Universidade do Minho
Escola de Engenharia

Bruna Silva Ramos

Models and Algorithms for Integrated Optimization Problems in Operations Management and Supply Chain

junho de 2018

UMinho | 2018

Bruna Silva Ramos

Models and Algorithms for Integrated Optimization Problems in Operations Management and Supply Chain

Universidade do Minho
Escola de Engenharia

Bruna Silva Ramos

Models and Algorithms for Integrated Optimization Problems in Operations Management and Supply Chain

Tese de Doutoramento em Engenharia Industrial e de Sistemas

Trabalho efetuado sob a orientação do
Professor Doutor Cláudio Alves
e do
Professor Doutor José Valério de Carvalho

junho de 2018

STATEMENT OF INTEGRITY

I hereby declare having conducted my thesis with integrity. I confirm that I have not used plagiarism or any form of falsification of results in the process of elaboration of this thesis. I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

Full Name: Bruna Silva Ramos

Signature: Bruna Silva Ramos

Acknowledgements

Walking with a friend in the dark is better than walking alone in the light.

— Helen Keller

The development of this thesis was only possible with the collaboration of several people who helped me to grow personally and professionally.

The most important acknowledgement of gratitude I wish to express is to my advisors Professor Cláudio Alves and Professor José Valério de Carvalho, giving me the opportunity to embrace this project.

I would like to thank you to Professor Cláudio to all the meeting hours, the patience and motivation that allowed this work to be developed with great pleasure. I learned by the example of quality, effort, rigour, seriousness and simultaneously through the enthusiasm and good humor. More than an advisor I consider Professor Cláudio an excellent person with a great heart. I would like to express my deepest sense of gratitude for all the human support that was fundamental to the completion of this project.

It's always a proud privilege to be able to work with Professor José Valério de Carvalho. I would like to express my sincere gratitude for the scientific support and for the wise advices. The calmness with which the knowledge is transmitted is undoubtedly a motivation that always makes work seem easier. I would like to thank you for the excellent human support.

To Professor Rita Macedo for all scientific support, the patience and the brainstorming.

I would also like to thank you to Professor Teresa Monteiro for directing me to this scientific area and for making me believe that problems can only get better. For all the faith and human support you gave me.

To my laboratory colleagues for all the fun and motivation, specially to Nuno Braga for the stimulating discussions and friendship. To Telmo for his suggestions.

All my friends and family who have always supported me. A deep hug to a very dear couple. Natália and Rente thank you for your unconditional friendship and dedication. To my brother André for the moments of relaxation and for his affection. I would also like to thank you for the patience and all the good advices.

I am grateful to my parents for having me become the person I am. The unconditional support, the motivation and the happiness you have given me are small examples of a list that is impossible for me to enumerate. Thank you for making my children your children. Mother, I admire your patience and affection so much. Dad, I love you.

Dear husband, without you it would not be possible to make the last thanks. You are the strongest and most fun person in my life. You always face the adversities of life with a smile and you are always available to help me. I will always love you.

Last but not the least, I would like to thank to my twin boys for the long nights I've been awake and for all smiles and kisses that leave me melted. All the time I've invested in you is still worth it. I love you unconditionally. All of you have been there to support me to create these two beautiful children: doctors, therapists, family and friends. A special thanks to the dear aunt and godmother Márcia Martins who has a huge heart. For their love, affection, dedication and unwavering ability to help.

Words cannot express how grateful I am to my closest friends and family.

The work of Bruna Ramos is supported by a doctoral grant by FCT - Fundação para a Ciência e a Tecnologia (SFRH/BD/90558/2012).

Resumo

A presente tese aborda problemas de otimização integrados e está dividida em duas partes principais. A primeira, refere-se a uma variante particular do problema de localização e encaminhamento, enquanto a segunda se foca num caso particular de um problema de produção, inventário e distribuição de bens. As variantes dos problemas tiram partido da múltipla utilização de veículos que se torna importante quando existe, por exemplo, uma rede geográfica pequena e densa, de forma a não a congestionar.

O problema de localização e encaminhamento com utilização múltipla de veículos combina dois problemas de otimização diferentes: um problema de localização e um problema de encaminhamento. A integração destes problemas é importante para que possam ser consideradas variáveis comuns aos dois problemas. O problema de localização e encaminhamento com utilização múltipla de veículos prevê a identificação de um determinado conjunto de instalações que devem funcionar e determina qual o conjunto de rotas que deve ser efetuado para satisfazer os pedidos de todos os clientes. Estas rotas estão associadas a uma frota homogénea de veículos, sendo esta frota atribuída a uma instalação funcional.

