therefore, these models either report the optimal solution or an upper bound. Similar to the results shown in Table 4.3, CPLEX reported the best solutions for more instances by using OPMVC-GG with the lowest coefficient of variation.

4.4.4.2 Computation time

Table 4.5 displays the execution time in seconds required to try to solve each instance class, even considering the instances for which the optimal solution was not reported. For a better visualization of the execution time, Figure 4.1 shows a heatmap that illustrates the results reported in Table 4.5.

It is worth noticing that despite the simplicity of OPMVC-DL, it requires more computation time than the other formulations. Furthermore, within this computation time, it is able to prove the optimality of only 55.18% of the instances. On

Table 4.4: Percentage of instances in which each model allowed CPLEX to find the best known integer solution

Class	**OPMVC-DL**	**OPMVC-GG**	**OPMVC-W**	**OPMVC-DFJ**	**OPMVC-C**
Class 1	62.90% (39/62)	72.58% (45/62)	58.06% (36/62)	69.35% (43/62)	62.90% (39/62)
Class 2	63.64% (35/55)	74.55% (41/55)	60.00% (33/55)	89.09% (49/55)	74.55% (41/55)
Class 3	71.70% (38/53)	75.47% (40/53)	58.49% (31/53)	98.11% (52/53)	88.68% (47/53)
Class 4	56.45% (35/62)	70.97% (44/62)	51.61% (32/62)	50.00% (31/62)	51.61% (32/62)
Class 5	52.73% (29/55)	76.36% (42/55)	52.73% (29/55)	61.82% (34/55)	61.82% (34/55)
Class 6	43.40% (23/53)	75.47% (40/53)	52.83% (28/53)	64.15% (34/53)	67.92% (36/53)
Class 7	58.06% (36/62)	66.13% (41/62)	50.00% (31/62)	41.94% (26/62)	46.77% (29/62)
Class 8	52.73% (29/55)	69.09% (38/55)	47.27% (26/55)	52.73% (29/55)	61.82% (34/55)
Class 9	43.40% (23/53)	67.92% (36/53)	49.06% (26/53)	56.60% (30/53)	56.60% (30/53)
Average percentage	56.11%	72.06%	53.34%	64.87%	63.63%
Coefficient of variation	16.62%	5.16%	8.47%	28.24%	19.64%
Total best solutions	287/510	367/510	271/510	328/510	322/510

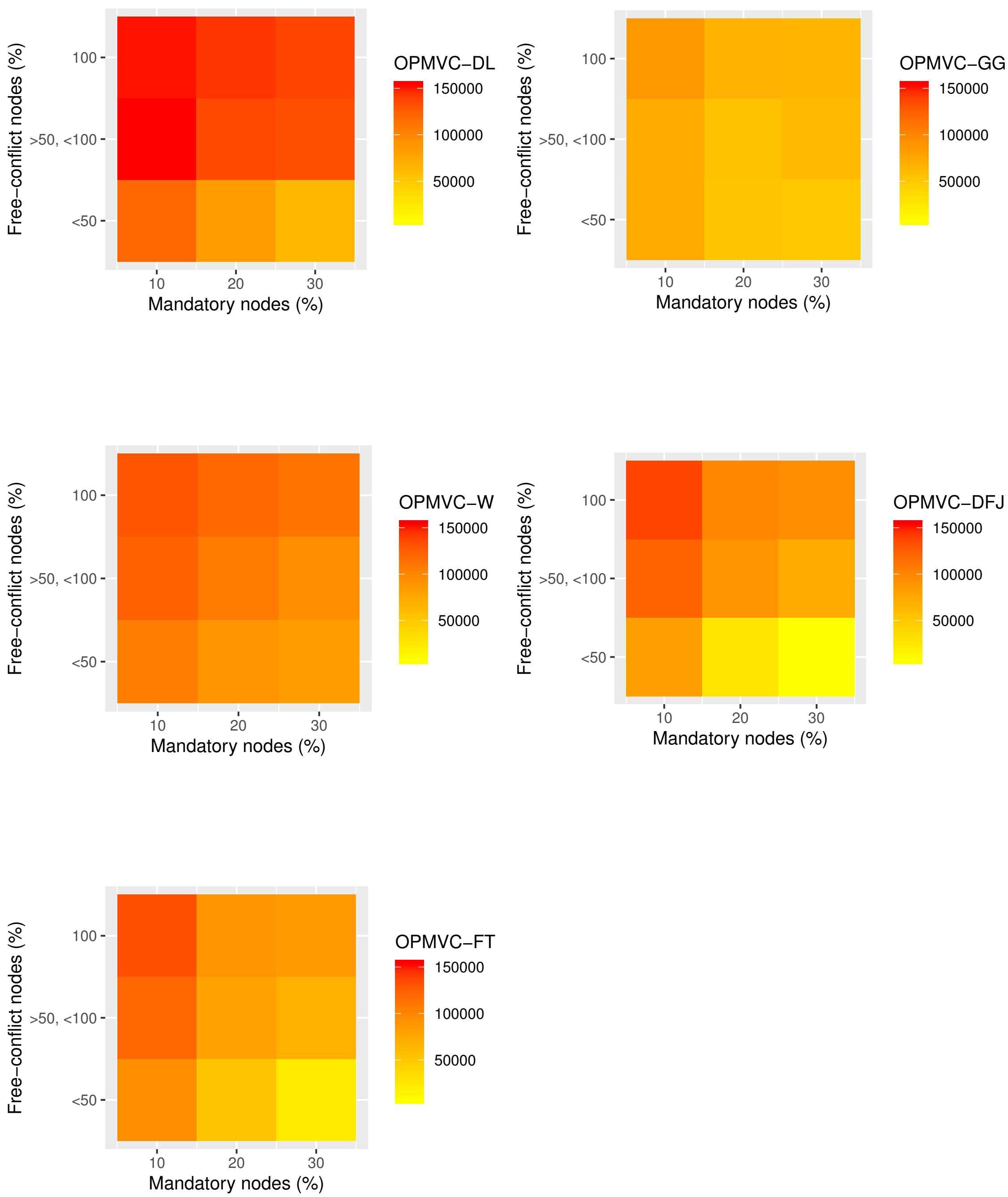

Figure 4.1: Computation time variation

Table 4.5: Execution time required to solve each instance class

Class	**OPMVC-DL**	**OPMVC-GG**	**OPMVC-W**	**OPMVC-DFJ**	**OPMVC-FT**
Class 1	120064.47	74142.79	106278.46	82201.7	95193.67
Class 2	84218.19	53650.74	91236.79	26705.55	54726.23
Class 3	63955.02	49953.04	83677.31	6782.01	23835.98
Class 4	153903.1	72498.27	124148.48	122295.5	119556.99
Class 5	135949.61	55296.94	108052.09	90589.71	81178.33
Class 6	133027.59	61920.41	93530.05	74438.2	69352.62
Class 7	152128.31	88041.26	129292.98	137733.75	132378.98
Class 8	143424.63	67656.44	119030.01	98714.28	89365.78
Class 9	138432	64859.43	112660.19	95328.69	85315.77

the other hand, the heatmap suggests that OPMVC-GG is not as affected by the variations of the tested instances as the other formulations.

The smallest computation times are observed for OPMVC-DFJ and OPMVC-FT in instances with small percentage of free-conflict nodes and high percentage of mandatory nodes. As mentioned before, the large number of incompatible nodes causes a reduction in the potential number of nodes to be included in the route, thus reducing the cliques size. As a consequence, the cut generation algorithms are able to quickly find the violated constraints and incorporate them into the model.

Finally, notice that the computation time required to solve each model seems to be affected both by the level of conflicts and the percentage of mandatory nodes. In fact, the instances in which all nodes are conflict-free and the percentage of mandatory nodes is small require more computation time to be solved, since the search space increases.

4.5 Chapter conclusions

Different formulations have been proposed in the literature to deal with subtour elimination in the TSP. In this chapter, five of the most known formulations have been adapted to model the OPMVC. Formulation OPMVC-DL uses the subtour elimination constraints proposed by Desrochers and Laporte (1991) to strengthen the ones proposed by Miller et al. (1960); formulation OPMVC-GG uses the subtour elimination constraints based on the single-commodity flow formulation proposed by Gavish and Graves (1978); formulation OPMVC-W avoids subtours by adapting the multi-commodity flow formulation proposed by Wong (1980) and later by Claus (1984); formulation OPMVC-DFJ contains clique constraints based on those introduced by Dantzig et al. (1954); finally, OPMVC-FT contains connectivity constraints adapted from those proposed by Fischetti and Toth (1988) to avoid subtours in the PCTSP.

OPMVC-DL, OPMVC-GG, and OPMVC-W contain a polynomial number of constraints and additional variables, while OPMVC-DFJ and OPMVC-FT do not require to introduce additional variables but they contain an exponential number of constraints, thus cut generation procedures were used to systematically find violated members of the constraints.

All formulations were coded and solved through CPLEX 12.6 for a set of 510 instances of the problem. Experimental results show that OPMVC-GG is able to solve more instances to optimality than the other formulations (71%, approximately) and the computation time required to try solve the instances does not seem to be significantly affected by the variations of the instances. Nevertheless, under particular configurations of the instances (high percentage of mandatory nodes and low percentage of free-conflict nodes), OPMVC-DFJ allows CPLEX to solve more instances to optimality in a shorter computation time.

The computation time required to solve all models seems to be affected by the percentage of mandatory nodes and the percentage of free-conflict nodes. Notice that both the increment of nodes that are not incompatible with any other and the decrement of mandatory nodes cause an increase on the search space, thus increasing the computation time.

Chapter 5

Conclusions and further research

This chapter contains general conclusions of the work developed in this thesis. In addition, it is described further research that would extend the results here presented.

5.1 Conclusions

Selective vehicle routing problems have been less studied than the classical Vehicle Routing Problems (VRPs) despite their practical importance due to the existence of many real-life applications in which it is not possible or necessary to provide a service to the complete set of customers. In this thesis, three rich selective VRPs, motivated by real-life situations were analyzed, modeled, and solved: the bi-objective Traveling Purchaser Problem with Deliveries (2-TPPD), the rich Team Orienteering Problem (rTOP), and the Orienteering Problem with Madatory Visits and Conflicts (OPMVC). The 2-TPPD generalizes the Traveling Purchaser Problem (TPP), while the rTOP and the OPMVC belong to the family of the Orienteering Problem (OP), which in turn belongs to the family of the VRPs with profits.

In Chapter 2, the 2-TPPD was introduced. An ϵ-constraint in combination with CPLEX 12.6 was not able to find Pareto optimal solutions for instances containing more than 10 nodes. Then, three versions of a Relinked Variable Neighborhood Search (RVNS) were proposed to solve large instances of the problem. Computational results show that the version in which the initial solution of every cycle of relinked Variable Neighborhood Searchs (VNSs) is chosen at random, provides Pareto front approximations that cover the Pareto front approximations reported by the other variants for some instances, despite requiring a larger execution time. Besides, the performance of some of the local search algorithms used in the RVNS is instance-dependent. This fact remarks the importance of using multiple local search algorithms when dealing with difficult combinatorial problems, since some of them can compensate the weaknesses of others under different configurations of the instances.

In Chapter 3, it was introduced a rich Team Orienteering Problem (TOP) that takes into account several features that have not been considered in the OP literature, such as the distribution of multiple products, the existence of a heterogeneous fleet of vehicles and soft time windows. Furthermore, to the best of our knowledge, no other VRP studied in the literature considers soft time windows in which the penalty is reflected in a waiting time rather than in the objective function. The rTOP was solved through a multi-start Adaptive Large Neighborhood Search (ALNS) which performance was experimentally proven to be better than the classical ALNS implementation. In addition, a mixed integer linear programming formulation for the problem was coded and solved through CPLEX 12.6. The multi-start ALNS provided better solutions than the feasible ones found by the solver in 186 out of 195 cases. In addition, it was found that in 50 out of 195 cases, the gap of the objective function of the solutions reported by the multi-start ALNS is, at most, 10% worse than the optimal solution.

Finally, the OPMVC was studied in Chapter 4. Five mixed integer linear programming formulations of the problem were proposed by adapting subtour elimination constraints from the Traveling Salesman Problem (TSP) literature and connectivity constraints originally proposed to avoid subtours in the Prize Collecting Traveling Salesman Problem (PCTSP). The models were coded and solved with CPLEX 12.6 over a large set of instances of the problem. The model in which subtours are eliminated by means of a single-commodity flow formulation based on the subtour elimination constraints proposed by Gavish and Graves (1978) showed to provide the largest number of optimal solutions and its performance does not seem to be affected significantly by the variations of the test instances.

5.2 Further research

This section describes further research guidelines that would improve the scope of the results obtained through the development of this work.

5.2.1 The bi-objective traveling purchaser problem with deliveries

The RVNS provides Pareto front approximations within a reasonable computation time. Nevertheless, it is not possible to assess their closeness to the Pareto fronts without computing Pareto optimal solutions. In Section 2.6, Pareto optimal solutions of large instances were not found due to the complexity of the optimization model used to solve the single-objective problems within the ϵ-constraint scheme. Then, a proposal is to model the problem using a multi-level network as done by Angel-Bello et al. (2013) to model the minimum latency problem. Another proposed research line is to solve the single-objective problems through a column generation scheme in which the subproblem is solved by a heuristic method to speed up its execution.

Additionally, considering the real-life problem that motivated the 2-TPPD, additional teams of technicians may be hired. Thus, the 2-TPPD could be extended to a multi-vehicle version.

On the other hand, it is realistic to think that new repairs may arise during the route execution. Then, a proposal is to study a dynamic version of the problem in which the customers requirements are known as time advances.

5.2.2 The rich team orienteering problem

As shown in Table 3.1, the rTOP comprises many characteristics of some TOPs variants which, in turn, generalize several OP variants. Then, further adaptations of the multi-start ALNS would allow us to find solutions for a wide range of OPs.

Another aim is to study a more realistic way to model the soft time windows penalizations. In real life, if a driver arrives after the closing of the time window but the customer still allows to perform the delivery, the driver will not know for sure the starting hour of the service, thus the waiting time becomes a stochastic variable.

5.2.3 The orienteering problem with mandatory visits and conflicts

Several exact methods that solve VRPs with profits exploit the characteristics of mathematical models. Then, further work consists of using the proposed subtour elimination constraints within exact methods to find optimal solutions for the OPMVC.

Appendix A

Detailed results for the rich team orienteering problem

Tables A.1–A.5 display the objective function value reported by each version of the multi-start Adaptive Large Neighborhood Search (ALNS), as well as the best value of the objective function reported by all of them, per instance.

Table A.1: Objective function values reported by each version of the multi-start ALNS for instances of class 1

Instance	**mALNS (1, 100, 4900)**	**mALNS (12, 100, 3800)**	**mALNS (25, 100, 2500)**	**mALNS (37, 100, 1300)**	**mALNS (1, 100, 9900)**	**mALNS (25, 100, 7500)**	**mALNS (50, 100, 5000)**	**mALNS (75, 100, 2500)**	**Best**
Cordeau_pr01	536.279	534	557.776	564	553.986	566	585.712	577.176	585.712
Cordeau_pr02	1137	1151	1145	1127	1157	1158	1112	1147	1158
Cordeau_pr03	1590.67	1616.1	1525.65	1583.94	1534.56	1577.15	1570.65	1517.65	1616.1
Cordeau_pr04	2114.95	2098.26	2070.15	2076.23	2017.33	2068.31	2073.95	2069.74	2114.95
Cordeau_pr05	2841.95	2876.78	2826.92	2778.05	2815.52	2927.51	2869.33	2878.95	2927.51
Cordeau_pr06	3349.22	3495.6	3476.55	3466.22	3466.87	3319.02	3526.73	3491	3526.73
Cordeau_pr07	810.164	861.672	834.129	803.825	844.657	801.204	816.754	836.057	861.672
Cordeau_pr08	1769.45	1796.86	1861.69	1738.08	1775.6	1820.8	1749.93	1799.88	1861.69
Cordeau_pr09	2698.08	2629.32	2651.42	2631.91	2675.61	2724.33	2625.66	2673.6	2724.33
Cordeau_pr10	3527.35	3703.94	3679.63	3602.36	3565.88	3636.52	3648.18	3678.95	3703.94
Solomon_c101	1429.08	1413.05	1430.38	1421.22	1426.37	1453.56	1498.28	1470.37	1498.28
Solomon_c102	1596.75	1585.76	1578.59	1577.93	1568.72	1600.02	1613.59	1638.19	1638.19
Solomon_c103	1609.05	1600.74	1610.31	1620.03	1593.13	1633.92	1602.83	1588.8	1633.92

Continues on next page

Continued from previous page

Instance	mALNS (1, 100, 4900)	mALNS (12, 100, 3800)	mALNS (25, 100, 2500)	mALNS (37, 100, 1300)	mALNS (1, 100, 9900)	mALNS (25, 100, 7500)	mALNS (50, 100, 5000)	mALNS (75, 100, 2500)	Best
Solomon_c104	1470.18	1505.68	1496.75	1474.81	1500	1479.89	1513.4	1484	1513.4
Solomon_c105	1419.4	1435.99	1423.78	1413.04	1453.61	1425.44	1401.37	1484.16	1484.16
Solomon_c106	1537.29	1608.31	1588.08	1578.09	1597.95	1591.18	1596.02	1572.98	1608.31
Solomon_c107	1643.45	1611.72	1644.04	1647.94	1615.58	1652.44	1631.9	1629.28	1652.44
Solomon_c108	1603.37	1613.81	1622.17	1632.86	1581.45	1601.51	1646.93	1666.99	1666.99
Solomon_c109	1559.78	1585	1552.74	1511.35	1610.84	1541.09	1552.53	1580.91	1610.84
Solomon_r101	1257.89	1191.34	1244.69	1263.99	1265.24	1252.94	1221.73	1218.04	1265.24
Solomon_r102	1397.05	1361.21	1351.18	1363	1380.71	1360.23	1367.07	1390.91	1397.05
Solomon_r103	1376.54	1316.79	1354.43	1369.89	1357.22	1370.62	1358.86	1385.87	1385.87
Solomon_r104	1130.01	1185.79	1154.83	1125.42	1156.87	1174.6	1167.99	1210.07	1210.07
Solomon_r105	1144.33	1223.31	1204.82	1143.84	1186.63	1206.05	1179.26	1213.78	1223.31
Solomon_r106	1349.42	1280.3	1351.27	1307.92	1344.46	1306.79	1344.42	1329.37	1351.27
Solomon_r107	1285.56	1306	1255.4	1252	1315.83	1266.28	1258.36	1263.09	1315.83
Solomon_r108	1157.91	1172.31	1097.91	1156.27	1134.37	1150.49	1173.68	1169.83	1173.68
Solomon_r109	1282.73	1314.71	1343	1291	1283.77	1349.37	1312.98	1295.99	1349.37
Solomon_r110	1260.93	1283.28	1250.88	1220.97	1264.53	1198.59	1250.18	1249.5	1283.28
Solomon_r111	1184.92	1181.44	1167.54	1201.02	1184.5	1232.74	1227.42	1195.8	1232.74
Solomon_r112	1291.84	1273.97	1315	1278	1321	1321	1306	1310.29	1321
Solomon_rc101	1331.51	1435.97	1442.85	1396.07	1380.42	1392.55	1401.09	1407.42	1442.85
Solomon_rc102	1525.15	1474.74	1465.42	1483.55	1535.18	1459	1564.49	1508.71	1564.49
Solomon_rc103	1480.31	1522.13	1464.52	1513.94	1558.97	1530.15	1544.52	1544.21	1558.97
Solomon_rc104	1364.47	1367.4	1324.44	1399.47	1325.49	1375.85	1407.02	1450.13	1450.13
Solomon_rc105	1574.66	1557.84	1503.84	1508.06	1569.7	1546.27	1530.46	1530.87	1574.66
Solomon_rc106	1444.51	1462.23	1438.88	1400.82	1356.31	1378.96	1477.61	1456.86	1477.61
Solomon_rc107	1480	1442.06	1434.32	1487.97	1541.72	1472.15	1503.57	1482.65	1541.72
Solomon_rc108	1406.28	1400.49	1352.78	1349.46	1309.02	1426.3	1401.59	1390.32	1426.3

Table A.2: Objective function values reported by each version of the multi-start ALNS for instances of class 2

Instance	mALNS (1, 100, 4900)	mALNS (12, 100, 3800)	mALNS (25, 100, 2500)	mALNS (37, 100, 1300)	mALNS (1, 100, 9900)	mALNS (25, 100, 7500)	mALNS (50, 100, 5000)	mALNS (75, 100, 2500)	Best
Cordeau_pr01	538.774	555.472	544.649	556.469	555.952	551.97	559.009	553.777	559.009
Cordeau_pr02	1060	1055.68	1058.23	1054.46	1050.04	1066.06	1044.89	1042.46	1066.06
Cordeau_pr03	1469.44	1518.38	1545.77	1496.03	1546.16	1560.2	1512.89	1557.51	1560.2

Continues on next page

Continued from previous page

Instance	mALNS (1, 100, 4900)	mALNS (12, 100, 3800)	mALNS (25, 100, 2500)	mALNS (37, 100, 1300)	mALNS (1, 100, 9900)	mALNS (25, 100, 7500)	mALNS (50, 100, 5000)	mALNS (75, 100, 2500)	Best
Cordeau_pr04	2024.49	1984.52	1882.33	1954.25	1955.45	2002.71	1985.51	1991.4	2024.49
Cordeau_pr05	2814.01	2846.15	2799.95	2762.13	2704.03	2864.42	2799.54	2778.94	2864.42
Cordeau_pr06	3099.88	3061.88	3103.13	3095.63	2988.17	3122.6	3138.04	3116.8	3138.04
Cordeau_pr07	778.838	778.4	792.374	769.277	792.803	786.927	798.732	780.074	798.732
Cordeau_pr08	1659.16	1706.65	1733.9	1692.13	1680.16	1693.8	1698.62	1690.79	1733.9
Cordeau_pr09	2401.02	2400.85	2417.36	2411.26	2343.55	2393.96	2431.64	2384.32	2431.64
Cordeau_pr10	3231	3338.22	3346.21	3340.46	3261.24	3364.43	3360.42	3426	3426
Solomon_c101	1361.1	1373.82	1411.93	1360.27	1387.92	1390.04	1429.02	1414.98	1429.02
Solomon_c102	1500.81	1498.52	1546.32	1559.98	1507.91	1561.53	1533.84	1528.66	1561.53
Solomon_c103	1479.27	1488.16	1505.92	1479.31	1459.51	1506.77	1532.16	1523.85	1532.16
Solomon_c104	1434.44	1485.98	1466.14	1454.8	1460.77	1450.84	1465.88	1452.71	1485.98
Solomon_c105	1343.1	1413.63	1452.36	1415.99	1413.53	1429.26	1382.3	1414.15	1452.36
Solomon_c106	1509.83	1511.05	1525.87	1506.64	1531.96	1538.19	1537.95	1535.01	1538.19
Solomon_c107	1507.99	1534.67	1545.83	1568.34	1549.84	1558.6	1566.66	1533.8	1568.34
Solomon_c108	1560.53	1525.65	1557.1	1529.24	1579.05	1552.65	1599.45	1592.2	1599.45
Solomon_c109	1467.6	1488.12	1507.48	1508.83	1544.45	1535.88	1525.8	1508.03	1544.45
Solomon_r101	1185.95	1223.52	1211.15	1211.26	1190.17	1176.43	1183.99	1168.88	1223.52
Solomon_r102	1266.64	1294.33	1287.01	1310.81	1290.95	1317.12	1314.08	1304.83	1317.12
Solomon_r103	1228.72	1258.56	1251.21	1250.84	1223.14	1235.19	1214.26	1242.91	1258.56
Solomon_r104	1124.97	1128.45	1103.4	1087.52	1119.79	1098.91	1147.19	1122.35	1147.19
Solomon_r105	1158.45	1174.77	1149.32	1139.58	1141.56	1191.24	1155.08	1180.29	1191.24
Solomon_r106	1253.05	1252.3	1276.63	1263	1287.69	1311.68	1253.39	1262.32	1311.68
Solomon_r107	1217.72	1192.86	1211.27	1223.14	1192.5	1209.35	1226.2	1231.97	1231.97
Solomon_r108	1097.36	1071.03	1148.93	1121.41	1087.63	1135.62	1089.27	1159.55	1159.55
Solomon_r109	1180.59	1233	1235.07	1209.93	1234	1230.01	1225.63	1221.06	1235.07
Solomon_r110	1175.13	1142.09	1186.73	1180.25	1197.54	1217.28	1152.85	1209.56	1217.28
Solomon_r111	1165.71	1135.72	1163.8	1153.21	1102.1	1122.94	1140.66	1153.85	1165.71
Solomon_r112	1195.12	1232.01	1191.58	1218.98	1205.51	1221.77	1216.21	1232.07	1232.07
Solomon_rc101	1355.55	1408.9	1357.52	1372.95	1355.23	1367.73	1416.65	1358.76	1416.65
Solomon_rc102	1431.66	1437.4	1474.46	1460.9	1487.73	1469.76	1489.69	1456.67	1489.69
Solomon_rc103	1363.26	1430.64	1397.49	1397.55	1390.02	1450.32	1398.17	1477.74	1477.74
Solomon_rc104	1370.25	1368.92	1338.26	1345.64	1323.4	1382.26	1406.25	1376.32	1406.25
Solomon_rc105	1496.67	1498.24	1518.91	1504	1498.63	1541.77	1527.24	1503.03	1541.77
Solomon_rc106	1365.3	1390.32	1341.49	1413.45	1386.2	1391.35	1402.04	1352.16	1413.45
Solomon_rc107	1421.69	1359.81	1379.95	1395.48	1420.93	1394.97	1441.84	1389.16	1441.84
Solomon_rc108	1344.48	1386.15	1362.14	1397.14	1370.02	1360.56	1382.63	1354.8	1397.14

Table A.3: Objective function values reported by each version of the multi-start ALNS for instances of class 3

Instance	mALNS (1, 100, 4900)	mALNS (12, 100, 3800)	mALNS (25, 100, 2500)	mALNS (37, 100, 1300)	mALNS (1, 100, 9900)	mALNS (25, 100, 7500)	mALNS (50, 100, 5000)	mALNS (75, 100, 2500)	Best
Cordeau_pr01	556.263	549.554	543.873	548.313	540.547	566.695	554.54	557.075	566.695
Cordeau_pr02	1038.35	1046.43	1054.1	1035.23	1054.12	1043.41	1043.78	1051.97	1054.12
Cordeau_pr03	1517.64	1476.47	1475.35	1514.15	1457.32	1508.1	1496.23	1539.31	1539.31
Cordeau_pr04	1961.18	1942.19	1914.85	1902.39	2007.98	1984.24	1990.19	1969.31	2007.98
Cordeau_pr05	2737.92	2694.87	2765.5	2676.55	2641.03	2696.79	2832.63	2776.86	2832.63
Cordeau_pr06	3053.29	2980.88	3039.19	3061.17	3080.78	3119.62	2993.71	3079.94	3119.62
Cordeau_pr07	770.505	773.227	780.689	780.875	798.257	785.667	796.753	788.893	798.257
Cordeau_pr08	1636.79	1671.32	1681.62	1663.81	1667.93	1721.4	1660.44	1643.9	1721.4
Cordeau_pr09	2384.15	2376.34	2366.73	2357.44	2315.5	2370.92	2391.83	2354.09	2391.83
Cordeau_pr10	3293.59	3332.65	3283.12	3277.21	3336.32	3345.16	3383.15	3295.37	3383.15
Solomon_c101	1374.94	1407.79	1401.37	1374.21	1381.17	1433.07	1399.58	1398.59	1433.07
Solomon_c102	1535.04	1531.73	1523.07	1522.34	1532.82	1555.46	1543.14	1508.49	1555.46
Solomon_c103	1464.16	1490.5	1511.65	1483.29	1483.63	1491.75	1473.12	1502.64	1511.65
Solomon_c104	1449.28	1422.9	1486.91	1442.15	1441.5	1479.84	1469.06	1450.84	1486.91
Solomon_c105	1338.01	1382.14	1403.71	1407.48	1405.22	1392.99	1398.4	1411.87	1411.87
Solomon_c106	1500.35	1507.69	1483.64	1485.66	1506.06	1528.06	1519.99	1529.85	1529.85
Solomon_c107	1502.87	1554.52	1543.96	1524.52	1515.49	1552.01	1530.73	1549.53	1554.52
Solomon_c108	1539.49	1559.21	1521.32	1572.42	1552.23	1567.96	1577.62	1560.54	1577.62
Solomon_c109	1492.98	1510.9	1501.74	1475.71	1487.99	1515.79	1536.87	1523.01	1536.87
Solomon_r101	1192.72	1179.24	1182.03	1211.6	1186.53	1201.62	1213.71	1190.51	1213.71
Solomon_r102	1258.7	1257.45	1302.34	1266.49	1274.36	1301.7	1303.7	1281.05	1303.7
Solomon_r103	1203.14	1202.39	1222.1	1207.6	1197.82	1248.4	1257.79	1229.46	1257.79
Solomon_r104	1126.6	1088.66	1091.52	1105.52	1123.37	1127.61	1095.14	1110.04	1127.61
Solomon_r105	1183.34	1168.44	1141.69	1198.73	1173.88	1160.62	1183.67	1155.09	1198.73
Solomon_r106	1256.63	1264.85	1255.28	1245.77	1259.78	1270.43	1301.9	1288.02	1301.9
Solomon_r107	1200.56	1183.95	1151.65	1178.6	1204.21	1209	1167.97	1179.71	1209
Solomon_r108	1115.79	1109.75	1119.59	1143.93	1111.66	1083.67	1121.1	1121.33	1143.93
Solomon_r109	1178.77	1193.21	1194.95	1193.96	1212.99	1213.45	1168.44	1220.13	1220.13
Solomon_r110	1174.06	1183.82	1161.05	1195.63	1192.91	1201.27	1147.76	1176.18	1201.27
Solomon_r111	1115.36	1135.87	1156.65	1157.39	1138.07	1170.84	1146.87	1137.6	1170.84
Solomon_r112	1163.28	1223	1207.14	1193.67	1204.54	1204.17	1193.62	1225.23	1225.23
Solomon_rc101	1344.27	1369.09	1377.42	1352.21	1379.61	1350.87	1371.75	1365.55	1379.61
Solomon_rc102	1413.62	1460.69	1429.93	1425.66	1429.26	1482.94	1444.92	1470.47	1482.94
Solomon_rc103	1413.14	1399.79	1382.76	1428.33	1368.39	1405.13	1405.91	1381.86	1428.33
Solomon_rc104	1345.95	1351.12	1329.75	1365.72	1333.3	1421.88	1385.78	1329.29	1421.88
Solomon_rc105	1447.76	1444.11	1497.16	1490.66	1441.95	1524.75	1516.05	1530.75	1530.75

Continues on next page

Continued from previous page

Instance	mALNS (1, 100, 4900)	mALNS (12, 100, 3800)	mALNS (25, 100, 2500)	mALNS (37, 100, 1300)	mALNS (1, 100, 9900)	mALNS (25, 100, 7500)	mALNS (50, 100, 5000)	mALNS (75, 100, 2500)	Best
Solomon_rc106	1343.7	1355.1	1331.3	1358.58	1365.44	1401.34	1330.72	1389.17	1401.34
Solomon_rc107	1369.07	1383.74	1352.89	1347.65	1347.59	1367.53	1389.3	1408.18	1408.18
Solomon_rc108	1358.01	1360.26	1323.66	1312.53	1360.97	1337.7	1363.82	1342.02	1363.82

Table A.4: Objective function values reported by each version of the multi-start ALNS for instances of class 4

Instance	mALNS (1, 100, 4900)	mALNS (12, 100, 3800)	mALNS (25, 100, 2500)	mALNS (37, 100, 1300)	mALNS (1, 100, 9900)	mALNS (25, 100, 7500)	mALNS (50, 100, 5000)	mALNS (75, 100, 2500)	Best
Cordeau_pr01	516.683	517.013	520.227	521.499	512.636	517.009	522.96	523.19	523.19
Cordeau_pr02	942.388	921.395	937.427	957.478	912.279	943.756	938.324	944.5	957.478
Cordeau_pr03	1310.6	1316.57	1310.73	1328.39	1322.23	1313.08	1337.36	1331.35	1337.36
Cordeau_pr04	1751.67	1754.87	1760.65	1740	1734.66	1796.86	1809.21	1766.36	1809.21
Cordeau_pr05	2330.51	2371.39	2249.45	2307.41	2232.86	2338.32	2270.41	2336.12	2371.39
Cordeau_pr06	2649.86	2703.49	2697.55	2655.74	2611.73	2726.83	2666.35	2685.99	2726.83
Cordeau_pr07	684.016	696.952	697.775	696.411	703.1	703.492	699.736	699.083	703.492
Cordeau_pr08	1461.77	1488.79	1453.31	1463.11	1498.83	1499.35	1498.94	1496.08	1499.35
Cordeau_pr09	2009.65	2027.49	2021.66	1997.47	2017.67	2016.74	2030.63	2017.8	2030.63
Cordeau_pr10	2805.94	2780.87	2781.68	2785.7	2811.45	2854.95	2821.58	2829.95	2854.95
Solomon_c101	1201.7	1213.26	1205.26	1224.37	1193.11	1210.47	1266.07	1238.42	1266.07
Solomon_c102	1309.53	1343.33	1335.51	1336.47	1319.38	1336.89	1347.15	1352.45	1352.45
Solomon_c103	1326.53	1317.34	1362.78	1361.89	1326.32	1381.89	1370	1333.92	1381.89
Solomon_c104	1265.5	1272.72	1276.31	1274.48	1291.48	1290.93	1284.3	1299.1	1299.1
Solomon_c105	1221.23	1234.6	1245.98	1226.11	1189.56	1211.59	1207.51	1239.44	1245.98
Solomon_c106	1210.57	1263.57	1273.63	1251.41	1226.53	1242.16	1264.49	1245.99	1273.63
Solomon_c107	1258.67	1237.34	1265.56	1275.63	1281.01	1290.67	1262.74	1241.73	1290.67
Solomon_c108	1315.99	1345.15	1350.69	1386.45	1310.94	1372.77	1385.55	1370.1	1386.45
Solomon_c109	1246.44	1229.37	1254.8	1262.34	1267.81	1272.8	1260.89	1261.23	1272.8
Solomon_r101	986.118	1039.87	1016.77	1036.79	963.239	1070.71	1047.42	1017.01	1070.71
Solomon_r102	1113.14	1123.63	1118.04	1111.04	1110.23	1144.21	1131.48	1122.12	1144.21
Solomon_r103	1060.58	1097.8	1076.31	1090.31	1095.28	1093.36	1092.21	1103.8	1103.8
Solomon_r104	1029.02	1047.14	1076.08	1038.7	1035.95	1060.22	1084.95	1055.15	1084.95
Solomon_r105	1074.43	1108.57	1121.92	1132.47	1097.7	1107.71	1097.9	1107.98	1132.47
Solomon_r106	1053.76	1018.74	1084.58	1060.89	1047.61	1074.1	1114.72	1059.93	1114.72
Solomon_r107	1079.48	1115.72	1098.38	1116.41	1060.79	1101.63	1129.19	1124.28	1129.19

Continues on next page

Continued from previous page

Instance	mALNS (1, 100, 4900)	mALNS (12, 100, 3800)	mALNS (25, 100, 2500)	mALNS (37, 100, 1300)	mALNS (1, 100, 9900)	mALNS (25, 100, 7500)	mALNS (50, 100, 5000)	mALNS (75, 100, 2500)	Best
Solomon_r108	986.918	1004.48	1009.07	1033.77	955.677	1056.19	1048.17	1008.74	1056.19
Solomon_r109	1072.77	1070.29	1059.58	1061.07	1038.05	1088.76	1070.91	1077.83	1088.76
Solomon_r110	1073.24	1078.97	1056.78	1061.34	1098.54	1088.28	1074.53	1074.77	1098.54
Solomon_r111	1079.29	1082.45	1088.79	1060.43	1065.09	1101.79	1112.2	1109.22	1112.2
Solomon_r112	1016	1028.05	1041.57	997.383	1036.7	1055.49	1019.83	1018.51	1055.49
Solomon_rc101	1243.85	1279.19	1274.46	1263.01	1241.52	1274.85	1275.91	1255.14	1279.19
Solomon_rc102	1205.59	1257.58	1236.66	1227.98	1218.8	1253.25	1254.47	1274.06	1274.06
Solomon_rc103	1284.5	1291.68	1244.83	1240.61	1277.94	1272.43	1286.58	1265.81	1291.68
Solomon_rc104	1229.07	1254.35	1263.07	1273.72	1239.76	1252.43	1278.41	1240.78	1278.41
Solomon_rc105	1228.95	1234.1	1268.8	1210.75	1169.51	1261.37	1229.45	1248.72	1268.8
Solomon_rc106	1140.51	1154.3	1168.52	1161.93	1154.45	1185.54	1189.47	1151.6	1189.47
Solomon_rc107	1306.44	1338	1332.37	1306.7	1275.22	1316.67	1342.97	1323.06	1342.97
Solomon_rc108	1256.61	1178.59	1194.25	1200.83	1162.79	1198.53	1198.11	1213.35	1256.61

Table A.5: Objective function values reported by each version of the multi-start ALNS for instances of class 5

Instance	mALNS (1, 100, 4900)	mALNS (12, 100, 3800)	mALNS (25, 100, 2500)	mALNS (37, 100, 1300)	mALNS (1, 100, 9900)	mALNS (25, 100, 7500)	mALNS (50, 100, 5000)	mALNS (75, 100, 2500)	Best
Cordeau_pr01	501.399	519.364	508.129	508.549	511.101	517.312	510.28	506.538	519.364
Cordeau_pr02	921.312	920.842	916.22	908.363	905.696	926.616	928.955	924.735	928.955
Cordeau_pr03	1295.55	1313.16	1310.66	1313.76	1289.17	1295.55	1315.37	1326.49	1326.49
Cordeau_pr04	1718.98	1720.07	1742.14	1728.16	1710.18	1746.34	1717.92	1723.54	1746.34
Cordeau_pr05	2209.05	2236.52	2225.83	2234.73	2180.22	2257.88	2234.49	2235.29	2257.88
Cordeau_pr06	2644.5	2661.87	2634.4	2637	2617.59	2645.14	2654.93	2650.22	2661.87
Cordeau_pr07	689.49	685.143	683.181	687.927	695.042	689.882	700.528	685.444	700.528
Cordeau_pr08	1380.72	1419.49	1440.62	1423.64	1432.06	1440.81	1434.21	1438.69	1440.81
Cordeau_pr09	1939.52	1975.17	1985.25	1992.57	1955.33	2008.78	1999.44	1969.69	2008.78
Cordeau_pr10	2746.51	2747.79	2738.38	2736.47	2763.58	2741.09	2735.3	2749.99	2763.58
Solomon_c101	1153.64	1189.4	1183.98	1190.85	1162.91	1216.6	1232.09	1220.13	1232.09
Solomon_c102	1314.86	1303.15	1308.06	1309.61	1273.29	1325.04	1309.92	1305.58	1325.04
Solomon_c103	1319.25	1292.76	1306.19	1319.67	1331.05	1338.31	1331.3	1325.27	1338.31
Solomon_c104	1222.64	1219.06	1248.65	1230.9	1259.67	1263.89	1242.59	1235.96	1263.89
Solomon_c105	1182.77	1165.79	1207.84	1181.93	1192.53	1211.38	1196.63	1164.08	1211.38
Solomon_c106	1168.81	1186.53	1213.07	1169.23	1165.37	1193.73	1207.79	1172.17	1213.07

Continues on next page

Continued from previous page

Instance	mALNS (1, 100, 4900)	mALNS (12, 100, 3800)	mALNS (25, 100, 2500)	mALNS (37, 100, 1300)	mALNS (1, 100, 9900)	mALNS (25, 100, 7500)	mALNS (50, 100, 5000)	mALNS (75, 100, 2500)	Best
Solomon_c107	1166.1	1219.56	1183.57	1203.21	1205.56	1210.03	1239.13	1205.35	1239.13
Solomon_c108	1315.08	1367.11	1318.19	1301.59	1291.22	1355.62	1328.21	1351	1367.11
Solomon_c109	1190.08	1218.16	1225.43	1206.5	1210.57	1248.3	1237.69	1220.92	1248.3
Solomon_r101	1001.41	990.438	1010.43	973.739	958.018	1022.03	1006	997.275	1022.03
Solomon_r102	1104.03	1101.15	1114.85	1096.6	1098.43	1106.75	1104.1	1104.53	1114.85
Solomon_r103	1070.26	1067.63	1064.95	1070.74	1056.02	1067.6	1076.04	1077.08	1077.08
Solomon_r104	1040.81	1047.55	1039.24	1004.39	1019.57	1057.91	1020.52	1045.58	1057.91
Solomon_r105	1101.99	1080.15	1085.07	1056.62	1092.28	1108.28	1103.65	1103.23	1108.28
Solomon_r106	1022.76	1047.85	1054.11	1013.31	1041.37	1037.27	1062.48	1051.55	1062.48
Solomon_r107	1071.42	1098.49	1106.7	1085.24	1065.77	1075.68	1117.84	1099.38	1117.84
Solomon_r108	1031.35	999.736	1005.21	986.225	998.489	1011.38	997.476	1004.52	1031.35
Solomon_r109	1031.55	1057.97	1062.13	1058.72	1061.8	1059.71	1054.21	1051.7	1062.13
Solomon_r110	1017.33	1036.82	1037.82	1024.86	1060.91	1075.21	1078.78	1043.18	1078.78
Solomon_r111	1071.45	1034.19	1037.67	1077.05	1074.33	1084.32	1066.26	1060.35	1084.32
Solomon_r112	979.542	984.377	973.441	993.47	976.513	1010.34	996.88	988.33	1010.34
Solomon_rc101	1253.66	1267.22	1276.96	1259.11	1222.6	1232.57	1279.65	1249.24	1279.65
Solomon_rc102	1179.84	1195.58	1211.2	1215.12	1194.62	1212.19	1175.16	1230.09	1230.09
Solomon_rc103	1188.61	1217.07	1236.76	1242.25	1220.29	1249.58	1250.96	1241.96	1250.96
Solomon_rc104	1227.42	1239.75	1250.78	1217.96	1245.9	1221.47	1248.62	1217.4	1250.78
Solomon_rc105	1203.09	1223.17	1224.85	1223.13	1189.26	1218.56	1217.16	1219.98	1224.85
Solomon_rc106	1136.76	1147.05	1141.28	1118.93	1140.37	1139.74	1140.98	1160.66	1160.66
Solomon_rc107	1267.4	1282.03	1272.94	1306.02	1314.63	1305.82	1293.52	1283.03	1314.63
Solomon_rc108	1160.39	1178.11	1197.18	1199.33	1200.06	1204.09	1153.67	1188.02	1204.09

Tables A.6–A.10 display the relative gap in percentage of objective function value reported by each version of the multi-start ALNS with respect to the best one, per instance.