Para a resolução do problema de localização e encaminhamento com utilização múltipla de veículos são utilizados três métodos exatos diferentes: um modelo de fluxo com três índices, um modelo de geração de colunas e um modelo de fluxo nos arcos. No modelo de fluxos com três índices é definido um grafo explícito que inclui um grande número de variáveis relacionadas com a utilização de um arco por um determinado veículo e com a quantidade de fluxo associada a esse mesmo arco. A geração de colunas é dividida no problema mestre que inclui as restrições

associadas ao problema de localização de instalações e no sub-problema que agrupa restrições que têm uma estrutura especial, neste caso, o problema do caminho elementar mais curto. Estes problemas vão trocando informação de forma a encontrar a solução global ótima. O modelo de fluxo em arcos é uma abordagem baseada em grafos, mas menos intuitiva, uma vez que os nodos representam instantes de tempo, em vez de clientes. Foram ainda propostos dois métodos heurísticos para a resolução do problema de localização e encaminhamento com utilização múltipla de veículos. Foi proposta uma heurística de arredondamento onde os valores fracionários da relaxação linear das variáveis são arredondados de acordo com determinados critérios e técnicas de arredondamento, e uma heurística de pesquisa em vizinhança variável que explora um conjunto de estruturas de vizinhança de forma definida e sistemática.

O problema de produção, inventário e encaminhamento com janelas temporais e utilização múltipla dos veículos é um problema integrado que concilia o problema de gestão da produção e encaminhamento com o problema de gestão de inventários. Neste problema integrado um conjunto de clientes, com pedidos que variam de acordo com o horizonte de planeamento finito, é servido por uma única instalação. A distribuição é feita por uma frota homogénea de veículos que entregam os pedidos de acordo com a janela temporal dos clientes. A gestão da produção é feita de acordo com os inventários existentes quer na instalação, quer no cliente.

Para a resolução do problema de produção, inventário e encaminhamento com janelas temporais e utilização múltipla dos veículos foi proposto um modelo exato de fluxos em arcos que tem como base um grafo que considera que os nodos são instantes de tempo. Foram ainda apresentadas duas heurísticas de pesquisa em vizinhança variável baseadas no modelo de fluxo em arcos que de forma sistemática explora um conjunto de estruturas de vizinhança.

O principal objetivo dos problemas abordados é minimizar o custo associado às decisões que envolvem todo o sistema. As abordagens propostas foram implementadas e testadas através de vários testes computacionais que tiveram por base um conjunto de instâncias da literatura. Os resultados finais são apresentados e analisados.

Abstract

The present thesis addresses integrated optimization problems and is divided into two main parts. The first refers to a particular variant of the location routing problem, while the second focuses on a particular case of a production, inventory, distribution and routing problem. The variants of the problems take advantage of the multiple use of vehicles that becomes important when, for example, there is a small and dense geographic network in an attempt to decongest the network.

The multi-trip location routing problem combines two different optimization problems: a facility location problem and a multi-trip vehicle routing problem. The multi-trip location routing problem consists in the selection of a set of facilities to be opened and the determination of a set of routes used to serve a set of customers. These routes are associated to a homogeneous fleet of vehicles, which is associated to a given facility.

To solve the multi-trip location routing problem three different exact methods are proposed: a three-index commodity flow model, a column generation and a network flow model. In the three-index commodity flow model the graph is defined in an explicit way which yields a higher number of variables. The model has variables related to the use of an arc and vehicle, and others representing the flow through the arcs. The column generation process includes two important problems. The restricted master problem that includes restrictions related to the facility location problem, and the sub-problem including constraints which have a special structure related to the elementary shortest path problem. These two problems exchange information in order to find the optimal solution. The network flow model is a graph-based approach, but less intuitive, since nodes represent instants of time instead of clients. Two heuristic methods to solve the multi-trip

location routing problem are also proposed. An iterative rounding heuristic where the fractional value of the linear relaxation of the decision variable is rounded according to some parameters and rounding techniques, and a skewed variable neighborhood search heuristic which explores a set of neighborhood structures in a defined and systematic way.

The multi-trip production, inventory, distribution and routing problem with time windows is an integrated problem that combines a production and distribution problem, a multi-trip vehicle routing problem and a inventory routing problem. In the multi-trip production, inventory, distribution and routing problem with time windows, a set of clients, which have a time varying demand during a finite planning horizon, is served by a single production facility. The distribution is accomplished by a fleet of homogeneous vehicles that deliver the clients orders within their specific time windows. Production management has to be done according to the inventories at the facility and at the customers.

To solve the multi-trip production, inventory, distribution and routing problem with time windows an exact arc flow model based on a graph is proposed, where the nodes represent instants of time. Two model-based variable neighborhood search that systematically explores a set of neighborhood structures exchanging information with the arc flow model are also proposed.