Table A.6: Relative gap of the objective function value reported by each version of the multi-start ALNS with respect to the best one for Class 1

Instance	mALNS (1, 100, 4900)	mALNS (12, 100, 3800)	mALNS (25, 100, 2500)	mALNS (37, 100, 1300)	mALNS (1, 100, 9900)	mALNS (25, 100, 7500)	mALNS (50, 100, 5000)	mALNS (75, 100, 2500)
Cordeau_pr01	8.44	8.83	4.77	3.71	5.42	3.37	0.00	1.46
Cordeau_pr02	1.81	0.60	1.12	2.68	0.09	0.00	3.97	0.95
Cordeau_pr03	1.57	0.00	5.60	1.99	5.05	2.41	2.81	6.09
Cordeau_pr04	0.00	0.79	2.12	1.83	4.62	2.21	1.94	2.14
Cordeau_pr05	2.92	1.73	3.44	5.11	3.83	0.00	1.99	1.66
Cordeau_pr06	5.03	0.88	1.42	1.72	1.70	5.89	0.00	1.01
Cordeau_pr07	5.98	0.00	3.20	6.71	1.97	7.02	5.21	2.97
Cordeau_pr08	4.95	3.48	0.00	6.64	4.62	2.20	6.00	3.32
Cordeau_pr09	0.96	3.49	2.68	3.39	1.79	0.00	3.62	1.86
Cordeau_pr10	4.77	0.00	0.66	2.74	3.73	1.82	1.51	0.67
Solomon_c101	4.62	5.69	4.53	5.14	4.80	2.98	0.00	1.86
Solomon_c102	2.53	3.20	3.64	3.68	4.24	2.33	1.50	0.00
Solomon_c103	1.52	2.03	1.44	0.85	2.50	0.00	1.90	2.76
Solomon_c104	2.86	0.51	1.10	2.55	0.89	2.21	0.00	1.94
Solomon_c105	4.36	3.25	4.07	4.79	2.06	3.96	5.58	0.00
Solomon_c106	4.42	0.00	1.26	1.88	0.64	1.07	0.76	2.20
Solomon_c107	0.54	2.46	0.51	0.27	2.23	0.00	1.24	1.40
Solomon_c108	3.82	3.19	2.69	2.05	5.13	3.93	1.20	0.00
Solomon_c109	3.17	1.60	3.61	6.18	0.00	4.33	3.62	1.86
Solomon_r101	0.58	5.84	1.62	0.10	0.00	0.97	3.44	3.73
Solomon_r102	0.00	2.57	3.28	2.44	1.17	2.64	2.15	0.44
Solomon_r103	0.67	4.98	2.27	1.15	2.07	1.10	1.95	0.00
Solomon_r104	6.62	2.01	4.57	7.00	4.40	2.93	3.48	0.00
Solomon_r105	6.46	0.00	1.51	6.50	3.00	1.41	3.60	0.78
Solomon_r106	0.14	5.25	0.00	3.21	0.50	3.29	0.51	1.62
Solomon_r107	2.30	0.75	4.59	4.85	0.00	3.77	4.37	4.01
Solomon_r108	1.34	0.12	6.46	1.48	3.35	1.98	0.00	0.33
Solomon_r109	4.94	2.57	0.47	4.33	4.86	0.00	2.70	3.96
Solomon_r110	1.74	0.00	2.52	4.86	1.46	6.60	2.58	2.63
Solomon_r111	3.88	4.16	5.29	2.57	3.91	0.00	0.43	3.00
Solomon_r112	2.21	3.56	0.45	3.26	0.00	0.00	1.14	0.81
Solomon_rc101	7.72	0.48	0.00	3.24	4.33	3.49	2.89	2.46
Solomon_rc102	2.51	5.74	6.33	5.17	1.87	6.74	0.00	3.57
Solomon_rc103	5.05	2.36	6.06	2.89	0.00	1.85	0.93	0.95
Solomon_rc104	5.91	5.71	8.67	3.49	8.60	5.12	2.97	0.00
Solomon_rc105	0.00	1.07	4.50	4.23	0.31	1.80	2.81	2.78

Continues on next page

Continued from previous page

Instance	**mALNS (1, 100, 4900)**	**mALNS (12, 100, 3800)**	**mALNS (25, 100, 2500)**	**mALNS (37, 100, 1300)**	**mALNS (1, 100, 9900)**	**mALNS (25, 100, 7500)**	**mALNS (50, 100, 5000)**	**mALNS (75, 100, 2500)**
Solomon_rc106	2.24	1.04	2.62	5.20	8.21	6.68	0.00	1.40
Solomon_rc107	4.00	6.46	6.97	3.49	0.00	4.51	2.47	3.83
Solomon_rc108	1.40	1.81	5.15	5.39	8.22	0.00	1.73	2.52
Average	3.18	2.52	3.11	3.56	2.86	2.58	2.13	1.87

Table A.7: Relative gap of the objective function value reported by each version of the multi-start ALNS with respect to the best one for Class 2

Instance	**mALNS (1, 100, 4900)**	**mALNS (12, 100, 3800)**	**mALNS (25, 100, 2500)**	**mALNS (37, 100, 1300)**	**mALNS (1, 100, 9900)**	**mALNS (25, 100, 7500)**	**mALNS (50, 100, 5000)**	**mALNS (75, 100, 2500)**
Cordeau_pr01	3.62	0.63	2.57	0.45	0.55	1.26	0.00	0.94
Cordeau_pr02	0.57	0.97	0.73	1.09	1.50	0.00	1.99	2.21
Cordeau_pr03	5.82	2.68	0.92	4.11	0.90	0.00	3.03	0.17
Cordeau_pr04	0.00	1.97	7.02	3.47	3.41	1.08	1.93	1.63
Cordeau_pr05	1.76	0.64	2.25	3.57	5.60	0.00	2.27	2.98
Cordeau_pr06	1.22	2.43	1.11	1.35	4.78	0.49	0.00	0.68
Cordeau_pr07	2.49	2.55	0.80	3.69	0.74	1.48	0.00	2.34
Cordeau_pr08	4.31	1.57	0.00	2.41	3.10	2.31	2.03	2.49
Cordeau_pr09	1.26	1.27	0.59	0.84	3.62	1.55	0.00	1.95
Cordeau_pr10	5.69	2.56	2.33	2.50	4.81	1.80	1.91	0.00
Solomon_c101	4.75	3.86	1.20	4.81	2.88	2.73	0.00	0.98
Solomon_c102	3.89	4.04	0.97	0.10	3.43	0.00	1.77	2.10
Solomon_c103	3.45	2.87	1.71	3.45	4.74	1.66	0.00	0.54
Solomon_c104	3.47	0.00	1.34	2.10	1.70	2.36	1.35	2.24
Solomon_c105	7.52	2.67	0.00	2.50	2.67	1.59	4.82	2.63
Solomon_c106	1.84	1.76	0.80	2.05	0.41	0.00	0.02	0.21
Solomon_c107	3.85	2.15	1.44	0.00	1.18	0.62	0.11	2.20
Solomon_c108	2.43	4.61	2.65	4.39	1.28	2.93	0.00	0.45
Solomon_c109	4.98	3.65	2.39	2.31	0.00	0.55	1.21	2.36
Solomon_r101	3.07	0.00	1.01	1.00	2.73	3.85	3.23	4.47
Solomon_r102	3.83	1.73	2.29	0.48	1.99	0.00	0.23	0.93
Solomon_r103	2.37	0.00	0.58	0.61	2.81	1.86	3.52	1.24
Solomon_r104	1.94	1.63	3.82	5.20	2.39	4.21	0.00	2.17
Solomon_r105	2.75	1.38	3.52	4.34	4.17	0.00	3.04	0.92
Solomon_r106	4.47	4.53	2.67	3.71	1.83	0.00	4.44	3.76

Continues on next page

Continued from previous page

Instance	mALNS (1, 100, 4900)	mALNS (12, 100, 3800)	mALNS (25, 100, 2500)	mALNS (37, 100, 1300)	mALNS (1, 100, 9900)	mALNS (25, 100, 7500)	mALNS (50, 100, 5000)	mALNS (75, 100, 2500)
Solomon_r107	1.16	3.17	1.68	0.72	3.20	1.84	0.47	0.00
Solomon_r108	5.36	7.63	0.92	3.29	6.20	2.06	6.06	0.00
Solomon_r109	4.41	0.17	0.00	2.04	0.09	0.41	0.76	1.13
Solomon_r110	3.46	6.18	2.51	3.04	1.62	0.00	5.29	0.63
Solomon_r111	0.00	2.57	0.16	1.07	5.46	3.67	2.15	1.02
Solomon_r112	3.00	0.00	3.29	1.06	2.16	0.84	1.29	0.00
Solomon_rc101	4.31	0.55	4.17	3.08	4.34	3.45	0.00	4.09
Solomon_rc102	3.90	3.51	1.02	1.93	0.13	1.34	0.00	2.22
Solomon_rc103	7.75	3.19	5.43	5.43	5.94	1.86	5.38	0.00
Solomon_rc104	2.56	2.65	4.83	4.31	5.89	1.71	0.00	2.13
Solomon_rc105	2.93	2.82	1.48	2.45	2.80	0.00	0.94	2.51
Solomon_rc106	3.41	1.64	5.09	0.00	1.93	1.56	0.81	4.34
Solomon_rc107	1.40	5.69	4.29	3.22	1.45	3.25	0.00	3.65
Solomon_rc108	3.77	0.79	2.51	0.00	1.94	2.62	1.04	3.03
Average	3.30	2.38	2.11	2.36	2.73	1.46	1.57	1.73

Table A.8: Relative gap of the objective function value reported by each version of the multi-start ALNS with respect to the best one for Class 3

Instance	mALNS (1, 100, 4900)	mALNS (12, 100, 3800)	mALNS (25, 100, 2500)	mALNS (37, 100, 1300)	mALNS (1, 100, 9900)	mALNS (25, 100, 7500)	mALNS (50, 100, 5000)	mALNS (75, 100, 2500)
Cordeau_pr01	1.84	3.02	4.03	3.24	4.61	0.00	2.14	1.70
Cordeau_pr02	1.50	0.73	0.00	1.79	0.00	1.02	0.98	0.20
Cordeau_pr03	1.41	4.08	4.16	1.63	5.33	2.03	2.80	0.00
Cordeau_pr04	2.33	3.28	4.64	5.26	0.00	1.18	0.89	1.93
Cordeau_pr05	3.34	4.86	2.37	5.51	6.76	4.80	0.00	1.97
Cordeau_pr06	2.13	4.45	2.58	1.87	1.25	0.00	4.04	1.27
Cordeau_pr07	3.48	3.14	2.20	2.18	0.00	1.58	0.19	1.17
Cordeau_pr08	4.92	2.91	2.31	3.35	3.11	0.00	3.54	4.50
Cordeau_pr09	0.32	0.65	1.05	1.44	3.19	0.87	0.00	1.58
Cordeau_pr10	2.65	1.49	2.96	3.13	1.38	1.12	0.00	2.59
Solomon_c101	4.06	1.76	2.21	4.11	3.62	0.00	2.34	2.41
Solomon_c102	1.31	1.53	2.08	2.13	1.46	0.00	0.79	3.02
Solomon_c103	3.14	1.40	0.00	1.88	1.85	1.32	2.55	0.60
Solomon_c104	2.53	4.30	0.00	3.01	3.05	0.48	1.20	2.43

Continues on next page

Continued from previous page

Instance	mALNS (1, 100, 4900)	mALNS (12, 100, 3800)	mALNS (25, 100, 2500)	mALNS (37, 100, 1300)	mALNS (1, 100, 9900)	mALNS (25, 100, 7500)	mALNS (50, 100, 5000)	mALNS (75, 100, 2500)
Solomon_c105	5.23	2.11	0.58	0.31	0.47	1.34	0.95	0.00
Solomon_c106	1.93	1.45	3.02	2.89	1.56	0.12	0.64	0.00
Solomon_c107	3.32	0.00	0.68	1.93	2.51	0.16	1.53	0.32
Solomon_c108	2.42	1.17	3.57	0.33	1.61	0.61	0.00	1.08
Solomon_c109	2.86	1.69	2.29	3.98	3.18	1.37	0.00	0.90
Solomon_r101	1.73	2.84	2.61	0.17	2.24	1.00	0.00	1.91
Solomon_r102	3.45	3.55	0.10	2.85	2.25	0.15	0.00	1.74
Solomon_r103	4.34	4.40	2.84	3.99	4.77	0.75	0.00	2.25
Solomon_r104	0.09	3.45	3.20	1.96	0.38	0.00	2.88	1.56
Solomon_r105	1.28	2.53	4.76	0.00	2.07	3.18	1.26	3.64
Solomon_r106	3.48	2.85	3.58	4.31	3.24	2.42	0.00	1.07
Solomon_r107	0.70	2.07	4.74	2.51	0.40	0.00	3.39	2.42
Solomon_r108	2.46	2.99	2.13	0.00	2.82	5.27	2.00	1.98
Solomon_r109	3.39	2.21	2.06	2.14	0.59	0.55	4.24	0.00
Solomon_r110	2.27	1.45	3.35	0.47	0.70	0.00	4.45	2.09
Solomon_r111	4.74	2.99	1.21	1.15	2.80	0.00	2.05	2.84
Solomon_r112	5.06	0.18	1.48	2.58	1.69	1.72	2.58	0.00
Solomon_rc101	2.56	0.76	0.16	1.99	0.00	2.08	0.57	1.02
Solomon_rc102	4.67	1.50	3.57	3.86	3.62	0.00	2.56	0.84
Solomon_rc103	1.06	2.00	3.19	0.00	4.20	1.62	1.57	3.25
Solomon_rc104	5.34	4.98	6.48	3.95	6.23	0.00	2.54	6.51
Solomon_rc105	5.42	5.66	2.19	2.62	5.80	0.39	0.96	0.00
Solomon_rc106	4.11	3.30	5.00	3.05	2.56	0.00	5.04	0.87
Solomon_rc107	2.78	1.74	3.93	4.30	4.30	2.89	1.34	0.00
Solomon_rc108	0.43	0.26	2.94	3.76	0.21	1.92	0.00	1.60
Average	2.82	2.45	2.57	2.45	2.46	1.07	1.59	1.62

Table A.9: Relative gap of the objective function value reported by each version of the multi-start ALNS with respect to the best one for Class 4

Instance	mALNS (1, 100, 4900)	mALNS (12, 100, 3800)	mALNS (25, 100, 2500)	mALNS (37, 100, 1300)	mALNS (1, 100, 9900)	mALNS (25, 100, 7500)	mALNS (50, 100, 5000)	mALNS (75, 100, 2500)
Cordeau_pr01	1.24	1.18	0.57	0.32	2.02	1.18	0.04	0.00
Cordeau_pr02	1.58	3.77	2.09	0.00	4.72	1.43	2.00	1.36
Cordeau_pr03	2.00	1.55	1.99	0.67	1.13	1.82	0.00	0.45

Continues on next page

Continued from previous page

Instance	mALNS (1, 100, 4900)	mALNS (12, 100, 3800)	mALNS (25, 100, 2500)	mALNS (37, 100, 1300)	mALNS (1, 100, 9900)	mALNS (25, 100, 7500)	mALNS (50, 100, 5000)	mALNS (75, 100, 2500)
Cordeau_pr04	3.18	3.00	2.68	3.83	4.12	0.68	0.00	2.37
Cordeau_pr05	1.72	0.00	5.14	2.70	5.84	1.39	4.26	1.49
Cordeau_pr06	2.82	0.86	1.07	2.61	4.22	0.00	2.22	1.50
Cordeau_pr07	2.77	0.93	0.81	1.01	0.06	0.00	0.53	0.63
Cordeau_pr08	2.51	0.70	3.07	2.42	0.03	0.00	0.03	0.22
Cordeau_pr09	1.03	0.15	0.44	1.63	0.64	0.68	0.00	0.63
Cordeau_pr10	1.72	2.59	2.57	2.43	1.52	0.00	1.17	0.88
Solomon_c101	5.08	4.17	4.80	3.29	5.76	4.39	0.00	2.18
Solomon_c102	3.17	0.67	1.25	1.18	2.45	1.15	0.39	0.00
Solomon_c103	4.01	4.67	1.38	1.45	4.02	0.00	0.86	3.47
Solomon_c104	2.59	2.03	1.75	1.90	0.59	0.63	1.14	0.00
Solomon_c105	1.99	0.91	0.00	1.59	4.53	2.76	3.09	0.52
Solomon_c106	4.95	0.79	0.00	1.74	3.70	2.47	0.72	2.17
Solomon_c107	2.48	4.13	1.95	1.17	0.75	0.00	2.16	3.79
Solomon_c108	5.08	2.98	2.58	0.00	5.45	0.99	0.06	1.18
Solomon_c109	2.07	3.41	1.41	0.82	0.39	0.00	0.94	0.91
Solomon_r101	7.90	2.88	5.04	3.17	10.04	0.00	2.18	5.02
Solomon_r102	2.72	1.80	2.29	2.90	2.97	0.00	1.11	1.93
Solomon_r103	3.92	0.54	2.49	1.22	0.77	0.95	1.05	0.00
Solomon_r104	5.16	3.48	0.82	4.26	4.52	2.28	0.00	2.75
Solomon_r105	5.13	2.11	0.93	0.00	3.07	2.19	3.05	2.16
Solomon_r106	5.47	8.61	2.70	4.83	6.02	3.64	0.00	4.92
Solomon_r107	4.40	1.19	2.73	1.13	6.06	2.44	0.00	0.43
Solomon_r108	6.56	4.90	4.46	2.12	9.52	0.00	0.76	4.49
Solomon_r109	1.47	1.70	2.68	2.54	4.66	0.00	1.64	1.00
Solomon_r110	2.30	1.78	3.80	3.39	0.00	0.93	2.19	2.16
Solomon_r111	2.96	2.67	2.10	4.65	4.24	0.94	0.00	0.27
Solomon_r112	3.74	2.60	1.32	5.51	1.78	0.00	3.38	3.50
Solomon_rc101	2.76	0.00	0.37	1.26	2.94	0.34	0.26	1.88
Solomon_rc102	5.37	1.29	2.94	3.62	4.34	1.63	1.54	0.00
Solomon_rc103	0.56	0.00	3.63	3.95	1.06	1.49	0.39	2.00
Solomon_rc104	3.86	1.88	1.20	0.37	3.02	2.03	0.00	2.94
Solomon_rc105	3.14	2.73	0.00	4.58	7.83	0.59	3.10	1.58
Solomon_rc106	4.12	2.96	1.76	2.32	2.94	0.33	0.00	3.18
Solomon_rc107	2.72	0.37	0.79	2.70	5.04	1.96	0.00	1.48
Solomon_rc108	0.00	6.21	4.96	4.44	7.47	4.62	4.66	3.44
Average	3.24	2.26	2.12	2.30	3.60	1.18	1.15	1.77

Table A.10: Relative gap of the objective function value reported by each version of the multi-start ALNS with respect to the best one for Class 5

Instance	**mALNS (1, 100, 4900)**	**mALNS (12, 100, 3800)**	**mALNS (25, 100, 2500)**	**mALNS (37, 100, 1300)**	**mALNS (1, 100, 9900)**	**mALNS (25, 100, 7500)**	**mALNS (50, 100, 5000)**	**mALNS (75, 100, 2500)**
Cordeau_pr01	3.46	0.00	2.16	2.08	1.59	0.40	1.75	2.47
Cordeau_pr02	0.82	0.87	1.37	2.22	2.50	0.25	0.00	0.45
Cordeau_pr03	2.33	1.00	1.19	0.96	2.81	2.33	0.84	0.00
Cordeau_pr04	1.57	1.50	0.24	1.04	2.07	0.00	1.63	1.31
Cordeau_pr05	2.16	0.95	1.42	1.03	3.44	0.00	1.04	1.00
Cordeau_pr06	0.65	0.00	1.03	0.93	1.66	0.63	0.26	0.44
Cordeau_pr07	1.58	2.20	2.48	1.80	0.78	1.52	0.00	2.15
Cordeau_pr08	4.17	1.48	0.01	1.19	0.61	0.00	0.46	0.15
Cordeau_pr09	3.45	1.67	1.17	0.81	2.66	0.00	0.46	1.95
Cordeau_pr10	0.62	0.57	0.91	0.98	0.00	0.81	1.02	0.49
Solomon_c101	6.37	3.46	3.90	3.35	5.61	1.26	0.00	0.97
Solomon_c102	0.77	1.65	1.28	1.16	3.91	0.00	1.14	1.47
Solomon_c103	1.42	3.40	2.40	1.39	0.54	0.00	0.52	0.97
Solomon_c104	3.26	3.55	1.21	2.61	0.33	0.00	1.69	2.21
Solomon_c105	2.36	3.76	0.29	2.43	1.56	0.00	1.22	3.90
Solomon_c106	3.65	2.19	0.00	3.61	3.93	1.59	0.44	3.37
Solomon_c107	5.89	1.58	4.48	2.90	2.71	2.35	0.00	2.73
Solomon_c108	3.81	0.00	3.58	4.79	5.55	0.84	2.85	1.18
Solomon_c109	4.66	2.41	1.83	3.35	3.02	0.00	0.85	2.19
Solomon_r101	2.02	3.09	1.13	4.73	6.26	0.00	1.57	2.42
Solomon_r102	0.97	1.23	0.00	1.64	1.47	0.73	0.96	0.93
Solomon_r103	0.63	0.88	1.13	0.59	1.96	0.88	0.10	0.00
Solomon_r104	1.62	0.98	1.76	5.06	3.62	0.00	3.53	1.17
Solomon_r105	0.57	2.54	2.09	4.66	1.44	0.00	0.42	0.46
Solomon_r106	3.74	1.38	0.79	4.63	1.99	2.37	0.00	1.03
Solomon_r107	4.15	1.73	1.00	2.92	4.66	3.77	0.00	1.65
Solomon_r108	0.00	3.07	2.53	4.38	3.19	1.94	3.28	2.60
Solomon_r109	2.88	0.39	0.00	0.32	0.03	0.23	0.75	0.98
Solomon_r110	5.70	3.89	3.80	5.00	1.66	0.33	0.00	3.30
Solomon_r111	1.19	4.62	4.30	0.67	0.92	0.00	1.67	2.21
Solomon_r112	3.05	2.57	3.65	1.67	3.35	0.00	1.33	2.18
Solomon_rc101	2.03	0.97	0.21	1.61	4.46	3.68	0.00	2.38
Solomon_rc102	4.09	2.81	1.54	1.22	2.88	1.46	4.47	0.00
Solomon_rc103	4.98	2.71	1.14	0.70	2.45	0.11	0.00	0.72
Solomon_rc104	1.87	0.88	0.00	2.62	0.39	2.34	0.17	2.67
Solomon_rc105	1.78	0.14	0.00	0.14	2.91	0.51	0.63	0.40

Continues on next page

Continued from previous page

Instance	mALNS (1, 100, 4900)	mALNS (12, 100, 3800)	mALNS (25, 100, 2500)	mALNS (37, 100, 1300)	mALNS (1, 100, 9900)	mALNS (25, 100, 7500)	mALNS (50, 100, 5000)	mALNS (75, 100, 2500)
Solomon_rc106	2.06	1.17	1.67	3.60	1.75	1.80	1.70	0.00
Solomon_rc107	3.59	2.48	3.17	0.65	0.00	0.67	1.61	2.40
Solomon_rc108	3.63	2.16	0.57	0.40	0.33	0.00	4.19	1.33
Average	2.65	1.84	1.58	2.20	2.33	0.84	1.09	1.49

Tables A.11–A.15 display the objective function value reported by mALNS (25,100,7500) by removing each operator.

On the other hand, Tables A.16–A.20 display the percent gap of the objective function value reported by mALNS (25,100,7500) by removing each operator with respect to the objective function value found by mALNS (25,100,7500) applying all operators.

Table A.11: Objective function value reported by mALNS (25,100,7500) by removing each operator individually for Class 1

Instance	EAS	RE	ES-1	ES-2	EH	IntraE	InterE	IntraR	InterR	IEF-1	IEF-2	IS-1	IS-2	IRDI-1	IRDI-2	IH-1	IH-2
Cordeau_pr01	552.164	572.645	562	549	595	538.315	565	561.921	567.4	564.659	557.047	559.396	587	543.425	575.5	575.283	568.273
Cordeau_pr02	1125	1136	1153	1132	1161	1159	1151	1131	1149	1183	1158.78	1160	1167	1127	1157	1157	1142
Cordeau_pr03	1573.09	1560.88	1606.12	1581.94	1585.26	1577.04	1570.89	1591.99	1604.06	1545.88	1540.66	1583.19	1570.22	1580.97	1541.79	1535.69	1514.75
Cordeau_pr04	1964.15	2075.21	2167.15	2119.47	2122.12	2161.83	2113.93	2086.01	2067.83	2150.73	2098.14	2056.95	2033.67	2143.72	2042.18	2078.38	1993.74
Cordeau_pr05	2763.64	2860.38	2946.59	2768.19	2845.28	2810.68	2807.08	2696.12	2919.13	2965.42	2871.77	2800.22	2748.13	2836.14	2772.58	2855.81	2763.35
Cordeau_pr06	3362.08	3491.36	3452.04	3493.34	3463.4	3472.25	3502.1	3525.31	3391.68	3420.66	3471.13	3452.49	3504.76	3508.93	3408.62	3514.86	3386.09
Cordeau_pr07	801.56	811.41	815.836	807.704	827.078	818.811	837.366	818.567	803.394	838.03	823.182	811.465	814.901	808.547	810.681	835.123	815.188
Cordeau_pr08	1698.98	1798.09	1785.61	1863.51	1884.67	1828.8	1847.59	1784.21	1771.05	1701.07	1774.43	1769.29	1769.56	1824.05	1762.41	1765.27	1777.61
Cordeau_pr09	2574.3	2687.25	2671.91	2698.24	2686	2647.88	2708.28	2653.44	2705	2681.53	2654.84	2689.29	2629.7	2654.96	2696	2674.46	2675.98
Cordeau_pr10	3513.97	3660.62	3645.82	3712.08	3690.12	3708.37	3692.39	3698.26	3721.09	3512.01	3533.76	3556.23	3656.37	3658.26	3692.4	3576.55	3678.22
Solomon_c101	1414.26	1460.9	1426.94	1470.73	1419.74	1443.44	1489.28	1490.91	1455.85	1479.91	1478.16	1451.67	1449.75	1464.83	1407.45	1468.71	1439.86
Solomon_c102	1568.65	1633.72	1604.36	1623.61	1628.07	1582.72	1598.48	1572.34	1583.89	1604.69	1617.43	1542.24	1600.34	1580.27	1607.07	1604.22	1619.52
Solomon_c103	1570.98	1650.55	1635.09	1591.25	1587.06	1625.54	1623.2	1570.05	1593.88	1616.37	1646.36	1619.62	1576.28	1632.08	1654.36	1577.86	1615.6
Solomon_c104	1447.18	1535.05	1492.87	1461.34	1508.7	1490.81	1498	1489.57	1502.38	1525.18	1524.38	1545.55	1525.84	1523.7	1515.3	1500.81	1580.9
Solomon_c105	1387.77	1448.22	1415	1435.62	1479.96	1424.7	1434.59	1450.66	1411	1374.98	1447.75	1363.83	1394.48	1435.73	1410.91	1426.78	1447.94
Solomon_c106	1587.37	1592.74	1590.21	1623.37	1563.37	1595.1	1598.58	1615.78	1576.3	1579.83	1607.62	1597.16	1539.56	1615.62	1601.53	1587.82	1562.74
Solomon_c107	1590	1637.55	1670	1660	1654.46	1669.03	1639.88	1641.01	1652.86	1665.5	1638.52	1628.52	1638.81	1634.25	1645.99	1651.78	1624.76
Solomon_c108	1630	1640.67	1676.71	1640	1665.74	1660.35	1640.83	1629.52	1670.47	1647.51	1631.09	1663.44	1654.62	1603.01	1675.77	1645	1661.85
Solomon_c109	1530.65	1586.29	1558.36	1590.51	1532.89	1638.71	1580.26	1568.72	1547.79	1569.56	1572.63	1549.14	1597.86	1594.24	1571.58	1599.27	1611.55
Solomon_r101	1223.2	1277.86	1293.07	1230.63	1230.56	1244.17	1239.07	1289.6	1292.89	1276.14	1267.47	1220.58	1265.74	1270.57	1216.79	1282.82	1284.87
Solomon_r102	1353.3	1390	1391.19	1385.15	1404.52	1353.17	1386.98	1362.1	1417.69	1397	1386.14	1380.34	1407.71	1382.22	1386.03	1380.93	1403.54
Solomon_r103	1350.3	1378.87	1380.31	1368.28	1347.23	1360.26	1409.12	1407.05	1413.58	1380	1327	1355.94	1349.51	1380.25	1408.93	1369.94	1353.99
Solomon_r104	1080.97	1185.89	1150.67	1135.16	1175.44	1136.32	1180.06	1116.5	1134.67	1174.18	1157.02	1174.84	1135.46	1127.11	1131.89	1188.63	1142.65
Solomon_r105	1145.6	1214.98	1224.45	1170.56	1173.62	1182.79	1202.31	1207.27	1211.32	1236.38	1208.18	1196.52	1213.5	1226.94	1177.47	1189.88	1197.4

Continues on next page

Continued from previous page

Instance	EAS	RE	ES-1	ES-2	EH	IntraE	InterE	IntraR	InterR	IEF-1	IEF-2	IS-1	IS-2	IRDI-1	IRDI-2	IH-1	IH-2
Solomon_r106	1306.36	1334.06	1364.87	1322.92	1344.75	1331.57	1351.37	1350.05	1318.83	1360.8	1309.12	1321.01	1345.4	1368.57	1292.24	1346.95	1314.83
Solomon_r107	1253.32	1301.66	1296.66	1316	1351.49	1314.09	1311.98	1265.86	1317	1268.64	1272.02	1269	1253.52	1286.05	1238.6	1303.84	1323.13
Solomon_r108	1138.78	1176.99	1122.75	1173.11	1173.56	1189.21	1121.3	1208.03	1155.13	1177.11	1176.28	1153.33	1187.8	1121.25	1219.26	1196.83	1179.8
Solomon_r109	1316.11	1345.19	1273.94	1278.18	1270.12	1319.12	1351.94	1326	1345.51	1323	1335.53	1315.82	1297.45	1367.22	1298.79	1337.29	1336.52
Solomon_r110	1237.38	1306.02	1270.51	1277	1260.28	1279.35	1277.85	1272.77	1273.16	1272.82	1275.12	1229.01	1231.01	1259.51	1268.5	1265.53	1280.87
Solomon_r111	1191.08	1220.43	1202.07	1145.95	1250.6	1187.27	1209.13	1197.84	1135.91	1168.85	1224.88	1153.73	1249.17	1170.98	1171.35	1182.25	1199.99
Solomon_r112	1235	1318.04	1324	1293	1326	1303	1299.65	1334	1308	1308	1302	1252.41	1286	1300	1284	1331	1301.19
Solomon_rc101	1362.73	1408.97	1448.56	1468.08	1412.36	1388.26	1449.84	1411.14	1419.35	1474	1412.37	1436.52	1432.4	1413.34	1401.7	1407.72	1431.52
Solomon_rc102	1508.68	1535.48	1491.09	1430.69	1517.52	1509.38	1523.22	1539.22	1510.12	1495.26	1504	1527.12	1471.09	1440.8	1448.67	1620.37	1462.53
Solomon_rc103	1485.55	1544.38	1573.07	1439.02	1546.82	1498.27	1473.19	1526.47	1502.67	1524	1522.18	1488.88	1552.6	1557.51	1501.94	1549.94	1554.67
Solomon_rc104	1359.15	1392.96	1374.25	1402.74	1393.31	1427.08	1466.61	1420.49	1390.07	1350.65	1430.36	1426.92	1420.28	1459.26	1486.81	1416.39	1445.82
Solomon_rc105	1467.41	1569.88	1503	1522.28	1554.59	1569.05	1534.98	1521.72	1510	1595.24	1557.64	1565.83	1603.31	1492.84	1564.75	1572.75	1526.96
Solomon_rc106	1386.57	1463.82	1444.78	1518.29	1441.82	1449.8	1486.5	1473.78	1431.9	1464.49	1446.39	1472.37	1473.77	1464.37	1416.07	1434.23	1508.75
Solomon_rc107	1440.99	1510.49	1539.05	1492.35	1494.81	1536	1574.74	1517.58	1524.1	1543.57	1556.7	1541.53	1552.5	1496.44	1463.68	1495.72	1516.69
Solomon_rc108	1376.44	1378	1436.33	1400	1416.67	1373.06	1396.66	1348.62	1415.32	1394.94	1441.56	1360	1406.5	1374.22	1385.79	1355.42	1350.44

Table A.12: Objective function value reported by mALNS (25,100,7500) by removing each operator individually for Class 2

Instance	EAS	RE	ES-1	ES-2	EH	IntraE	InterE	IntraR	InterR	IEF-1	IEF-2	IS-1	IS-2	IRDI-1	IRDI-2	IH-1	IH-2
Cordeau_pr01	548.264	561.913	561.592	549.297	556.355	539.415	560.523	541.946	536.619	564.662	550.324	546.594	543.252	550.028	541.866	537.882	556.742
Cordeau_pr02	1048.88	1065.12	1061.69	1060.24	1074.53	1054.86	1063.68	1067	1053.36	1080.83	1041.03	1050.06	1040.16	1059.1	1067.6	1053.96	1070.56
Cordeau_pr03	1480.71	1528.51	1518.86	1518.38	1477.65	1541.54	1487.88	1521.02	1527.6	1482.6	1524.67	1487.79	1475.15	1548.21	1549.44	1529.55	1532.26
Cordeau_pr04	1903.8	2018.72	1994.44	1988.94	2030.3	2060.12	1972.67	1949.11	1956.75	1948.29	1958.03	2009.91	2030.71	1968.05	2017.26	1954.82	2005.69
Cordeau_pr05	2647.11	2862.02	2821.37	2830.87	2917.13	2761.39	2822.12	2835.83	2783.99	2704.2	2887.91	2864.59	2876.13	2796.43	2819.62	2799.35	2797.18
Cordeau_pr06	2985.39	3111.4	3113.1	3003.45	2976.49	3123.13	3063.72	3106.14	3152.23	2994.48	3165.09	3101.6	3111.12	3134.18	3079.76	3127.3	3147.91
Cordeau_pr07	783.651	798.594	781.495	788.366	806.635	808.964	768.78	802.07	786.313	778.846	798.39	783.149	785.229	796.123	783.027	804.445	792.33
Cordeau_pr08	1614.7	1726.3	1702.83	1727.64	1708.06	1709.21	1751.75	1738.48	1716.36	1728.13	1633.75	1730.83	1745.16	1752.16	1748.36	1722.85	1725.42
Cordeau_pr09	2330.04	2425.45	2417.13	2435.43	2406.81	2392.76	2386.73	2443.81	2378.67	2393.46	2421.46	2422.24	2407.54	2423.86	2376.16	2417.43	2409.62
Cordeau_pr10	3263.58	3438.85	3282.09	3404.63	3386.79	3339.23	3284.84	3381.08	3232.41	3388.94	3375.6	3391.87	3381.33	3386.85	3407.68	3408.73	3398.08
Solomon_c101	1379.93	1419.91	1420.08	1427.62	1407.49	1415.27	1383.92	1461.44	1421.63	1425.64	1382.8	1412.44	1373.56	1425.22	1417.75	1412.89	1403.62
Solomon_c102	1508.03	1566.77	1561.99	1533.27	1522.84	1536.11	1563.42	1531.24	1526.55	1549.62	1544.18	1553.04	1528.33	1563.46	1574.97	1548.01	1501.59
Solomon_c103	1474.83	1520.6	1485.78	1536.67	1536.45	1495.99	1533.83	1516.44	1509.89	1500.33	1456.83	1463.64	1485.75	1497.03	1536.44	1523.16	1514.64
Solomon_c104	1451.33	1499.84	1484.75	1468.22	1486.01	1461.99	1489.3	1468	1472.77	1458.41	1450.23	1471.07	1433.96	1462.25	1471.38	1463.08	1433
Solomon_c105	1382.78	1405.28	1412.08	1430.57	1443.65	1421.75	1442.21	1383.32	1410.16	1418.86	1415.37	1413.35	1384.78	1439.24	1400.26	1401.65	1460.12
Solomon_c106	1494	1542.28	1533.68	1496.85	1465.78	1532.12	1502.37	1533.93	1469.01	1528.83	1546.45	1535.1	1486.39	1494.31	1537.24	1562.1	1535.36
Solomon_c107	1503.93	1553.41	1561.38	1514.25	1542.3	1584.61	1551.46	1518.22	1535.21	1569.78	1556.07	1552.9	1575.12	1547.26	1536.01	1560.35	1581.41
Solomon_c108	1521.87	1607.42	1572.47	1584.29	1532.35	1602.9	1591.96	1537	1543.47	1576.49	1543.34	1583.83	1545.3	1557.13	1578.04	1567.03	1548.34
Solomon_c109	1465	1585.53	1543.03	1549.44	1545.61	1533.46	1555.25	1520.73	1507.51	1569.01	1499.76	1503.87	1541.57	1528.14	1542.12	1543.11	1550.13
Solomon_r101	1164.74	1218.68	1210.1	1215.06	1228.85	1232.96	1202.12	1215.19	1240.63	1237.89	1190.94	1234.89	1233.63	1199.41	1209.41	1209.03	1198.22
Solomon_r102	1281.83	1294.1	1324.98	1283.91	1309.76	1273.09	1277.67	1305.62	1272.75	1313.66	1309.87	1315.28	1288.64	1318.6	1271.8	1287.54	1315.15
Solomon_r103	1221.96	1265.4	1242.79	1280.43	1277.31	1254.68	1276.44	1273.28	1268.91	1244.49	1259.01	1227.64	1271.05	1286.15	1256.62	1252.79	1264.3
Solomon_r104	1077.23	1142.58	1114.14	1137.85	1134.14	1117.91	1134.98	1153.36	1095.07	1118.24	1141.44	1139.4	1124.88	1105.79	1084.34	1118.89	1152.98
Solomon_r105	1178.04	1200.65	1162.98	1128.63	1206.21	1199.8	1192.15	1168.6	1166.81	1151.32	1165.81	1169.88	1160.86	1173.63	1221.56	1154.28	1179.55