The main goal of the presented problems is to minimize the costs associated to the entire system. The proposed approaches were implemented and a set of experimental tests were conducted. Several computational tests were performed based on a set of benchmark instances from the literature. The final results are presented and analyzed.

Contents

Nomenclature

2MVNS Two-phase Model-Based Variable Neighborhood Search

3MVNS Three-phase Model-Based Variable Neighborhood Search

A

ALNS Adaptive Large Neighborhood Search

AMP Adaptive Memory Procedure

ARP Arc Routing Problem

C

CARP Capacitated Arc Routing Problem

CCG Column-and-Cut Generation

CDCPLP Capacity and Distance Constrained Plant Location Problem

CG Column Generation

CLRP Capacitated Location Routing Problem

E

ESPPRC Elementary Shortest Path Problem with Resource Constraints

F

FIFO First in First Out

FLP Facility Location Problem

G

GRASP Greedy Randomized Adaptive Search Procedure

GVNS General Variable Neighborhood Search

I

IP Integer Programming

IPDP Integrated Production and Distribution Problem

IRP Inventory Routing Problems

L

LARP Location Arc Routing Problem

LC Load Consolidation

LP Linear Programming

LRP Location Routing Problem

LRSP Location Routing and Scheduling Problem

LTL Less-than Transporter Load

M

MIP Mixed Integer Programming

ML Maximum Level

MLRP Multi-trip Location Routing Problem

MPIDRP Multi-trip, Production, Inventory, Distribution and Routing Problem

MPIDRPTW Multi-trip Production, Inventory, Distribution and Routing Problem with Time Windows

MVND Model-Based Variable Neighborhood Descent

MVRP Multi-trip Vehicle Routing Problem

MVRPTW Vehicle Routing Problem with Time Windows and Multiple Routes

O

OU Order-Up

P

PDP Production and Distribution Problems

PRP Production-Routing Problem

R

RMP Restricted Master Problem

S

SA Simulated Annealing

SVNS Skewed Variable Neighborhood Search

T

TA Threshold Accepting

TSP Traveling Salesman Problem

V

VND Variable Neighborhood Descendent

VNS Variable Neighborhood Search

VRP Vehicle Routing Problem

VRPMT Vehicle Routing Problem with Multiple Trips

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The integrated planning of operations, coordinated throughout the different functions of the enterprise, is essential for industrial companies to achieve higher levels of competitiveness. Although the significant progress in the area of discrete optimization and the important contributions reported in solving applied optimization problems, current approaches have obvious limitations. In most cases, the problems are solved independently without any concern with the strong integration that may exist between them. In practice, these approaches lead to solutions that are sub-optimal from the global perspective of companies.

This thesis is a contribution for the efficient resolution of integrated optimization problems in the area of operations management and supply chains. The research focus on two classes of problems in the distribution area: facility location and vehicle routing problems; and production scheduling and distribution problems. Integer Programming models based on original formulations are explored and optimization algorithms based on dynamic management of models, exact methods and hybridization strategies with heuristic methods are developed. The approaches are based on innovative techniques for Integer Programming with a particular emphasis on methods such as decomposition and hybridization with heuristics based on relaxations of the models.

The thesis is divided in 7 main chapters which include the present one (Chapter 1). In Chapter 2, different variants and specificities of integrated optimization problems are explored, namely with regard to the location routing problem and scheduling and distribution problem. This chapter is organized in 4 sections. Section 2.1 presents different variants and approaches for solving the location routing problem, while Section 2.2 focuses on a particular variant of the location routing problem with multiple usage of a vehicle during a planning horizon. In this last section, there is an effort to synthesize the details of this variant addressed in the literature. Section 2.3 provides a literature review of integrated scheduling and distribution problems and Section 2.4 highlights a particular scheduling and distribution problem: the production, inventory, distribution and routing problem. The main characteristics of this particular problem are explored.

For a clear understanding of the structure, Chapter 3 and Chapter 4 address the multi-trip location routing problem while Chapter 5 and Chapter 6 refers to the multi-trip, production, inventory, distribution and routing problem with time windows.

Chapter 3 starts with a detailed description of the multi-trip location routing problem. During this chapter three exact methods are proposed: a three-index commodity flow model, a column generation and a network flow model. The three-index commodity flow model is described in Section 3.2 and leads to a higher number of variables since the load associated to a vehicle and to a facility is represented in an explicit way through the graph. Section 3.3 addresses the column generation approach where the restricted master problem exchanges information with the elementary shortest path problem with resource constraints as sub-problem with the aim of finding the global optimal solution. The restricted master problem includes constraints related to the facility location problem. The network flow model is presented in Section 3.4 and represents a less intuitive graph-based approach, since the nodes of the graph represent time instants instead of clients. At the end of this chapter some implementation details, computational results and conclusions with comparative analysis are presented.