Continues on next page

Continued from previous page

Instance	EAS	RE	ES-1	ES-2	EH	IntraE	InterE	IntraR	InterR	IEF-1	IEF-2	IS-1	IS-2	IRDI-1	IRDI-2	IH-1	IH-2
Solomon_r106	1245.87	1301.02	1307.5	1239.56	1292.6	1277.38	1256.38	1319.83	1286.14	1244.09	1288.36	1262.43	1253.19	1290.26	1294.09	1296.94	1293.83
Solomon_r107	1157.45	1218.61	1210.33	1208.56	1221.66	1229.07	1229.63	1194.9	1223.59	1218.16	1224	1187.37	1201.54	1213.07	1214.06	1216.79	1167.28
Solomon_r108	1116.98	1141.46	1164.78	1118.2	1125.92	1129.58	1140.53	1157.86	1156.18	1142.25	1103.72	1138.98	1145.13	1145.04	1178.76	1163.02	1148.19
Solomon_r109	1200.85	1220.23	1248.57	1222.94	1270.06	1261.79	1203.43	1227.17	1195.78	1241.88	1231.7	1244.26	1165.25	1242.65	1235.16	1222.65	1219.1
Solomon_r110	1168.45	1182.02	1210.96	1162.49	1146.45	1187.82	1173.42	1202.9	1197.81	1223.64	1226.63	1188.04	1191.25	1193.68	1213.06	1222.04	1217.93
Solomon_r111	1127.73	1193.59	1160.61	1156.14	1174.69	1176.66	1146.79	1124.07	1132.35	1117.66	1165.12	1132.98	1168.86	1176.39	1200.91	1152.11	1193.49
Solomon_r112	1163.91	1217.08	1201.41	1202	1234.1	1219.31	1212.07	1227.78	1217.41	1244.91	1247.75	1209.62	1234.18	1213.61	1212.45	1229.85	1205.86
Solomon_rc101	1362.76	1366.02	1380.29	1347.95	1375.05	1378.82	1390.26	1375.17	1386.92	1341.38	1365.47	1389.37	1356.55	1366.77	1403.38	1391.08	1381.46
Solomon_rc102	1410.21	1454.43	1434.11	1412.26	1429.96	1502.28	1462.97	1402.1	1435.33	1419.57	1426.43	1486.93	1419.56	1476.52	1456.98	1456.71	1465.76
Solomon_rc103	1370.14	1405.43	1418.26	1433.75	1411.06	1419.07	1422.26	1417.71	1435.84	1432.2	1446.31	1402.84	1406.84	1432.15	1410.67	1422.1	1435.77
Solomon_rc104	1326.21	1317.92	1423.47	1349.03	1382.61	1361.59	1373.51	1380.85	1334.13	1361.3	1364.55	1350.4	1401.88	1390.85	1413.02	1351.3	1323.72
Solomon_rc105	1463.52	1509.67	1543.8	1529.09	1455.22	1542.51	1524.93	1527.98	1543.2	1545.73	1452.27	1523.13	1515.1	1543.22	1526.01	1474.97	1453.32
Solomon_rc106	1337.94	1375.32	1377.26	1403.33	1399.02	1372.32	1402.14	1378.33	1394.19	1407.03	1378.74	1382.47	1354.57	1387.94	1357.16	1391.44	1409.37
Solomon_rc107	1365.95	1423.82	1402.6	1421.6	1425.93	1418.89	1379.31	1371.01	1421.25	1423.35	1411.15	1443.99	1403.09	1389.23	1371.85	1382.49	1415.72
Solomon_rc108	1344.38	1389.55	1377.05	1376.12	1405.34	1382.2	1362.83	1392.95	1341.06	1371.26	1394.66	1326.73	1369.75	1411.36	1380.11	1302.25	1412.36

Table A.13: Objective function value reported by mALNS (25,100,7500) by removing each operator individually for Class 3

Instance	EAS	RE	ES-1	ES-2	EH	IntraE	InterE	IntraR	InterR	IEF-1	IEF-2	IS-1	IS-2	IRDI-1	IRDI-2	IH-1	IH-2
Cordeau_pr01	547.239	554.924	553.69	545.851	556.209	547.051	549.436	551.212	530.249	552.958	546.958	545.063	548.631	551.582	538.968	562.601	546.931
Cordeau_pr02	1023.53	1049.24	1056.76	1048.38	1042.88	1034.22	1049.02	1057.42	1047.95	1051.67	1053.4	1046.58	1051.05	1051.06	1038.43	1054.17	1046.32
Cordeau_pr03	1454.05	1518.41	1537.07	1498.19	1486.61	1534.67	1514.9	1529.99	1510.6	1528.78	1512.75	1514.68	1522.53	1542.46	1512.77	1504.84	1518.72
Cordeau_pr04	1858.53	1983.57	1979.04	1949.71	1945.02	1904.6	1938.79	1973.77	1946.16	1943.59	1965.04	1864.9	1913.8	1995.91	1965.49	1951.53	1961.64
Cordeau_pr05	2704.63	2885.03	2776.52	2752.54	2850.27	2675.65	2816.52	2733.6	2809.63	2819.29	2876.36	2713.59	2841.27	2794.08	2787.37	2816.71	2852.24
Cordeau_pr06	2992.86	3077.86	3056.25	3012.94	3130.78	3068.59	3114.86	3108.46	3015.12	3058.67	3095.58	3076.97	3037.33	3112.83	3073.27	3055.29	3073.77
Cordeau_pr07	770.718	776.715	784.576	782.592	771.403	771.309	769.531	778.973	777.248	787.005	779.659	782.414	782.406	793.264	788.559	785.858	775.375
Cordeau_pr08	1672.09	1727.54	1700.05	1689.51	1723.15	1707.52	1719.96	1685.83	1710.83	1737.23	1710.69	1662.42	1685.49	1731.59	1717.41	1717.21	1716.08
Cordeau_pr09	2311.49	2371.9	2325.07	2383.02	2368.39	2374.98	2358.59	2377.27	2365.35	2396.97	2364.01	2387.07	2356.7	2367.56	2352.95	2376.85	2375.24
Cordeau_pr10	3245.25	3345.02	3412	3320.18	3330.31	3320.76	3343.53	3348.38	3340.57	3279.34	3341.06	3313.53	3324.65	3273.42	3344.54	3262.81	3349.04
Solomon_c101	1375.09	1418.45	1388.59	1404.74	1370.3	1413.31	1389.59	1400.43	1412.84	1389.17	1398.15	1434.67	1404.45	1431.59	1442.16	1421.99	1418.28
Solomon_c102	1495.43	1541.44	1557.7	1547.9	1554.03	1558.1	1544.95	1519.26	1537.9	1503.52	1544.6	1530.39	1537.48	1550.23	1512.83	1518.81	1500.8
Solomon_c103	1462.4	1497.31	1502.78	1512.48	1499.36	1479.31	1491.97	1467.05	1479.85	1501.41	1486.84	1489.1	1504.41	1484.06	1494.71	1466.33	1492.78
Solomon_c104	1434.79	1435.12	1488.68	1457.97	1498.31	1474.65	1434.41	1448.36	1484.52	1467.72	1478.21	1429.86	1453.34	1433.33	1436.2	1482.99	1465.81
Solomon_c105	1384.81	1396.32	1414.7	1430.88	1380.88	1442.67	1391.02	1389.54	1385.3	1381.99	1419.22	1404.26	1388.21	1404.9	1421.18	1403.29	1418.24
Solomon_c106	1476.78	1534.7	1497.34	1519.08	1525.02	1519.3	1525.74	1520.66	1538.31	1533.32	1517.37	1520.18	1502.85	1487.39	1538.53	1489.49	1515.03
Solomon_c107	1491.65	1532.64	1549.14	1545.22	1550.1	1540.35	1533.05	1550.95	1525.06	1553.55	1534.8	1539.14	1542.77	1540.37	1534.76	1524.21	1552.79
Solomon_c108	1543.77	1564.14	1551.24	1554.12	1523.64	1542.05	1560.63	1572.18	1555.31	1542.14	1520.77	1574.48	1564.46	1557.34	1546.57	1559.63	1535.97
Solomon_c109	1467.64	1540.93	1530.11	1543.61	1538.44	1530.56	1534.39	1513.66	1519.54	1495.87	1522.76	1537.14	1499.76	1532.19	1499.55	1543.7	1521.92
Solomon_r101	1166.35	1199.19	1193.27	1152.74	1175.37	1216.37	1210.15	1184.53	1215.75	1221.13	1195.26	1205.44	1211.32	1204.7	1230.16	1219.17	1202.95
Solomon_r102	1262.94	1293.03	1271.8	1296.83	1310.71	1300.51	1288.88	1268.06	1254.12	1296.34	1293.14	1293.81	1265.95	1279.22	1263.61	1276.07	1292.21
Solomon_r103	1206.62	1245.72	1249.14	1195.71	1250.11	1226.88	1255.63	1254.56	1209.75	1245.84	1207.71	1253.78	1221.38	1206.47	1235.04	1248.01	1199.59
Solomon_r104	1123.59	1097.97	1161.89	1125.51	1150.85	1107.39	1134.27	1093.79	1086.3	1135.56	1184.73	1117.91	1116.08	1113.66	1092.71	1123.36	1127.3
Solomon_r105	1147.04	1162.89	1216.14	1165.08	1172.72	1181.95	1193.92	1149.68	1158.17	1195.81	1169.57	1146.88	1179.52	1165.33	1181.49	1191.07	1204.09

Continues on next page

Continued from previous page

Instance	EAS	RE	ES-1	ES-2	EH	IntraE	InterE	IntraR	InterR	IEF-1	IEF-2	IS-1	IS-2	IRDI-1	IRDI-2	IH-1	IH-2
Solomon_r106	1248.78	1309.15	1297.8	1294.48	1302.83	1299.96	1270.42	1282.35	1291.7	1252.06	1267.05	1260.24	1272.93	1269.65	1271.35	1300.85	1261.74
Solomon_r107	1171.67	1167.97	1217.09	1199.77	1211.83	1188.6	1174.71	1172.48	1167.27	1194.83	1215.06	1162.51	1202.82	1185.12	1213.5	1211.55	1204.78
Solomon_r108	1107.52	1085.02	1121.71	1158.92	1126.87	1125.92	1122.01	1101.47	1128	1153.69	1152.99	1147.35	1121.85	1165.24	1141.82	1118.73	1162.19
Solomon_r109	1173.72	1214.64	1182.01	1185.61	1219.26	1231.22	1206.47	1187.8	1222.14	1214.16	1215.56	1219.09	1218.14	1227.19	1221.02	1215.45	1207.71
Solomon_r110	1153.81	1195.94	1192.24	1181.6	1198.69	1162.25	1168.43	1180.21	1208.85	1148.05	1191.11	1175.39	1178.54	1198.89	1191.5	1187.47	1156.47
Solomon_r111	1124.27	1111.66	1159.16	1106.42	1176.53	1175.18	1127.45	1148.01	1163.73	1125.2	1180.82	1137.31	1138.54	1168.71	1137.6	1163.83	1148.4
Solomon_r112	1194.45	1202.01	1213.78	1204.3	1200.16	1214.69	1220.45	1206.1	1231.01	1220.48	1231.73	1203.97	1186.25	1207.28	1202.63	1221.1	1226.1
Solomon_rc101	1349.52	1390.55	1378.3	1354.13	1386.15	1378.83	1405.17	1391.11	1383.64	1381.74	1369.87	1370.55	1341.83	1359.4	1358.28	1377.38	1354.16
Solomon_rc102	1426.11	1447.73	1462.97	1447.05	1428.47	1468.08	1396.78	1457.68	1406.27	1434.77	1456.54	1468.63	1423.07	1499.18	1465.24	1398.99	1478.92
Solomon_rc103	1405.14	1442.96	1456.53	1406.99	1428.85	1414.27	1417.97	1388.5	1424.58	1365.49	1369.87	1423.64	1434.36	1390.41	1410.78	1386.17	1430.22
Solomon_rc104	1305.59	1382.28	1344.85	1387.51	1389.65	1386.28	1374.79	1385.8	1382.79	1405.04	1348.31	1327.57	1379.2	1361.65	1390.79	1369.64	1382.01
Solomon_rc105	1457.66	1501.21	1535.58	1527.11	1547.09	1508.84	1543.32	1511.92	1502.84	1522.89	1523.99	1495.73	1537.81	1529.02	1550.66	1513.02	1510.91
Solomon_rc106	1308.97	1392.53	1374.3	1340.68	1405.31	1414.5	1376.29	1352.12	1348.89	1399.15	1339.71	1338.35	1337.45	1350.67	1383.48	1389.7	1349.54
Solomon_rc107	1364.3	1378.78	1415.75	1398.28	1411.29	1398.94	1382.82	1375.27	1411.59	1426.18	1391.18	1340.98	1356.12	1397.71	1430.21	1372	1394.61
Solomon_rc108	1343.75	1368.98	1383.58	1385.57	1359.91	1385.2	1344.5	1391.27	1339.02	1309.54	1382.69	1386.12	1418.45	1330.37	1362.01	1334.33	1378.2

Table A.14: Objective function value reported by mALNS (25,100,7500) by removing each operator individually for Class 4

Instance	EAS	RE	ES-1	ES-2	EH	IntraE	InterE	IntraR	InterR	IEF-1	IEF-2	IS-1	IS-2	IRDI-1	IRDI-2	IH-1	IH-2
Cordeau_pr01	518.552	519.091	521.975	514.699	513.199	508.116	516.107	517.978	522.95	523.479	522.104	518.124	514.563	520.589	529.409	516.138	517.095
Cordeau_pr02	919.194	948.153	950.122	947.703	944.812	935.295	946.204	946.644	938.004	938.226	939.381	961.018	947.709	941.588	938.571	939.759	943.555
Cordeau_pr03	1295.12	1297.7	1334.78	1328.21	1341.7	1325.22	1330.67	1302.19	1338.82	1340.89	1329.06	1349.3	1324.36	1324.95	1295.79	1310.97	1341.08
Cordeau_pr04	1727.26	1809.43	1797.24	1780.9	1782.77	1798.28	1768.27	1800.99	1750.14	1756.89	1749.13	1782.61	1818.39	1746.03	1787.4	1803.58	1775.69
Cordeau_pr05	2192.09	2386.57	2392.7	2336.84	2273.62	2443.25	2361.74	2337.98	2269.65	2360.9	2361.21	2402.06	2438.53	2350.15	2299.74	2395.09	2349.94
Cordeau_pr06	2659.75	2739.13	2716.77	2769.06	2725.7	2726.5	2702.03	2678.48	2726.93	2706.31	2670.46	2706.56	2724.06	2628.1	2710.21	2718.75	2658.72
Cordeau_pr07	693.434	718.914	722.263	719.711	704.69	704.589	716.9	696.345	707.408	712.167	694.758	720.109	714.367	711.603	716.556	712.367	719.089
Cordeau_pr08	1401.87	1476.52	1515.91	1461.64	1460.77	1470.15	1517.03	1467.59	1465.84	1488.52	1476.05	1506.1	1479.32	1526.43	1498.36	1501.88	1481.91
Cordeau_pr09	1948.63	2026	2051.85	2011.75	2014.29	2026.54	2018.1	2016.23	2015.38	2005.69	2002.96	2023.93	2034.32	2030.17	2051.57	2036.22	2022.56
Cordeau_pr10	2712.53	2844.73	2855.9	2811.38	2823.65	2809.62	2798.18	2829.21	2780.69	2828.18	2783.05	2835.79	2814.18	2829.64	2864.44	2818.08	2843.09
Solomon_c101	1195.81	1273.58	1233.79	1254.23	1217.67	1280.29	1202.45	1262.84	1249.9	1216.5	1267.8	1273.16	1256.48	1222.57	1249.01	1243.94	1180.7
Solomon_c102	1335.75	1325.33	1338	1328.25	1347.16	1357.64	1331.34	1330.73	1331.18	1347.07	1333.76	1356.51	1346.28	1334.95	1342.68	1346.01	1340.43
Solomon_c103	1342.33	1361.91	1344.89	1384.87	1387.95	1383.85	1385.72	1382.75	1366.34	1362.11	1374.63	1329.74	1381.61	1368.59	1395.16	1376.88	1358.08
Solomon_c104	1272.57	1270.71	1280.7	1281.93	1301.42	1251.76	1274.63	1276.02	1305.05	1264.81	1258.25	1289.45	1288.94	1277.61	1255.81	1289.81	1281.65
Solomon_c105	1182.94	1264.3	1241.89	1261.33	1250.58	1226.22	1236.14	1235.85	1247.8	1251.29	1223.37	1253.2	1279.68	1245.59	1239.07	1246.07	1235.29
Solomon_c106	1211.43	1241.93	1254.19	1269.58	1265.47	1267.96	1262.49	1243.26	1271.46	1265.09	1234.8	1254.13	1254.49	1238.36	1263.7	1259.5	1215.42
Solomon_c107	1212.87	1270.28	1273.83	1262.62	1258.77	1216.26	1261.19	1299.99	1252.26	1303.05	1274.37	1244.54	1282.17	1286.64	1292.21	1263.59	1247.18
Solomon_c108	1325.37	1387.78	1368.21	1389.54	1393.48	1375.07	1369.96	1360.85	1371.53	1354.63	1342.41	1343.86	1365.58	1363.74	1373.45	1368.25	1371.02
Solomon_c109	1231.29	1268.2	1260.08	1273.74	1269.15	1267.26	1273.14	1252.11	1252.3	1280	1238.56	1265.68	1258.83	1256.16	1286.78	1245.48	1283.43
Solomon_r101	1021.28	1088.78	1095.86	1083.37	1038.78	1027.31	1058.2	1071.62	1019.07	1019.89	1020.98	1045.12	1066.15	1068.18	1049.08	1040.92	1037.11
Solomon_r102	1110.16	1124.59	1134.2	1133.95	1136.23	1117.3	1134.99	1138.28	1131.16	1134.28	1124.77	1127.99	1123.18	1136.3	1144.17	1130.34	1130.29
Solomon_r103	1050.48	1100.99	1101.35	1109.05	1093.4	1100.55	1104	1083	1093.2	1095.55	1101.56	1112.77	1111.91	1097.26	1096.22	1097.86	1098.3
Solomon_r104	997.214	1073.3	1040.26	1037.02	1094.85	1063.65	1039.07	1088.82	1087.7	1070.03	1079.33	1064.25	1057.89	1059.97	1060.79	1076.36	1071.42
Solomon_r105	1070.81	1136.41	1134.15	1092.56	1120.2	1084.86	1066.48	1085.29	1090.79	1101.28	1118.9	1097.32	1131.25	1083.14	1094.85	1131.8	1055.56

Continues on next page

Continued from previous page

Instance	EAS	RE	ES-1	ES-2	EH	IntraE	InterE	IntraR	InterR	IEF-1	IEF-2	IS-1	IS-2	IRDI-1	IRDI-2	IH-1	IH-2
Solomon_r106	1017.53	1083.72	1112.01	1078.58	1119.72	1088.87	1068.66	1084.76	1075.28	1089.48	1096.92	1094.53	1104.96	1049.14	1050.76	1073.09	1087.01
Solomon_r107	1071.2	1125.76	1130.68	1127.6	1099.94	1088.75	1136.11	1114.19	1124.15	1130.57	1111.45	1079.97	1108.42	1124.98	1111.84	1139.36	1099.64
Solomon_r108	988.421	1052.36	1039.84	1014.58	1033.94	1043.78	1031.53	1030.62	1052.93	1025.89	1019.29	1058.52	1020.26	1031.24	1023.73	1027.81	1038.83
Solomon_r109	1041.73	1100.98	1089.2	1082	1068.71	1090.91	1086.4	1108.75	1082.29	1060.97	1034.49	1090.51	1106.45	1085.56	1098.04	1102.53	1071.22
Solomon_r110	1034.23	1099.44	1082.67	1082.89	1086.82	1108.61	1038.65	1092.06	1090.14	1076.17	1074.73	1090.79	1064.51	1078.29	1077.25	1100.13	1095.96
Solomon_r111	1026.92	1098.83	1104.05	1107.61	1102.73	1087.73	1080.23	1041.71	1092.63	1027.77	1100.02	1088.19	1087.42	1064.96	1067.26	1097.37	1087.78
Solomon_r112	989.204	1018.66	1051.82	1024.87	1014.42	1047.28	1056.66	1042.57	1028.12	1028.73	1032.62	1048.6	1039.41	1038.29	1015.27	1041.02	1015.29
Solomon_rc101	1249.77	1257.73	1284.41	1249.63	1279.33	1284.83	1299.73	1251.75	1260.47	1302.11	1262.68	1290.52	1290.44	1293.51	1310.34	1293.15	1277.57
Solomon_rc102	1198.24	1265.76	1259.04	1262.96	1261.76	1264.73	1239.64	1250.29	1239.2	1249.03	1269.39	1260.19	1238.58	1269.34	1242.78	1240.88	1245.4
Solomon_rc103	1238.49	1301.38	1279.27	1278.81	1273.47	1260.7	1291.57	1286.48	1257.39	1293.1	1276.03	1220.63	1253.08	1292.16	1290.11	1299.31	1281.33
Solomon_rc104	1200.33	1275.7	1291.28	1262.45	1285.18	1240.67	1252.76	1231.81	1229.84	1275.9	1264.66	1280.55	1325.72	1290.84	1287.54	1272.17	1304.54
Solomon_rc105	1199.83	1287.11	1229.88	1278.99	1268.12	1261.62	1300.02	1260.21	1259.94	1281.26	1300.41	1236.1	1232.37	1274.43	1242.84	1279.89	1246.1
Solomon_rc106	1118.54	1196.03	1200.4	1180.95	1182.02	1183.32	1191.65	1179.55	1189.27	1126	1144.61	1172.31	1178.05	1171.65	1167.83	1190.77	1178.81
Solomon_rc107	1289.47	1361.75	1339.33	1335.88	1346.66	1316.67	1331.09	1320.08	1326.61	1344.43	1340.17	1305.94	1336.97	1321.48	1299.33	1342.97	1334.15
Solomon_rc108	1189.93	1164.53	1237.01	1195.77	1276	1227.52	1188.71	1190.51	1203.88	1176.97	1191.28	1191.36	1206.42	1229.61	1231.89	1236.99	1210.16

Table A.15: Objective function value reported by mALNS (25,100,7500) by removing each operator individually for Class 5

Instance	EAS	RE	ES-1	ES-2	EH	IntraE	InterE	IntraR	InterR	IEF-1	IEF-2	IS-1	IS-2	IRDI-1	IRDI-2	IH-1	IH-2
Cordeau_pr01	508.475	510.55	519.798	511.419	514.766	515.218	512.675	517.077	513.456	522.566	508.395	510.211	514.708	509.685	516.82	513.01	508.578
Cordeau_pr02	894.417	932.524	928.796	941.42	912.328	930.418	923.984	915.159	930.296	928.095	926.501	931.497	914.49	923.459	931.09	928.976	916.672
Cordeau_pr03	1286.64	1325.31	1307.58	1312.06	1315.59	1310.46	1316.18	1300.32	1318.6	1316.51	1316.7	1319.45	1315.9	1318.86	1275.26	1315.71	1314.79
Cordeau_pr04	1676.23	1757.43	1740.84	1768.57	1696.4	1715.34	1728.18	1741.82	1717.58	1749.23	1758.6	1716.51	1732.28	1755.82	1714.75	1710.68	1749.34
Cordeau_pr05	2211.69	2334.55	2264.34	2233.11	2295.08	2253.25	2337.27	2290.69	2278.21	2305.84	2286.47	2272.59	2255.82	2251.54	2295.07	2305.46	2266.26
Cordeau_pr06	2617.82	2680.92	2669.08	2654.89	2686.5	2649.8	2642.64	2633.42	2653.78	2660.93	2615.73	2651.46	2632.15	2666.89	2665.15	2649.64	2665.12
Cordeau_pr07	667.191	686.584	688.766	700.548	695.155	700.189	686.378	686.818	690.789	694.653	685.219	696.292	688.046	690.747	699.285	706.43	691.381
Cordeau_pr08	1414.48	1460.08	1451.78	1437.05	1445.23	1435.88	1443.21	1438.37	1440.99	1472.15	1442.06	1452.68	1397.52	1441.8	1452.09	1459.95	1423.49
Cordeau_pr09	1937.22	2003.59	1997.69	1975.38	1999.85	1987.25	1965.2	1990.35	1963.94	1987.07	1981.75	1990.5	1977.29	2002	1982.51	1995.66	1977.41
Cordeau_pr10	2688.94	2783.16	2799.09	2768.39	2780.95	2753.11	2754.33	2794.12	2717.51	2779.51	2759.16	2820.78	2747.3	2702.63	2735.54	2747.11	2775.43
Solomon_c101	1162.05	1213.94	1216.06	1206.37	1228.38	1205.24	1190.36	1213.53	1192.34	1185.65	1232.68	1214.37	1209.3	1217.95	1201.43	1204.8	1217.39
Solomon_c102	1285.9	1325.04	1327.24	1313.39	1323.41	1315.34	1312.23	1322.67	1317.93	1306.95	1306.03	1305.82	1278.4	1321.47	1322.42	1307.61	1290.22
Solomon_c103	1297.65	1347.27	1313.32	1305.38	1352.96	1332.18	1342.25	1314.18	1332.19	1320.1	1333.12	1326.08	1330.19	1296.38	1331.67	1347.64	1321.76
Solomon_c104	1267.4	1255.52	1236.54	1240.88	1269.33	1256.63	1257.16	1254.28	1269.15	1253.38	1234.74	1261.84	1271.3	1236.69	1241.33	1251.01	1248.25
Solomon_c105	1154.11	1207.04	1203.31	1198.38	1211.89	1225.57	1186.09	1206.02	1196.29	1197.19	1187.34	1200.99	1201.71	1173.22	1193.21	1201.76	1174.86
Solomon_c106	1157.96	1204.45	1231.17	1226.98	1219.88	1193.53	1199.68	1187.99	1204.36	1206.55	1182.77	1203.62	1195.78	1191.09	1204.3	1210.8	1177.76
Solomon_c107	1184.33	1227.41	1230.43	1220.14	1212.7	1235.41	1243.57	1213.75	1209.53	1212.97	1203.51	1201.06	1227.69	1213.19	1192.91	1239.08	1206.98
Solomon_c108	1317.46	1355.81	1344.84	1337.17	1352.11	1356.49	1351.01	1351.92	1352.93	1302.53	1342.55	1337.09	1356.29	1347.31	1337.48	1352.67	1325.77
Solomon_c109	1194.82	1230.34	1242.2	1233.9	1198	1238.53	1234.69	1218.49	1238.38	1233.45	1230.03	1216.25	1231.98	1229.5	1231.34	1239.19	1236.32
Solomon_r101	1001.85	1026.76	1004.79	1035.13	1020.13	1006.66	1009.56	1011.4	985.266	1025.32	1011.86	1014.68	1010.84	985.469	1021.09	998.092	1009.12
Solomon_r102	1102.71	1115.25	1109.38	1103.22	1115.52	1127.46	1121.15	1111.34	1108.94	1104.18	1105.68	1114.85	1114.76	1099.14	1109.39	1113.92	1109.23
Solomon_r103	1050.01	1087.53	1080.1	1072.33	1073.37	1061.18	1077.72	1077.32	1058.74	1059.73	1066.38	1077.53	1066.83	1083.39	1080.39	1070.24	1068.21
Solomon_r104	1032.68	1059.51	1053.7	1060.69	1078.58	1028.86	1029.91	1022.06	1046.96	1042.51	1045.78	1052.27	1043.01	1023.24	1033.35	1043.87	1045.23
Solomon_r105	1068.15	1112.92	1110.18	1114.19	1117.51	1110.3	1090.6	1124.44	1092.34	1124.57	1117.24	1118.11	1089.88	1069.33	1096.44	1075.84	1115.38

Continues on next page

Continued from previous page

Instance	EAS	RE	ES-1	ES-2	EH	IntraE	InterE	IntraR	InterR	IEF-1	IEF-2	IS-1	IS-2	IRDI-1	IRDI-2	IH-1	IH-2
Solomon_r106	1005.89	1059.48	1049.79	1041.56	1061.58	1036.1	1020.91	1029.41	1041.75	1044.26	1056.13	1048.42	1045.54	1055.69	1046.95	1055.41	1065.15
Solomon_r107	1085.62	1087.79	1107.06	1058.52	1088.02	1104.32	1105.8	1096.44	1067.46	1101.18	1090.27	1097.47	1107.08	1103.52	1082.97	1101.61	1104.85
Solomon_r108	994.617	1025.1	997.021	1004.17	1000.86	1010.85	1006.03	999.657	1034.03	1016.06	1029.36	966.57	1001.89	996.178	1021.04	1003.51	1023.32
Solomon_r109	1064.12	1045.43	1068.03	1050.72	1056.13	1052.46	1057.49	1072.54	1068.02	1061.71	1072.78	1064.05	1065	1066.07	1071.49	1077.2	1069.34
Solomon_r110	1019.42	1056.87	1062.93	1046.81	1066.23	1062.57	1067.87	1058.62	1054.25	1074.52	1066.59	1066.83	1022.91	1066.13	1030.62	1067.62	1059.28
Solomon_r111	1052.44	1078.64	1054.06	1059.33	1039.94	1074.62	1047.73	1058.71	1034.66	1073.74	1088.91	1070.78	1077.46	1072.4	1043.72	1037.77	1075.22
Solomon_r112	982.017	1010.47	1002.87	1002.48	983.956	994.498	999.842	1013.77	1014.93	1006.78	1015.41	1018.55	993.053	995.281	976.381	1019.03	1023.35
Solomon_rc101	1226.37	1276.03	1261.15	1268.56	1250.15	1245.36	1239.51	1273.29	1270.78	1227.74	1258.24	1255.99	1272.58	1254.94	1252.47	1260.61	1265.95
Solomon_rc102	1183.61	1244.29	1214.43	1223.01	1233.36	1219.09	1223.01	1215.4	1203.29	1231.74	1213.19	1186.61	1223.93	1221.45	1229.39	1200.94	1205.05
Solomon_rc103	1213.94	1235.8	1229.32	1219.64	1245.66	1230.18	1240.92	1240.65	1216.51	1236	1237.57	1245.72	1244.84	1239.79	1237.6	1226.34	1255.31
Solomon_rc104	1202.95	1255.2	1247.28	1219.8	1243.26	1226.06	1252.2	1256.64	1219.64	1236.06	1233.69	1242.46	1228.24	1259.31	1196.68	1248.16	1269.67
Solomon_rc105	1172.49	1231.47	1242.27	1230.37	1241.56	1233.48	1204.93	1213.85	1205.94	1210.41	1225.53	1225.17	1230.09	1227.53	1224.14	1224.16	1215.05
Solomon_rc106	1143.23	1156.21	1157.52	1147.75	1165.05	1150.72	1142.75	1158.75	1152.81	1152.92	1148.12	1150.71	1124.26	1153.57	1149.51	1152.78	1128.26
Solomon_rc107	1265.93	1320.11	1318.69	1302.96	1306.5	1311.84	1327.26	1276	1332.63	1301.3	1281.35	1310.36	1306.52	1333.52	1311.19	1322.15	1323.22
Solomon_rc108	1155.76	1171.65	1209.47	1154.92	1171.8	1199.84	1195.19	1218.7	1210.11	1177.75	1166.6	1167.66	1156.81	1202.05	1197.65	1203.36	1217.54

Table A.16: Relative gap of the objective function value reported by mALNS (25,100,7500) by removing each operator individually with respect to one reported by mALNS (25,100,7500) for Class 1

Instance	EAS	RE	ES-1	ES-2	EH	IntraE	InterE	IntraR	InterR	IEF-1	IEF-2	IS-1	IS-2	IRDI-1	IRDI-2	IH-1	IH-2
Cordeau_pr01	2.4	-1.2	0.7	3.0	-5.1	4.9	0.2	0.7	-0.2	0.2	1.6	1.2	-3.7	4.0	-1.7	-1.6	-0.4
Cordeau_pr02	2.8	1.9	0.4	2.2	-0.3	-0.1	0.6	2.3	0.8	-2.2	-0.1	-0.2	-0.8	2.7	0.1	0.1	1.4
Cordeau_pr03	0.3	1.0	-1.8	-0.3	-0.5	0.0	0.4	-0.9	-1.7	2.0	2.3	-0.4	0.4	-0.2	2.2	2.6	4.0
Cordeau_pr04	5.0	-0.3	-4.8	-2.5	-2.6	-4.5	-2.2	-0.9	0.0	-4.0	-1.4	0.5	1.7	-3.6	1.3	-0.5	3.6
Cordeau_pr05	5.6	2.3	-0.7	5.4	2.8	4.0	4.1	7.9	0.3	-1.3	1.9	4.3	6.1	3.1	5.3	2.4	5.6
Cordeau_pr06	-1.3	-5.2	-4.0	-5.3	-4.4	-4.6	-5.5	-6.2	-2.2	-3.1	-4.6	-4.0	-5.6	-5.7	-2.7	-5.9	-2.0
Cordeau_pr07	0.0	-1.3	-1.8	-0.8	-3.2	-2.2	-4.5	-2.2	-0.3	-4.6	-2.7	-1.3	-1.7	-0.9	-1.2	-4.2	-1.7
Cordeau_pr08	6.7	1.2	1.9	-2.3	-3.5	-0.4	-1.5	2.0	2.7	6.6	2.5	2.8	2.8	-0.2	3.2	3.0	2.4
Cordeau_pr09	5.5	1.4	1.9	1.0	1.4	2.8	0.6	2.6	0.7	1.6	2.6	1.3	3.5	2.5	1.0	1.8	1.8
Cordeau_pr10	3.4	-0.7	-0.3	-2.1	-1.5	-2.0	-1.5	-1.7	-2.3	3.4	2.8	2.2	-0.5	-0.6	-1.5	1.6	-1.1
Solomon_c101	2.7	-0.5	1.8	-1.2	2.3	0.7	-2.5	-2.6	-0.2	-1.8	-1.7	0.1	0.3	-0.8	3.2	-1.0	0.9
Solomon_c102	2.0	-2.1	-0.3	-1.5	-1.8	1.1	0.1	1.7	1.0	-0.3	-1.1	3.6	0.0	1.2	-0.4	-0.3	-1.2
Solomon_c103	3.9	-1.0	-0.1	2.6	2.9	0.5	0.7	3.9	2.5	1.1	-0.8	0.9	3.5	0.1	-1.3	3.4	1.1
Solomon_c104	2.2	-3.7	-0.9	1.3	-1.9	-0.7	-1.2	-0.7	-1.5	-3.1	-3.0	-4.4	-3.1	-3.0	-2.4	-1.4	-6.8
Solomon_c105	2.6	-1.6	0.7	-0.7	-3.8	0.1	-0.6	-1.8	1.0	3.5	-1.6	4.3	2.2	-0.7	1.0	-0.1	-1.6
Solomon_c106	0.2	-0.1	0.1	-2.0	1.7	-0.2	-0.5	-1.5	0.9	0.7	-1.0	-0.4	3.2	-1.5	-0.7	0.2	1.8
Solomon_c107	3.8	0.9	-1.1	-0.5	-0.1	-1.0	0.8	0.7	0.0	-0.8	0.8	1.4	0.8	1.1	0.4	0.0	1.7
Solomon_c108	-1.8	-2.4	-4.7	-2.4	-4.0	-3.7	-2.5	-1.7	-4.3	-2.9	-1.8	-3.9	-3.3	-0.1	-4.6	-2.7	-3.8
Solomon_c109	0.7	-2.9	-1.1	-3.2	0.5	-6.3	-2.5	-1.8	-0.4	-1.8	-2.0	-0.5	-3.7	-3.4	-2.0	-3.8	-4.6
Solomon_r101	2.4	-2.0	-3.2	1.8	1.8	0.7	1.1	-2.9	-3.2	-1.9	-1.2	2.6	-1.0	-1.4	2.9	-2.4	-2.5
Solomon_r102	0.5	-2.2	-2.3	-1.8	-3.3	0.5	-2.0	-0.1	-4.2	-2.7	-1.9	-1.5	-3.5	-1.6	-1.9	-1.5	-3.2
Solomon_r103	1.5	-0.6	-0.7	0.2	1.7	0.8	-2.8	-2.7	-3.1	-0.7	3.2	1.1	1.5	-0.7	-2.8	0.0	1.2
Solomon_r104	8.0	-1.0	2.0	3.4	-0.1	3.3	-0.5	4.9	3.4	0.0	1.5	0.0	3.3	4.0	3.6	-1.2	2.7

Continues on next page

Continued from previous page

Instance	EAS	RE	ES-1	ES-2	EH	IntraE	InterE	IntraR	InterR	IEF-1	IEF-2	IS-1	IS-2	IRDI-1	IRDI-2	IH-1	IH-2
Solomon_r105	5.0	-0.7	-1.5	2.9	2.7	1.9	0.3	-0.1	-0.4	-2.5	-0.2	0.8	-0.6	-1.7	2.4	1.3	0.7
Solomon_r106	0.0	-2.1	-4.4	-1.2	-2.9	-1.9	-3.4	-3.3	-0.9	-4.1	-0.2	-1.1	-3.0	-4.7	1.1	-3.1	-0.6
Solomon_r107	1.0	-2.8	-2.4	-3.9	-6.7	-3.8	-3.6	0.0	-4.0	-0.2	-0.5	-0.2	1.0	-1.6	2.2	-3.0	-4.5
Solomon_r108	1.0	-2.3	2.4	-2.0	-2.0	-3.4	2.5	-5.0	-0.4	-2.3	-2.2	-0.2	-3.2	2.5	-6.0	-4.0	-2.5
Solomon_r109	2.5	0.3	5.6	5.3	5.9	2.2	-0.2	1.7	0.3	2.0	1.0	2.5	3.8	-1.3	3.7	0.9	1.0
Solomon_r110	-3.2	-9.0	-6.0	-6.5	-5.1	-6.7	-6.6	-6.2	-6.2	-6.2	-6.4	-2.5	-2.7	-5.1	-5.8	-5.6	-6.9
Solomon_r111	3.4	1.0	2.5	7.0	-1.4	3.7	1.9	2.8	7.9	5.2	0.6	6.4	-1.3	5.0	5.0	4.1	2.7
Solomon_r112	6.5	0.2	-0.2	2.1	-0.4	1.4	1.6	-1.0	1.0	1.0	1.4	5.2	2.6	1.6	2.8	-0.8	1.5
Solomon_rc101	2.1	-1.2	-4.0	-5.4	-1.4	0.3	-4.1	-1.3	-1.9	-5.8	-1.4	-3.2	-2.9	-1.5	-0.7	-1.1	-2.8
Solomon_rc102	-3.4	-5.2	-2.2	1.9	-4.0	-3.5	-4.4	-5.5	-3.5	-2.5	-3.1	-4.7	-0.8	1.2	0.7	-11.1	-0.2
Solomon_rc103	2.9	-0.9	-2.8	6.0	-1.1	2.1	3.7	0.2	1.8	0.4	0.5	2.7	-1.5	-1.8	1.8	-1.3	-1.6
Solomon_rc104	1.2	-1.2	0.1	-2.0	-1.3	-3.7	-6.6	-3.2	-1.0	1.8	-4.0	-3.7	-3.2	-6.1	-8.1	-2.9	-5.1
Solomon_rc105	5.1	-1.5	2.8	1.6	-0.5	-1.5	0.7	1.6	2.3	-3.2	-0.7	-1.3	-3.7	3.5	-1.2	-1.7	1.2
Solomon_rc106	-0.6	-6.2	-4.8	-10.1	-4.6	-5.1	-7.8	-6.9	-3.8	-6.2	-4.9	-6.8	-6.9	-6.2	-2.7	-4.0	-9.4
Solomon_rc107	2.1	-2.6	-4.5	-1.4	-1.5	-4.3	-7.0	-3.1	-3.5	-4.9	-5.7	-4.7	-5.5	-1.6	0.6	-1.6	-3.0
Solomon_rc108	3.5	3.4	-0.7	1.8	0.7	3.7	2.1	5.4	0.8	2.2	-1.1	4.6	1.4	3.7	2.8	5.0	5.3
Average	2.3	-1.3	-1.0	-0.2	-1.1	-0.6	-1.3	-0.6	-0.6	-1.0	-0.8	0.1	-0.6	-0.5	0.0	-1.0	-0.6

Table A.17: Relative gap of the objective function value reported by mALNS (25,100,7500) by removing each operator individually with respect to one reported by mALNS (25,100,7500) for Class 2

Instance	EAS	RE	ES-1	ES-2	EH	IntraE	InterE	IntraR	InterR	IEF-1	IEF-2	IS-1	IS-2	IRDI-1	IRDI-2	IH-1	IH-2
Cordeau_pr01	0.7	-1.8	-1.7	0.5	-0.8	2.3	-1.5	1.8	2.8	-2.3	0.3	1.0	1.6	0.4	1.8	2.6	-0.9
Cordeau_pr02	1.6	0.1	0.4	0.5	-0.8	1.1	0.2	-0.1	1.2	-1.4	2.3	1.5	2.4	0.7	-0.1	1.1	-0.4
Cordeau_pr03	5.1	2.0	2.6	2.7	5.3	1.2	4.6	2.5	2.1	5.0	2.3	4.6	5.5	0.8	0.7	2.0	1.8
Cordeau_pr04	4.9	-0.8	0.4	0.7	-1.4	-2.9	1.5	2.7	2.3	2.7	2.2	-0.4	-1.4	1.7	-0.7	2.4	-0.1
Cordeau_pr05	7.6	0.1	1.5	1.2	-1.8	3.6	1.5	1.0	2.8	5.6	-0.8	0.0	-0.4	2.4	1.6	2.3	2.3
Cordeau_pr06	4.4	0.4	0.3	3.8	4.7	0.0	1.9	0.5	-0.9	4.1	-1.4	0.7	0.4	-0.4	1.4	-0.2	-0.8
Cordeau_pr07	0.4	-1.5	0.7	-0.2	-2.5	-2.8	2.3	-1.9	0.1	1.0	-1.5	0.5	0.2	-1.2	0.5	-2.2	-0.7
Cordeau_pr08	4.7	-1.9	-0.5	-2.0	-0.8	-0.9	-3.4	-2.6	-1.3	-2.0	3.5	-2.2	-3.0	-3.4	-3.2	-1.7	-1.9
Cordeau_pr09	2.7	-1.3	-1.0	-1.7	-0.5	0.1	0.3	-2.1	0.6	0.0	-1.1	-1.2	-0.6	-1.2	0.7	-1.0	-0.7
Cordeau_pr10	3.0	-2.2	2.4	-1.2	-0.7	0.7	2.4	-0.5	3.9	-0.7	-0.3	-0.8	-0.5	-0.7	-1.3	-1.3	-1.0
Solomon_c101	0.7	-2.1	-2.2	-2.7	-1.3	-1.8	0.4	-5.1	-2.3	-2.6	0.5	-1.6	1.2	-2.5	-2.0	-1.6	-1.0
Solomon_c102	3.4	-0.3	0.0	1.8	2.5	1.6	-0.1	1.9	2.2	0.8	1.1	0.5	2.1	-0.1	-0.9	0.9	3.8
Solomon_c103	2.1	-0.9	1.4	-2.0	-2.0	0.7	-1.8	-0.6	-0.2	0.4	3.3	2.9	1.4	0.6	-2.0	-1.1	-0.5
Solomon_c104	0.0	-3.4	-2.3	-1.2	-2.4	-0.8	-2.7	-1.2	-1.5	-0.5	0.0	-1.4	1.2	-0.8	-1.4	-0.8	1.2
Solomon_c105	3.3	1.7	1.2	-0.1	-1.0	0.5	-0.9	3.2	1.3	0.7	1.0	1.1	3.1	-0.7	2.0	1.9	-2.2
Solomon_c106	2.9	-0.3	0.3	2.7	4.7	0.4	2.3	0.3	4.5	0.6	-0.5	0.2	3.4	2.9	0.1	-1.6	0.2
Solomon_c107	3.5	0.3	-0.2	2.8	1.0	-1.7	0.5	2.6	1.5	-0.7	0.2	0.4	-1.1	0.7	1.4	-0.1	-1.5
Solomon_c108	2.0	-3.5	-1.3	-2.0	1.3	-3.2	-2.5	1.0	0.6	-1.5	0.6	-2.0	0.5	-0.3	-1.6	-0.9	0.3
Solomon_c109	4.6	-3.2	-0.5	-0.9	-0.6	0.2	-1.3	1.0	1.8	-2.2	2.4	2.1	-0.4	0.5	-0.4	-0.5	-0.9
Solomon_r101	1.0	-3.6	-2.9	-3.3	-4.5	-4.8	-2.2	-3.3	-5.5	-5.2	-1.2	-5.0	-4.9	-2.0	-2.8	-2.8	-1.9
Solomon_r102	2.7	1.7	-0.6	2.5	0.6	3.3	3.0	0.9	3.4	0.3	0.6	0.1	2.2	-0.1	3.4	2.2	0.1
Solomon_r103	1.1	-2.4	-0.6	-3.7	-3.4	-1.6	-3.3	-3.1	-2.7	-0.8	-1.9	0.6	-2.9	-4.1	-1.7	-1.4	-2.4
Solomon_r104	2.0	-4.0	-1.4	-3.5	-3.2	-1.7	-3.3	-5.0	0.3	-1.8	-3.9	-3.7	-2.4	-0.6	1.3	-1.8	-4.9

Continues on next page

Continued from previous page

Instance	EAS	RE	ES-1	ES-2	EH	IntraE	InterE	IntraR	InterR	IEF-1	IEF-2	IS-1	IS-2	IRDI-1	IRDI-2	IH-1	IH-2
Solomon_r105	1.1	-0.8	2.4	5.3	-1.3	-0.7	-0.1	1.9	2.1	3.4	2.1	1.8	2.6	1.5	-2.5	3.1	1.0
Solomon_r106	5.0	0.8	0.3	5.5	1.5	2.6	4.2	-0.6	1.9	5.2	1.8	3.8	4.5	1.6	1.3	1.1	1.4
Solomon_r107	4.3	-0.8	-0.1	0.1	-1.0	-1.6	-1.7	1.2	-1.2	-0.7	-1.2	1.8	0.6	-0.3	-0.4	-0.6	3.5
Solomon_r108	1.6	-0.5	-2.6	1.5	0.9	0.5	-0.4	-2.0	-1.8	-0.6	2.8	-0.3	-0.8	-0.8	-3.8	-2.4	-1.1
Solomon_r109	2.4	0.8	-1.5	0.6	-3.3	-2.6	2.2	0.2	2.8	-1.0	-0.1	-1.2	5.3	-1.0	-0.4	0.6	0.9
Solomon_r110	4.0	2.9	0.5	4.5	5.8	2.4	3.6	1.2	1.6	-0.5	-0.8	2.4	2.1	1.9	0.3	-0.4	-0.1
Solomon_r111	-0.4	-6.3	-3.4	-3.0	-4.6	-4.8	-2.1	-0.1	-0.8	0.5	-3.8	-0.9	-4.1	-4.8	-6.9	-2.6	-6.3
Solomon_r112	4.7	0.4	1.7	1.6	-1.0	0.2	0.8	-0.5	0.4	-1.9	-2.1	1.0	-1.0	0.7	0.8	-0.7	1.3
Solomon_rc101	0.4	0.1	-0.9	1.4	-0.5	-0.8	-1.6	-0.5	-1.4	1.9	0.2	-1.6	0.8	0.1	-2.6	-1.7	-1.0
Solomon_rc102	4.1	1.0	2.4	3.9	2.7	-2.2	0.5	4.6	2.3	3.4	2.9	-1.2	3.4	-0.5	0.9	0.9	0.3
Solomon_rc103	5.5	3.1	2.2	1.1	2.7	2.2	1.9	2.2	1.0	1.2	0.3	3.3	3.0	1.3	2.7	1.9	1.0
Solomon_rc104	4.1	4.7	-3.0	2.4	0.0	1.5	0.6	0.1	3.5	1.5	1.3	2.3	-1.4	-0.6	-2.2	2.2	4.2
Solomon_rc105	5.1	2.1	-0.1	0.8	5.6	0.0	1.1	0.9	-0.1	-0.3	5.8	1.2	1.7	-0.1	1.0	4.3	5.7
Solomon_rc106	3.8	1.2	1.0	-0.9	-0.6	1.4	-0.8	0.9	-0.2	-1.1	0.9	0.6	2.6	0.2	2.5	0.0	-1.3
Solomon_rc107	2.1	-2.1	-0.5	-1.9	-2.2	-1.7	1.1	1.7	-1.9	-2.0	-1.2	-3.5	-0.6	0.4	1.7	0.9	-1.5
Solomon_rc108	1.2	-2.1	-1.2	-1.1	-3.3	-1.6	-0.2	-2.4	1.4	-0.8	-2.5	2.5	-0.7	-3.7	-1.4	4.3	-3.8
Average	2.9	-0.6	-0.2	0.4	-0.2	-0.3	0.2	0.1	0.7	0.2	0.4	0.3	0.7	-0.3	-0.3	0.2	-0.2

Table A.18: Relative gap of the objective function value reported by mALNS (25,100,7500) by removing each operator individually with respect to one reported by mALNS (25,100,7500) for Class 3

Instance	EAS	RE	ES-1	ES-2	EH	IntraE	InterE	IntraR	InterR	IEF-1	IEF-2	IS-1	IS-2	IRDI-1	IRDI-2	IH-1	IH-2
Cordeau_pr01	3.4	2.1	2.3	3.7	1.9	3.5	3.0	2.7	6.4	2.4	3.5	3.8	3.2	2.7	4.9	0.7	3.5
Cordeau_pr02	1.9	-0.6	-1.3	-0.5	0.1	0.9	-0.5	-1.3	-0.4	-0.8	-1.0	-0.3	-0.7	-0.7	0.5	-1.0	-0.3
Cordeau_pr03	3.6	-0.7	-1.9	0.7	1.4	-1.8	-0.5	-1.5	-0.2	-1.4	-0.3	-0.4	-1.0	-2.3	-0.3	0.2	-0.7
Cordeau_pr04	6.3	0.0	0.3	1.7	2.0	4.0	2.3	0.5	1.9	2.0	1.0	6.0	3.5	-0.6	0.9	1.6	1.1
Cordeau_pr05	-0.3	-7.0	-3.0	-2.1	-5.7	0.8	-4.4	-1.4	-4.2	-4.5	-6.7	-0.6	-5.4	-3.6	-3.4	-4.4	-5.8
Cordeau_pr06	4.1	1.3	2.0	3.4	-0.4	1.6	0.2	0.4	3.3	2.0	0.8	1.4	2.6	0.2	1.5	2.1	1.5
Cordeau_pr07	1.9	1.1	0.1	0.4	1.8	1.8	2.1	0.9	1.1	-0.2	0.8	0.4	0.4	-1.0	-0.4	0.0	1.3
Cordeau_pr08	2.9	-0.4	1.2	1.9	-0.1	0.8	0.1	2.1	0.6	-0.9	0.6	3.4	2.1	-0.6	0.2	0.2	0.3
Cordeau_pr09	2.5	0.0	1.9	-0.5	0.1	-0.2	0.5	-0.3	0.2	-1.1	0.3	-0.7	0.6	0.1	0.8	-0.3	-0.2
Cordeau_pr10	3.0	0.0	-2.0	0.7	0.4	0.7	0.0	-0.1	0.1	2.0	0.1	0.9	0.6	2.1	0.0	2.5	-0.1
Solomon_c101	4.0	1.0	3.1	2.0	4.4	1.4	3.0	2.3	1.4	3.1	2.4	-0.1	2.0	0.1	-0.6	0.8	1.0
Solomon_c102	3.9	0.9	-0.1	0.5	0.1	-0.2	0.7	2.3	1.1	3.3	0.7	1.6	1.2	0.3	2.7	2.4	3.5
Solomon_c103	2.0	-0.4	-0.7	-1.4	-0.5	0.8	0.0	1.7	0.8	-0.6	0.3	0.2	-0.8	0.5	-0.2	1.7	-0.1
Solomon_c104	3.0	3.0	-0.6	1.5	-1.2	0.4	3.1	2.1	-0.3	0.8	0.1	3.4	1.8	3.1	2.9	-0.2	0.9
Solomon_c105	0.6	-0.2	-1.6	-2.7	0.9	-3.6	0.1	0.2	0.6	0.8	-1.9	-0.8	0.3	-0.9	-2.0	-0.7	-1.8
Solomon_c106	3.4	-0.4	2.0	0.6	0.2	0.6	0.2	0.5	-0.7	-0.3	0.7	0.5	1.6	2.7	-0.7	2.5	0.9
Solomon_c107	3.9	1.2	0.2	0.4	0.1	0.8	1.2	0.1	1.7	-0.1	1.1	0.8	0.6	0.7	1.1	1.8	-0.1
Solomon_c108	1.5	0.2	1.1	0.9	2.8	1.7	0.5	-0.3	0.8	1.6	3.0	-0.4	0.2	0.7	1.4	0.5	2.0
Solomon_c109	3.2	-1.7	-0.9	-1.8	-1.5	-1.0	-1.2	0.1	-0.2	1.3	-0.5	-1.4	1.1	-1.1	1.1	-1.8	-0.4
Solomon_r101	2.9	0.2	0.7	4.1	2.2	-1.2	-0.7	1.4	-1.2	-1.6	0.5	-0.3	-0.8	-0.3	-2.4	-1.5	-0.1
Solomon_r102	3.0	0.7	2.3	0.4	-0.7	0.1	1.0	2.6	3.7	0.4	0.7	0.6	2.7	1.7	2.9	2.0	0.7
Solomon_r103	3.3	0.2	-0.1	4.2	-0.1	1.7	-0.6	-0.5	3.1	0.2	3.3	-0.4	2.2	3.4	1.1	0.0	3.9
Solomon_r104	0.4	2.6	-3.0	0.2	-2.1	1.8	-0.6	3.0	3.7	-0.7	-5.1	0.9	1.0	1.2	3.1	0.4	0.0

Continues on next page

Continued from previous page

Instance	EAS	RE	ES-1	ES-2	EH	IntraE	InterE	IntraR	InterR	IEF-1	IEF-2	IS-1	IS-2	IRDI-1	IRDI-2	IH-1	IH-2
Solomon_r105	1.2	-0.2	-4.8	-0.4	-1.0	-1.8	-2.9	0.9	0.2	-3.0	-0.8	1.2	-1.6	-0.4	-1.8	-2.6	-3.7
Solomon_r106	1.7	-3.0	-2.2	-1.9	-2.6	-2.3	0.0	-0.9	-1.7	1.4	0.3	0.8	-0.2	0.1	-0.1	-2.4	0.7
Solomon_r107	3.1	3.4	-0.7	0.8	-0.2	1.7	2.8	3.0	3.5	1.2	-0.5	3.8	0.5	2.0	-0.4	-0.2	0.3
Solomon_r108	-2.2	-0.1	-3.5	-6.9	-4.0	-3.9	-3.5	-1.6	-4.1	-6.5	-6.4	-5.9	-3.5	-7.5	-5.4	-3.2	-7.2
Solomon_r109	3.3	-0.1	2.6	2.3	-0.5	-1.5	0.6	2.1	-0.7	-0.1	-0.2	-0.5	-0.4	-1.1	-0.6	-0.2	0.5
Solomon_r110	4.0	0.4	0.8	1.6	0.2	3.2	2.7	1.8	-0.6	4.4	0.8	2.2	1.9	0.2	0.8	1.1	3.7
Solomon_r111	4.0	5.1	1.0	5.5	-0.5	-0.4	3.7	1.9	0.6	3.9	-0.9	2.9	2.8	0.2	2.8	0.6	1.9
Solomon_r112	0.8	0.2	-0.8	0.0	0.3	-0.9	-1.4	-0.2	-2.2	-1.4	-2.3	0.0	1.5	-0.3	0.1	-1.4	-1.8
Solomon_rc101	0.1	-2.9	-2.0	-0.2	-2.6	-2.1	-4.0	-3.0	-2.4	-2.3	-1.4	-1.5	0.7	-0.6	-0.5	-2.0	-0.2
Solomon_rc102	3.8	2.4	1.3	2.4	3.7	1.0	5.8	1.7	5.2	3.2	1.8	1.0	4.0	-1.1	1.2	5.7	0.3
Solomon_rc103	0.0	-2.7	-3.7	-0.1	-1.7	-0.7	-0.9	1.2	-1.4	2.8	2.5	-1.3	-2.1	1.0	-0.4	1.3	-1.8
Solomon_rc104	8.2	2.8	5.4	2.4	2.3	2.5	3.3	2.5	2.7	1.2	5.2	6.6	3.0	4.2	2.2	3.7	2.8
Solomon_rc105	4.4	1.5	-0.7	-0.2	-1.5	1.0	-1.2	0.8	1.4	0.1	0.0	1.9	-0.9	-0.3	-1.7	0.8	0.9
Solomon_rc106	6.6	0.6	1.9	4.3	-0.3	-0.9	1.8	3.5	3.7	0.2	4.4	4.5	4.6	3.6	1.3	0.8	3.7
Solomon_rc107	0.2	-0.8	-3.5	-2.2	-3.2	-2.3	-1.1	-0.6	-3.2	-4.3	-1.7	1.9	0.8	-2.2	-4.6	-0.3	-2.0
Solomon_rc108	-0.5	-2.3	-3.4	-3.6	-1.7	-3.6	-0.5	-4.0	-0.1	2.1	-3.4	-3.6	-6.0	0.5	-1.8	0.3	-3.0
Average	2.6	0.2	-0.3	0.6	-0.2	0.1	0.4	0.7	0.6	0.3	0.1	0.8	0.6	0.2	0.2	0.3	0.2

Table A.19: Relative gap of the objective function value reported by mALNS (25,100,7500) by removing each operator individually with respect to one reported by mALNS (25,100,7500) for Class 4

Instance	EAS	RE	ES-1	ES-2	EH	IntraE	InterE	IntraR	InterR	IEF-1	IEF-2	IS-1	IS-2	IRDI-1	IRDI-2	IH-1	IH-2
Cordeau_pr01	-0.3	-0.4	-1.0	0.4	0.7	1.7	0.2	-0.2	-1.1	-1.3	-1.0	-0.2	0.5	-0.7	-2.4	0.2	0.0
Cordeau_pr02	2.6	-0.5	-0.7	-0.4	-0.1	0.9	-0.3	-0.3	0.6	0.6	0.5	-1.8	-0.4	0.2	0.5	0.4	0.0
Cordeau_pr03	1.4	1.2	-1.7	-1.2	-2.2	-0.9	-1.3	0.8	-2.0	-2.1	-1.2	-2.8	-0.9	-0.9	1.3	0.2	-2.1
Cordeau_pr04	3.9	-0.7	0.0	0.9	0.8	-0.1	1.6	-0.2	2.6	2.2	2.7	0.8	-1.2	2.8	0.5	-0.4	1.2
Cordeau_pr05	6.3	-2.1	-2.3	0.1	2.8	-4.5	-1.0	0.0	2.9	-1.0	-1.0	-2.7	-4.3	-0.5	1.6	-2.4	-0.5
Cordeau_pr06	2.5	-0.5	0.4	-1.5	0.0	0.0	0.9	1.8	0.0	0.8	2.1	0.7	0.1	3.6	0.6	0.3	2.5
Cordeau_pr07	1.4	-2.2	-2.7	-2.3	-0.2	-0.2	-1.9	1.0	-0.6	-1.2	1.2	-2.4	-1.5	-1.2	-1.9	-1.3	-2.2
Cordeau_pr08	6.5	1.5	-1.1	2.5	2.6	1.9	-1.2	2.1	2.2	0.7	1.6	-0.5	1.3	-1.8	0.1	-0.2	1.2
Cordeau_pr09	3.4	-0.5	-1.7	0.2	0.1	-0.5	-0.1	0.0	0.1	0.5	0.7	-0.4	-0.9	-0.7	-1.7	-1.0	-0.3
Cordeau_pr10	5.0	0.4	0.0	1.5	1.1	1.6	2.0	0.9	2.6	0.9	2.5	0.7	1.4	0.9	-0.3	1.3	0.4
Solomon_c101	1.2	-5.2	-1.9	-3.6	-0.6	-5.8	0.7	-4.3	-3.3	-0.5	-4.7	-5.2	-3.8	-1.0	-3.2	-2.8	2.5
Solomon_c102	0.1	0.9	-0.1	0.6	-0.8	-1.6	0.4	0.5	0.4	-0.8	0.2	-1.5	-0.7	0.1	-0.4	-0.7	-0.3
Solomon_c103	2.9	1.4	2.7	-0.2	-0.4	-0.1	-0.3	-0.1	1.1	1.4	0.5	3.8	0.0	1.0	-1.0	0.4	1.7
Solomon_c104	1.4	1.6	0.8	0.7	-0.8	3.0	1.3	1.2	-1.1	2.0	2.5	0.1	0.2	1.0	2.7	0.1	0.7
Solomon_c105	2.4	-4.4	-2.5	-4.1	-3.2	-1.2	-2.0	-2.0	-3.0	-3.3	-1.0	-3.4	-5.6	-2.8	-2.3	-2.8	-2.0
Solomon_c106	2.5	0.0	-1.0	-2.2	-1.9	-2.1	-1.6	-0.1	-2.4	-1.8	0.6	-1.0	-1.0	0.3	-1.7	-1.4	2.2
Solomon_c107	6.0	1.6	1.3	2.2	2.5	5.8	2.3	-0.7	3.0	-1.0	1.3	3.6	0.7	0.3	-0.1	2.1	3.4
Solomon_c108	3.5	-1.1	0.3	-1.2	-1.5	-0.2	0.2	0.9	0.1	1.3	2.2	2.1	0.5	0.7	0.0	0.3	0.1
Solomon_c109	3.3	0.4	1.0	-0.1	0.3	0.4	0.0	1.6	1.6	-0.6	2.7	0.6	1.1	1.3	-1.1	2.1	-0.8
Solomon_r101	4.6	-1.7	-2.3	-1.2	3.0	4.1	1.2	-0.1	4.8	4.7	4.6	2.4	0.4	0.2	2.0	2.8	3.1
Solomon_r102	3.0	1.7	0.9	0.9	0.7	2.4	0.8	0.5	1.1	0.9	1.7	1.4	1.8	0.7	0.0	1.2	1.2
Solomon_r103	3.9	-0.7	-0.7	-1.4	0.0	-0.7	-1.0	0.9	0.0	-0.2	-0.7	-1.8	-1.7	-0.4	-0.3	-0.4	-0.5
Solomon_r104	5.9	-1.2	1.9	2.2	-3.3	-0.3	2.0	-2.7	-2.6	-0.9	-1.8	-0.4	0.2	0.0	-0.1	-1.5	-1.1

Continues on next page

Continued from previous page

Instance	EAS	RE	ES-1	ES-2	EH	IntraE	InterE	IntraR	InterR	IEF-1	IEF-2	IS-1	IS-2	IRDI-1	IRDI-2	IH-1	IH-2
Solomon_r105	3.3	-2.6	-2.4	1.4	-1.1	2.1	3.7	2.0	1.5	0.6	-1.0	0.9	-2.1	2.2	1.2	-2.2	4.7
Solomon_r106	5.3	-0.9	-3.5	-0.4	-4.2	-1.4	0.5	-1.0	-0.1	-1.4	-2.1	-1.9	-2.9	2.3	2.2	0.1	-1.2
Solomon_r107	2.8	-2.2	-2.6	-2.4	0.2	1.2	-3.1	-1.1	-2.0	-2.6	-0.9	2.0	-0.6	-2.1	-0.9	-3.4	0.2
Solomon_r108	6.4	0.4	1.5	3.9	2.1	1.2	2.3	2.4	0.3	2.9	3.5	-0.2	3.4	2.4	3.1	2.7	1.6
Solomon_r109	4.3	-1.1	0.0	0.6	1.8	-0.2	0.2	-1.8	0.6	2.6	5.0	-0.2	-1.6	0.3	-0.9	-1.3	1.6
Solomon_r110	5.0	-1.0	0.5	0.5	0.1	-1.9	4.6	-0.3	-0.2	1.1	1.2	-0.2	2.2	0.9	1.0	-1.1	-0.7
Solomon_r111	6.8	0.3	-0.2	-0.5	-0.1	1.3	2.0	5.5	0.8	6.7	0.2	1.2	1.3	3.3	3.1	0.4	1.3
Solomon_r112	6.3	3.5	0.3	2.9	3.9	0.8	-0.1	1.2	2.6	2.5	2.2	0.7	1.5	1.6	3.8	1.4	3.8
Solomon_rc101	2.0	1.3	-0.7	2.0	-0.4	-0.8	-2.0	1.8	1.1	-2.1	1.0	-1.2	-1.2	-1.5	-2.8	-1.4	-0.2
Solomon_rc102	4.4	-1.0	-0.5	-0.8	-0.7	-0.9	1.1	0.2	1.1	0.3	-1.3	-0.6	1.2	-1.3	0.8	1.0	0.6
Solomon_rc103	2.7	-2.3	-0.5	-0.5	-0.1	0.9	-1.5	-1.1	1.2	-1.6	-0.3	4.1	1.5	-1.6	-1.4	-2.1	-0.7
Solomon_rc104	4.2	-1.9	-3.1	-0.8	-2.6	0.9	0.0	1.6	1.8	-1.9	-1.0	-2.2	-5.9	-3.1	-2.8	-1.6	-4.2
Solomon_rc105	4.9	-2.0	2.5	-1.4	-0.5	0.0	-3.1	0.1	0.1	-1.6	-3.1	2.0	2.3	-1.0	1.5	-1.5	1.2
Solomon_rc106	5.7	-0.9	-1.3	0.4	0.3	0.2	-0.5	0.5	-0.3	5.0	3.5	1.1	0.6	1.2	1.5	-0.4	0.6
Solomon_rc107	2.1	-3.4	-1.7	-1.5	-2.3	0.0	-1.1	-0.3	-0.8	-2.1	-1.8	0.8	-1.5	-0.4	1.3	-2.0	-1.3
Solomon_rc108	0.7	2.8	-3.2	0.2	-6.5	-2.4	0.8	0.7	-0.4	1.8	0.6	0.6	-0.7	-2.6	-2.8	-3.2	-1.0
Average	3.6	-0.5	-0.7	-0.1	-0.3	0.1	0.2	0.3	0.4	0.3	0.6	0.0	-0.4	0.1	0.0	-0.5	0.4

Table A.20: Relative gap of the objective function value reported by mALNS (25,100,7500) by removing each operator individually with respect to the one reported by mALNS (25,100,7500) for Class 5

Instance	EAS	RE	ES-1	ES-2	EH	IntraE	InterE	IntraR	InterR	IEF-1	IEF-2	IS-1	IS-2	IRDI-1	IRDI-2	IH-1	IH-2
Cordeau_pr01	1.7	1.3	-0.5	1.1	0.5	0.4	0.9	0.0	0.7	-1.0	1.7	1.4	0.5	1.5	0.1	0.8	1.7
Cordeau_pr02	3.5	-0.6	-0.2	-1.6	1.5	-0.4	0.3	1.2	-0.4	-0.2	0.0	-0.5	1.3	0.3	-0.5	-0.3	1.1
Cordeau_pr03	0.7	-2.3	-0.9	-1.3	-1.5	-1.2	-1.6	-0.4	-1.8	-1.6	-1.6	-1.8	-1.6	-1.8	1.6	-1.6	-1.5
Cordeau_pr04	4.0	-0.6	0.3	-1.3	2.9	1.8	1.0	0.3	1.6	-0.2	-0.7	1.7	0.8	-0.5	1.8	2.0	-0.2
Cordeau_pr05	2.0	-3.4	-0.3	1.1	-1.6	0.2	-3.5	-1.5	-0.9	-2.1	-1.3	-0.7	0.1	0.3	-1.6	-2.1	-0.4
Cordeau_pr06	1.0	-1.4	-0.9	-0.4	-1.6	-0.2	0.1	0.4	-0.3	-0.6	1.1	-0.2	0.5	-0.8	-0.8	-0.2	-0.8
Cordeau_pr07	3.3	0.5	0.2	-1.5	-0.8	-1.5	0.5	0.4	-0.1	-0.7	0.7	-0.9	0.3	-0.1	-1.4	-2.4	-0.2
Cordeau_pr08	1.8	-1.3	-0.8	0.3	-0.3	0.3	-0.2	0.2	0.0	-2.2	-0.1	-0.8	3.0	-0.1	-0.8	-1.3	1.2
Cordeau_pr09	3.6	0.3	0.6	1.7	0.4	1.1	2.2	0.9	2.2	1.1	1.3	0.9	1.6	0.3	1.3	0.7	1.6
Cordeau_pr10	1.9	-1.5	-2.1	-1.0	-1.5	-0.4	-0.5	-1.9	0.9	-1.4	-0.7	-2.9	-0.2	1.4	0.2	-0.2	-1.3
Solomon_c101	4.5	0.2	0.0	0.8	-1.0	0.9	2.2	0.3	2.0	2.5	-1.3	0.2	0.6	-0.1	1.2	1.0	-0.1
Solomon_c102	3.0	0.0	-0.2	0.9	0.1	0.7	1.0	0.2	0.5	1.4	1.4	1.5	3.5	0.3	0.2	1.3	2.6
Solomon_c103	3.0	-0.7	1.9	2.5	-1.1	0.5	-0.3	1.8	0.5	1.4	0.4	0.9	0.6	3.1	0.5	-0.7	1.2
Solomon_c104	-0.3	0.7	2.2	1.8	-0.4	0.6	0.5	0.8	-0.4	0.8	2.3	0.2	-0.6	2.2	1.8	1.0	1.2
Solomon_c105	4.7	0.4	0.7	1.1	0.0	-1.2	2.1	0.4	1.2	1.2	2.0	0.9	0.8	3.2	1.5	0.8	3.0
Solomon_c106	3.0	-0.9	-3.1	-2.8	-2.2	0.0	-0.5	0.5	-0.9	-1.1	0.9	-0.8	-0.2	0.2	-0.9	-1.4	1.3
Solomon_c107	2.1	-1.4	-1.7	-0.8	-0.2	-2.1	-2.8	-0.3	0.0	-0.2	0.5	0.7	-1.5	-0.3	1.4	-2.4	0.3
Solomon_c108	2.8	0.0	0.8	1.4	0.3	-0.1	0.3	0.3	0.2	3.9	1.0	1.4	0.0	0.6	1.3	0.2	2.2
Solomon_c109	4.3	1.4	0.5	1.2	4.0	0.8	1.1	2.4	0.8	1.2	1.5	2.6	1.3	1.5	1.4	0.7	1.0
Solomon_r101	2.0	-0.5	1.7	-1.3	0.2	1.5	1.2	1.0	3.6	-0.3	1.0	0.7	1.1	3.6	0.1	2.3	1.3
Solomon_r102	0.4	-0.8	-0.2	0.3	-0.8	-1.9	-1.3	-0.4	-0.2	0.2	0.1	-0.7	-0.7	0.7	-0.2	-0.6	-0.2
Solomon_r103	1.6	-1.9	-1.2	-0.4	-0.5	0.6	-0.9	-0.9	0.8	0.7	0.1	-0.9	0.1	-1.5	-1.2	-0.2	-0.1
Solomon_r104	2.4	-0.2	0.4	-0.3	-2.0	2.7	2.6	3.4	1.0	1.5	1.1	0.5	1.4	3.3	2.3	1.3	1.2

Continues on next page

Continued from previous page

Instance	EAS	RE	ES-1	ES-2	EH	IntraE	InterE	IntraR	InterR	IEF-1	IEF-2	IS-1	IS-2	IRDI-1	IRDI-2	IH-1	IH-2
Solomon_r105	3.6	-0.4	-0.2	-0.5	-0.8	-0.2	1.6	-1.5	1.4	-1.5	-0.8	-0.9	1.7	3.5	1.1	2.9	-0.6
Solomon_r106	3.0	-2.1	-1.2	-0.4	-2.3	0.1	1.6	0.8	-0.4	-0.7	-1.8	-1.1	-0.8	-1.8	-0.9	-1.7	-2.7
Solomon_r107	-0.9	-1.1	-2.9	1.6	-1.1	-2.7	-2.8	-1.9	0.8	-2.4	-1.4	-2.0	-2.9	-2.6	-0.7	-2.4	-2.7
Solomon_r108	1.7	-1.4	1.4	0.7	1.0	0.1	0.5	1.2	-2.2	-0.5	-1.8	4.4	0.9	1.5	-1.0	0.8	-1.2
Solomon_r109	-0.4	1.3	-0.8	0.8	0.3	0.7	0.2	-1.2	-0.8	-0.2	-1.2	-0.4	-0.5	-0.6	-1.1	-1.7	-0.9
Solomon_r110	5.2	1.7	1.1	2.6	0.8	1.2	0.7	1.5	1.9	0.1	0.8	0.8	4.9	0.8	4.1	0.7	1.5
Solomon_r111	2.9	0.5	2.8	2.3	4.1	0.9	3.4	2.4	4.6	1.0	-0.4	1.2	0.6	1.1	3.7	4.3	0.8
Solomon_r112	2.8	0.0	0.7	0.8	2.6	1.6	1.0	-0.3	-0.5	0.4	-0.5	-0.8	1.7	1.5	3.4	-0.9	-1.3
Solomon_rc101	0.5	-3.5	-2.3	-2.9	-1.4	-1.0	-0.6	-3.3	-3.1	0.4	-2.1	-1.9	-3.2	-1.8	-1.6	-2.3	-2.7
Solomon_rc102	2.4	-2.6	-0.2	-0.9	-1.7	-0.6	-0.9	-0.3	0.7	-1.6	-0.1	2.1	-1.0	-0.8	-1.4	0.9	0.6
Solomon_rc103	2.9	1.1	1.6	2.4	0.3	1.6	0.7	0.7	2.6	1.1	1.0	0.3	0.4	0.8	1.0	1.9	-0.5
Solomon_rc104	1.5	-2.8	-2.1	0.1	-1.8	-0.4	-2.5	-2.9	0.1	-1.2	-1.0	-1.7	-0.6	-3.1	2.0	-2.2	-3.9
Solomon_rc105	3.8	-1.1	-1.9	-1.0	-1.9	-1.2	1.1	0.4	1.0	0.7	-0.6	-0.5	-0.9	-0.7	-0.5	-0.5	0.3
Solomon_rc106	-0.3	-1.4	-1.6	-0.7	-2.2	-1.0	-0.3	-1.7	-1.1	-1.2	-0.7	-1.0	1.4	-1.2	-0.9	-1.1	1.0
Solomon_rc107	3.1	-1.1	-1.0	0.2	-0.1	-0.5	-1.6	2.3	-2.1	0.3	1.9	-0.3	-0.1	-2.1	-0.4	-1.3	-1.3
Solomon_rc108	4.0	2.7	-0.4	4.1	2.7	0.4	0.7	-1.2	-0.5	2.2	3.1	3.0	3.9	0.2	0.5	0.1	-1.1
Average	2.4	-0.6	-0.3	0.3	-0.2	0.1	0.2	0.1	0.4	0.0	0.2	0.1	0.5	0.3	0.4	-0.1	0.0

Tables A.21–A.25 display the objective function values of the solutions reached by mALNS (25,100,7500) by removing more than one operator at a time per instance.