In Chapter 4, two heuristics are proposed: an iterative rounding heuristic and the general skewed variable neighborhood search. At the beginning of the chapter the advantages of the heuristics are discussed in a brief introduction. Section 4.2 addresses an iterative rounding heuristic that takes advantage of linear relaxations. Through rounding techniques the heuristic converts the fractional value of the linear relaxation of the decision variables into integer values according to some parameters and constraints. The skewed general variable neighborhood search is explained in Section 4.3. This heuristic method explores a group of neighborhoods in an attempt to find the optimal solution of the problem or a very good solution. A set of neighborhoods is exploited in a systematic way in order to create perturbations in the final solution, allowing to escape from local optima. Computational results are presented at the end of the chapter.

Chapter 5 addresses the multi-trip production, inventory, distribution and routing problem with time windows. This problem is described in Section 5.1 where some details and specificities are presented. To solve this problem an exact arc flow model based on a graph is proposed, where the nodes represent instants of time instead of clients. Section 5.2 presents the definition of the arc flow model. Some computational results are presented through Section 5.4.

In Chapter 6, two model-based variable neighborhood search are proposed. These approaches explores a set of neighborhoods in an attempt to find good routing and distribution decisions. The arc flow model optimizes the decisions of production and inventory at the facility. The local search method and the arc flow model exchange information in order to find a good solution.

The main contributions and some conclusions are presented in Chapter 7. In addition, some future research directions are introduced.

Chapter 2

Integrated optimization

Outline

Integrated optimization

Nowadays, in contrast with a distant past, one is constantly dealing with permanent technological evolution. Due to this fact, some problems that appeared to have no solution in acceptable computer science terms, may be addressed with new and fresh perspectives. Technological advances are arguably an aid, however they may not constitute the basis of the research. In order to solve a complex investigation issue, a systematic examination of the real environment, in which the problems are embedded, is required. Thus it is possible to perceive the entire problem and apply integrated optimization techniques. These integrated optimization techniques should lead to better results than addressing the problem in an individual way.

Integer Programming (IP) techniques have been successfully used in the last years to solve problems of planning and management in various application domains [1, 2, 3, 4, 5]. Progress has been remarkable: obtained theoretical results are important and computationally efficient algorithms have been proposed to relevant and complex problems. These results are translated into substantial savings which are a consequence of a better use of resources. Economic agents have recognized the relevance of these contributions and their practical importance. A clear example is the strong commitment of some of the most prestigious companies in business analytics, which combines contributions arising essentially from computer science, optimization and operations research.

In this research work, optimization problems in the area of integrated management operations and supply chains are addressed. Despite the practical importance of these problems, the first scientific results in this field have emerged only in the last years [2, 6, 7, 8, 9, 10, 11]. The contributions focus mainly on heuristic methods and do not use the advantages of the latest advances in IP techniques. However, some preliminary experiments clearly illustrate the potential of IP methods in solving specific problems of integrated optimization [6, 8]. In this project it is intended to contribute to the resolution of two classes of integrated optimization problems: facility location and vehicle routing problems; and production scheduling and distribution problems.

2.1 Location routing

The Location Routing Problem (LRP) is a difficult optimization problem that integrates the facility location and the routing problems. The routing problem allows the determination of a set of optimal routes such that the demands of multiple clients are fulfilled. In the location problem, one has to determine the location of a set of depots to be used. The integration of these two problems has as main objective the minimization of the total cost of the system, thereby allowing the reduction of unnecessary costs. During the last years, several authors analyzed this problem and its specificities in an attempt to reduce the complexity of the entire system expecting good solutions in acceptable computer science terms.

A review of methods for the location and vehicle routing problems and their variants has been proposed by Nagy and Salhi in [12]. Most of the proposed methods are purely heuristic approaches. Nagy and Salhi [12] classify these heuristics into three categories: clustering techniques, iterative heuristics and heuristics based on hierarchies [13, 14, 15]. Prodhon et al. [16] and Drexl et al. [17] propose complementary surveys in an effort to update the review offered by Nagy and Salhi [12]. The exact number of approaches to variants of this problem is still very low [13, 18, 19]. Belenguer et al. [18] describe a branch-and-cut algorithm to instances of the problem with capacity constraints in depots and a maximum of 50 clients. Akca et al. [13] propose a branch-and-price algorithm combined with heuristics for solving the column generation subproblem. Another branch-and-price algorithm has been proposed by Berger et al. [19] to solve a version of the problem without capacity constraints on depots and constraints on the distances travelled.