Table A.21: Objective function values reported by mALNS (25,100,7500) by removing more than one operator at a time for Class 1

Instance	**RE, ES-1**	**RE, ES-1, EH**	**RE, ES-1, EH, IH-1**	**RE, ES-1, EH, IH-1, IntraE**	**RE, ES-1, EH, IH-1, IntraE, InterE**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1**
Cordeau_pr01	569.75	566.00	573.38	544.75	568.318	579.873	552.833	585.727
Cordeau_pr02	1154.00	1158.00	1136.00	1149.00	1147	1153	1166	1176
Cordeau_pr03	1538.63	1582.23	1570.83	1584.48	1588.97	1571.66	1614.03	1641.43
Cordeau_pr04	2086.15	2105.18	2053.49	2157.30	2185.26	2010.92	2136.61	2126.66
Cordeau_pr05	2954.25	2773.53	2887.65	2921.22	2927.17	2955.33	2906.31	2982.14
Cordeau_pr06	3542.14	3526.17	3540.48	3562.95	3553.6	3575.85	3520.59	3521.71
Cordeau_pr07	864.19	813.04	825.19	842.09	847.774	816.135	840.893	859.479
Cordeau_pr08	1861.51	1869.27	1827.54	1844.74	1880.81	1900.28	1885.55	1834.12
Cordeau_pr09	2679.18	2679.75	2706.63	2683.20	2713	2690.01	2701.47	2709.1
Cordeau_pr10	3714.69	3758.78	3729.55	3682.54	3697.98	3713.18	3716.36	3760.63
Solomon_c101	1469.74	1432.97	1450.23	1493.45	1491.28	1439.34	1423.28	1473.29
Solomon_c102	1620.00	1592.38	1597.56	1589.41	1633.09	1621.83	1629.27	1614.27
Solomon_c103	1590.84	1629.65	1646.30	1631.68	1627.17	1645.76	1654.82	1627.41
Solomon_c104	1518.84	1583.56	1555.70	1539.96	1540.34	1523.77	1561.89	1508.79
Solomon_c105	1491.30	1400.23	1484.23	1462.56	1425.24	1418.69	1419.97	1436.76
Solomon_c106	1568.49	1635.74	1622.29	1584.64	1604.99	1622.58	1590.51	1569.58
Solomon_c107	1643.30	1630.77	1650.39	1641.43	1649.26	1671.25	1665.61	1667.11
Solomon_c108	1646.94	1659.24	1606.94	1665.37	1680.52	1688.32	1663.07	1619.87
Solomon_c109	1558.40	1559.71	1589.44	1607.96	1573.05	1559.12	1592.09	1603.02
Solomon_r101	1288.83	1293.54	1280.18	1307.77	1284.15	1289.2	1267.57	1302.83
Solomon_r102	1418.00	1414.80	1409.69	1419.51	1411.06	1426.49	1416	1410.77
Solomon_r103	1383.72	1390.78	1384.29	1377.43	1403.18	1404.82	1405.38	1346.66
Solomon_r104	1191.34	1188.31	1184.40	1173.47	1244.86	1173.93	1199.09	1143.83
Solomon_r105	1200.69	1195.15	1210.59	1210.45	1187.07	1212.61	1216.83	1245.04
Solomon_r106	1370.60	1357.22	1359.05	1382.41	1351.7	1350.55	1387.07	1376.18

Continues on next page

Continued from previous page

Instance	**RE, ES-1**	**RE, ES-1, EH**	**RE, ES-1, EH, IH-1**	**RE, ES-1, EH, IH-1, IntraE**	**RE, ES-1, EH, IH-1, IntraE, InterE**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1**
Solomon_r107	1309.67	1316.79	1304.13	1336.98	1327.56	1337.18	1323.37	1319.31
Solomon_r108	1182.59	1201.68	1198.95	1190.36	1146.84	1155.03	1208.38	1153.74
Solomon_r109	1328.75	1356.10	1370.47	1247.52	1321.7	1362.01	1349.27	1372.93
Solomon_r110	1282.21	1322.24	1270.90	1302.34	1314.28	1328.11	1286.63	1264.16
Solomon_r111	1206.92	1161.30	1174.93	1213.92	1214.78	1211.58	1232.58	1228.23
Solomon_r112	1309.00	1347.00	1318.00	1309.00	1333	1345	1348	1346.27
Solomon_rc101	1403.01	1422.32	1458.28	1459.11	1435.39	1462.07	1388.52	1479.15
Solomon_rc102	1500.37	1518.83	1565.86	1501.19	1545.85	1579.12	1492.8	1519.96
Solomon_rc103	1548.19	1522.54	1417.75	1594.71	1521.24	1509.54	1537.64	1549.38
Solomon_rc104	1437.70	1386.65	1467.99	1443.34	1465.79	1434.21	1452.03	1437.78
Solomon_rc105	1589.98	1524.69	1564.93	1596.41	1637.17	1593	1620.31	1587
Solomon_rc106	1483.33	1527.27	1513.59	1491.60	1477.89	1537.44	1456.69	1466.19
Solomon_rc107	1491.01	1570.74	1516.32	1513.32	1515.12	1471.42	1549.19	1527.41
Solomon_rc108	1321.88	1440.07	1420.31	1404.90	1432.92	1412.13	1401.87	1435.46

Table A.22: Objective function values reported by mALNS (25,100,7500) by removing more than one operator at a time for Class 2

Instance	**RE, ES-1**	**RE, ES-1, EH**	**RE, ES-1, EH, IH-1**	**RE, ES-1, EH, IH-1, IntraE**	**RE, ES-1, EH, IH-1, IntraE, InterE**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1**
Cordeau_pr01	573.94	559.93	545.328	558.506	553.759	570.668	566.994	527.474
Cordeau_pr02	1064.94	1055.09	1055.51	1068.89	1062.88	1067.67	1057.51	1073.89
Cordeau_pr03	1549.05	1556.16	1520.94	1539.3	1536.63	1571.58	1536.26	1554.8
Cordeau_pr04	2026.79	2008.96	2034.72	1960.02	2069.18	2037.76	2034.97	2017.14
Cordeau_pr05	2884.16	2898.40	2827.1	2873.1	2876.75	2884.87	2895.87	2883.48
Cordeau_pr06	3131.34	3171.14	3186.95	3176.7	3179.6	3164.74	3186.39	3122.31
Cordeau_pr07	800.98	778.61	785.068	789.141	795.771	820.228	814.748	806.309
Cordeau_pr08	1725.35	1714.52	1761.27	1768.31	1746.99	1745.86	1757.31	1746.38
Cordeau_pr09	2412.07	2438.80	2421.98	2453.49	2412.17	2455.85	2453.75	2452.94
Cordeau_pr10	3470.76	3430.98	3485.99	3420.93	3466.73	3463.59	3469.97	3414.35
Solomon_c101	1414.10	1440.25	1434.18	1440.87	1437.62	1424.18	1443.67	1424.92
Solomon_c102	1522.04	1545.37	1551.78	1559.97	1557.67	1551.09	1576.8	1552.67
Solomon_c103	1510.42	1520.19	1541.24	1517.76	1547.86	1524.35	1532.3	1553.38
Solomon_c104	1476.98	1482.33	1480.2	1502.41	1477.58	1501.53	1532.8	1486
Solomon_c105	1450.62	1439.40	1407.74	1450.94	1424.32	1419.61	1427.31	1411.54
Solomon_c106	1523.19	1545.45	1539.28	1531.54	1558.28	1558.76	1527.95	1516.54
Solomon_c107	1578.42	1572.39	1564.32	1569.31	1581.44	1578.94	1581.51	1549.17
Solomon_c108	1569.71	1599.80	1593.18	1603.97	1602.41	1582.99	1594.1	1587.29
Solomon_c109	1531.81	1550.05	1536.07	1583.42	1548.48	1537.52	1553.21	1563.29
Solomon_r101	1250.56	1232.64	1210.3	1231.3	1203.69	1253.29	1223.45	1216.63
Solomon_r102	1321.48	1306.87	1325.08	1319.61	1309.55	1319.61	1312.76	1321.3
Solomon_r103	1273.06	1255.60	1288.42	1289.63	1294.48	1282.71	1298.86	1283.47
Solomon_r104	1128.69	1154.94	1159.9	1176.16	1181.91	1182	1164	1159.34
Solomon_r105	1181.57	1193.43	1207.85	1193.51	1170.77	1215.14	1195	1209.93
Solomon_r106	1311.28	1309.80	1294.53	1314.15	1329.61	1306.58	1311.37	1318.29
Solomon_r107	1222.92	1229.16	1238.48	1232.08	1233.58	1211.11	1229.25	1247.21
Solomon_r108	1175.04	1159.42	1194.6	1114.8	1161.69	1153.48	1174.96	1140.64
Solomon_r109	1228.13	1246.21	1241.09	1267.25	1249.81	1250.19	1243.18	1245.9

Continues on next page

Continued from previous page

Instance	RE, ES-1	RE, ES-1, EH	RE, ES-1, EH, IH-1	RE, ES-1, EH, IH-1, IntraE	RE, ES-1, EH, IH-1, IntraE, InterE	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1
Solomon_r110	1214.16	1218.44	1222.07	1234.46	1223.68	1227.96	1231.4	1234.84
Solomon_r111	1156.19	1140.53	1197.79	1182.87	1198.78	1195.85	1153.35	1188.44
Solomon_r112	1223.55	1235.29	1234.15	1231.48	1237.22	1242.62	1230.86	1227.49
Solomon_rc101	1408.56	1349.88	1388.49	1356.25	1375.86	1389.86	1404.66	1381.62
Solomon_rc102	1488.62	1427.78	1463.84	1461.97	1503.13	1443.45	1485.4	1460.7
Solomon_rc103	1460.94	1436.00	1441.86	1448.89	1458.62	1406.04	1469.22	1441.69
Solomon_rc104	1387.79	1429.80	1415.55	1348.12	1417.83	1394.92	1378.21	1410.57
Solomon_rc105	1548.51	1555.77	1543.51	1513.24	1561.85	1560.26	1548.92	1572.46
Solomon_rc106	1399.27	1395.79	1375.36	1382.7	1392.09	1406.84	1430.46	1413.72
Solomon_rc107	1440.11	1466.06	1434.98	1440.78	1449.75	1429.87	1457.85	1446.5
Solomon_rc108	1343.68	1382.53	1387.93	1415.68	1381.46	1348.12	1407.99	1397.51

Table A.23: Objective function values reported by mALNS (25,100,7500) by removing more than one operator at a time for Class 3

Instance	RE, ES-1	RE, ES-1, EH	RE, ES-1, EH, IH-1	RE, ES-1, EH, IH-1, IntraE	RE, ES-1, EH, IH-1, IntraE, InterE	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1
Cordeau_pr01	552.30	541.702	574.804	536.762	551.562	556.3	540.648	566.246
Cordeau_pr02	1053.65	1057.02	1059.67	1035.89	1057.94	1052.8	1052.63	1070.53

Continues on next page

Continued from previous page

Instance	RE, ES-1	RE, ES-1, EH	RE, ES-1, EH, IH-1	RE, ES-1, EH, IH-1, IntraE	RE, ES-1, EH, IH-1, IntraE, InterE	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1
Cordeau_pr03	1532.39	1529.65	1518.41	1563.26	1534.68	1541.48	1570.04	1518.46
Cordeau_pr04	1957.03	1990.5	2000.04	1986.47	2027.37	2003.68	2001.91	2013.2
Cordeau_pr05	2793.29	2830.99	2816.9	2857.17	2878.14	2846.84	2852.12	2802.01
Cordeau_pr06	3087.36	3106.72	3113.34	3097.47	3128.29	3142.44	3132.84	3072.17
Cordeau_pr07	783.07	791.537	794.292	787.271	799.219	806.064	784.441	784.699
Cordeau_pr08	1741.36	1730.27	1724.45	1729.34	1714.55	1730.03	1744.22	1736.24
Cordeau_pr09	2403.40	2371.35	2395.53	2396.75	2407.72	2413.48	2380.07	2400.18
Cordeau_pr10	3370.72	3424.08	3410.84	3435.78	3431.69	3384.15	3394.44	3373.92
Solomon_c101	1384.68	1398.15	1412.71	1436.47	1403.2	1393.32	1413.71	1378.93
Solomon_c102	1539.07	1565.1	1579.2	1523.44	1542.66	1545.11	1556.8	1540.67
Solomon_c103	1492.11	1497.38	1498.92	1519.73	1517.94	1511.28	1512.06	1521.31
Solomon_c104	1471.71	1473.87	1457.94	1493.72	1443.87	1474.45	1481.26	1449.74
Solomon_c105	1450.52	1439.79	1446.05	1399.26	1397.81	1400.25	1409.89	1458.02
Solomon_c106	1543.80	1529.09	1539.07	1565.15	1525.89	1512.88	1532.55	1530.78
Solomon_c107	1543.37	1553.07	1553.58	1564.22	1567.89	1543.6	1551.9	1561.01
Solomon_c108	1565.50	1550.9	1551.64	1568.82	1568.28	1581.19	1565.13	1582.17
Solomon_c109	1550.70	1541.77	1551.84	1508.93	1557.93	1521.17	1562.28	1530.5
Solomon_r101	1205.05	1197.48	1211.38	1223.61	1223.62	1241.7	1227.16	1215.62
Solomon_r102	1295.22	1279.55	1292.05	1306.17	1300.45	1311.01	1294.08	1299.11
Solomon_r103	1259.76	1257.72	1258.96	1242.85	1253.24	1259.87	1258.46	1236.75
Solomon_r104	1149.39	1138.69	1169.8	1152.85	1123.42	1148.28	1145.54	1158.79
Solomon_r105	1172.21	1184.06	1189.65	1171.92	1163.32	1173.48	1229.4	1203.7
Solomon_r106	1302.48	1285.66	1265.5	1298.97	1292.99	1305.13	1297.88	1305.95
Solomon_r107	1211.03	1199.85	1218.23	1215.8	1204.45	1222.35	1207.25	1210.14
Solomon_r108	1138.44	1135.86	1135.91	1140.19	1147.34	1178.3	1134.05	1150.04
Solomon_r109	1226.54	1221.07	1222.76	1226.36	1226.11	1231.42	1225.22	1229.73
Solomon_r110	1202.63	1205.85	1199.09	1220.21	1144.71	1211.03	1183.78	1208.51
Solomon_r111	1128.93	1150.59	1142.83	1169.62	1174.11	1168.96	1166.36	1169.97
Solomon_r112	1190.45	1206.33	1236.93	1222.13	1217.78	1214.07	1233.72	1215.58
Solomon_rc101	1355.26	1383.73	1357.91	1401.37	1380.89	1325.25	1385.74	1381.29
Solomon_rc102	1476.18	1508.5	1489.01	1494.68	1490.96	1432.27	1510.62	1486.69

Continues on next page

Continued from previous page

Instance	RE, ES-1	RE, ES-1, EH	RE, ES-1, EH, IH-1	RE, ES-1, EH, IH-1, IntraE	RE, ES-1, EH, IH-1, IntraE, InterE	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1
Solomon_rc103	1426.94	1459.48	1442.73	1461.27	1461.08	1401.84	1459.72	1429.38
Solomon_rc104	1355.51	1407.04	1396.23	1422	1362.83	1389.32	1374.28	1395.53
Solomon_rc105	1536.85	1538.8	1553.88	1514.86	1545.54	1533.36	1520.12	1531.69
Solomon_rc106	1386.01	1438.68	1389.18	1413.76	1399	1373.84	1411.75	1382.95
Solomon_rc107	1404.33	1410.6	1433.19	1417.71	1425.07	1412.38	1434.48	1437.4
Solomon_rc108	1375.51	1377.35	1376.92	1334.2	1390.14	1350.84	1371.75	1401.21

Table A.24: Objective function values reported by mALNS (25,100,7500) by removing more than one operator at a time for Class 4

Instance	RE, ES-1	RE, ES-1, EH	RE, ES-1, EH, IH-1	RE, ES-1, EH, IH-1, IntraE	RE, ES-1, EH, IH-1, IntraE, InterE	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1
Cordeau_pr01	523.80	518.839	520.685	520.621	525.978	518.602	520.204	513.928
Cordeau_pr02	949.60	954.684	950.153	945.133	941.195	963.238	946.508	957.699
Cordeau_pr03	1344.64	1357.95	1349.56	1335.84	1349.17	1352.77	1359.11	1342.98
Cordeau_pr04	1810.66	1835.77	1830.8	1811.81	1823.25	1835.64	1832.72	1812.31
Cordeau_pr05	2434.45	2380.41	2406.18	2417.98	2350.15	2410.44	2442.49	2433.75
Cordeau_pr06	2740.18	2735.91	2749.05	2757.67	2764.09	2737.28	2704.42	2765.1
Cordeau_pr07	717.05	708.881	734.831	720.574	724.701	722.489	714.631	722.517

Continues on next page

Continued from previous page

Instance	**RE, ES-1**	**RE, ES-1, EH**	**RE, ES-1, EH, IH-1**	**RE, ES-1, EH, IH-1, IntraE**	**RE, ES-1, EH, IH-1, IntraE, InterE**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1**
Cordeau_pr08	1481.63	1515.55	1526.6	1471.74	1524.69	1531.6	1517.11	1505.39
Cordeau_pr09	2053.34	2061.2	2045.2	2061.23	2064.31	2049.1	2058.49	2047.01
Cordeau_pr10	2859.91	2839.23	2892.82	2873.82	2888.73	2885.84	2853.8	2837.51
Solomon_c101	1250.23	1293.54	1265.54	1270.07	1271.83	1272.67	1243.44	1257
Solomon_c102	1356.14	1348.4	1351.66	1361.55	1358.43	1340.83	1346.14	1333.19
Solomon_c103	1391.15	1398.96	1390.85	1382.69	1395.57	1391.26	1399.38	1383.33
Solomon_c104	1297.22	1307.26	1319.99	1303.18	1285.89	1318.2	1312.01	1290.55
Solomon_c105	1269.84	1280.51	1285.33	1280.37	1264.47	1285.25	1272.9	1286.92
Solomon_c106	1259.58	1249.66	1258.29	1303.76	1299.53	1267.01	1259.37	1278.33
Solomon_c107	1272.66	1296.14	1308.92	1254.31	1318.58	1296.34	1301.87	1295.3
Solomon_c108	1385.79	1398.39	1383.41	1389.38	1377.19	1392.43	1383.91	1381.07
Solomon_c109	1280.19	1277.43	1290.58	1277.31	1284.59	1275.3	1295.63	1264.69
Solomon_r101	1055.71	1087.68	1096.91	1080.54	1079.62	1090.81	1044.79	1083.16
Solomon_r102	1140.68	1130.04	1136.82	1137.57	1144.07	1143.74	1144.59	1138.21
Solomon_r103	1096.36	1105.74	1112.81	1122.29	1108.81	1104.36	1105.39	1104.98
Solomon_r104	1042.75	1068.76	1094.54	1084.65	1071.79	1077.11	1082.95	1080.38
Solomon_r105	1137.65	1123.11	1120.92	1145.67	1138.35	1116.57	1164.17	1122.88
Solomon_r106	1111.47	1104.82	1113.49	1077.91	1107.45	1081.23	1095.25	1090.06
Solomon_r107	1132.41	1132.5	1130.51	1136.83	1125.67	1124.76	1126.14	1132.12
Solomon_r108	1055.34	1053.86	1048.11	1040.57	1045.65	1065.98	1036.44	1040.62
Solomon_r109	1108.96	1080.57	1105.91	1111.98	1098.84	1116.14	1110.07	1094.64
Solomon_r110	1070.04	1092.19	1085.66	1100.97	1103.17	1090.58	1099.09	1108.86
Solomon_r111	1113.22	1066.61	1124.56	1110.9	1099.57	1122.68	1110.39	1111.43
Solomon_r112	1011.10	1024.88	1041.01	1048.04	1059.2	1041.52	1025.78	1028.46
Solomon_rc101	1318.52	1286.42	1288.27	1334.06	1311.51	1315.66	1311.72	1306.9
Solomon_rc102	1267.26	1256.1	1269.45	1274.05	1285.86	1288.06	1268.03	1255.64
Solomon_rc103	1280.88	1291.8	1288.74	1301.23	1299	1273.3	1300.95	1305.34
Solomon_rc104	1292.01	1280.89	1295.21	1312.96	1257.03	1307.99	1248.96	1301.3
Solomon_rc105	1286.94	1314.35	1289.26	1287.58	1278.54	1275.12	1294.27	1298.47
Solomon_rc106	1207.01	1196.39	1188.13	1206.26	1177.84	1183.79	1188.88	1206.88
Solomon_rc107	1349.78	1355.89	1352.19	1344.64	1343.49	1335.63	1343.58	1335.12

Continues on next page

Continued from previous page

Instance	RE, ES-1	RE, ES-1, EH	RE, ES-1, EH, IH-1	RE, ES-1, EH, IH-1, IntraE	RE, ES-1, EH, IH-1, IntraE, InterE	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1
Solomon_rc108	1187.77	1247.01	1152.49	1252.4	1245.36	1259.9	1261.14	1247.03

Table A.25: Objective function values reported by mALNS (25,100,7500) by removing more than one operator at a time for Class 5

Instance	RE, ES-1	RE, ES-1, EH	RE, ES-1, EH, IH-1	RE, ES-1, EH, IH-1, IntraE	RE, ES-1, EH, IH-1, IntraE, InterE	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1
Cordeau_pr01	512.13	521.334	515.615	521.888	517.081	503.671	516.556	517.048
Cordeau_pr02	929.52	926.293	933.232	931.499	932.075	934.074	933.313	935.929
Cordeau_pr03	1328.35	1306.57	1331.84	1322.66	1330.93	1317.65	1338.08	1312.02
Cordeau_pr04	1711.42	1764.09	1783.43	1765.69	1784.67	1762.97	1772.2	1759.37
Cordeau_pr05	2319.92	2318.01	2322.32	2338.88	2314.14	2320.15	2308.91	2321.27
Cordeau_pr06	2689.85	2685.65	2679.66	2690.29	2701.45	2689.68	2683.74	2680.39
Cordeau_pr07	697.91	712.064	710.297	704.57	693.557	700.43	687.616	702.884
Cordeau_pr08	1469.08	1464.29	1458	1451.57	1469.06	1465.63	1479.82	1471.11
Cordeau_pr09	1995.09	2005.96	2008.96	2014.55	2008.19	2013.55	2001.72	1999.23
Cordeau_pr10	2783.25	2801.93	2804.85	2813.39	2799.51	2785.88	2789.46	2817.34
Solomon_c101	1204.41	1220.48	1217.57	1243.67	1204.21	1212.65	1217.04	1219.36
Solomon_c102	1315.87	1325.35	1323.67	1332.62	1316.08	1321.91	1325.8	1327.49

Continues on next page

Continued from previous page

Instance	RE, ES-1	RE, ES-1, EH	RE, ES-1, EH, IH-1	RE, ES-1, EH, IH-1, IntraE	RE, ES-1, EH, IH-1, IntraE, InterE	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1
Solomon_c103	1346.17	1346.25	1342.46	1339.26	1352.4	1345.39	1344.07	1348.81
Solomon_c104	1259.58	1268.43	1268.44	1266.51	1253.56	1246.61	1261.79	1250.44
Solomon_c105	1220.57	1201.97	1209.97	1197.39	1203.95	1222.38	1221.76	1213.89
Solomon_c106	1205.85	1220.83	1214.91	1226.54	1189.77	1199.84	1226.32	1221.85
Solomon_c107	1221.83	1242.73	1234.01	1226.28	1251.29	1239.68	1233.37	1234.71
Solomon_c108	1358.32	1350.94	1353.38	1347.19	1356.87	1361.39	1352.33	1352.56
Solomon_c109	1230.08	1255.39	1247.61	1250.51	1239.75	1250.69	1248.76	1247.28
Solomon_r101	1022.35	1028.88	1012.88	1021.63	1027.71	1036.55	1022.29	1034.24
Solomon_r102	1116.02	1113.01	1119.74	1118.79	1118.6	1116.92	1107.99	1108.93
Solomon_r103	1079.34	1081.71	1073.2	1076.88	1080.72	1069.92	1069.73	1088.66
Solomon_r104	1054.82	1068.92	1060.8	1058.52	1044.63	1056.08	1053.02	1060.82
Solomon_r105	1107.80	1128.09	1116.37	1125.64	1110.93	1127.82	1106.85	1124.14
Solomon_r106	1058.79	1051.21	1050.97	1065.71	1058.02	1068.83	1060.38	1057.82
Solomon_r107	1088.62	1115.86	1098.95	1107.9	1108.06	1109.96	1110.51	1114.47
Solomon_r108	1014.73	1039.69	1014.95	1023.63	1022.84	1040.72	1022.77	1016.44
Solomon_r109	1053.42	1079.17	1069.29	1074.86	1071.31	1079.89	1074.45	1067.28
Solomon_r110	1078.12	1069.11	1064.02	1068.65	1070.38	1068.23	1060.19	1075.6
Solomon_r111	1085.35	1080.9	1072.08	1084.48	1082.37	1089.13	1084.97	1082.1
Solomon_r112	998.12	981.557	1018.16	1017.72	992.641	1014.34	1000.61	1015.52
Solomon_rc101	1257.76	1268.74	1269.16	1232.88	1285.67	1284.72	1279.73	1274.26
Solomon_rc102	1248.38	1240.43	1225.38	1238.55	1243.51	1246.31	1243.48	1234.84
Solomon_rc103	1257.53	1267	1256.77	1259	1261.39	1248.95	1271.1	1252.39
Solomon_rc104	1215.18	1244.01	1259.26	1270.7	1250.13	1257.81	1242.21	1290.78
Solomon_rc105	1247.90	1246.61	1234.4	1247.81	1249.32	1246.37	1260.53	1232.4
Solomon_rc106	1151.76	1137.46	1146.9	1170.6	1131.19	1158.7	1165.51	1156.55
Solomon_rc107	1302.58	1333.19	1325.94	1338.4	1331.89	1317.55	1331.82	1312.74
Solomon_rc108	1172.78	1222.79	1219.07	1231.11	1220.58	1213.18	1219.18	1229.95

Tables A.26–A.30 display the relative gap in percentage of the objective function value reported by mALNS(25,100,7500) by removing more than one operator at a time with respect to the results obtained by mALNS (25,100,7500) considering all operators, per instance.

Table A.26: Relative gap of the objective function value reported by mALNS (25,100,7500) by removing more than one operator at a time with respect to the ones reported by mALNS (25,100,7500) for Class 1

Instance	**RE, ES-1**	**RE, ES-1, EH**	**RE, ES-1, EH, IH-1**	**RE, ES-1, EH, IH-1, IntraE**	**RE, ES-1, EH, IH-1, IntraE, InterE**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1**
Cordeau_pr01	-0.66	0.00	-1.30	3.75	-0.41	-2.45	2.33	-3.49
Cordeau_pr02	0.35	0.00	1.90	0.78	0.95	0.43	-0.69	-1.55
Cordeau_pr03	2.44	-0.32	0.40	-0.46	-0.75	0.35	-2.34	-4.08
Cordeau_pr04	-0.86	-1.78	0.72	-4.30	-5.65	2.77	-3.30	-2.82
Cordeau_pr05	-0.91	5.26	1.36	0.21	0.01	-0.95	0.72	-1.87
Cordeau_pr06	-6.72	-6.24	-6.67	-7.35	-7.07	-7.74	-6.07	-6.11
Cordeau_pr07	-7.86	-1.48	-2.99	-5.10	-5.81	-1.86	-4.95	-7.27
Cordeau_pr08	-2.24	-2.66	-0.37	-1.31	-3.30	-4.37	-3.56	-0.73
Cordeau_pr09	1.66	1.64	0.65	1.51	0.42	1.26	0.84	0.56
Cordeau_pr10	-2.15	-3.36	-2.56	-1.27	-1.69	-2.11	-2.20	-3.41
Solomon_c101	-1.11	1.42	0.23	-2.74	-2.60	0.98	2.08	-1.36
Solomon_c102	-1.25	0.48	0.15	0.66	-2.07	-1.36	-1.83	-0.89
Solomon_c103	2.64	0.26	-0.76	0.14	0.41	-0.72	-1.28	0.40
Solomon_c104	-2.63	-7.01	-5.12	-4.06	-4.08	-2.97	-5.54	-1.95
Solomon_c105	-4.62	1.77	-4.12	-2.60	0.01	0.47	0.38	-0.79
Solomon_c106	1.43	-2.80	-1.96	0.41	-0.87	-1.97	0.04	1.36
Solomon_c107	0.55	1.31	0.12	0.67	0.19	-1.14	-0.80	-0.89
Solomon_c108	-2.84	-3.60	-0.34	-3.99	-4.93	-5.42	-3.84	-1.15
Solomon_c109	-1.12	-1.21	-3.14	-4.34	-2.07	-1.17	-3.31	-4.02
Solomon_r101	-2.86	-3.24	-2.17	-4.38	-2.49	-2.89	-1.17	-3.98
Solomon_r102	-4.25	-4.01	-3.64	-4.36	-3.74	-4.87	-4.10	-3.72
Solomon_r103	-0.96	-1.47	-1.00	-0.50	-2.38	-2.50	-2.54	1.75
Solomon_r104	-1.43	-1.17	-0.83	0.10	-5.98	0.06	-2.08	2.62
Solomon_r105	0.44	0.90	-0.38	-0.36	1.57	-0.54	-0.89	-3.23

Continues on next page

Continued from previous page

Instance	**RE, ES-1**	**RE, ES-1, EH**	**RE, ES-1, EH, IH-1**	**RE, ES-1, EH, IH-1, IntraE**	**RE, ES-1, EH, IH-1, IntraE, InterE**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1**
Solomon_r106	-4.88	-3.86	-4.00	-5.79	-3.44	-3.35	-6.14	-5.31
Solomon_r107	-3.43	-3.99	-2.99	-5.58	-4.84	-5.60	-4.51	-4.19
Solomon_r108	-2.79	-4.45	-4.21	-3.47	0.32	-0.39	-5.03	-0.28
Solomon_r109	1.53	-0.50	-1.56	7.55	2.05	-0.94	0.01	-1.75
Solomon_r110	-6.98	-10.32	-6.03	-8.66	-9.65	-10.81	-7.35	-5.47
Solomon_r111	2.09	5.80	4.69	1.53	1.46	1.72	0.01	0.37
Solomon_r112	0.91	-1.97	0.23	0.91	-0.91	-1.82	-2.04	-1.91
Solomon_rc101	-0.75	-2.14	-4.72	-4.78	-3.08	-4.99	0.29	-6.22
Solomon_rc102	-2.84	-4.10	-7.32	-2.89	-5.95	-8.23	-2.32	-4.18
Solomon_rc103	-1.18	0.50	7.35	-4.22	0.58	1.35	-0.49	-1.26
Solomon_rc104	-4.50	-0.78	-6.70	-4.91	-6.54	-4.24	-5.54	-4.50
Solomon_rc105	-2.83	1.40	-1.21	-3.24	-5.88	-3.02	-4.79	-2.63
Solomon_rc106	-7.57	-10.76	-9.76	-8.17	-7.17	-11.49	-5.64	-6.33
Solomon_rc107	-1.28	-6.70	-3.00	-2.80	-2.92	0.05	-5.23	-3.75
Solomon_rc108	7.32	-0.97	0.42	1.50	-0.46	0.99	1.71	-0.64
Average	-1.59	-1.80	-1.81	-2.10	-2.53	-2.29	-2.34	-2.43

Table A.27: Relative gap of the objective function value reported by mALNS (25,100,7500) by removing more than one operator at a time with respect to the ones reported by mALNS (25,100,7500) for Class 2

Instance	**RE, ES-1**	**RE, ES-1, EH**	**RE, ES-1, EH, IH-1**	**RE, ES-1, EH, IH-1, IntraE**	**RE, ES-1, EH, IH-1, IntraE, InterE**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1**
Cordeau_pr01	-3.98	-1.44	1.20	-1.18	-0.32	-3.39	-2.72	4.44
Cordeau_pr02	0.11	1.03	0.99	-0.27	0.30	-0.15	0.80	-0.73
Cordeau_pr03	0.71	0.26	2.52	1.34	1.51	-0.73	1.53	0.35
Cordeau_pr04	-1.20	-0.31	-1.60	2.13	-3.32	-1.75	-1.61	-0.72
Cordeau_pr05	-0.69	-1.19	1.30	-0.30	-0.43	-0.71	-1.10	-0.67
Cordeau_pr06	-0.28	-1.55	-2.06	-1.73	-1.83	-1.35	-2.04	0.01
Cordeau_pr07	-1.79	1.06	0.24	-0.28	-1.12	-4.23	-3.54	-2.46
Cordeau_pr08	-1.86	-1.22	-3.98	-4.40	-3.14	-3.07	-3.75	-3.10
Cordeau_pr09	-0.76	-1.87	-1.17	-2.49	-0.76	-2.59	-2.50	-2.46
Cordeau_pr10	-3.16	-1.98	-3.61	-1.68	-3.04	-2.95	-3.14	-1.48
Solomon_c101	-1.73	-3.61	-3.18	-3.66	-3.42	-2.46	-3.86	-2.51
Solomon_c102	2.53	1.03	0.62	0.10	0.25	0.67	-0.98	0.57
Solomon_c103	-0.24	-0.89	-2.29	-0.73	-2.73	-1.17	-1.69	-3.09
Solomon_c104	-1.80	-2.17	-2.02	-3.55	-1.84	-3.49	-5.65	-2.42
Solomon_c105	-1.49	-0.71	1.51	-1.52	0.35	0.68	0.14	1.24
Solomon_c106	0.98	-0.47	-0.07	0.43	-1.31	-1.34	0.67	1.41
Solomon_c107	-1.27	-0.88	-0.37	-0.69	-1.47	-1.31	-1.47	0.61
Solomon_c108	-1.10	-3.04	-2.61	-3.31	-3.20	-1.95	-2.67	-2.23
Solomon_c109	0.26	-0.92	-0.01	-3.10	-0.82	-0.11	-1.13	-1.78
Solomon_r101	-6.30	-4.78	-2.88	-4.66	-2.32	-6.53	-4.00	-3.42
Solomon_r102	-0.33	0.78	-0.60	-0.19	0.57	-0.19	0.33	-0.32
Solomon_r103	-3.07	-1.65	-4.31	-4.41	-4.80	-3.85	-5.15	-3.91
Solomon_r104	-2.71	-5.10	-5.55	-7.03	-7.55	-7.56	-5.92	-5.50
Solomon_r105	0.81	-0.18	-1.39	-0.19	1.72	-2.01	-0.32	-1.57
Solomon_r106	0.03	0.14	1.31	-0.19	-1.37	0.39	0.02	-0.50
Solomon_r107	-1.12	-1.64	-2.41	-1.88	-2.00	-0.15	-1.65	-3.13
Solomon_r108	-3.47	-2.10	-5.19	1.83	-2.30	-1.57	-3.46	-0.44

Continues on next page

Continued from previous page

Instance	RE, ES-1	RE, ES-1, EH	RE, ES-1, EH, IH-1	RE, ES-1, EH, IH-1, IntraE	RE, ES-1, EH, IH-1, IntraE, InterE	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1
Solomon_r109	0.15	-1.32	-0.90	-3.03	-1.61	-1.64	-1.07	-1.29
Solomon_r110	0.26	-0.10	-0.39	-1.41	-0.53	-0.88	-1.16	-1.44
Solomon_r111	-2.96	-1.57	-6.67	-5.34	-6.75	-6.49	-2.71	-5.83
Solomon_r112	-0.15	-1.11	-1.01	-0.79	-1.26	-1.71	-0.74	-0.47
Solomon_rc101	-2.99	1.31	-1.52	0.84	-0.59	-1.62	-2.70	-1.02
Solomon_rc102	-1.28	2.86	0.40	0.53	-2.27	1.79	-1.06	0.62
Solomon_rc103	-0.73	0.99	0.58	0.10	-0.57	3.05	-1.30	0.60
Solomon_rc104	-0.40	-3.44	-2.41	2.47	-2.57	-0.92	0.29	-2.05
Solomon_rc105	-0.44	-0.91	-0.11	1.85	-1.30	-1.20	-0.46	-1.99
Solomon_rc106	-0.57	-0.32	1.15	0.62	-0.05	-1.11	-2.81	-1.61
Solomon_rc107	-3.24	-5.10	-2.87	-3.28	-3.93	-2.50	-4.51	-3.69
Solomon_rc108	1.24	-1.61	-2.01	-4.05	-1.54	0.91	-3.49	-2.72
Average	-1.13	-1.12	-1.32	-1.36	-1.73	-1.67	-1.96	-1.40

Table A.28: Relative gap of the objective function value reported by mALNS (25,100,7500) by removing more than one operator at a time with respect to the ones reported by mALNS (25,100,7500) for Class 3

Instance	**RE, ES-1**	**RE, ES-1, EH**	**RE, ES-1, EH, IH-1**	**RE, ES-1, EH, IH-1, IntraE**	**RE, ES-1, EH, IH-1, IntraE, InterE**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1**
Cordeau_pr01	2.54	4.41	-1.43	5.28	2.67	1.83	4.60	0.08
Cordeau_pr02	-0.98	-1.30	-1.56	0.72	-1.39	-0.90	-0.88	-2.60
Cordeau_pr03	-1.61	-1.43	-0.68	-3.66	-1.76	-2.21	-4.11	-0.69
Cordeau_pr04	1.37	-0.32	-0.80	-0.11	-2.17	-0.98	-0.89	-1.46
Cordeau_pr05	-3.58	-4.98	-4.45	-5.95	-6.72	-5.56	-5.76	-3.90
Cordeau_pr06	1.03	0.41	0.20	0.71	-0.28	-0.73	-0.42	1.52
Cordeau_pr07	0.33	-0.75	-1.10	-0.20	-1.72	-2.60	0.16	0.12
Cordeau_pr08	-1.16	-0.52	-0.18	-0.46	0.40	-0.50	-1.33	-0.86
Cordeau_pr09	-1.37	-0.02	-1.04	-1.09	-1.55	-1.80	-0.39	-1.23
Cordeau_pr10	-0.76	-2.36	-1.96	-2.71	-2.59	-1.17	-1.47	-0.86
Solomon_c101	3.38	2.44	1.42	-0.24	2.08	2.77	1.35	3.78
Solomon_c102	1.05	-0.62	-1.53	2.06	0.82	0.67	-0.09	0.95
Solomon_c103	-0.02	-0.38	-0.48	-1.88	-1.76	-1.31	-1.36	-1.98
Solomon_c104	0.55	0.40	1.48	-0.94	2.43	0.36	-0.10	2.03
Solomon_c105	-4.13	-3.36	-3.81	-0.45	-0.35	-0.52	-1.21	-4.67
Solomon_c106	-1.03	-0.07	-0.72	-2.43	0.14	0.99	-0.29	-0.18
Solomon_c107	0.56	-0.07	-0.10	-0.79	-1.02	0.54	0.01	-0.58
Solomon_c108	0.16	1.09	1.04	-0.05	-0.02	-0.84	0.18	-0.91
Solomon_c109	-2.30	-1.71	-2.38	0.45	-2.78	-0.35	-3.07	-0.97
Solomon_r101	-0.29	0.34	-0.81	-1.83	-1.83	-3.34	-2.13	-1.17
Solomon_r102	0.50	1.70	0.74	-0.34	0.10	-0.72	0.59	0.20
Solomon_r103	-0.91	-0.75	-0.85	0.44	-0.39	-0.92	-0.81	0.93
Solomon_r104	-1.93	-0.98	-3.74	-2.24	0.37	-1.83	-1.59	-2.77
Solomon_r105	-1.00	-2.02	-2.50	-0.97	-0.23	-1.11	-5.93	-3.71
Solomon_r106	-2.52	-1.20	0.39	-2.25	-1.78	-2.73	-2.16	-2.80
Solomon_r107	-0.17	0.76	-0.76	-0.56	0.38	-1.10	0.14	-0.09
Solomon_r108	-5.05	-4.82	-4.82	-5.22	-5.88	-8.73	-4.65	-6.12

Continues on next page

Continued from previous page

Instance	RE, ES-1	RE, ES-1, EH	RE, ES-1, EH, IH-1	RE, ES-1, EH, IH-1, IntraE	RE, ES-1, EH, IH-1, IntraE, InterE	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1
Solomon_r109	-1.08	-0.63	-0.77	-1.06	-1.04	-1.48	-0.97	-1.34
Solomon_r110	-0.11	-0.38	0.18	-1.58	4.71	-0.81	1.46	-0.60
Solomon_r111	3.58	1.73	2.39	0.10	-0.28	0.16	0.38	0.07
Solomon_r112	1.14	-0.18	-2.72	-1.49	-1.13	-0.82	-2.45	-0.95
Solomon_rc101	-0.32	-2.43	-0.52	-3.74	-2.22	1.90	-2.58	-2.25
Solomon_rc102	0.46	-1.72	-0.41	-0.79	-0.54	3.42	-1.87	-0.25
Solomon_rc103	-1.55	-3.87	-2.68	-4.00	-3.98	0.23	-3.89	-1.73
Solomon_rc104	4.67	1.04	1.80	-0.01	4.15	2.29	3.35	1.85
Solomon_rc105	-0.79	-0.92	-1.91	0.65	-1.36	-0.56	0.30	-0.46
Solomon_rc106	1.09	-2.66	0.87	-0.89	0.17	1.96	-0.74	1.31
Solomon_rc107	-2.69	-3.15	-4.80	-3.67	-4.21	-3.28	-4.90	-5.11
Solomon_rc108	-2.83	-2.96	-2.93	0.26	-3.92	-0.98	-2.55	-4.75
Average	-0.41	-0.83	-1.07	-1.05	-0.88	-0.79	-1.18	-1.08

Table A.29: Relative gap of the objective function value reported by mALNS (25,100,7500) by removing more than one operator at a time with respect to the ones reported by mALNS (25,100,7500) for Class 4

Instance	**RE, ES-1**	**RE, ES-1, EH**	**RE, ES-1, EH, IH-1**	**RE, ES-1, EH, IH-1, IntraE**	**RE, ES-1, EH, IH-1, IntraE, InterE**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2**	**RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1**
Cordeau_pr01	-1.31	-0.35	-0.71	-0.70	-1.73	-0.31	-0.62	0.60
Cordeau_pr02	-0.62	-1.16	-0.68	-0.15	0.27	-2.06	-0.29	-1.48
Cordeau_pr03	-2.40	-3.42	-2.78	-1.73	-2.75	-3.02	-3.51	-2.28
Cordeau_pr04	-0.77	-2.17	-1.89	-0.83	-1.47	-2.16	-2.00	-0.86
Cordeau_pr05	-4.11	-1.80	-2.90	-3.41	-0.51	-3.08	-4.45	-4.08
Cordeau_pr06	-0.49	-0.33	-0.81	-1.13	-1.37	-0.38	0.82	-1.40
Cordeau_pr07	-1.93	-0.77	-4.45	-2.43	-3.01	-2.70	-1.58	-2.70
Cordeau_pr08	1.18	-1.08	-1.82	1.84	-1.69	-2.15	-1.18	-0.40
Cordeau_pr09	-1.81	-2.20	-1.41	-2.21	-2.36	-1.60	-2.07	-1.50
Cordeau_pr10	-0.17	0.55	-1.33	-0.66	-1.18	-1.08	0.04	0.61
Solomon_c101	-3.28	-6.86	-4.55	-4.92	-5.07	-5.14	-2.72	-3.84
Solomon_c102	-1.44	-0.86	-1.10	-1.84	-1.61	-0.29	-0.69	0.28
Solomon_c103	-0.67	-1.24	-0.65	-0.06	-0.99	-0.68	-1.27	-0.10
Solomon_c104	-0.49	-1.26	-2.25	-0.95	0.39	-2.11	-1.63	0.03
Solomon_c105	-4.81	-5.69	-6.09	-5.68	-4.36	-6.08	-5.06	-6.22
Solomon_c106	-1.40	-0.60	-1.30	-4.96	-4.62	-2.00	-1.39	-2.91
Solomon_c107	1.40	-0.42	-1.41	2.82	-2.16	-0.44	-0.87	-0.36
Solomon_c108	-0.95	-1.87	-0.78	-1.21	-0.32	-1.43	-0.81	-0.60
Solomon_c109	-0.58	-0.36	-1.40	-0.35	-0.93	-0.20	-1.79	0.64
Solomon_r101	1.40	-1.58	-2.45	-0.92	-0.83	-1.88	2.42	-1.16
Solomon_r102	0.31	1.24	0.65	0.58	0.01	0.04	-0.03	0.52
Solomon_r103	-0.27	-1.13	-1.78	-2.65	-1.41	-1.01	-1.10	-1.06
Solomon_r104	1.65	-0.81	-3.24	-2.30	-1.09	-1.59	-2.14	-1.90
Solomon_r105	-2.70	-1.39	-1.19	-3.43	-2.77	-0.80	-5.10	-1.37
Solomon_r106	-3.48	-2.86	-3.67	-0.35	-3.10	-0.66	-1.97	-1.49
Solomon_r107	-2.79	-2.80	-2.62	-3.20	-2.18	-2.10	-2.22	-2.77
Solomon_r108	0.08	0.22	0.77	1.48	1.00	-0.93	1.87	1.47