Albareda-Sambola et al. [20] define a combined LRP problem solved by a Tabu Search heuristic that solves the facility location and distribution problem simultaneously. The authors presented a deterministic formulation for one single period. In their approach the capacity of the depots is considered with one vehicle associated to each depot. The main goal is to determine which depots should be opened and the distribution routes in order to minimize the cost related to opening a depot and the cost of the different routes. Using a Linear Programming (LP) problem the authors are able to obtain an initial lower bound which is considered as the starting point of the Tabu Search process. The main constraints associated to the problem ensure that a route starts and ends at the same depot, the vehicles capacity is not excedeed and a client is served by exactly one route, which is not associated to a closed depot. In order to obtain the initial point for the heuristic method the authors used a linear relaxation of the initial model. This relaxation is strengthened through the inclusion of new constraints ensuring that the total client demand is lower than the total capacity of the open depots. After a first step for obtaining the initial point, the authors used a Tabu Search heuristic. In this iterative process, each iteration has two phases: an intensification phase followed by a diversification one. In the former, a local search is done in order to find a better solution than the current one. This process consists in the re-assignment of at least one client to another depot, which could be done in two distinct ways. One consists in re-assigning a client from a depot to another depot, while the other consists in the exchange of a client from one depot with a client from a different depot. The intensification phase ends when the maximum number of iterations is reached or when it is not possible to find a valid solution in the neighborhood after a consecutive number of iterations. In this phase, the solution is changed but the number of open depots remains unchanged. In the diversification phase, one has to increase the solution space by searching solutions with different open depots. For this phase, there are three different methods. One method is to close a depot and to re-assign the associated clients. A different approach consists in replacing an open depot by a closed one that is able to ensure the demand of the clients. The third method is to open a depot and re-assign part of the clients of other depots using a proximity criteria. These three phases are executed sequentially only if in the current phase it is not possible to obtain a valid solution. The authors conclude that this approach provides good results and is trustworthy.

Nagy and Salhi [12] perform a survey of issues, models and methods for the LRP problem. They consider LRP a relatively new branch of research. They also propose a potential classification scheme according to the different variants of the initial problem. The research work conducted by the authors uses exact and heuristics methods. The authors underlined the importance of their research whose objective is to conduct a review that can be considered a guide to future researchers who want to start up in this area. Being the LRP an approach to model and solve location problems, Nagy and Salhi [12] define the problem using a hierarchy. The top-level goal is to solve facility location problems, being necessary to address the vehicle routing problem at a level immediately below. The facility location and vehicle routing problems have a strong interrelationship. This relationship is sometimes neglected by researchers and professionals that seek an optimal solution for the facility location without taking into account the distribution routes. The LRP can become the traditional problem of facility location if one considers that all clients are directly connected to the depot. If one consider that the depots have a fixed location, then the LRP becomes the classic Vehicle Routing Problem (VRP). Areas such as health, military, and communications are targets for the application of LRP, with the majority of research focused on the distribution of goods. Nagy and Salhi [12] emphasize that the LRP is not purely academic having practical application in the mentioned areas. Classifying LRP problems is a difficult task due to several variants of the location problem, with the addition of the various variants of the routing problem. The classification proposed by Nagy and Salhi [12] is divided into nine key aspects of the LRP:

- ▷ **Hierarchical base structure:** customers are served through routes associated with depots. Each route depends only on a depot;
- ▷ **Data type:** the demand of a customer can be stochastic or deterministic;
- ▷ **Planning horizon:** the planning horizon can contain a single period (static) or multi-period (dynamic);
- ▷ **Solution method:** can be exact or heuristic. Exact methods are efficient for particular cases of the LRP. Heuristic methods allow solving larger instances in acceptable compu-

tational times;

- ▷ **Objective function:** the minimization of the overall costs associated with depots and vehicles is the main objective of the LRP;
- ▷ **Solution space:** can be discrete, in network or continuous;
- ▷ **Number of depots:** there may be considered one (single) or more (multiple) depots;
- ▷ **Number and type of vehicles:** the fleet can be homogeneous or heterogeneous with a varying number of vehicles;
- ▷ **Structure of the route:** the route starts and ends in the same depot visiting various customers.

In the first deterministic problem solved, the route started in a depot, the load was brought from a client, delivered to another and ended later in the same depot where the route started. This problem is a particular case named round trip location problem. The problem definition through exact methods using mathematical formulation frequently involves relaxation and reintroduction of constraints such as the elimination of sub-routes, supply chain constraints and integrality.

Berger et al. [19] present the LRP problem with constraints only to limit the distance that vehicles may travel. Through an alternative set of constraints, the authors create a formulation that improves the value of the relaxation of the linear programming model. In conventional approaches a depot location model is used, and the distribution routes serve one client at a time. For this reason, the authors state that the cost of each route is independent from the others. However, in real cases, each route usually visits more than one client, which makes the cost dependent on the location of the various clients and the sequence in which they are satisfied. The facility location problem and routing problem must be solved simultaneously in order to perform the most accurate representation of reality. The authors consider a version of the LRP without capacity associated to the depot, but consider the load carried by the vehicle, adding constraints on the duration or length of the routes. This version fits, for example, in the delivery of goods with short lifetime and delivery of products at pre-established intervals. LRP formulations using exact methods are still rare and solve relatively small instances. Berger et al. [19] use set partitioning and branch-and-price algorithms for the LRP resolution through exact algorithms, evaluating their efficiency. In partitioning set approaches the goal is to select a group of open facilities and to calculate the distribution routes where the total costs are minimized. Each customer is visited only once by a route where its length is limited. The formulation includes an exponential number of variables and constraints, which is why the application of the formulation to real cases is not feasible. To overcome these difficulties, Berger et al. [19] use a column generation model with branch-and-bound. In the formulation, sets of constraints are replaced in order to strengthen the lower bound of the linear relaxation, allowing for better results in the application of the branch-and-bound algorithm. The formulation, which is solved by branch-and-price, is efficient for the resolution of problems such as crew airlines scheduling and delivery of goods with defined time interval. With the proposed formulations, the authors can obtain optimal solutions for instances with 10 candidate facilities and 100 customers with various distance constraints.

Barreto et al. [21] consider a particular LRP problem with a set of distribution centers with an associated capacity and a set of customer. The main objectives are to determine the set of facilities that will work effectively and to optimize the distribution routes that will start and end at the same depot. The vehicles are homogeneous, have a finite capacity and vehicles carry only one type of product. Each client is visited exactly once. The minimization of total costs associated with routes and depots location is the main purpose of the authors. The division of entities with similar characteristics into groups are commonly referred to as cluster, which represents an approach used by several authors in the LRP. Barreto et al. [21] integrate hierarchical and non-hierarchical techniques for sequential heuristics to find optimal solutions in LRP. The particular cases of LRP which deal with capacities are called Capacitated Location Routing Problem (CLRP). The use of heuristics for these problems allows to obtain good solutions within an acceptable computational time. The heuristics presented by the authors are easy to understand, to implement and to modify and allow the definition of some specificities associated to the problem. They also enables the execution of large instances in acceptable computational times. According to Jain [22], a cluster might be defined as:

> *"Cluster may be described as connected regions of a multi-dimensional space containing a relatively high density of points, separated from other such regions by a region containing a relatively low density of points"*

This definition has revealed to be important since it demonstrates a good reason to use the analysis of clusters in LRP. This analysis is composed by several methodologies which may be used on heuristics for the particular CLRP problem. A significant number of authors have proposed their approaches using integrated grouping techniques in LRP. However, the comparison of these techniques to determine the real capabilities of their approaches is rarely addressed. In order to determine the proximity between points on the plane, Barreto et al. [21] used some different techniques such as determining the shortest distance, the greater distance, the average distances, the distances from the centers of gravity and the saving method. According to Barreto et al. [21], the lack of consensus of several authors demonstrates that there are no adequate measures to determine the proximity between points on the plane for generalized problems. These measures are commonly selected after testing several alternatives. Heuristic methods for LRP can be defined using sequential or iterative strategies. The authors proposed a sequential approach since it produces perfectly acceptable errors associated to the obtained solution and it is preferable from a computational perspective. The sequential heuristic strategy addresses, in a first step, the distribution problem and then the location problem. The heuristic proposed by the authors is defined into 4 essential stages: construction of customer groups with capacity limit, determination of the distribution route for each customer group, improvement of the routes and location of the depots and their assigned routes. The authors used 19 instances adapted from literature in order to evaluate the performance of the four versions of the heuristics and six proximity measures, concluding that the first version of the heuristic provides better results.