Continues on next page

Continued from previous page

Instance	RE, ES-1	RE, ES-1, EH	RE, ES-1, EH, IH-1	RE, ES-1, EH, IH-1, IntraE	RE, ES-1, EH, IH-1, IntraE, InterE	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1
Solomon_r109	-1.86	0.75	-1.58	-2.13	-0.93	-2.51	-1.96	-0.54
Solomon_r110	1.68	-0.36	0.24	-1.17	-1.37	-0.21	-0.99	-1.89
Solomon_r111	-1.04	3.19	-2.07	-0.83	0.20	-1.90	-0.78	-0.87
Solomon_r112	4.21	2.90	1.37	0.71	-0.35	1.32	2.81	2.56
Solomon_rc101	-3.43	-0.91	-1.05	-4.64	-2.88	-3.20	-2.89	-2.51
Solomon_rc102	-1.12	-0.23	-1.29	-1.66	-2.60	-2.78	-1.18	-0.19
Solomon_rc103	-0.66	-1.52	-1.28	-2.26	-2.09	-0.07	-2.24	-2.59
Solomon_rc104	-3.16	-2.27	-3.42	-4.83	-0.37	-4.44	0.28	-3.90
Solomon_rc105	-2.03	-4.20	-2.21	-2.08	-1.36	-1.09	-2.61	-2.94
Solomon_rc106	-1.81	-0.92	-0.22	-1.75	0.65	0.15	-0.28	-1.80
Solomon_rc107	-2.51	-2.98	-2.70	-2.12	-2.04	-1.44	-2.04	-1.40
Solomon_rc108	0.90	-4.04	3.84	-4.49	-3.91	-5.12	-5.22	-4.05
Average	-1.06	-1.32	-1.59	-1.71	-1.61	-1.72	-1.45	-1.40

Table A.30: Relative gap of the objective function value reported by mALNS (25,100,7500) by removing more than one operator at a time with respect to the ones reported by mALNS (25,100,7500) for Class 5

Instance	RE, ES-1	RE, ES-1, EH	RE, ES-1, EH, IH-1	RE, ES-1, EH, IH-1, IntraE	RE, ES-1, EH, IH-1, IntraE, InterE	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1
Cordeau_pr01	1.00	-0.78	0.33	-0.88	0.04	2.64	0.15	0.05
Cordeau_pr02	-0.31	0.03	-0.71	-0.53	-0.59	-0.80	-0.72	-1.01
Cordeau_pr03	-2.53	-0.85	-2.80	-2.09	-2.73	-1.71	-3.28	-1.27
Cordeau_pr04	2.00	-1.02	-2.12	-1.11	-2.19	-0.95	-1.48	-0.75
Cordeau_pr05	-2.75	-2.66	-2.85	-3.59	-2.49	-2.76	-2.26	-2.81
Cordeau_pr06	-1.69	-1.53	-1.31	-1.71	-2.13	-1.68	-1.46	-1.33
Cordeau_pr07	-1.16	-3.22	-2.96	-2.13	-0.53	-1.53	0.33	-1.88
Cordeau_pr08	-1.96	-1.63	-1.19	-0.75	-1.96	-1.72	-2.71	-2.10
Cordeau_pr09	0.68	0.14	-0.01	-0.29	0.03	-0.24	0.35	0.48
Cordeau_pr10	-1.54	-2.22	-2.33	-2.64	-2.13	-1.63	-1.76	-2.78
Solomon_c101	1.00	-0.32	-0.08	-2.23	1.02	0.32	-0.04	-0.23
Solomon_c102	0.69	-0.02	0.10	-0.57	0.68	0.24	-0.06	-0.18
Solomon_c103	-0.59	-0.59	-0.31	-0.07	-1.05	-0.53	-0.43	-0.78
Solomon_c104	0.34	-0.36	-0.36	-0.21	0.82	1.37	0.17	1.06
Solomon_c105	-0.76	0.78	0.12	1.15	0.61	-0.91	-0.86	-0.21
Solomon_c106	-1.02	-2.27	-1.77	-2.75	0.33	-0.51	-2.73	-2.36
Solomon_c107	-0.98	-2.70	-1.98	-1.34	-3.41	-2.45	-1.93	-2.04
Solomon_c108	-0.20	0.35	0.17	0.62	-0.09	-0.43	0.24	0.23
Solomon_c109	1.46	-0.57	0.06	-0.18	0.68	-0.19	-0.04	0.08
Solomon_r101	-0.03	-0.67	0.90	0.04	-0.56	-1.42	-0.03	-1.19
Solomon_r102	-0.84	-0.57	-1.17	-1.09	-1.07	-0.92	-0.11	-0.20
Solomon_r103	-1.10	-1.32	-0.52	-0.87	-1.23	-0.22	-0.20	-1.97
Solomon_r104	0.29	-1.04	-0.27	-0.06	1.26	0.17	0.46	-0.28
Solomon_r105	0.04	-1.79	-0.73	-1.57	-0.24	-1.76	0.13	-1.43
Solomon_r106	-2.07	-1.34	-1.32	-2.74	-2.00	-3.04	-2.23	-1.98
Solomon_r107	-1.20	-3.74	-2.16	-3.00	-3.01	-3.19	-3.24	-3.61
Solomon_r108	-0.33	-2.80	-0.35	-1.21	-1.13	-2.90	-1.13	-0.50

Continues on next page

Continued from previous page

Instance	RE, ES-1	RE, ES-1, EH	RE, ES-1, EH, IH-1	RE, ES-1, EH, IH-1, IntraE	RE, ES-1, EH, IH-1, IntraE, InterE	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2	RE, ES-1, EH, IH-1, IntraE, InterE, IRDI-1, IH-2, IEF-1
Solomon_r109	0.59	-1.84	-0.90	-1.43	-1.09	-1.90	-1.39	-0.71
Solomon_r110	-0.27	0.57	1.04	0.61	0.45	0.65	1.40	-0.04
Solomon_r111	-0.09	0.32	1.13	-0.01	0.18	-0.44	-0.06	0.20
Solomon_r112	1.21	2.85	-0.77	-0.73	1.75	-0.40	0.96	-0.51
Solomon_rc101	-2.04	-2.93	-2.97	-0.03	-4.31	-4.23	-3.83	-3.38
Solomon_rc102	-2.99	-2.33	-1.09	-2.17	-2.58	-2.81	-2.58	-1.87
Solomon_rc103	-0.64	-1.39	-0.58	-0.75	-0.95	0.05	-1.72	-0.22
Solomon_rc104	0.51	-1.85	-3.09	-4.03	-2.35	-2.98	-1.70	-5.67
Solomon_rc105	-2.41	-2.30	-1.30	-2.40	-2.52	-2.28	-3.44	-1.14
Solomon_rc106	-1.05	0.20	-0.63	-2.71	0.75	-1.66	-2.26	-1.47
Solomon_rc107	0.25	-2.10	-1.54	-2.49	-2.00	-0.90	-1.99	-0.53
Solomon_rc108	2.60	-1.55	-1.24	-2.24	-1.37	-0.75	-1.25	-2.15
Average	-0.46	-1.16	-0.96	-1.29	-0.95	-1.14	-1.10	-1.19

Tables A.31–A.35 report the the lower bound and the upper bounds found through CPLEX, as well as the percent gap of the objective function value reported by mALNS* with respect to the bounds, per instance.

Table A.31: Results reported by mALNS* and lower and upper bounds reported by CPLEX 12.6 for Class 1

	mALNS*	Lower bound		Upper bound	
Instance	**Objective**	**Gap**	**Bound**	**Gap**	**Bound**
Cordeau_pr01	552.833	4.87	527.143	15.85	657
Cordeau_pr02	1166	19.59	975	4.43	1220
Cordeau_pr03	1614.03	64.36	982	9.73	1788

Continues on next page

Continued from previous page

	mALNS*	Lower bound		Upper bound	
Instance	**Objective**	**Gap**	**Bound**	**Gap**	**Bound**
Cordeau_pr04	2136.61	76.37	1211.47	13.73	2476.78
Cordeau_pr05	2906.31	164.63	1098.27	13.26	3350.79
Cordeau_pr06	3520.59	176.09	1275.14	4.10	3671
Cordeau_pr07	840.893	32.22	636	11.30	948
Cordeau_pr08	1885.55	63.56	1152.81	6.00	2006
Cordeau_pr09	2701.47	97.25	1369.54	1.26	2736
Cordeau_pr10	3716.36	183.59	1310.49	3.47	3850
Solomon_c101	1423.28	15.30	1234.37	21.37	1810
Solomon_c102	1629.27	23.97	1314.26	9.99	1810
Solomon_c103	1654.82	42.03	1165.11	8.57	1810
Solomon_c104	1561.89	33.66	1168.59	10.33	1741.75
Solomon_c105	1419.97	-2.49	1456.17	18.54	1743.18
Solomon_c106	1590.51	5.52	1507.33	12.13	1810
Solomon_c107	1665.61	12.68	1478.13	7.98	1810
Solomon_c108	1663.07	39.97	1188.14	8.12	1810
Solomon_c109	1592.09	29.96	1225.04	12.04	1810
Solomon_r101	1267.57	0.41	1262.34	9.51	1400.85
Solomon_r102	1416	44.52	979.768	2.88	1458
Solomon_r103	1405.38	48.09	949.034	3.61	1458
Solomon_r104	1199.09	57.79	759.911	17.76	1458
Solomon_r105	1216.83	30.52	932.26	8.84	1334.76
Solomon_r106	1387.07	71.78	807.459	4.86	1458
Solomon_r107	1323.37	72.37	767.735	9.23	1458
Solomon_r108	1208.38	94.90	620	17.12	1458
Solomon_r109	1349.27	59.68	845	7.46	1458
Solomon_r110	1286.63	55.77	826	11.75	1458
Solomon_r111	1232.58	73.96	708.526	15.46	1458
Solomon_r112	1348	55.61	866.291	7.54	1458
Solomon_rc101	1388.52	72.94	802.89	19.46	1724
Solomon_rc102	1492.8	82.38	818.527	13.41	1724
Solomon_rc103	1537.64	80.37	852.47	10.81	1724

Continues on next page

Continued from previous page

	mALNS*	Lower bound		Upper bound	
Instance	**Objective**	**Gap**	**Bound**	**Gap**	**Bound**
Solomon_rc104	1452.03	105.25	707.437	15.78	1724
Solomon_rc105	1620.31	47.11	1101.46	6.01	1724
Solomon_rc106	1456.69	52.37	956	15.51	1724
Solomon_rc107	1549.19	84.62	839.106	10.14	1724
Solomon_rc108	1401.87	77.92	787.916	18.69	1724
Average		60.30		10.72	

Table A.32: Results reported by mALNS* and lower and upper bounds reported by CPLEX 12.6 for Class 2

	mALNS*	Lower bound		Upper bound	
Instance	**Objective**	**Gap**	**Bound**	**Gap**	**Bound**
Cordeau_pr01	566.994	6.21	533.832	11.75	642.482
Cordeau_pr02	1057.51	11.88	945.245	11.03	1188.55
Cordeau_pr03	1536.26	79.07	857.926	11.93	1744.29
Cordeau_pr04	2034.97	56.29	1302.04	14.65	2384.22
Cordeau_pr05	2895.87	118.03	1328.17	11.64	3277.48
Cordeau_pr06	3186.39	142.50	1314	10.01	3540.98
Cordeau_pr07	814.748	16.29	700.621	10.99	915.329
Cordeau_pr08	1757.31	86.55	942	9.74	1946.89
Cordeau_pr09	2453.75	96.68	1247.56	7.49	2652.49
Cordeau_pr10	3469.97	163.58	1316.5	7.26	3741.71
Solomon_c101	1443.67	20.56	1197.49	17.83	1756.99
Solomon_c102	1576.8	38.50	1138.47	10.96	1770.84
Solomon_c103	1532.3	19.77	1279.4	12.30	1747.17
Solomon_c104	1532.8	20.26	1274.57	10.03	1703.68
Solomon_c105	1427.31	3.78	1375.38	16.89	1717.44
Solomon_c106	1527.95	10.39	1384.1	13.50	1766.49
Solomon_c107	1581.51	5.52	1498.71	10.41	1765.19
Solomon_c108	1594.1	26.87	1256.53	10.01	1771.46

Continues on next page

Continued from previous page

	mALNS*	Lower bound		Upper bound	
Instance	**Objective**	**Gap**	**Bound**	**Gap**	**Bound**
Solomon_c109	1553.21	31.31	1182.85	12.08	1766.61
Solomon_r101	1223.45	9.39	1118.47	10.64	1369.17
Solomon_r102	1312.76	49.42	878.557	7.58	1420.38
Solomon_r103	1298.86	22.96	1056.3	7.79	1408.64
Solomon_r104	1164	45.87	797.974	17.82	1416.44
Solomon_r105	1195	33.17	897.338	9.47	1320.04
Solomon_r106	1311.37	39.28	941.519	8.44	1432.22
Solomon_r107	1229.25	64.44	747.534	12.54	1405.44
Solomon_r108	1174.96	78.50	658.25	16.44	1406.1
Solomon_r109	1243.18	39.68	890	11.83	1409.98
Solomon_r110	1231.4	57.95	779.605	12.35	1404.83
Solomon_r111	1153.35	51.50	761.28	18.03	1407.04
Solomon_r112	1230.86	39.55	882	12.88	1412.85
Solomon_rc101	1404.66	45.52	965.257	15.05	1653.44
Solomon_rc102	1485.4	48.84	998	11.95	1687.09
Solomon_rc103	1469.22	69.46	867	11.96	1668.8
Solomon_rc104	1378.21	51.04	912.504	17.72	1675.09
Solomon_rc105	1548.92	64.52	941.454	8.59	1694.52
Solomon_rc106	1430.46	58.41	903	14.40	1671.14
Solomon_rc107	1457.85	73.09	842.238	12.05	1657.56
Solomon_rc108	1407.99	69.63	830.031	16.18	1679.74
Average		50.42		12.16	

Table A.33: Results reported by mALNS* and lower and upper bounds reported by CPLEX 12.6 for Class 3

	mALNS*	Lower bound		Upper bound	
Instance	**Objective**	**Gap**	**Bound**	**Gap**	**Bound**
Cordeau_pr01	540.648	17.28	461	13.84	627.484
Cordeau_pr02	1052.63	11.05	947.867	8.91	1155.64

Continues on next page

Continued from previous page

	mALNS*	Lower bound		Upper bound	
Instance	**Objective**	**Gap**	**Bound**	**Gap**	**Bound**
Cordeau_pr03	1570.04	61.03	975	7.50	1697.39
Cordeau_pr04	2001.91	60.93	1243.99	12.41	2285.64
Cordeau_pr05	2852.12	117.15	1313.45	10.85	3199.35
Cordeau_pr06	3132.84	149.03	1258	7.99	3404.79
Cordeau_pr07	784.441	6.78	734.606	10.90	880.439
Cordeau_pr08	1744.22	51.94	1147.93	7.34	1882.44
Cordeau_pr09	2380.07	74.36	1365.05	7.22	2565.42
Cordeau_pr10	3394.44	155.84	1326.77	6.44	3628.12
Solomon_c101	1413.71	148.02	570	16.51	1693.26
Solomon_c102	1556.8	18.52	1313.52	9.64	1722.91
Solomon_c103	1512.06	17.44	1287.56	9.58	1672.2
Solomon_c104	1481.26	17.46	1261.12	10.63	1657.4
Solomon_c105	1409.89	-0.93	1423.08	16.40	1686.46
Solomon_c106	1532.55	8.13	1417.35	10.58	1713.9
Solomon_c107	1551.9	4.39	1486.61	9.29	1710.91
Solomon_c108	1565.13	25.84	1243.78	9.25	1724.6
Solomon_c109	1562.28	31.19	1190.84	8.87	1714.32
Solomon_r101	1227.16	9.11	1124.67	8.16	1336.21
Solomon_r102	1294.08	78.74	723.989	6.32	1381.31
Solomon_r103	1258.46	42.13	885.42	7.29	1357.45
Solomon_r104	1145.54	61.12	711	16.52	1372.2
Solomon_r105	1229.4	40.98	872.067	5.81	1305.21
Solomon_r106	1297.88	50.76	860.897	7.64	1405.19
Solomon_r107	1207.25	46.19	825.791	10.64	1351.07
Solomon_r108	1134.05	57.11	721.819	16.08	1351.31
Solomon_r109	1225.22	29.11	949	9.82	1358.62
Solomon_r110	1183.78	41.13	838.792	12.28	1349.51
Solomon_r111	1166.36	62.49	717.819	13.92	1354.98
Solomon_r112	1233.72	36.34	904.857	9.69	1366.17
Solomon_rc101	1385.74	36.59	1014.51	12.16	1577.52
Solomon_rc102	1510.62	67.24	903.24	8.27	1646.8

Continues on next page

Continued from previous page

	mALNS*	**Lower bound**		**Upper bound**	
Instance	**Objective**	**Gap**	**Bound**	**Gap**	**Bound**
Solomon_rc103	1459.72	44.30	1011.61	9.29	1609.25
Solomon_rc104	1374.28	59.15	863.51	15.24	1621.47
Solomon_rc105	1520.12	53.26	991.848	8.52	1661.68
Solomon_rc106	1411.75	51.96	929	12.47	1612.81
Solomon_rc107	1434.48	68.43	851.652	9.46	1584.31
Solomon_rc108	1371.75	65.14	830.667	15.86	1630.34
Average		50.69		10.50	

Table A.34: Results reported by mALNS* and lower and upper bounds reported by CPLEX 12.6 for Class 4

	mALNS*	**Lower bound**		**Upper bound**	
Instance	**Objective**	**Gap**	**Bound**	**Gap**	**Bound**
Cordeau_pr01	520.204	1.54	512.329	16.94	626.296
Cordeau_pr02	946.508	10.74	854.675	17.95	1153.59
Cordeau_pr03	1359.11	61.30	842.593	18.89	1675.71
Cordeau_pr04	1832.72	50.76	1215.66	20.13	2294.65
Cordeau_pr05	2442.49	89.24	1290.66	20.37	3067.22
Cordeau_pr06	2704.42	138.07	1136	20.97	3422.1
Cordeau_pr07	714.631	7.60	664.131	19.07	883.038
Cordeau_pr08	1517.11	19.46	1269.99	18.66	1865.19
Cordeau_pr09	2058.49	44.68	1422.75	19.27	2550
Cordeau_pr10	2853.8	96.81	1450	20.28	3579.79
Solomon_c101	1243.44	-1.25	1259.15	25.13	1660.73
Solomon_c102	1346.14	14.58	1174.83	20.65	1696.48
Solomon_c103	1399.38	18.99	1176.02	17.32	1692.49
Solomon_c104	1312.01	21.45	1080.29	19.32	1626.16
Solomon_c105	1272.9	-6.58	1362.5	20.99	1611.06
Solomon_c106	1259.37	-0.16	1261.41	23.60	1648.3
Solomon_c107	1301.87	-0.31	1305.94	21.32	1654.62

Continues on next page

Continued from previous page

	mALNS*	Lower bound		Upper bound	
Instance	**Objective**	**Gap**	**Bound**	**Gap**	**Bound**
Solomon_c108	1383.91	17.40	1178.81	18.60	1700.05
Solomon_c109	1295.63	7.27	1207.84	22.25	1666.51
Solomon_r101	1044.79	1044790000.00	0	19.72	1301.39
Solomon_r102	1144.59	46.99	778.686	16.46	1370.14
Solomon_r103	1105.39	13.68	972.395	18.63	1358.41
Solomon_r104	1082.95	28.42	843.291	20.36	1359.77
Solomon_r105	1164.17	24.01	938.766	9.49	1286.18
Solomon_r106	1095.25	35.41	808.863	18.57	1345.08
Solomon_r107	1126.14	61.34	698	17.89	1371.45
Solomon_r108	1036.44	43.70	721.262	23.23	1349.98
Solomon_r109	1110.07	42.01	781.668	18.37	1359.93
Solomon_r110	1099.09	37.08	801.786	19.11	1358.75
Solomon_r111	1110.39	47.46	753	18.80	1367.43
Solomon_r112	1025.78	47.82	693.923	23.45	1340.02
Solomon_rc101	1311.72	24.41	1054.37	18.72	1613.81
Solomon_rc102	1268.03	47.92	857.257	20.68	1598.66
Solomon_rc103	1300.95	58.27	822	18.83	1602.8
Solomon_rc104	1248.96	67.89	743.917	22.27	1606.88
Solomon_rc105	1294.27	29.92	996.187	18.73	1592.47
Solomon_rc106	1188.88	31.07	907.073	24.73	1579.51
Solomon_rc107	1343.58	57.23	854.53	17.63	1631.11
Solomon_rc108	1261.14	63.45	771.591	20.78	1591.91
Average		26789523.07		19.70	

Table A.35: Results reported by mALNS* and lower and upper bounds reported by CPLEX 12.6 for Class 5

	mALNS*	Lower bound		Upper bound	
Instance	**Objective**	**Gap**	**Bound**	**Gap**	**Bound**
Cordeau_pr01	516.556	3.73	497.989	13.14	594.708

Continues on next page

Continued from previous page

	mALNS*	Lower bound		Upper bound	
Instance	**Objective**	**Gap**	**Bound**	**Gap**	**Bound**
Cordeau_pr02	933.313	7.92	864.802	14.05	1085.94
Cordeau_pr03	1338.08	32.61	1009.05	14.18	1559.17
Cordeau_pr04	1772.2	73.54	1021.21	16.03	2110.4
Cordeau_pr05	2308.91	79.94	1283.16	17.29	2791.48
Cordeau_pr06	2683.74	128.60	1174	15.39	3172
Cordeau_pr07	687.616	21.96	563.786	15.85	817.1
Cordeau_pr08	1479.82	26.34	1171.34	14.04	1721.59
Cordeau_pr09	2001.72	55.01	1291.37	15.44	2367.35
Cordeau_pr10	2789.46	89.71	1470.37	15.73	3310.29
Solomon_c101	1217.04	-1.59	1236.66	18.41	1491.61
Solomon_c102	1325.8	52.74	868.028	15.29	1565.15
Solomon_c103	1344.07	11.76	1202.6	13.47	1553.37
Solomon_c104	1261.79	4.43	1208.21	15.28	1489.35
Solomon_c105	1221.76	-4.73	1282.47	15.96	1453.83
Solomon_c106	1226.32	2.74	1193.63	16.37	1466.33
Solomon_c107	1233.37	-1.58	1253.11	16.54	1477.76
Solomon_c108	1352.33	18.51	1141.07	13.92	1570.98
Solomon_c109	1248.76	8.30	1153.05	17.00	1504.52
Solomon_r101	1022.29	2172.15	44.9921	14.83	1200.23
Solomon_r102	1107.99	12.11	988.289	13.63	1282.8
Solomon_r103	1069.73	6.17	1007.54	15.02	1258.77
Solomon_r104	1053.02	49.32	705.199	16.53	1261.55
Solomon_r105	1106.85	26.53	874.786	10.43	1235.67
Solomon_r106	1060.38	32.34	801.246	14.08	1234.08
Solomon_r107	1110.51	62.92	681.62	13.54	1284.42
Solomon_r108	1022.77	36.47	749.469	17.80	1244.19
Solomon_r109	1074.45	39.90	768.037	14.73	1260.02
Solomon_r110	1060.19	28.94	822.233	15.83	1259.61
Solomon_r111	1084.97	47.82	733.977	15.06	1277.39
Solomon_r112	1000.61	38.71	721.351	18.27	1224.28
Solomon_rc101	1279.73	14.64	1116.33	14.56	1497.73

Continues on next page

Continued from previous page

	mALNS*	**Lower bound**		**Upper bound**	
Instance	**Objective**	**Gap**	**Bound**	**Gap**	**Bound**
Solomon_rc102	1243.48	54.53	804.668	15.17	1465.88
Solomon_rc103	1271.1	59.41	797.364	13.93	1476.76
Solomon_rc104	1242.21	41.17	879.96	16.30	1484.07
Solomon_rc105	1260.53	20.30	1047.78	13.30	1453.95
Solomon_rc106	1165.51	20.62	966.28	18.46	1429.45
Solomon_rc107	1331.82	26.68	1051.31	12.95	1529.87
Solomon_rc108	1219.18	71.96	709	16.33	1457.06
Average		89.04		15.23	

Table A.36 shows the execution time in seconds required to report a solution by mALNS*, per instance.

Table A.36: Execution time in seconds required by mALNS* per instance

Instance	**Class 1**	**Class 2**	**Class 3**	**Class 4**	**Class 5**
Cordeau_pr01	6.09772	6.53486	7.09934	7.71522	8.41386
Cordeau_pr02	18.08	27.0591	27.716	37.7257	34.1416
Cordeau_pr03	36.2629	42.9195	43.3343	69.955	65.4295
Cordeau_pr04	72.1257	102.429	103.6	192.215	166.49
Cordeau_pr05	111.311	137.403	119.576	394.615	333.108
Cordeau_pr06	147.982	300.465	294.635	1406.26	802.776
Cordeau_pr07	13.4367	17.6944	18.6864	28.9433	30.3201
Cordeau_pr08	47.4603	63.8482	71.7079	142.604	101.936
Cordeau_pr09	90.1344	220.622	181.771	596.667	441.774
Cordeau_pr10	160.62	299.296	339.189	1402.6	1118.18
Solomon_c101	27.8287	32.4637	33.7495	71.6341	57.6768
Solomon_c102	37.3712	37.8328	37.1378	71.4798	69.3197
Solomon_c103	32.314	46.7183	52.0384	78.8501	81.2322
Solomon_c104	39.5737	52.0469	44.7179	92.2204	90.8017
Solomon_c105	29.8863	30.6707	33.1673	55.5037	52.2787
Solomon_c106	30.1046	36.6667	36.4233	75.8611	76.6414

Continues on next page

Continued from previous page

Instance	Class 1	Class 2	Class 3	Class 4	Class 5
Solomon_c107	30.6708	42.4646	45.5872	86.1521	80.6047
Solomon_c108	30.5744	42.294	38.3461	67.5447	60.2113
Solomon_c109	34.6888	37.533	37.2555	81.0837	75.552
Solomon_r101	46.7891	58.471	59.8922	106.439	117.377
Solomon_r102	48.611	76.8	76.3808	182.444	147.285
Solomon_r103	37.368	54.2028	62.2297	106.796	96.3156
Solomon_r104	28.1691	28.4527	29.6812	37.0861	35.8057
Solomon_r105	35.4199	37.0437	38.6028	57.8873	55.8351
Solomon_r106	32.0727	41.9416	45.057	89.0416	78.1855
Solomon_r107	30.3844	34.8411	41.7337	48.3919	45.7084
Solomon_r108	26.2905	30.992	28.1818	40.9784	36.4527
Solomon_r109	30.4567	40.2915	40.6819	68.6004	69.4367
Solomon_r110	29.2327	39.4112	36.8361	52.184	54.516
Solomon_r111	30.8314	33.7528	33.0375	42.2111	41.6222
Solomon_r112	27.0335	34.7878	33.7108	56.4949	50.461
Solomon_rc101	39.2602	41.9719	44.8926	53.6889	71.3989
Solomon_rc102	34.2463	36.6468	43.2251	63.7678	67.8866
Solomon_rc103	30.0294	39.6386	39.2275	53.616	57.3354
Solomon_rc104	29.2001	33.6577	32.1133	42.2541	46.5312
Solomon_rc105	34.3637	38.6198	40.7033	78.4448	78.3671
Solomon_rc106	31.8376	36.7894	36.0072	54.8232	60.8844
Solomon_rc107	32.0083	39.9831	40.2074	47.9405	48.1589
Solomon_rc108	29.6238	32.6162	29.209	42.5346	37.7637
Total	1659.75	2387.87	2397.35	6285.25	5044.21

Appendix B

Detailed results for the orienteering problem with mandatory visits and conflicts

Tables B.1 to B.9 display the results reported by CPLEX for each instance of the OPMVC, by using the OPMVC-DL, OPMVC-GG, OPMVC-W, OPMVC-DFJ, and OPMVC-C formulations. For each instance, it is shown its name, the reported objective function value, the number of visited nodes in the reported solution, and the execution time.

An objective function value followed by an * indicates that the solver reached the optimal solution but it was not able to prove its optimality. If the time limit is smaller than 3600 seconds, the reported solution is optimal.

Table B.1: Solutions reported by CPLEX for Class 1

Instance	OPMVC-DL			OPMVC-GG			OPMVC-W			OPMVC-DFJ			OPMVC-FT		
	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**
att48A	15	19	143.26	15	19	15.74	15	19	317.55	15	19	10.32	15	19	654.35
att48B	17	21	15.74	17	21	13.44	-	-	>3600	17	21	23.12	17	21	1645.4
att48C	9*	13	>3600	9	13	33.77	9	13	208.55	9	13	207.05	9	13	352.22
att48D	13	17	365.38	13	17	5.31	13	17	285.21	13	17	12.99	13	17	65.43

Continues on next page

Continued from previous page

Instance	OPMVC-DL			OPMVC-GG			OPMVC-W			OPMVC-DFJ			OPMVC-FT		
	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**
att48E	14	18	638.34	14	18	16.49	14	18	189.48	14	18	11.07	14	18	182.83
cmt121A	-	-	>3600	-	-	>3600	-	-	>3600	535	48	16.52	535	48	160.24
cmt121B	-	-	>3600	-	-	>3600	-	-	>3600	498	-	>3600	513	-	>3600
cmt121C	-	-	>3600	476	46	>3600	-	-	>3600	514	-	>3600	521	-	>3600
cmt121D	-	-	>3600	-	-	>3600	-	-	>3600	530	48	193.58	530*	-	>3600
cmt151A	-	-	>3600	815	53	1433.01	-	-	>3600	818	-	>3600	835	-	>3600
cmt151B	872	55	124.51	872	55	3396.9	-	-	>3600	872	55	285.65	-	-	>3600
cmt151C	-	-	>3600	-	-	>3600	-	-	>3600	484	-	>3600	546	-	>3600
cmt151D	-	-	>3600	-	-	>3600	-	-	>3600	611	-	>3600	649	-	>3600
cmt151E	-	-	>3600	-	-	>3600	-	-	>3600	693	-	>3600	722	-	>3600
cmt200A	-	-	>3600	-	-	>3600	-	-	>3600	749	-	>3600	877	-	>3600
cmt200B	-	-	>3600	-	-	>3600	-	-	>3600	1352	84	1529.4	-	-	>3600
cmt200C	-	-	>3600	-	-	>3600	-	-	>3600	835	-	>3600	956	-	>3600
cmt200D	-	-	>3600	-	-	>3600	-	-	>3600	954	-	>3600	1075	-	>3600
cmt200E	-	-	>3600	-	-	>3600	-	-	>3600	1045	-	>3600	1159	-	>3600
eil30A	6375	12	0.15	6375	12	2.83	6375	12	14.76	6375	12	0.56	6375	12	0.58
eil30B	5125*	9	>3600	5125	9	10.87	5125	9	104.69	5125	9	9.28	5125	9	43.22
eil30C	5775	10	1321.67	5775	10	8.5	5775	10	46.16	5775	10	4.7	5775	10	6.16
eil30D	6275	11	46.85	6275	11	3.89	6275	11	12.02	6275	11	0.61	6275	11	1.21
eil33A	5230*	10	>3600	5230	10	88.85	5230	10	162.98	9780	-	>3600	5230	10	105.39
eil33B	14380	14	111.55	14380	14	1.21	14380	14	131.4	14380	14	1.32	14380	14	2.89
eil33C	7430*	13	>3600	7430	13	34.55	7430	13	66.71	10320	-	>3600	7430	13	122.92
eil33D	11630*	12	>3600	11630	12	5.67	11630	12	14.29	11630	12	1277.64	11630	12	6.15
eil33E	12830*	14	>3600	12830	14	6.19	12830	14	62.15	12830	14	31.12	12830	14	9.17
eil51A	245	19	199.1	245	19	7.33	245	19	182.67	245	18	35.99	248	-	>3600
eil51B	287	20	2.3	287	20	0.88	287	20	469.28	287	20	2.74	287	20	23.8
eil51C	122	11	248.74	122	11	35.54	122	11	977.85	122	11	69.56	122	11	41.26
eil51D	150	14	3250.1	150	14	27.53	150	14	967.31	150	14	418.37	150	14	127.62
eil51E	177	15	345.34	177	15	32.26	177	15	216.12	177	15	1194.27	177	15	722.3
eil76A	520*	30	>3600	520	30	93.09	520	30	2446.14	520	30	66.57	530	-	>3600
eil76B	599	32	37.15	599	32	48.25	-	-	>3600	599	32	11.04	599	32	203.54
eil76C	232*	18	>3600	232	18	60.49	232	18	3552.34	232	18	2769.67	232	18	1183.97
eil76D	305	20	>3600	312	21	28.06	312	21	1285.68	312	21	1456.77	312	21	1345.75
eil76E	367	23	1479.08	367	23	50.32	367	23	891.04	367	23	208.31	369	-	>3600
eil101A	556	38	>3600	570	40	2608.02	-	-	>3600	570	40	192.86	589	-	>3600
eil101B	612	40	3.18	612	40	5.27	-	-	>3600	612	40	132.84	-	-	>3600
eil101C	-	-	>3600	-	-	>3600	-	-	>3600	294	-	>3600	281	24	1013.78
eil101D	-	-	>3600	367	29	796.14	-	-	>3600	367*	-	>3600	367	29	750.78
eil101E	-	-	>3600	414	32	438.97	-	-	>3600	415	-	>3600	429	-	>3600
gil262A	-	-	>3600	-	-	>3600	-	-	>3600	4463	-	>3600	4723	-	>3600
gil262B	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600

Continues on next page

Continued from previous page

Instance	OPMVC-DL			OPMVC-GG			OPMVC-W			OPMVC-DFJ			OPMVC-FT		
	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**
gil262C	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	3711	-	>3600
gil262D	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	4086	-	>3600
gil262E	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	4369	-	>3600
op21A	165	6	0.58	165	6	0.53	165	6	1.15	165	6	0.9	165	6	0.4
op21B	135	6	0.37	135	6	1.55	135	6	1.3	135	6	1.19	135	6	0.86
op21C	150	7	0.32	150	7	0.96	150	7	0.81	150	7	0.73	150	7	0.35
op21D	155	7	0.57	155	7	1.39	155	7	1.96	155	7	0.85	155	7	0.84
op32A	85	11	5.95	85	11	1.95	85	11	7.13	85	11	1.01	85	11	2.19
op32B	115	13	0.15	115	13	1.1	115	13	25.52	115	13	0.46	115	13	0.29
op32C	35	7	65.63	35	7	6.24	35	7	2.56	35	7	10.51	35	7	4.42
op32D	60	9	32.79	60	9	3.16	60	9	3.56	60	9	5.29	60	9	1.6
op32E	75	10	13.37	75	10	2.69	75	10	4.47	75	10	2.5	75	10	0.85
op33A	260	12	0.33	260	12	1.02	260	12	3.27	260	12	0.63	260	12	2.05
op33B	330	13	1.22	330	13	1.27	330	13	7.71	330	13	0.33	330	13	1.44
op33C	110	9	8.71	110	9	7.57	110	9	3.92	110	9	1.18	110	9	2.65
op33D	160	10	0.85	160	10	1.72	160	10	5.37	160	10	1.24	160	10	1.35
op33E	180	10	1.19	180	10	2.27	180	10	5.35	180	10	0.96	180	10	3.42

Table B.2: Solutions reported by CPLEX for Class 2

Instance	OPMVC-DL			OPMVC-GG			OPMVC-W			OPMVC-DFJ			OPMVC-FT		
	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**
att48A	12	21	3.72	12	21	15.23	12	21	503.08	12	21	4.3	12	21	5.1
att48B	11*	20	>3600	11	20	853.03	11	20	88.64	11	20	3.26	11	20	5.54
att48C	12	21	3.03	12	21	7.22	12	21	419.3	12	21	2.6	12	21	4.2
att48D	12	21	5.07	12	21	0.74	12	21	478.07	12	21	3.21	12	21	2.91
cmt121A	-	-	>3600	-	-	>3600	-	-	>3600	330	48	6.89	330	48	44.58
cmt121B	-	-	>3600	-	-	>3600	-	-	>3600	305	47	211.71	310	-	>3600
cmt121C	-	-	>3600	-	-	>3600	-	-	>3600	330	48	24.36	330	48	24.29
cmt121D	-	-	>3600	-	-	>3600	-	-	>3600	330	48	35.84	330	48	161.63
cmt151A	-	-	>3600	462	53	657.95	-	-	>3600	462	53	162.9	481	-	>3600
cmt151B	561	55	504.64	-	-	>3600	-	-	>3600	561	55	102.49	-	-	>3600
cmt151C	-	-	>3600	433	52	374.75	-	-	>3600	433	52	401.8	441	-	>3600
cmt151D	-	-	>3600	-	-	>3600	-	-	>3600	496	53	182.83	509	-	>3600
cmt151E	-	-	>3600	541	55	899.58	-	-	>3600	541	55	57.39	548	-	>3600
cmt200A	-	-	>3600	-	-	>3600	-	-	>3600	825	84	968.45	-	-	>3600
cmt200B	-	-	>3600	-	-	>3600	-	-	>3600	528	-	>3600	554	-	>3600
cmt200C	-	-	>3600	-	-	>3600	-	-	>3600	641	-	>3600	-	-	>3600
cmt200D	-	-	>3600	-	-	>3600	-	-	>3600	732	-	>3600	758	-	>3600