Lopes et al. [23] developed a decision support system which includes a LRP with limits on the capacities of the vehicles and depots. The authors highlight the fact that the decision support system has to consider other aspects in addition to the optimization problem. A feature considered important, which allows an assertive decision in this type of systems, is the problem presentation in a complete, simple and easy to understand interface. The integration of the LRP in decision support systems is scarce and the authors try to fulfill this flaw. Lopes et al. [23] deal with a LRP with capacitated depots and an homogeneous fleet, in which all costs are known or may be calculated, allowing for their minimization. Due to the larger size of the practical problems the authors resort to heuristics that provided a good solution in reasonable time. The authors developed a sequential approach of the LRP due to the advantages at a computational level. The problem was divided in two sub-problems (distribution routes and depots localization) which were sequentially solved according to four steps:

- ▷ **Group clients**: according to the limit of the capacity and analysis of possible neighborhoods;
- ▷ **Determine the distribution route**: to each group of clients through the Travelling Salesman Problem using linear and integer programming for problems with less than 40 clients and heuristics to set the neighbor and local search, otherwise;
- ▷ **Improve routes**: defined in the previous item through a local search heuristic, reducing their costs;
- ▷ **Locate the depots**: assign routes to the depots and corresponding costs to the group of clients that should be served through an exact method.

Both in the determination of the distribution routes and in their assignment to the depots, the commercial optimization solver CPLEX was used. Through the integration of several technologies, the authors provided a graphical interface which allows an easier and intuitive definition of the problem as well as the interpretation of the obtained results.

According to Belenguer et al. [18], there are few studies under the LRP topic where exact algorithms are addressed to enable the resolution of this type of problem. The LRP model developed by the authors uses integer and linear programming and has points in common with the work carried by Laporte et al. [24]. However, new constraints that limit the depots capacity are included through the use of binary variables. The present case consists of opening one or more depots, for which it is necessary to associate a number of routes, where the total demand of customers cannot exceed the depot capacity. Each route must start and end at the same depot and the total demand must be fulfilled. In their approach, Belenguer et al. [18] analyze a generic problem which has capacity constraints in the load of vehicles and in the quantity stored in the depots. These constraints involve more complex decisions. The integer programming models that the authors presented are composed by an objective function and several initial constraints. Generally, in the objective function, the authors represent all costs associated to the transportation and location problem, in order to minimize them. These costs are reflected in fixed costs for opening depots, fixed costs for the use of vehicles and also the cost associated with each route. Each customer is visited exactly once and sub-routes are eliminated. The depot capacity cannot be exceeded, a depot is only used if it is open, and a route must start and finish in the same depot. It is noteworthy that some of these constraints grow exponentially with the number of customers, which is why the authors use branch-and-cut that introduces specific constraints only when necessary. In order to strengthen the LP relaxation of the formulation the authors include additional constraints to the problem. The LRP formulation with the new improved constraints allows to obtain a lower bound for the LRP. In the first iteration, the LP problem is initialized with the objective function and a subset of constraints. In each iteration, the LP problem is solved and the valid inequalities violated by the optimal solution of the LP problem are identified and added to the LP problem. These constraints are then added to the LP problem and the iterative process continues, finishing when there are no more violated inequalities. In order to test the algorithm, Belenguer et al. [18] used three sets of instances in the literature, corresponding to 34 instances with 20 to 88 customers and 5 to 10 possible locations for the depots. The computational results showed that the developed method solved optimally 26 instances with 5 potential locations for depots including all instances up to 40 customers and only 3 with 50 customers.

Albareda-Sambola et al. [25] address the LRP problem with capacity constraints limiting the distance that the vehicles may travel. This variant is named as Capacity and Distance Constrained Plant Location Problem (CDCPLP). When a depot is open, a fixed open cost is associated to it as well as the number of identical vehicles. Each vehicle can make several one-way routes to and from its depot if they do not exceed the maximum limit of distance traveled. The CDCPLP has as main objective to determine which depots will be opened and the number of associated vehicles, minimizing the fixed costs of opening the depots, the fixed costs of vehicle usage and the allocation costs. Several models proposed for the CDCPLP problem are compared although all the integer programming models considered have as main objective the minimization of all costs incurred. In the models presented by the authors the first constraint ensures that the customer orders are not neglected. A second constraint ensures that the depots capacity is not exceeded by the number of requests from the customers that it serves. The sum of the distances in the various round-trip routes must be less than or equal to a maximum distance traveled by the vehicle. The authors also ensured that a customer is not served by a closed depot. An additional constraint is introduced to ensure that a customer is served by only one vehicle and that the vehicles are used in the order of association with the depots. The versions of the models presented by Albareda-Sambola et al. [25] differ due to several factors that are summarized next:

- ▷ **Strengthening the initial model:** when a client is served by a vehicle then the previous customer has been served by the same vehicle or a previous one. This approach prevents the replication of solutions since the vehicles are homogeneous;
- ▷ **Removing the explicit variable associated to a vehicle:** vehicles are now assigned

to a lower index client avoiding the use of negligible variables and symmetrical solutions;

- ▷ **Setting upper and lower bounds for the distance traveled:** the depots will be opened only if the total distance that their assigned vehicles perform is limited between an established upper and lower bound.