Continues on next page

Continued from previous page

Instance	OPMVC-DL			OPMVC-GG			OPMVC-W			OPMVC-DFJ			OPMVC-FT		
	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**
eil30A	2925	11	19.84	2925	11	1.75	2925	11	6.05	2925	11	0.24	2925	11	0.53
eil30B	1625	10	32.79	1625	10	1.86	1625	10	6.32	1625	10	1.38	1625	10	2.15
eil30C	1625	10	138.8	1625	10	2.42	1625	10	12.39	1625	10	3.55	1625	10	3.1
eil30D	2475	10	81.67	2475	10	1.34	2475	10	7.8	2475	10	0.88	2475	10	0.53
eil33A	8480	14	424.29	8480	14	1.29	8480	14	9.81	8480	14	4.21	8480	14	0.72
eil33B	8480	14	57.65	8480	14	3.4	8480	14	8.68	8480	14	2.94	8480	14	0.59
eil33C	8480	14	2378.94	8480	14	2.22	8480	14	15.05	8480	14	3.32	8480	14	1.53
eil33D	9430	14	1.51	9430	14	0.88	9430	14	5.97	9430	14	0.78	9430	14	0.44
eil51A	168	20	0.31	168	20	0.59	168	20	88.55	168	20	1.56	168	20	0.53
eil51B	148	19	521.77	148	19	24.12	148	19	186.63	148	19	15.84	148	19	16.15
eil51C	168	20	1.02	168	20	5.35	168	20	39.42	168	20	0.81	168	20	1.33
eil51D	168	20	2.03	168	20	0.62	168	20	70.91	168	20	0.66	168	20	1.25
eil76A	377	31	602.01	377	31	8.42	377	31	3046.36	377	31	18.93	377	31	173.05
eil76B	409	32	1.75	409	32	1.7	409	32	3205.73	409	32	3.98	409	32	0.78
eil76C	-	-	>3600	201	24	20.33	201	24	763.1	201	24	57.28	201	24	19.17
eil76D	294	28	63.82	294	28	11.57	294	28	1107.52	294	28	28.35	294	28	27.52
eil76E	345	30	126.07	345	30	7.31	345	30	1945.19	345	30	21.13	345	30	26.77
eil101A	383	40	>3600	386	40	135.39	-	-	>3600	386	40	25.81	386	40	266.41
eil101B	409	40	32.84	409	40	54.49	-	-	>3600	409	40	85.53	409	40	141.41
eil101C	-	-	>3600	260	35	6.91	-	-	>3600	260	35	7.34	260	35	12.49
eil101D	326*	38	>3600	326	38	55.32	-	-	>3600	326	38	9.72	326	38	852.51
eil101E	-	-	>3600	370	40	75.44	-	-	>3600	370	40	30.8	370	40	2524.06
gil262A	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600
gil262B	-	-	>3600	-	-	>3600	-	-	>3600	2502	-	>3600	2658	-	>3600
gil262C	-	-	>3600	-	-	>3600	-	-	>3600	2867	-	>3600	-	-	>3600
gil262D	-	-	>3600	-	-	>3600	-	-	>3600	3000	103	2604.54	-	-	>3600
op21A	60	7	0.1	60	7	0.43	60	7	0.29	60	7	0.56	60	7	0.58
op21B	125	7	0.21	125	7	0.34	125	7	0.45	125	7	0.21	125	7	0.08
op21C	60	7	0.13	60	7	0.4	60	7	0.31	60	7	0.96	60	7	0.57
op21D	75	7	0.18	75	7	2.88	75	7	1.38	75	7	0.89	75	7	0.37
op21E	80	7	0.33	80	7	3.13	80	7	0.96	80	7	1.39	80	7	0.5
op32A	80	13	1.23	80	13	1.97	80	13	4.02	80	13	0.29	80	13	0.25
op32B	70	12	4.48	70	12	1.81	70	12	2.77	70	12	0.85	70	12	0.93
op32C	80	13	1.46	80	13	0.36	80	13	4.37	80	13	0.29	80	13	0.25
op33A	210	13	1.86	210	13	0.59	210	13	1.52	210	13	0.25	210	13	0.19
op33B	170	12	4.05	170	12	4.33	170	12	1.56	170	12	1.56	170	12	0.68
op33C	210	13	0.49	210	13	0.92	210	13	1.64	210	13	0.38	210	13	0.18
op33D	220	13	0.1	220	13	2.36	220	13	2.95	220	13	0.31	220	13	0.38

Table B.3: Solutions reported by CPLEX for Class 3

Instance	OPMVC-DL			OPMVC-GG			OPMVC-W			OPMVC-DFJ			OPMVC-FT		
	z	$\sum y$	Time	z	$\sum y$	Time	z	$\sum y$	Time	z	$\sum y$	Time	z	$\sum y$	Time
att48A	4	18	462.26	4	18	3.93	4	18	42.87	4	18	9.57	4	18	9.76
att48B	7	21	0.32	7	21	1.12	7	21	521.83	7	21	2.89	7	21	13.18
att48C	4	18	2559.55	4	18	6.54	4	18	169.92	4	18	6.27	4	18	10.92
att48D	5	19	158.76	5	19	8.44	5	19	30.08	5	19	6.35	5	19	13.62
att48E	5	19	669.28	5	19	8.09	5	19	47.93	5	19	7.67	5	19	41.44
cmt121A	-	-	>3600	-	-	>3600	-	-	>3600	152	48	5.89	152	48	10
cmt121B	-	-	>3600	152	48	67.6	-	-	>3600	152	48	5.31	152	48	7.1
cmt121C	-	-	>3600	-	-	>3600	-	-	>3600	152	48	23.22	152	48	10.78
cmt121D	152	48	173.1	152	48	64.93	-	-	>3600	152	48	15.89	152	48	46.65
cmt151A	-	-	>3600	-	-	>3600	-	-	>3600	199	55	103.22	199	55	265.54
cmt151B	-	-	>3600	-	-	>3600	-	-	>3600	179	54	60.12	179	54	23.81
cmt151C	-	-	>3600	-	-	>3600	-	-	>3600	199	55	29.28	199	55	443.73
cmt151D	-	-	>3600	-	-	>3600	-	-	>3600	199	55	66.23	199	55	835.32
cmt200A	-	-	>3600	-	-	>3600	-	-	>3600	463	84	117.29	-	-	>3600
cmt200B	-	-	>3600	391	77	370.57	-	-	>3600	391	77	937.63	391	77	66.73
cmt200C	-	-	>3600	-	-	>3600	-	-	>3600	452	83	149.27	452	83	337.15
cmt200D	-	-	>3600	-	-	>3600	-	-	>3600	463	84	62.34	-	-	>3600
eil30A	950	11	172.99	950	11	5.79	950	11	2.5	950	11	0.87	950	11	1.12
eil30B	1950	11	48.25	1950	11	5.39	1950	11	3.74	1950	11	0.47	1950	11	0.32
eil30C	2075	12	0.04	2075	12	0.17	2075	12	1.38	2075	12	0.33	2075	12	0.09
eil30D	2075	12	0.04	2075	12	0.11	2075	12	1.13	2075	12	0.24	2075	12	0.07
eil33A	5530	14	1.18	5530	14	0.44	5530	14	1.33	5530	14	0.28	5530	14	0.11
eil33B	5530	14	1.91	5530	14	0.7	5530	14	1.23	5530	14	0.23	5530	14	0.22
eil33C	5530	14	0.27	5530	14	0.25	5530	14	1.32	5530	14	0.6	5530	14	0.21
eil51A	99	20	0.39	99	20	1.4	99	20	28.36	99	20	0.37	99	20	0.68
eil51B	90	20	68.7	90	20	10.32	90	20	191.43	90	20	3.31	90	20	1.23
eil51C	99	20	1.76	99	20	5.11	99	20	19.69	99	20	0.49	99	20	0.5
eil51D	99	20	0.71	99	20	0.59	99	20	22.87	99	20	0.91	99	20	0.29
eil76A	167	30	30.03	167	30	5.55	167	30	199.36	167	30	4.92	167	30	2.12
eil76B	202	32	0.32	202	32	0.88	-	-	>3600	202	32	6.13	202	32	11.12
eil76C	142	29	525.68	142	29	4.78	142	29	655.83	142	29	17.82	142	29	4.13
eil76D	193	32	107.31	193	32	23.63	192	31	>3600	193	32	15.9	193	32	6.16
eil76E	202	32	0.81	202	32	47.25	202	32	2522.54	202	32	6.64	202	32	5.72
eil101A	175	40	16.28	175	40	34.64	-	-	>3600	175	40	6.26	175	40	18.66
eil101B	155	37	3375.58	155	37	11.62	-	-	>3600	155	37	13.03	155	37	4.29
eil101C	175	40	1208.89	175	40	2416.59	-	-	>3600	175	40	6.59	175	40	27.61
eil101D	175	40	367.8	175	40	42.7	-	-	>3600	175	40	30.4	175	40	14.13
gil262A	-	-	>3600	-	-	>3600	-	-	>3600	1369	106	671.04	-	-	>3600
gil262B	-	-	>3600	-	-	>3600	-	-	>3600	1239	-	>3600	-	-	>3600

Continues on next page

Continued from previous page

Instance	OPMVC-DL			OPMVC-GG			OPMVC-W			OPMVC-DFJ			OPMVC-FT		
	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**
gil262C	-	-	>3600	-	-	>3600	-	-	>3600	1369	106	362.12	-	-	>3600
gil262D	-	-	>3600	-	-	>3600	-	-	>3600	1369	106	423.25	-	-	>3600
op21A	45	7	0.04	45	7	0.06	45	7	0.09	45	7	0.01	45	7	0.04
op21B	30	6	0.03	30	6	0.1	30	6	0.12	30	6	0.04	30	6	0.03
op21C	45	7	0.03	45	7	0.08	45	7	0.1	45	7	0.01	45	7	0.01
op21D	45	7	0.01	45	7	0.04	45	7	0.09	45	7	0.05	45	7	0.06
op32A	45	13	0.11	45	13	0.5	45	13	2.21	45	13	0.13	45	13	0.23
op32B	45	13	0.12	45	13	1.07	45	13	1.46	45	13	0.15	45	13	0.05
op32C	45	13	0.18	45	13	0.35	45	13	2.02	45	13	0.2	45	13	0.15
op32D	45	13	0.21	45	13	0.13	45	13	1.95	45	13	0.31	45	13	0.44
op33A	130	13	0.03	130	13	0.18	130	13	1.04	130	13	0.18	130	13	0.14
op33B	80	11	0.44	80	11	0.22	80	11	0.74	80	11	0.08	80	11	0.17
op33C	120	13	1.34	120	13	0.36	120	13	1.25	120	13	0.13	120	13	0.11
op33D	130	13	0.27	130	13	0.82	130	13	0.9	130	13	0.08	130	13	0.04

Table B.4: Solutions reported by CPLEX for Class 4

Instance	OPMVC-DL			OPMVC-GG			OPMVC-W			OPMVC-DFJ			OPMVC-FT		
	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**
att48A	21	25	2493.92	21	25	5.45	21	25	373.37	21	25	47.81	21	25	35.88
att48B	28	32	>3600	29	33	19.52	29	33	1149.53	29	33	8.41	30	-	>3600
att48C	13*	17	>3600	13	17	122.29	13	17	336.79	13*	-	>3600	13	17	78.56
att48D	17*	21	>3600	17	21	8.76	17	21	386	17	21	25.71	17	21	38.01
att48E	20*	24	>3600	20	24	12.09	20	24	740.4	20	24	7.2	20	24	54.76
cmt121A	-	-	>3600	-	-	>3600	-	-	>3600	817	-	>3600	911	-	>3600
cmt121B	-	-	>3600	-	-	>3600	-	-	>3600	594	-	>3600	824	-	>3600
cmt121C	-	-	>3600	-	-	>3600	-	-	>3600	678	-	>3600	872	-	>3600
cmt121D	-	-	>3600	-	-	>3600	-	-	>3600	778	-	>3600	884	-	>3600
cmt151A	995	67	>3600	1087	72	2547.29	-	-	>3600	1096	-	>3600	1201	-	>3600
cmt151B	1418		>3600	1406	95	>3600	-	-	>3600	1485	-	>3600	-	-	>3600
cmt151C	-	-	>3600	-	-	>3600	-	-	>3600	554	-	>3600	662	-	>3600
cmt151D	-	-	>3600	-	-	>3600	-	-	>3600	700	-	>3600	790	-	>3600
cmt151E	-	-	>3600	-	-	>3600	-	-	>3600	829	-	>3600	897	-	>3600
cmt200A	-	-	>3600	-	-	>3600	-	-	>3600	899	-	>3600	1202	-	>3600
cmt200B	-	-	>3600	-	-	>3600	-	-	>3600	2175	-	>3600	2270	-	>3600
cmt200C	-	-	>3600	-	-	>3600	-	-	>3600	1018	-	>3600	1294	-	>3600
cmt200D	-	-	>3600	-	-	>3600	-	-	>3600	1208	-	>3600	1465	-	>3600
cmt200E	-	-	>3600	-	-	>3600	-	-	>3600	1346	-	>3600	1609	-	>3600
eil30A	9350	18	>3600	9375	17	18.7	9375	17	132.46	9375	17	16.36	9375	17	516.7

Continues on next page

Continued from previous page

Instance	OPMVC-DL			OPMVC-GG			OPMVC-W			OPMVC-DFJ			OPMVC-FT		
	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**
eil30B	5450*	12	>3600	5450	12	60.45	5450	12	172.28	5450	12	131.77	5450	12	1796.22
eil30C	7300*	13	>3600	7300	13	30.27	7300	13	169.24	7300	13	188.59	7300	13	2220.18
eil30D	7450	9	>3600	8575	13	31.44	8575	13	48.49	8575	13	35.45	8575	13	1148.82
eil33A	6250	11	>3600	6950	11	41.85	6950	11	36.83	15070	-	>3600	6950	11	57.96
eil33B	19530	22	>3600	19730	23	9.26	19730	23	68.9	20020	-	>3600	19730	23	102.41
eil33C	10630*	17	>3600	10630	17	77.43	10630	17	23.13	-	-	>3600	10630	17	30.51
eil33D	14180*	17	>3600	14180	17	9.65	14180	17	46.62	17480	-	>3600	14180	17	807.33
eil33E	17430*	20	>3600	17430	20	7.95	17430	20	131.71	18180	-	>3600	17430	20	28.04
eil51A	328	22	907.01	328	22	45.76	328*	22	>3600	329	-	>3600	356	-	>3600
eil51B	509	34	230	509	34	21.85	509	34	2094.1	509	34	9.51	-	-	>3600
eil51C	168	13	167.82	168	13	11.53	168	13	1478.53	168	13	26.58	168	13	18.67
eil51D	201	16	222.61	201	16	39.25	200	14	>3600	201	16	245.25	201	16	110.5
eil51E	232	18	365.07	232	18	28.31	232	18	1499	232	18	891.03	232	18	1447.2
eil76A	629*	37	>3600	629	37	103.62	-	-	>3600	629	37	45.03	659	-	>3600
eil76B	923*	52	>3600	923	52	343.88	-	-	>3600	923	52	96.96	935	-	>3600
eil76C	274*	21	>3600	274	21	76.58	274*	21	>3600	276	-	>3600	274	21	1194.28
eil76D	366*	26	>3600	366	26	73.19	-	-	>3600	375	-	>3600	372	-	>3600
eil76E	442	27	1180.38	442	27	20.45	-	-	>3600	442*	-	>3600	445	-	>3600
eil101A	785	51	>3600	789	51	781.72	-	-	>3600	795	-	>3600	824	-	>3600
eil101B	1067	70	>3600	1071	70	1244.04	-	-	>3600	1071	70	1684.17	1090	-	>3600
eil101C	-	-	>3600	326	25	660.61	-	-	>3600	344	-	>3600	326	25	1665.09
eil101D	422*	32	>3600	422	32	342.76	-	-	>3600	435	-	>3600	427	-	>3600
eil101E	445	33	>3600	491	35	842.24	-	-	>3600	-	-	>3600	513	-	>3600
gil262A	-	-	>3600	-	-	>3600	-	-	>3600	5813	-	>3600	6910	-	>3600
gil262B	-	-	>3600	-	-	>3600	-	-	>3600	8036	-	>3600	-	-	>3600
gil262C	-	-	>3600	-	-	>3600	-	-	>3600	3406	-	>3600	4969	-	>3600
gil262D	-	-	>3600	-	-	>3600	-	-	>3600	4180	-	>3600	5432	-	>3600
gil262E	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	5998	-	>3600
op21A	260	14	9.61	260	14	4.72	260	14	12.09	260	13	1.63	260	14	37.08
op21B	180	9	5.98	180	9	8.07	180	9	1.37	180	9	1.49	180	9	2.68
op21C	215	11	6.01	215	11	2.87	215	11	1.92	215	11	1.53	215	11	4.27
op21D	260	14	5.36	260	14	2.77	260	14	1.54	260	14	1.08	260	14	4.76
op32A	110	15	18.71	110	15	1.32	110	15	3.62	110	15	1.53	110	15	3.07
op32B	180	22	6.5	180	22	5.72	180	22	4.68	180	21	1.01	180	22	32.47
op32C	35	7	446.29	35	7	10.83	35	7	3.19	35	7	14.71	35	7	5.69
op32D	75	11	41.1	75	11	1.03	75	11	3.27	75	11	2.94	75	11	1
op32E	90	13	46.07	90	13	2.85	90	13	2.9	90	13	4.17	90	13	2.07
op33A	310	16	5.96	310	17	4.13	310	16	12.99	310	16	0.87	310	16	36.5
op33B	460	21	6.1	460	22	3.42	-	-	>3600	460	21	0.22	460	20	71.85
op33C	120	9	131.66	120	9	4.14	120	9	4.36	120	9	2.36	120	9	1.99
op33D	180	10	4.07	180	10	5.74	180	11	4.95	180	10	1.32	180	11	1.14

Continues on next page

Continued from previous page

Instance	OPMVC-DL			OPMVC-GG			OPMVC-W			OPMVC-DFJ			OPMVC-FT		
	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**
op33E	220	12	2.87	220	12	2.47	220	12	4.22	220	12	0.81	220	12	1.3

Table B.5: Solutions reported by CPLEX for Class 5

Instance	OPMVC-DL			OPMVC-GG			OPMVC-W			OPMVC-DFJ			OPMVC-FT		
	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**
att48A	24	33	2016.54	24	33	21.77	24	33	366.71	24	33	22.29	24	33	11.93
att48B	19*	28	>3600	19	28	21.3	19	28	371.23	19	28	22.22	19	28	22.72
att48C	23*	32	>3600	23	32	25.02	23	32	88.74	23	32	4.79	23	32	8.75
att48D	25	34	1403.44	25	34	21.72	25	34	315.08	25	34	6.04	25	34	12.27
cmt121A	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	716	-	>3600
cmt121B	-	-	>3600	-	-	>3600	-	-	>3600	494	-	>3600	669	-	>3600
cmt121C	-	-	>3600	-	-	>3600	-	-	>3600	643	-	>3600	708	-	>3600
cmt121D	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	758	-	>3600
cmt151A	-	-	>3600	681	65	2010.52	-	-	>3600	681	65	2146.73	721	-	>3600
cmt151B	-	-	>3600	-	-	>3600	-	-	>3600	1207	102	687.28	-	-	>3600
cmt151C	-	-	>3600	610	60	988.27	-	-	>3600	610	61	3550.11	637	-	>3600
cmt151D	-	-	>3600	-	-	>3600	-	-	>3600	776	-	>3600	-	-	>3600
cmt151E	-	-	>3600	914	81	829.27	-	-	>3600	914	81	31.04	-	-	>3600
cmt200A	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	1724	-	>3600
cmt200B	-	-	>3600	-	-	>3600	-	-	>3600	889	-	>3600	971	-	>3600
cmt200C	-	-	>3600	1053	95	2617.97	-	-	>3600	1111	-	>3600	1180	-	>3600
cmt200D	-	-	>3600	-	-	>3600	-	-	>3600	1272	-	>3600	-	-	>3600
eil30A	5475	16	>3600	6225	16	8.6	6225	16	66.57	6225	16	19.56	6225	16	166.99
eil30B	3350*	13	>3600	3350	13	8.59	3350	13	18.86	3350	13	49.89	3350	13	264.94
eil30C	4750*	13	>3600	4750	13	9.84	4750	13	18.85	4750	13	17.68	4750	13	182.14
eil30D	5075*	16	>3600	5075	16	15.73	5075	16	69.39	5075	16	35.24	5075	16	712.1
eil33A	14230*	22	>3600	14230	22	7.74	14230	22	45.49	14230	22	2513.2	14230	22	6.23
eil33B	-	-	>3600	12880	19	8.63	12880	19	63.08	13630	-	>3600	12880	19	10.74
eil33C	14930*	23	>3600	14930	23	3.09	14930	23	49.02	14930	23	938.85	14930	23	8.04
eil33D	15430	24	>3600	15680	25	19.13	15680	25	64.69	15680	25	9.3	15680	25	17.64
eil51A	402	35	54.55	402	35	3.63	402	35	1366.26	402	35	4.01	402	35	136.69
eil51B	236	24	1248.43	236	24	61.89	236	24	3334.86	244	-	>3600	236	24	61.82
eil51C	302	29	611.36	302	29	74.25	302	29	3471.34	305	-	>3600	302	28	333.64
eil51D	357	32	201.31	357	32	44.22	357	32	1059.95	357	32	7.72	357	32	79.92
eil76A	477	37	>3600	481	36	51.23	-	-	>3600	-	-	>3600	483	-	>3600
eil76B	754	53	2662.98	754	53	77.79	-	-	>3600	754	53	83.32	776	-	>3600
eil76C	-	-	2037.99	228	26	125.34	-	-	>3600	228	26	2756.03	228	26	307.57
eil76D	-	-	>3600	341	30	94.53	-	-	>3600	341	30	1552.32	341	30	182.9

Continues on next page

Continued from previous page

Instance	OPMVC-DL			OPMVC-GG			OPMVC-W			OPMVC-DFJ			OPMVC-FT		
	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**
eil76E	411	33	2037.99	411	33	128.51	-	-	>3600	-	-	>3600	411	33	1465
eil101A	559*	52	>3600	559	52	374.04	-	-	>3600	568	-	>3600	567	-	>3600
eil101B	845	69	>3600	857	69	312.21	-	-	>3600	857	69	297.69	871	-	>3600
eil101C	-	-	>3600	296	38	159.87	-	-	>3600	296	38	192.78	296	38	319.39
eil101D	-	-	>3600	418	43	235.67	-	-	>3600	427	-	>3600	418	43	345.59
eil101E	521	49	1692.44	521	49	104.89	-	-	>3600	522	-	>3600	521	49	869.75
gil262A	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	6572	-	>3600
gil262B	-	-	>3600	-	-	>3600	-	-	>3600	4090	-	>3600	5001	-	>3600
gil262C	-	-	>3600	-	-	>3600	-	-	>3600	4963	-	>3600	5585	-	>3600
gil262D	-	-	>3600	-	-	>3600	-	-	>3600	5647	-	>3600	6094	-	>3600
op21A	135	12	0.22	135	12	0.4	135	12	0.96	135	12	0.15	135	12	0.58
op21B	235	13	7.93	235	13	3.27	235	13	2.56	235	13	1.04	235	13	36.86
op21C	135	12	0.23	135	12	0.5	135	12	0.81	135	12	0.12	135	12	0.49
op21D	135	12	0.85	135	12	0.74	135	12	2.5	135	12	0.42	135	12	0.97
op21E	165	13	0.55	165	13	1.49	165	13	2.99	165	13	0.31	165	13	2.49
op32A	150	22	3.69	150	22	1.53	150	22	4.85	150	22	2.05	150	22	0.47
op32B	110	17	17.27	110	17	2.4	110	17	5.93	110	17	3.84	110	17	2.27
op32C	140	20	5.9	140	20	1.29	140	20	7.24	140	20	0.54	140	21	0.34
op33A	400	20	1.4	400	20	1.94	400	20	13.98	400	20	1.14	400	20	0.9
op33B	270	15	1580.71	270	15	15.3	270	15	17.37	270	15	30.25	270	15	4.37
op33C	410	21	0.55	410	21	0.89	410	21	9.95	410	21	1.04	410	21	0.31
op33D	440	23	1.27	440	23	1.93	440	23	12.83	440	23	0.72	440	23	1.52

Table B.6: Solutions reported by CPLEX for Class 6

Instance	OPMVC-DL			OPMVC-GG			OPMVC-W			OPMVC-DFJ			OPMVC-FT		
	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**
att48A	11*	25	>3600	11	25	2.14	11	25	231.31	11	25	4.92	11	25	8.16
att48B	18	32	>3600	19	33	25.27	19	33	695.52	19	33	16.26	19	33	400.66
att48C	12	26	1967.09	12	26	2.62	12	26	371.9	12	26	2.43	12	26	8.67
att48D	-	-	>3600	14	28	2.69	14	28	393.68	14	28	1.2	14	28	11.84
att48E	16	30	1013.86	16	30	2.39	16	30	272.31	16	30	2.07	16	30	18.57
cmt121A	-	-	>3600	-	-	>3600	-	-	>3600	507	-	>3600	518	-	>3600
cmt121B	-	-	>3600	-	-	>3600	-	-	>3600	465	-	>3600	470	-	>3600
cmt121C	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	575	-	>3600
cmt121D	-	-	>3600	-	-	>3600	-	-	>3600	618	88	331.56	623	-	>3600
cmt151A	847	98	>3600	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600
cmt151B	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	554	-	>3600
cmt151C	-	-	>3600	740	89	2341.16	-	-	>3600	-	-	>3600	759	-	>3600

Continues on next page

Continued from previous page

Instance	**OPMVC-DL**			**OPMVC-GG**			**OPMVC-W**			**OPMVC-DFJ**			**OPMVC-FT**		
	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**
cmt151D	-	-	>3600	852	98	551.02	-	-	>3600	857	-	>3600	867	-	>3600
cmt200A	-	-	>3600	-	-	>3600	-	-	>3600	1445	-	>3600	-	-	>3600
cmt200B	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	1060	-	>3600
cmt200C	-	-	>3600	1224*	122	>3600	-	-	>3600	1224	122	286.84	1294	-	>3600
cmt200D	-	-	>3600	1363	131	>3600	-	-	>3600	-	-	>3600	-	-	>3600
eil30A	-	-	>3600	1550	13	15.16	1550	13	19.85	1550	13	17.31	1550	13	93.16
eil30B	3950	14	>3600	4775	17	12.85	4775	17	24.72	4775	17	4.49	4775	17	31.05
eil30C	5550*	22	>3600	5550	22	1.32	5550	22	17.11	5550	22	0.72	5550	22	4.28
eil30D	5725	23	>3600	5850	24	7.25	5850	24	22.72	5850	24	0.44	5850	24	1.69
eil33A	-	-	>3600	10600	18	7.09	10600	18	44.53	10600	18	1038.28	10600	18	2.25
eil33B	13080	24	>3600	13330	25	6.55	13330	25	23.48	13330	25	3.57	13330	25	4.06
eil33C	14030	26	3328.02	14030	26	5.61	14030	26	31.22	14030	26	2.63	14030	26	2.83
eil51A	329	34	74.31	329	34	3.09	329	34	668.3	329	34	14.04	329	34	3.99
eil51B	-	-	>3600	212	25	6.44	212	25	191.39	-	-	>3600	212	25	6.03
eil51C	289	31	84.92	289	31	1.79	289	31	181.01	289	31	43.94	289	31	4.63
eil51D	344	35	74.09	344	35	12.56	344	35	284.19	344	35	13.15	344	35	2.88
eil76A	-	-	>3600	233	33	22.25	-	-	>3600	234	-	>3600	233	33	110.88
eil76B	613*	53	>3600	613	53	161.24	-	-	>3600	613	53	1386.07	615	-	>3600
eil76C	211*	33	>3600	211	33	19.78	-	-	>3600	211	33	1186.81	211	33	100.35
eil76D	289	36	>3600	305*	37	>3600	-	-	>3600	312	-	>3600	305	37	411.36
eil76E	383	40	>3600	386	40	247.35	-	-	>3600	392	-	>3600	386*	-	>3600
eil101A	-	-	>3600	633	71	211.25	-	-	>3600	633	71	307.33	633	71	1904.95
eil101B	-	-	>3600	344	51	363.78	-	-	>3600	-	-	>3600	344	52	572.3
eil101C	-	-	>3600	474	59	257.82	-	-	>3600	474	59	1325.3	474	59	396.97
eil101D	-	-	>3600	-	-	>3600	-	-	>3600	555	62	36.6	555	62	390.09
gil262A	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600
gil262B	-	-	>3600	-	-	>3600	-	-	>3600	3254	-	>3600	-	-	>3600
gil262C	-	-	>3600	-	-	>3600	-	-	>3600	4174	-	>3600	-	-	>3600
gil262D	-	-	>3600	-	-	>3600	-	-	>3600	4718	-	>3600	-	-	>3600
op21A	155	14	0.83	155	14	1.34	155	14	4.01	155	13	0.71	155	14	12.52
op21B	75	9	0.4	75	9	3.38	75	9	1.03	75	9	0.61	75	9	2.99
op21C	110	11	0.72	110	11	1.14	110	11	1.65	110	11	0.53	110	11	3.58
op21D	160	13	4.7	160	13	2.21	160	13	3.63	160	13	0.69	160	13	24.94
op32A	115	21	8.46	115	21	0.6	115	21	4.87	115	21	1.7	115	21	0.98
op32B	65	16	445.84	65	16	2.07	65	16	5.33	65	16	1.87	65	16	1.23
op32C	105	20	1.85	105	20	0.95	105	20	4.92	105	20	2.27	105	20	4.64
op32D	130	22	7.77	130	22	2.28	130	23	5.87	130	22	0.43	130	23	0.82
op33A	350	21	7.21	350	21	3.84	350	21	8.26	350	21	0.28	350	21	0.32
op33B	130	13	2.16	130	13	3.17	130	13	6.85	130	13	1.71	130	13	4.92
op33C	240	17	4.45	240	17	6.56	240	18	6.57	240	17	1.07	240	17	3.4
op33D	300	20	0.91	300	20	2.4	300	20	3.82	300	20	0.37	300	20	0.63

Table B.7: Solutions reported by CPLEX for Class 7

Instance	OPMVC-DL			OPMVC-GG			OPMVC-W			OPMVC-DFJ			OPMVC-FT		
	z	$\sum y$	Time	z	$\sum y$	Time	z	$\sum y$	Time	z	$\sum y$	Time	z	$\sum y$	Time
att48A	23*	28	>3600	-	-	>3600	23	28	1456.2	-	-	>3600	-	-	>3600
att48B	35	40	2084.03	-	-	>3600	35	40	2953.42	-	-	>3600	-	-	>3600
att48C	13	18	>3600	-	-	>3600	14	19	813.72	-	-	>3600	-	-	>3600
att48D	18*	23	>3600	18	23	65.93	18	23	1973.23	-	-	>3600	18	23	112.86
att48E	21	26	>3600	22	27	12.29	22	27	1204.17	22	27	21.67	22	27	199.78
cmt121A	845	62	>3600	-	-	>3600	-	-	>3600	958	-	>3600	1073	-	>3600
cmt121B	-	-	>3600	-	-	>3600	-	-	>3600	643	-	>3600	937	-	>3600
cmt121C	-	-	>3600	-	-	>3600	-	-	>3600	778	-	>3600	935	-	>3600
cmt121D	-	-	>3600	838	82	>3600	-	-	>3600	900	-	>3600	-	-	>3600
cmt151A	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	1336	-	>3600
cmt151B	1657	116	>3600	1742	119	3587.34	-	-	>3600	-	-	>3600	1896	-	>3600
cmt151C	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	664	-	>3600
cmt151D	-	-	>3600	-	-	>3600	-	-	>3600	739	-	>3600	-	-	>3600
cmt151E	-	-	>3600	-	-	>3600	-	-	>3600	878	-	>3600	979	-	>3600
cmt200A	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	1378	-	>3600
cmt200B	1877	116	>3600	-	-	>3600	-	-	>3600	2472	-	>3600	-	-	>3600
cmt200C	-	-	>3600	-	-	>3600	-	-	>3600	1093	-	>3600	1459	-	>3600
cmt200D	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	1649	-	>3600
cmt200E	-	-	>3600	-	-	>3600	-	-	>3600	1462	-	>3600	1798	-	>3600
eil30A	10700*	21	>3600	10700	21	15.65	10700	21	57.26	10700	21	21.41	10700	21	2778.12
eil30B	5750*	14	>3600	5750	14	152.62	5750	14	341.87	5750	14	326.16	6250	-	>3600
eil30C	7650*	15	>3600	7650	15	431.85	7650	12	324.94	7650	12	120.37	8025	-	>3600
eil30D	8875	13	>3600	9575	15	25.43	9575	15	39.54	9575	15	32.51	9575	15	1218.99
eil33A	6290	12	>3600	6990	13	319.59	6990	13	312.72	17380	-	>3600	6990	13	158.52
eil33B	22680*	27	>3600	22680	27	6.18	22680	27	28.45	23220	-	>3600	22680	27	434.57
eil33C	10630*	18	>3600	10630	18	44.96	10630	18	231.54	18540	-	>3600	10630	18	52.06
eil33D	15730*	19	>3600	15730	19	11.76	15730	19	507.58	20180	-	>3600	15730	19	68.11
eil33E	19230*	22	>3600	19230	22	5.83	19230	22	151.36	21490	-	>3600	19230	22	107.33
eil51A	373	25	791.73	373	25	60.22	331	23	>3600	373*	-	>3600	380	-	>3600
eil51B	586	38	1151.38	586	38	17.41	585	38	>3600	-	-	>3600	613	-	>3600
eil51C	177	15	677.22	177	15	42.25	177	15	2729.5	177	15	33.05	177	15	26.61
eil51D	223	17	361.28	223	17	32.68	223*	17	>3600	223	16	74.59	223	17	24.48
eil51E	265	19	264.15	265	19	10.93	265	19	761.43	265	19	61.28	265	19	79.69
eil76A	693	43	741.49	693	43	107.01	-	-	>3600	693	43	18.28	748	-	>3600
eil76B	1065	62	1011.13	1065	62	12.84	-	-	>3600	1065	62	1643.09	1116	-	>3600
eil76C	-	-	>3600	302	23	155.62	-	-	>3600	306	-	>3600	302	24	1309.16
eil76D	406	28	1568.1	406	28	93.28	-	-	>3600	406	28	2127.69	406	28	1423.94
eil76E	474*	31	>3600	474	31	45.1	-	-	>3600	-	-	>3600	484	-	>3600
eil101A	822	54	>3600	834	55	894.32	-	-	>3600	840	-	>3600	868	-	>3600

Continues on next page

Continued from previous page

Instance	OPMVC-DL			OPMVC-GG			OPMVC-W			OPMVC-DFJ			OPMVC-FT		
	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**
eil101B	1177	77	>3600	1178	78	395.36	-	-	>3600	1180	-	>3600	1203	-	>3600
eil101C	-	-	>3600	353	29	315.75	-	-	>3600	368	-	>3600	353	29	1088.64
eil101D	-	-	>3600	450	35	1889.51	-	-	>3600	459	-	>3600	454	-	>3600
eil101E	-	-	>3600	-	-	>3600	-	-	>3600	530	-	>3600	539	-	>3600
gil262A	-	-	>3600	-	-	>3600	-	-	>3600	6708	-	>3600	8303	-	>3600
gil262B	6999	143	>3600	-	-	>3600	-	-	>3600	9905	-	>3600	-	-	>3600
gil262C	-	-	>3600	-	-	>3600	-	-	>3600	3801	-	>3600	5559	-	>3600
gil262D	-	-	>3600	-	-	>3600	-	-	>3600	4668	-	>3600	-	-	>3600
gil262E	-	-	>3600	-	-	>3600	-	-	>3600	5333	-	>3600	7011	-	>3600
op21A	270	15	5.7	270	15	5.61	270	15	4.33	270	15	1.3	270	15	26.83
op21B	180	9	6.46	180	9	11.66	180	9	1.32	180	9	2.57	180	9	2.7
op21C	220	12	6.74	220	12	4.74	220	12	2.39	220	12	0.97	220	12	5.54
op21D	260	14	1.23	260	14	3.91	260	14	0.81	260	14	0.59	260	14	5.33
op32A	125	17	12.14	125	16	1.15	125	17	12.15	125	17	1.98	125	17	2.7
op32B	195	23	1.87	195	23	7.33	195	23	40.23	195	21	2.99	195	23	832.95
op32C	55	9	2896.96	55	9	17.82	55	9	9.49	55	9	26.05	55	9	2.05
op32D	95	13	6.16	95	13	1.05	95	13	6.79	95	13	3.74	95	13	0.61
op32E	110	15	24.46	110	15	1.36	110	15	7.94	110	15	2.03	110	15	0.85
op33A	400	19	6.43	400	19	2.8	400	19	30.19	400	19	0.61	400	19	9.38
op33B	550	27	12.6	550	27	6.82	550	25	35.28	550	27	2.75	550	-	>3600
op33C	150	10	41.28	150	10	10.96	150	10	7.02	150	10	4.41	150	10	3.22
op33D	210	210	34.36	210	12	8.39	210	12	11.17	210	12	2.71	210	12	2.43
op33E	280	14	21.41	280	14	5.95	280	14	36.94	280	14	0.95	280	14	1.53

Table B.8: Solutions reported by CPLEX for Class 8

Instance	OPMVC-DL			OPMVC-GG			OPMVC-W			OPMVC-DFJ			OPMVC-FT		
	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**
att48A	26	36	>3600	28	38	85.25	28	38	1133.26	28	38	42.61	28*	-	>3600
att48B	-	-	>3600	-	-	>3600	22	32	1610.69	22	32	560.23	22	32	293.31
att48C	26	36	>3600	27	37	46.72	27	37	1490.71	27	37	8.84	27	37	177.09
att48D	-	-	>3600	30	40	35.11	30	40	628.56	30	40	37.55	30	40	946.52
cmt121A	-	-	>3600	-	-	>3600	-	-	>3600	716	-	>3600	-	-	>3600
cmt121B	-	-	>3600	-	-	>3600	-	-	>3600	530	-	>3600	785	-	>3600
cmt121C	-	-	>3600	-	-	>3600	-	-	>3600	711	-	>3600	889	-	>3600
cmt121D	-	-	>3600	-	-	>3600	-	-	>3600	863	-	>3600	922	-	>3600
cmt151A	-	-	>3600	-	-	>3600	-	-	>3600	818	-	>3600	903	-	>3600
cmt151B	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	1672	-	>3600
cmt151C	-	-	>3600	-	-	>3600	-	-	>3600	731	-	>3600	-	-	>3600

Continues on next page

Continued from previous page

Instance	OPMVC-DL			OPMVC-GG			OPMVC-W			OPMVC-DFJ			OPMVC-FT		
	z	$\sum y$	Time	z	$\sum y$	Time	z	$\sum y$	Time	z	$\sum y$	Time	z	$\sum y$	Time
cmt151D	-	-	>3600	887	76	>3600	-	-	>3600	927	-	>3600	1028	-	>3600
cmt151E	-	-	>3600	-	-	>3600	-	-	>3600	1080	-	>3600	1210	-	>3600
cmt200A	-	-	>3600	-	-	>3600	-	-	>3600	2050	-	>3600	-	-	>3600
cmt200B	-	-	>3600	-	-	>3600	-	-	>3600	999	-	>3600	1215	-	>3600
cmt200C	-	-	>3600	-	-	>3600	-	-	>3600	1252	-	>3600	1457	-	>3600
cmt200D	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600
eil30A	5575	15	>3600	6225	17	22.42	6225	17	507.39	6225	17	29.62	6225	17	650.48
eil30B	>3600*	14	>3600	>3600	14	8.52	>3600	14	259.48	>3600	14	16.38	>3600	14	451.54
eil30C	5000*	14	>3600	5000	15	8.66	5000	14	145.5	5000	15	17.36	5000	14	611.02
eil30D	5375*	17	>3600	5375	17	16.63	5375	17	624.37	5375	17	17.14	5375	17	2040.75
eil33A	19370*	25	>3600	19370	25	9.73	19370	25	559.6	21180	-	>3600	19370	25	31.22
eil33B	16830	19	>3600	17000	19	50.76	17000	19	318.51	20530	-	>3600	17000	19	261.82
eil33C	20930*	27	>3600	20930	27	6.63	20930	27	57.52	21300	-	>3600	20930	27	21.2
eil33D	22280*	30	>3600	22280	30	6.52	22280	30	71.8	22470	-	>3600	22280	30	24.71
eil51A	507	38	688.69	507	38	41.1	505	39	>3600	-	-	>3600	-	-	>3600
eil51B	318	25	2538.98	318	25	89.6	318	25	1620.44	318	25	8.77	318	25	18.82
eil51C	391	30	705.4	391	30	18.74	391	30	1895.1	391	30	12.81	391	30	51.21
eil51D	448*	33	>3600	448	33	13.77	448	33	3544.44	448	33	32.21	448	33	173.12
eil76A	572	42	192.14	572	42	14	-	-	>3600	572	42	194	572	42	1191.14
eil76B	901	62	1820.05	901	62	55.08	-	-	>3600	901*	-	>3600	-	-	>3600
eil76C	263*	27	>3600	263	27	103.62	-	-	>3600	263	27	37.55	263	27	219.95
eil76D	393*	33	>3600	393	33	115.66	-	-	>3600	393	33	37.92	393	33	260.68
eil76E	488	38	600.25	488	38	144.21	-	-	>3600	488	38	22.76	488	38	184.24
eil101A	-	-	>3600	655	57	552.65	-	-	>3600	661	-	>3600	660	-	>3600
eil101B	1050*	80	>3600	1050	80	215.68	-	-	>3600	1052	-	>3600	-	-	>3600
eil101C	-	-	>3600	361	42	487.13	-	-	>3600	361	42	99.49	361	42	462.92
eil101D	-	-	>3600	496	49	117.11	-	-	>3600	508	-	>3600	496	49	653.82
eil101E	607*	53	>3600	607	53	555.63	-	-	>3600	615	-	>3600	607	54	1328.01
gil262A	-	-	>3600	-	-	>3600	-	-	>3600	8530	-	>3600	-	-	>3600
gil262B	-	-	>3600	-	-	>3600	-	-	>3600	5115	-	>3600	6724	-	>3600
gil262C	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	7597	-	>3600
gil262D	-	-	>3600	-	-	>3600	-	-	>3600	7276	-	>3600	-	-	>3600
op21A	140	11	0.3	140	11	0.32	140	11	0.8	140	11	0.15	140	11	0.78
op21B	245	14	4.49	245	14	5.2	245	14	6.99	245	14	0.49	245	13	86.09
op21C	140	11	0.25	140	11	0.47	140	11	0.81	140	11	0.14	140	11	0.74
op21D	160	13	0.19	160	13	0.63	-	-	>3600	160	13	0.21	160	13	2
op21E	165	13	0.75	165	13	0.71	165	13	8.27	165	13	0.61	165	13	5.91
op32A	165	23	9.52	165	23	0.86	165	23	11.42	165	22	4.99	165	23	1.29
op32B	125	18	15.96	125	18	4.77	125	18	10.47	125	18	2.14	125	18	2.68
op32C	155	21	0.63	155	21	0.48	155	21	8.1	155	21	0.87	155	21	0.82
op33A	440	21	24.87	440	21	3.32	440	21	18.06	440	21	3.78	440	21	1.46

Continues on next page

Continued from previous page

Instance	OPMVC-DL			OPMVC-GG			OPMVC-W			OPMVC-DFJ			OPMVC-FT		
	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**
op33B	280*	14	>3600	280	14	15.69	280	14	36.71	280	14	323.18	280	14	7.96
op33C	470	22	16.34	470	22	1.31	470	22	11.16	470	22	0.87	470	22	1
op33D	530	25	5.82	530	24	1.75	530	24	49.85	530	25	1.61	530	25	1.48

Table B.9: Solutions reported by CPLEX for Class 9

Instance	OPMVC-DL			OPMVC-GG			OPMVC-W			OPMVC-DFJ			OPMVC-FT		
	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**
att48A	-	-	>3600	14	29	3.62	14	29	758.09	14	29	5.9	14	29	24.63
att48B	24	39	>3600	25	40	59.33	25	40	2597.94	25	40	11.7	-	-	>3600
att48C	-	-	>3600	15	30	6.5	15	30	569.01	15	30	3.07	15	30	22.12
att48D	17	32	1023.07	17	32	4.44	17	32	1240.8	17	32	2.36	17	32	560.96
att48E	19*	34	>3600	19	34	16.39	19	34	2508.19	19	34	26.74	20	-	>3600
cmt121A	-	-	>3600	-	-	>3600	-	-	>3600	679	-	>3600	748	-	>3600
cmt121B	-	-	>3600	-	-	>3600	-	-	>3600	612	-	>3600	674	-	>3600
cmt121C	-	-	>3600	-	-	>3600	-	-	>3600	757	-	>3600	-	-	>3600
cmt121D	-	-	>3600	-	-	>3600	-	-	>3600	853	-	>3600	-	-	>3600
cmt151A	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600
cmt151B	-	-	>3600	-	-	>3600	-	-	>3600	705	-	>3600	-	-	>3600
cmt151C	-	-	>3600	-	-	>3600	-	-	>3600	925	-	>3600	968	-	>3600
cmt151D	-	-	>3600	-	-	>3600	-	-	>3600	1095	-	>3600	1159	-	>3600
cmt200A	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	1925	-	>3600
cmt200B	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600
cmt200C	-	-	>3600	-	-	>3600	-	-	>3600	1441	-	>3600	-	-	>3600
cmt200D	-	-	>3600	-	-	>3600	-	-	>3600	1692	-	>3600	-	-	>3600
eil30A	-	-	>3600	1800	15	274.38	1800	15	100.73	1800	15	49.29	1800	15	404.08
eil30B	5525*	24	>3600	5525	24	30.08	5525	24	154.53	5525	24	16.7	5525	24	189.16
eil30C	7050	25	>3600	7200	26	4.44	7200	26	99.38	7200	26	2.12	7200	26	10.06
eil30D	7600*	29	>3600	7600	29	7.25	7600	29	48.23	7600	29	0.69	7600	29	7.9
eil33A	-	-	>3600	16700	22	35.38	16700	22	290.49	20130	-	>3600	16700	22	97.32
eil33B	-	-	>3600	21180	30	7.39	21180	30	204.11	21180	30	25.66	21180	30	6.74
eil33C	22380*	32	>3600	22380	32	17.11	22380	32	361.34	22380	32	8.56	22380	32	34.03
eil51A	422	37	1047.57	422	37	27.05	422	37	3567.72	422	37	2306.27	422	37	262.04
eil51B	-	-	>3600	289	30	69.29	289	30	2791.64	296	-	>3600	289	30	13.37
eil51C	-	-	>3600	372	33	69.67	372	33	>3600	375	-	>3600	372	33	31.91
eil51D	441	38	697.35	441	38	44.2	441	38	>3600	441	38	2532.5	441	38	469.98
eil76A	288*	39	>3600	288	39	43.99	288	-	>3600	297	-	>3600	288	39	104.09
eil76B	720	58	>3600	723	59	23.01	723	-	>3600	723	59	2674.35	746	-	>3600
eil76C	260*	35	>3600	260	36	28.74	260	-	>3600	272	-	>3600	260	36	63.97

Continues on next page

Continued from previous page

Instance	OPMVC-DL			OPMVC-GG			OPMVC-W			OPMVC-DFJ			OPMVC-FT		
	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**	z	$\sum y$	**Time**
eil76D	383*	42	>3600	383	42	94.09	383	-	>3600	383	42	250.21	383	42	63.95
eil76E	476	46	2257.51	476	46	119.66	476	-	>3600	476	46	387.52	476	46	72.23
eil101A	-	-	>3600	785	78	715.33	785	-	>3600	785*	-	>3600	-	-	>3600
eil101B	-	-	>3600	-	-	>3600	-	-	>3600	415	-	>3600	399	-	>3600
eil101C	-	-	>3600	534	61	959.81	534	-	>3600	540	-	>3600	548	-	>3600
eil101D	-	-	>3600	652	67	942.75	652	-	>3600	652	67	599.49	-	-	>3600
gil262A	-	-	>3600	-	-	>3600	-	-	>3600	6965	-	>3600	7319	-	>3600
gil262B	-	-	>3600	-	-	>3600	-	-	>3600	-	-	>3600	5649	-	>3600
gil262C	-	-	>3600	-	-	>3600	-	-	>3600	5791	-	>3600	-	-	>3600
gil262D	-	-	>3600	-	-	>3600	-	-	>3600	6787	-	>3600	-	-	>3600
op21A	165	15	1.7	165	15	3.27	165	15	4.79	165	15	0.69	165	15	14.25
op21B	85	10	0.61	85	10	3.89	85	10	1.21	85	10	0.62	85	10	2.19
op21C	115	12	0.57	115	12	6.97	115	12	2.27	115	12	0.96	115	12	4.72
op21D	165	15	1.78	165	15	5.48	165	15	9.66	165	15	1.54	165	14	18.52
op32A	125	21	13.34	125	21	2.26	125	21	9.19	125	22	9.12	125	21	2.55
op32B	80	17	73.96	80	17	3.77	80	17	8.86	80	17	2.2	80	17	1.26
op32C	120	21	10.35	120	21	0.98	120	21	7.16	120	21	2.48	120	21	2.8
op32D	145	24	19.96	145	24	0.48	145	24	9.75	145	24	0.79	145	24	1.7
op33A	430	25	9.3	430	25	7.46	430	25	62.7	430	25	0.8	430	25	6.97
op33B	160	14	21.92	160	14	6.32	160	14	30.94	160	14	3.09	160	14	11.54
op33C	280	19	30.06	280	19	7.76	280	19	12.91	280	19	2.11	280	19	7.7
op33D	370	22	22.95	370	22	6.89	370	22	8.55	370	22	1.16	370	22	3.03

Bibliography

Abedzadeh, M., Mazinani, M., Moradinasab, N., and Roghanian, E. (2013). Parallel variable neighborhood search for solving fuzzy multi-objective dynamic facility layout problem. *The International Journal of Advanced Manufacturing Technology*, 65(1):197–211.

Adibi, M., Zandieh, M., and Amiri, M. (2010). Multi-objective scheduling of dynamic job shop using variable neighborhood search. *Expert Systems with Applications*, 37(1):282 – 287.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2014). Optimization-based adaptive large neighborhood search for the production routing problem. *Transportation Science*, 48(1):20–45.

Agarwal, A. and Gupta, S. K. (2008). Multiobjective optimal design of heat exchanger networks using new adaptations of the elitist nondominated sorting genetic algorithm, nsga-ii. *Industrial & Engineering Chemistry Research*, 47(10):3489–3501.

Alcalá, R., Ducange, P., Herrera, F., Lazzerini, B., and Marcelloni, F. (2009). A multiobjective evolutionary approach to concurrently learn rule and data bases of linguistic fuzzy-rule-based systems. *IEEE Transactions on Fuzzy Systems*, 17(5):1106–1122.

Almeida, C. P., Gonçalves, R. A., Goldbarg, E. F., Goldbarg, M. C., and Delgado,

M. R. (2012). An experimental analysis of evolutionary heuristics for the biobjective traveling purchaser problem. *Annals of Operations Research*, 199(1):305–341.

Angel-Bello, F., Alvarez, A., and García, I. (2013). Two improved formulations for the minimum latency problem. *Applied Mathematical Modelling*, 37(4):2257 – 2266.

Angelelli, E., Gendreau, M., Mansini, R., and Vindigni, M. (2017). The traveling purchaser problem with time-dependent quantities. *Computers & Operations Research*, 82:15 – 26.

Angelelli, E., Mansini, R., and Vindigni, M. (2008). Exploring greedy criteria for the dynamic traveling purchaser problem. *Central European Journal of Operations Research*, 17(2):141–158.

Angelelli, E., Mansini, R., and Vindigni, M. (2011). Look-ahead heuristics for the dynamic traveling purchaser problem. *Computers & Operations Research*, 38(12):1867 – 1876.

Angelelli, E., Mansini, R., and Vindigni, M. (2016). The stochastic and dynamic traveling purchaser problem. *Transportation Science*, 50(2):642–658.

Archetti, C., Bianchessi, N., and Speranza, M. G. (2013a). The capacitated team orienteering problem with incomplete service. *Optimization Letters*, 7(7):1405–1417.

Archetti, C., Bianchessi, N., and Speranza, M. G. (2013b). Optimal solutions for routing problems with profits. *Discrete Applied Mathematics*, 161(4–5):547–557.

Archetti, C., Bianchessi, N., Speranza, M. G., and Hertz, A. (2014a). Incomplete service and split deliveries in a routing problem with profits. *Networks*, 63(2):135–145.

Archetti, C., Bianchessi, N., Speranza, M. G., and Hertz, A. (2014b). The split delivery capacitated team orienteering problem. *Networks*, 63(1):16–33.

Archetti, C., Feillet, D., Hertz, A., and Speranza, M. G. (2009). The capacitated team orienteering and profitable tour problems. *Journal of the Operational Research Society*, 60:831–842.

Archetti, C., Speranza, M. G., and Vigo, D. (2014c). Vehicle routing problems with profits. In Vigo, D. and Toth, P., editors, *Vehicle Routing: problems, methods, and applications*, pages 273–297. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Azi, N., Gendreau, M., and Potvin, J.-Y. (2014). An adaptive large neighborhood search for a vehicle routing problem with multiple routes. *Computers & Operations Research*, 41:167 – 173.

Balas, E. (1989). The prize collecting traveling salesman problem. *Networks*, 19(6):621–636.

Baraldi, P., Pedroni, N., and Zio, E. (2009). Application of a niched pareto genetic algorithm for selecting features for nuclear transients classification. *International Journal of Intelligent Systems*, 24(2):118–151.

Basu, M. (2008). Dynamic economic emission dispatch using nondominated sorting genetic algorithm-ii. *International Journal of Electrical Power & Energy Systems*, 30(2):140 – 149.

Basu, M. (2011). Economic environmental dispatch of fixed head hydrothermal power systems using nondominated sorting genetic algorithm-ii. *Applied Soft Computing*, 11(3):3046 – 3055.

Basu, M. (2013). Combined heat and power economic emission dispatch using nondominated sorting genetic algorithm-ii. *International Journal of Electrical Power & Energy Systems*, 53:135 – 141.

Batista-Galván, M., Riera-Ledesma, J., and Salazar-González, J. J. (2013). The traveling purchaser problem, with multiple stacks and deliveries: A branch-and-cut approach. *Computers & Operations Research*, 40(8):2103 – 2115.

Bensmaine, A., Dahane, M., and Benyoucef, L. (2013). A non-dominated sorting genetic algorithm based approach for optimal machines selection in reconfigurable manufacturing environment. *Computers & Industrial Engineering*, 66(3):519 – 524. Special Issue: The International Conferences on Computers and Industrial Engineering (ICC&IEs) - series 41.

Beraldi, P., Bruni, M. E., Manerba, D., and Mansini, R. (2015). A stochastic programming approach for the traveling purchaser problem. *IMA Journal of Management Mathematics*, 28(1):41–63.

Bernardino, R. and Paias, A. (2016). Metaheuristics based on decision hierarchies for the traveling purchaser problem. *International Transactions in Operational Research*.

Bianchessi, N., Mansini, R., and Speranza, M. (2014). The distance constrained multiple vehicle traveling purchaser problem. *European Journal of Operational Research*, 235(1):73 – 87.

Boctor, F. F., Laporte, G., and Renaud, J. (2003). Heuristics for the traveling purchaser problem. *Computers & Operations Research*, 30(4):491 – 504.

Bontoux, B. and Feillet, D. (2008). Ant colony optimization for the traveling purchaser problem. *Computers & Operations Research*, 35(2):628 – 637.

Bruglieri, M., Pezzella, F., Pisacane, O., and Suraci, S. (2015). A variable neighborhood search branching for the electric vehicle routing problem with time windows. *Electronic Notes in Discrete Mathematics*, 47:221 – 228. The 3rd International Conference on Variable Neighborhood Search (VNS'14).

Burstall, R. M. (1966). A heuristic method for a job-scheduling problem. *OR*, 17(3):291–304.

Butt, S. E. and Cavalier, T. M. (1994). A heuristic for the multiple tour maximum collection problem. *Computers & Operations Research*, 21(1):101–111.

Cambazard, H. and Penz, B. (2012). *Principles and Practice of Constraint Programming: 18th International Conference, CP 2012, Québec City, QC, Canada, October 8-12, 2012. Proceedings*, chapter A Constraint Programming Approach for the Traveling Purchaser Problem, pages 735–749. Springer Berlin Heidelberg, Berlin, Heidelberg.

Cao, K., Batty, M., Huang, B., Liu, Y., Yu, L., and Chen, J. (2011). Spatial multi-objective land use optimization: extensions to the non-dominated sorting genetic algorithm-ii. *International Journal of Geographical Information Science*, 25(12):1949–1969.

Carlucci, S., Cattarin, G., Causone, F., and Pagliano, L. (2015). Multi-objective optimization of a nearly zero-energy building based on thermal and visual discomfort minimization using a non-dominated sorting genetic algorithm (nsga-ii). *Energy and Buildings*, 104:378 – 394.

Chao, I., Golden, B. L., and Wasil, E. A. (1996). The team orienteering problem. *European Journal of Operational Research*, 88:464–474.

Chen, Z., Yuan, X., Ji, B., Wang, P., and Tian, H. (2014). Design of a fractional order pid controller for hydraulic turbine regulating system using chaotic non-dominated sorting genetic algorithm ii. *Energy Conversion and Management*, 84:390 – 404.

Chitra, C. and Subbaraj, P. (2012). A nondominated sorting genetic algorithm solution for shortest path routing problem in computer networks. *Expert Systems with Applications*, 39(1):1518 – 1525.

Choi, M. J. and Lee, S. H. (2011). The multiple traveling purchaser problem for maximizing system's reliability with budget constraints. *Expert Systems with Applications*, 38(8):9848 – 9853.

Claus, A. (1984). A new formulation for the travelling salesman problem. *SIAM Journal on Algebraic Discrete Methods*, 5:21–25.

Croes, G. A. (1958). A method for solving traveling-salesman problems. *Operations Research*, 6(6):791–812.

Current, J. R. (1982). *Multiobjective Design of Transportation Networks*. PhD thesis, The Johns Hopkins University.

Dantzig, G., Fulkerson, R., and Johnson, S. (1954). Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2:393–410.

de Assumpção Drummond, L. M., Vianna, L. S., da Silva, M. B., and Ochi, L. S. (2002). Distributed parallel metaheuristics based on GRASP and VNS for solving the traveling purchaser problem. In *Parallel and Distributed Systems, 2002. Proceedings. Ninth International Conference on*, pages 257–263. IEEE.

Dell'Amico, M., Maffioli, F., and Värbrand, P. (1995). On prize-collecting tours and the asymmetric travelling salesman problem. *International Transactions in Operational Research*, 2(3):297 – 308.

Demir, E., Bektaş, T., and Laporte, G. (2012). An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, 223(2):346 – 359.

Desrochers, M. and Laporte, G. (1991). Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters*, 10(1):27 – 36.

dos Santos Coelho, L. and Alotto, P. (2008). Multiobjective electromagnetic optimization based on a nondominated sorting genetic approach with a chaotic crossover operator. *IEEE Transactions on Magnetics*, 44(6):1078–1081.

Dridi, L., Parizeau, M., Mailhot, A., and Villeneuve, J.-P. (2008). Using evolutionary optimization techniques for scheduling water pipe renewal considering a short planning horizon. *Computer-Aided Civil and Infrastructure Engineering*, 23(8):625–635.

Dufo-López, R., Bernal-Agustín, J. L., Yusta-Loyo, J. M., Domínguez-Navarro, J. A., Ramírez-Rosado, I. J., Lujano, J., and Aso, I. (2011). Multi-objective optimization minimizing cost and life cycle emissions of stand-alone pv–wind–diesel systems with batteries storage. *Applied Energy*, 88(11):4033 – 4041.

Emeç, U., Çatay, B., and Bozkaya, B. (2016). An adaptive large neighborhood search for an e-grocery delivery routing problem. *Computers & Operations Research*, 69:109 – 125.

Erdoğan, G., Cordeau, J.-F., and Laporte, G. (2010). The attractive traveling salesman problem. *European Journal of Operational Research*, 203(1):59 – 69.

Erdoğan, G. and Laporte, G. (2013). The orienteering problem with variable profits. *Networks*, 61(2):104–116.

Eskandarpour, M., Nikbakhsh, E., and Zegordi, S. H. (2014). Variable neighborhood search for the bi-objective post-sales network design problem: A fitness landscape analysis approach. *Computers & Operations Research*, 52:300 – 314.

Eskandarpour, M., Zegordi, S. H., and Nikbakhsh, E. (2013). A parallel variable neighborhood search for the multi-objective sustainable post-sales network design problem. *International Journal of Production Economics*, 145(1):117 – 131.

Falcon, R., Li, X., Nayak, A., and Stojmenovic, I. (2010). The one-commodity traveling salesman problem with selective pickup and delivery: An ant colony approach. In *IEEE Congress on Evolutionary Computation*, pages 1–8.

Feillet, D., Dejax, P., and Gendreau, M. (2005). Traveling salesman problems with profits. *Transportation science*, 39:188–205.

Fischetti, M., Salazar-Gonzalez, J. J., and Toth, P. (1998). Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing*, 10:133–148.

Fischetti, M. and Toth, P. (1988). An additive approach for the optimal solution of the prize collecting traveling salesman problem. *Vehicle Routing: Methods and Studies*, pages 319–343.

Fleszar, K., Osman, I. H., and Hindi, K. S. (2009). A variable neighbourhood search algorithm for the open vehicle routing problem. *European Journal of Operational Research*, 195(3):803 – 809.

Gao, J., Sun, L., and Gen, M. (2008). A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. *Computers & Operations Research*, 35(9):2892 – 2907.

García, A., Vansteenwegen, P., Souffriau, W., Arbelaitz, O., and Linaza, M. (2010). Solving multi constrained team orienteering problems to generate tourist routes. Technical report, Centre for Industrial Management/Traffic & Infrastructure, Katholieke Universiteit Leuven Leuven, Belgium.

Gavish, B. and Graves, S. C. (1978). The travelling salesman problem and related problems. Working Paper GR-078-78.

Gendreau, M., Laporte, G., and Semet, F. (1997). The covering tour problem. *Operations Research*, 45(4):568–576.

Gendreau, M., Manerba, D., and Mansini, R. (2016). The multi-vehicle traveling purchaser problem with pairwise incompatibility constraints and unitary demands: A branch-and-price approach. *European Journal of Operational Research*, 248(1):59 – 71.

Ghoddousi, P., Eshtehardian, E., Jooybanpour, S., and Javanmardi, A. (2013). Multi-mode resource-constrained discrete time–cost-resource optimization in project scheduling using non-dominated sorting genetic algorithm. *Automation in Construction*, 30:216 – 227.

Goldbarg, M., Bagi, L., and Goldbarg, E. (2009). Transgenetic algorithm for the traveling purchaser problem. *European Journal of Operational Research*, 199(1):36 – 45.

Golden, B., Levy, L., and Dahl, R. (1981). Two generalizations of the traveling salesman problem. *Omega*, 9(4):439–441.

Gouveia, L., Paias, A., and Voß, S. (2011). Models for a traveling purchaser problem with additional side-constraints. *Computers & Operations Research*, 38(2):550 – 558.

Gribkovskaia, I., Laporte, G., and Shyshou, A. (2008). The single vehicle routing problem with deliveries and selective pickups. *Computers & Operations Research*, 35(9):2908 – 2924. Part Special Issue: Bio-inspired Methods in Combinatorial Optimization.

Gunawan, A., Lau, H. C., and Vansteenwegen, P. (2016). Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*.

Gutiérrez-Jarpa, G., Desaulniers, G., Laporte, G., and Marianov, V. (2010). A branch-and-price algorithm for the vehicle routing problem with deliveries, selective pickups and time windows. *European Journal of Operational Research*, 206(2):341 – 349.

Gutiérrez-Jarpa, G., Marianov, V., and Obreque, C. (2009). A single vehicle routing problem with fixed delivery and optional collections. *IIE Transactions*, 41(12):1067–1079.

Haimes, Y., Lasdon, L., and Wisner, D. (1971). On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1(3):296–297.

Hamdan, S., Larbi, R., Cheaitou, A., and Alsyouf, I. (2017). Green traveling purchaser problem model: A bi-objective optimization approach. In *2017 7th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*, pages 1–6.

Hansen, P. and Mladenović, N. (2014). Variable neighborhood search. In Burke, E. K. and Kendall, G., editors, *Search Methodologies: Introductory Tutorials* *in Optimization and Decision Support Techniques*, pages 313–337. Springer US, Boston, MA.

Hemmelmayr, V. C., Cordeau, J.-F., and Crainic, T. G. (2012). An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & Operations Research*, 39(12):3215 – 3228.

Hemmelmayr, V. C., Doerner, K. F., and Hartl, R. F. (2009). A variable neighborhood search heuristic for periodic routing problems. *European Journal of Operational Research*, 195(3):791 – 802.

Hosseini, S., Khaled, A. A., and Vadlamani, S. (2014). Hybrid imperialist competitive algorithm, variable neighborhood search, and simulated annealing for dynamic facility layout problem. *Neural Computing and Applications*, 25(7):1871–1885.

Hu, Q. and Lim, A. (2014). An iterative three-component heuristic for the team orienteering problem with time windows. *European Journal of Operational Research*, 232(2):276–286.

Imran, A., Salhi, S., and Wassan, N. A. (2009). A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research*, 197(2):509 – 518.

Jiang, S., Ong, Y. S., Zhang, J., and Feng, L. (2014). Consistencies and contradictions of performance metrics in multiobjective optimization. *IEEE Transactions on Cybernetics*, 44(12):2391–2404.

Kalaivani, L., Subburaj, P., and Iruthayarajan, M. W. (2013). Speed control of switched reluctance motor with torque ripple reduction using non-dominated sorting genetic algorithm (nsga-ii). *International Journal of Electrical Power & Energy Systems*, 53:69 – 77.

Kanagarajan, D., Karthikeyan, R., Palanikumar, K., and Davim, J. P. (2008). Optimization of electrical discharge machining characteristics of wc/co composites using non-dominated sorting genetic algorithm (nsga-ii). *The International Journal of Advanced Manufacturing Technology*, 36(11-12):1124–1132.

Kang, S., Oh, S., and Kim, T. (2006). Heuristic algorithm for solving a multimodal location-based concierge service problem. *Transportation Research Record: Journal of the Transportation Research Board*, 1972:123–132.

Kang, S. and Ouyang, Y. (2011). The traveling purchaser problem with stochastic prices: Exact and approximate algorithms. *European Journal of Operational Research*, 209(3):265 – 272.

Knowles, J., Thiele, L., and Zitzler, E. (2006). A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. Technical report, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland.

Labadie, N., Mansini, R., Melechovský, J., and Calvo, R. W. (2012). The team orienteering problem with time windows: An lp-based granular variable neighborhood search. *European Journal of Operational Research*, 220(1):15 – 27.

Laporte, G., Riera-Ledesma, J., and Salazar-González, J. J. (2003). A branch-and-cut algorithm for the undirected traveling purchaser problem. *Operations Research*, 51(6):940–951.

Lin, S.-W. and Yu, V. F. (2015). A simulated annealing heuristic for the multi-constraint team orienteering problem with multiple time windows. *Applied Soft Computing*, 37:632–642.

Lu, Y., Benlic, U., and Wu, Q. (2018). A memetic algorithm for the orienteering problem with mandatory visits and exclusionary constraints. *European Journal of Operational Research*, 268(1):54 – 69.

Luo, Z., Cheang, B., Lim, A., and Zhu, W. (2013). An adaptive ejection pool with toggle-rule diversification approach for the capacitated team orienteering problem. *European Journal of Operational Research*, 229(3):673–682.

Luo, Z., Qin, H., Zhang, D., and Lim, A. (2016). Adaptive large neighborhood search heuristics for the vehicle routing problem with stochastic demands and weight-related cost. *Transportation Research Part E: Logistics and Transportation Review*, 85:69 – 89.

Mancini, S. (2016). A real-life multi depot multi period vehicle routing problem with a heterogeneous fleet: Formulation and adaptive large neighborhood search based matheuristic. *Transportation Research Part C: Emerging Technologies*, 70:100 – 112.

Manerba, D. and Mansini, R. (2015). A branch-and-cut algorithm for the multi-vehicle traveling purchaser problem with pairwise incompatibility constraints. *Networks*, 65(2):139–154.

Manerba, D., Mansini, R., and Riera-Ledesma, J. (2017). The traveling purchaser problem and its variants. *European Journal of Operational Research*, 259(1):1 – 18.

Mansini, R. and Tocchella, B. (2009). The traveling purchaser problem with budget constraint. *Computers & Operations Research*, 36(7):2263 – 2274.

Masson, R., Lehuédé, F., and Péton, O. (2013). An adaptive large neighborhood search for the pickup and delivery problem with transfers. *Transportation Science*, 47(3):344–355.

Miller, C. E., Tucker, A. W., and Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7:326–329.

Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, 24(11):1097 – 1100.

Molina, J., Laguna, M., Martí, R., and Caballero, R. (2007). SSPMO: A scatter tabu search procedure for non-linear multiobjective optimization. *INFORMS Journal on Computing*, 19(1):91–100.

Montoya, F. G., nos, R. B., Gil, C., Espín, A., Alcayde, A., and Gómez, J. (2010). Minimization of voltage deviation and power losses in power networks using pareto optimization methods. *Engineering Applications of Artificial Intelligence*, 23(5):695 – 703. Advances in metaheuristics for hard optimization: new trends and case studies.

Nabeta, S. I., Chabu, I. E., Lebensztajn, L., Correa, D. A. P., da Silva, W. M., and Hameyer, K. (2008). Mitigation of the torque ripple of a switched reluctance motor through a multiobjective optimization. *IEEE Transactions on Magnetics*, 44(6):1018–1021.

Ong, H. L. (1982). Approximate algorithms for the travelling purchaser problem. *Operations Research Letters*, 1(5):201 – 205.

Palomo-Martínez, P. J. (2015). Problema del agente viajero selectivo con restricciones adicionales. Master's thesis, Graduate Program in Systems Engineering. FIME, UANL, San Nicolás de los Garza, México.

Palomo-Martínez, P. J., Salazar-Aguilar, M. A., Laporte, G., and Langevin, A. (2017). A hybrid variable neighborhood search for the orienteering problem with mandatory visits and exclusionary constraints. *Computers & Operations Research*, 78:408 – 419.

Panda, S. (2011). Multi-objective pid controller tuning for a facts-based damping stabilizer using non-dominated sorting genetic algorithm-ii. *International Journal of Electrical Power & Energy Systems*, 33(7):1296 – 1308.

Panda, S. and Yegireddy, N. K. (2013). Automatic generation control of multi-area power system using multi-objective non-dominated sorting genetic algorithm-ii. *International Journal of Electrical Power & Energy Systems*, 53:54 – 63.

Paraskevopoulos, D. C., Repoussis, P. P., Tarantilis, C. D., Ioannou, G., and Prastacos, G. P. (2008). A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows. *Journal of Heuristics*, 14(5):425–455.

Pearn, W. and Chien, R. (1998). Improved solutions for the traveling purchaser problem. *Computers & Operations Research*, 25(11):879 – 885.

Pirkwieser, S. and Raidl, G. R. (2009). Multiple variable neighborhood search enriched with ilp techniques for the periodic vehicle routing problem with time windows. In Blesa, M. J., Blum, C., Di Gaspero, L., Roli, A., Sampels, M., and Schaerf, A., editors, *Hybrid Metaheuristics*, pages 45–59, Berlin, Heidelberg. Springer Berlin Heidelberg.

Polat, O., Kalayci, C. B., Kulak, O., and Günther, H.-O. (2015). A perturbation based variable neighborhood search heuristic for solving the vehicle routing problem with simultaneous pickup and delivery with time limit. *European Journal of Operational Research*, 242(2):369 – 382.

Ramesh, T. (1981). Traveling purchaser problem. *OPSEARCH*, 18(2):78–91.

Ravi, R. and Salman, F. S. (1999). *Algorithms - ESA' 99: 7th Annual European Symposium Prague, Czech Republic, July 16–18, 1999 Proceedings*, chapter Approximation Algorithms for the Traveling Purchaser Problem and Its Variants in Network Design, pages 29–40. Springer Berlin Heidelberg, Berlin, Heidelberg.

Ribeiro, G. M. and Laporte, G. (2012). An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, 39(3):728 – 735.

Riera-Ledesma, J. and Salazar-González, J. (2012). Solving school bus routing using the multiple vehicle traveling purchaser problem: A branch-and-cut approach. *Computers & Operations Research*, 39(2):391 – 404.

Riera-Ledesma, J. and Salazar-González, J. J. (2005a). The biobjective travelling purchaser problem. *European Journal of Operational Research*, 160(3):599 – 613.

Riera-Ledesma, J. and Salazar-González, J. J. (2005b). A heuristic approach for the travelling purchaser problem. *European Journal of Operational Research*, 162(1):142 – 152.

Riera-Ledesma, J. and Salazar-González, J.-J. (2006). Solving the asymmetric traveling purchaser problem. *Annals of Operations Research*, 144(1):83–97.

Riera-Ledesma, J. and Salazar-González, J. J. (2013). A column generation approach for a school bus routing problem with resource constraints. *Computers & Operations Research*, 40(2):566 – 583.

Ropke, S. and Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472.

Rostami, S. and Neri, F. (2016). Covariance matrix adaptation pareto archived evolution strategy with hypervolume-sorted adaptive grid algorithm. *Integrated Computer-Aided Engineering*, 23(4):313–329.

Salazar-Aguilar, M. A., Langevin, A., and Laporte, G. (2011). An adaptive large neighborhood search heuristic for a snow plowing problem with synchronized routes. In *Network Optimization*, pages 406–411. Springer.

Salazar-Aguilar, M. A., Langevin, A., and Laporte, G. (2014). The multi-district team orienteering problem. *Computers & Operations Research*, 41:76 – 82.

Schiffer, M. and Walther, G. (2018). An adaptive large neighborhood search for the location-routing problem with intra-route facilities. *Transportation Science*, 52(2):331–352.

Sheng, W., Liu, K., Liu, Y., Meng, X., and Li, Y. (2015). Optimal placement and sizing of distributed generation via an improved nondominated sorting genetic algorithm ii. *IEEE Transactions on Power Delivery*, 30(2):569–578.

Sheng, W., Liu, Y., Meng, X., and Zhang, T. (2012). An improved strength pareto evolutionary algorithm 2 with application to the optimization of distributed generations. *Computers & Mathematics with Applications*, 64(5):944 – 955. Advanced Technologies in Computer, Consumer and Control.

Souffriau, W., Vansteenwegen, P., Vanden Berghe, G., and Van Oudheusden, D. (2013). The multiconstraint team orienteering problem with multiple time windows. *Transportation Science*, 47(1):53–63.

Süral, H. and Bookbinder, J. H. (2003). The single-vehicle routing problem with unrestricted backhauls. *Networks*, 41(3):127–136.

Sylejmani, K., Dorn, J., and Musliu, N. (2012). A tabu search approach for multi constrained team orienteering problem and its application in touristic trip planning. In *2012 12th International Conference on Hybrid Intelligent Systems (HIS)*, pages 300–305, Pune, India. Institute of Electrical and Electronics Engineers (IEEE).

Tarantilis, C., Stavropoulou, F., and Repoussis, P. (2013). The capacitated team orienteering problem: A bi-level filter-and-fan method. *European Journal of Operational Research*, 224(1):65–78.

Teeninga, A. and Volgenant, A. (2004). Improved heuristics for the traveling purchaser problem. *Computers & Operations Research*, 31(1):139 – 150.

Ting, C.-K. and Liao, X.-L. (2013). The selective pickup and delivery problem: Formulation and a memetic algorithm. *International Journal of Production Economics*, 141(1):199 – 211. Meta-heuristics for manufacturing scheduling and logistics problems.

Ting, C.-K., Liao, X.-L., Huang, Y.-H., and Liaw, R.-T. (2017). Multi-vehicle selective pickup and delivery using metaheuristic algorithms. *Information Sciences*, 406-407:146 – 169.

Toth, P. and Vigo, D. (2014). *Vehicle Routing: problems, methods, and applications.* Society for Industrial and Applied Mathematics, Philadelphia, PA.

Tsiligirides, T. (1984). Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, 35:797–809.

Vansteenwegen, P., Souffriau, W., and Van Oudheusden, D. (2011a). The orienteering problem: A survey. *European Journal of Operational Research*, 209(1):1–10.

Vansteenwegen, P., Souffriau, W., and Van Oudheusden, D. (2011b). The orienteering problem: A survey. *European Journal of Operational Research*, 209:1–10.

Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., and Van Oudheusden, D. (2009). Iterated local search for the team orienteering problem with time windows. *Computers & Operations Research*, 36(12):3281–3290.

Veldhuizen, D. A. V. and Lamont, G. B. (2000). On measuring multiobjective evolutionary algorithm performance. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, volume 1, pages 204–211 vol.1.

Voß, S. (1996a). Add and drop-procedures for the traveling purchaser problem. *Methods of operations research*, 53:317–318.

Voß, S. (1996b). Dynamic tabu search strategies for the traveling purchaser problem. *Annals of Operations Research*, 63(2):253–275.

Wang, L., Wang, T.-g., and Luo, Y. (2011). Improved non-dominated sorting genetic algorithm (nsga)-ii in multi-objective optimization studies of wind turbine blades. *Applied Mathematics and Mechanics*, 32(6):739–748.

Wang, X., Golden, B. L., and Gulczynski, D. (2014). A worst-case analysis for the split delivery capacitated team orienteering problem with minimum delivery amounts. *Optimization Letters*, 8(8):2349–2356.

Wang, Y., Cheng, H., Wang, C., Hu, Z., Yao, L., Ma, Z., and Zhu, Z. (2008). Pareto optimality-based multi-objective transmission planning considering transmission congestion. *Electric Power Systems Research*, 78(9):1619 – 1626.

Wong, R. T. (1980). Integer programming formulations of the traveling salesman problem. *Proceedings of the IEEE International Conference on Circuits and Computers*, pages 149–152.

Yang, S. H. and Natarajan, U. (2010). Multi-objective optimization of cutting parameters in turning process using differential evolution and non-dominated sorting genetic algorithm-ii approaches. *The International Journal of Advanced Manufacturing Technology*, 49(5):773–784.

Yazdani, M., Amiri, M., and Zandieh, M. (2010). Flexible job-shop scheduling with parallel variable neighborhood search algorithm. *Expert Systems with Applications*, 37(1):678 – 687.

Zahraie, B. and Tavakolan, M. (2009). Stochastic time-cost-resource utilization optimization using nondominated sorting genetic algorithm and discrete fuzzy sets. *Journal of Construction Engineering and Management*, 135(11):1162–1171.

Zhang, C., Chen, Y., Shi, M., and Peterson, G. (2009). Optimization of heat pipe with axial "Ω"-shaped micro grooves based on a niched Pareto genetic algorithm (NPGA). *Applied Thermal Engineering*, 29(16):3340 – 3345.

Zitzler, E. (1999). *Evolutionary algorithms for multiobjective optimization: Methods and applications*, volume 63. Citeseer.

Zitzler, E., Laumanns, M., and Thiele, L. (2001). SPEA2: Improving the strength pareto evolutionary algorithm. Technical report, Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.

Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271.

Autobiography

Pamela Jocelyn Palomo Martínez

Candidato para obtener el grado de

Doctora en Ingeniería

con Especialidad en Ingeniería de Sistemas

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

Mathematical formulations and optimization algorithms for solving rich vehicle routing problems

I was born in Ciudad Madero, Mexico on February 10th, 1990. I am the first-born of Astolfo Palomo Solis and María Concepción Martínez Guerrero's four children.

I completed my primary studies in 2002 at "Escuela Ford #74". After that, I finished my secondary studies in 2005 at "Escuela Secundaria General #4 Profesor José Santos Valdés Salazar" and I earned the "José Santos Valdés Salazar" merit medal. In 2008, I earned a Technical Baccalaureate in Electronics from the "Centro de Bachillerato Tecnológico industrial y de servicios #24" (CBTis 24) and I was awarded the "Elvia Vázquez Flores" medal for academic achievements in mathematics.

In 2008 I moved to Monterrey, Mexico to start my studies in Mathematics at "Universidad Autónoma de Nuevo León" (UANL). I earned my Bachelor's degree in 2012 with the thesis "Uso de un algoritmo Stackelberg-Evolutivo para resolver el problema de fijación de cuotas en una red de transporte" (English translation: Using a Stackelberg-evolutionary algorithm to solve the toll optimization problem).

In 2013 I started my Master's studies at the Graduate Program on Systems Engineering (PISIS) at UANL. As a part of my education, I carried out a research stay at "Universidad Técnica Federico Santa María" (UTFSM) in Vitacura, Chile in 2014. I got my Master's degree in 2015 with the thesis entitled "Problema del agente viajero selectivo con restricciones adicionales" (English translation: The orienteering problem with additional constraints), which was awarded the best Master Thesis on Informatics Technology and/or Computing by the "Asociación Nacional de Instituciones de Educación en Tecnologías de Información" (ANIEI) in 2016.

I started my PhD studies at PISIS in 2015. In 2016 I carried out a research stay at "Centre interuniversitaire de recherche sur les reseaux d'entreprise, la logistique et le transport" (CIRRELT) in Montreal, Canada. In 2017 I earned the Sofía Kovalévskaia award, granted by the "Sociedad Matemática Mexicana" (SMM) and the Sofía Kovalévskaia Foundation. Besides, I got married to the love of my life in 2016 and we had our beautiful twins early this year.