- ▷ **Improving the limits:** the strengthening of the previous formulation is conducted through the addition of new inequalities based on bin packing and knapsack problems, respectively.

The modifications applied to the initial formulation presented good results. The latest improvement presents the most promising results for instances with the maximum distance associated to the vehicles between 40 and 100 and fixed costs of vehicle usage between 50 and 300. All instances have 10 possible depots and 20 customers. Albareda-Sambola et al. [25] concluded that getting better upper and lower limits may have a beneficial effect on the total computing time used.

Escobar et al. [26] proposed a two-phase hybrid heuristic algorithm to solve the Capacitated Location Routing Problem (CLRP). The problem considers a homogeneous fleet with capacity and fixed costs associated, a set of capacitated depots and their opening costs and a deterministic set of clients demands. It is necessary to determine which depots should remain open and associate to them the best distribution routes. The fixed costs of vehicles and depots and the costs of the routes should be minimized. In the CLRP each route must start and end at the same depot and it cannot connect to other depots. Each client is visited just once by one route and the total demand of the clients served by a route cannot exceed the vehicle capacity. The total demand of the clients associated to a depot cannot exceed its capacity. The heuristic has two distinct phases: construction phase and improvement phase. In the former, an initial valid solution is selected and used in the next phase in order to trying to avoid that a local optimum is not achieved. The improvement phase is based on a modified granular Tabu Search heuristic which considers five neighborhood structures. There are three diversification strategies and a perturbation procedure that ensures that the iterative process does not end on a local optima. For the computational tests, the authors used the five most effective instances from the literature. The proposed algorithm solved the various instances improving the known computational times. Escobar et al. [26] stated that the proposed algorithm may be extended to other versions of the CLRP.

According to Doulabi et al. [27], there are three important areas of research in logistical problems: facilities location, inventory management and vehicle routing problem. The authors presented integer programming models and heuristics for the resolution of location routing problems with multiple depots. The determination of the routes is made through the arcs. This approach, according to the authors, is not commonly reported in the literature for the LRP problem. This sub-problem, referred to as Arc Routing Problem (ARP) can be divided into three main problems:

- ▷ **Chinese Postman Problem**: all of the vertices or arcs of the graph must be traversed.
- ▷ **Rural Postman Problem**: only a subset of vertices, or arcs must be traversed.
- ▷ **Capacitated Arc Routing Problem**: similar to ARP problem but vehicles have capacities.

Doulabi et al. [27] proposed two integer programming models. One for problems with a single depot and another for problems with multiple depots. The upper and lower bounds are computed using integer programming models. The objective function aims to minimize fixed costs associated to opening the depots and the costs associated to the routes. The constraints ensure the continuity of routes and that each arc or edge is served in one route. The flow conservation in the graph is also ensured. The authors highlighted that the number of variables for the single depot problem is much lower than the number of variables of the multiple depots version, which is reflected in the computational time. Due to the complexity of the problem Doulabi et al. [27] developed heuristics that able to obtain very good solutions, close to the optimal one. For the Location Arc Routing Problem (LARP) the authors developed two heuristics based on simulated annealing: heuristics for determining routes through the arcs (arc routing) and location-allocation heuristic. The arc routing heuristic has as main objective the insertion of small sub-routes inside an existent route in order to reduce the costs. In an initial phase, a basic solution is created, with a route being generated for each arc or edge. Then the solution is improved by the combination of the various routes previously obtained, ending when the stopping criteria is reached, i.e., when the conjugation of the routes implies an increase in the objective function or the violation of the vehicle capacity. Location-allocation heuristic has the aim of decreasing the total costs by swapping the routes associated for each depot. Initially, the authors define the set with the location of the depots, then the pre-established routes are disconnected from their associated depot becoming closed circuits (union of the edges that connected the route to the depot). These new circuits are designated as clusters. Each cluster is joined to a random depot and the improvement in the objective function is verified. The routes of each depot are joined or disconnected, and the process continues in an iterative manner until no improvement in the solution is found. Doulabi et al. [27] concluded that the proposed heuristics can find good quality solutions in reasonable times to the LARP problem with multiple depots.

2.2 Multi-trip location routing

The multi-trip location routing problem is a variant rarely discussed in the literature nevertheless it has been previously addressed in [28, 29, 30, 13, 31, 32]. Although one may perceive the effort made by different authors to perform a review of methods for the location routing problems [12, 16, 17], the variant of the multi-trip location routing problems is poorly mentioned. During this research work, special attention is given to this variant of the problem. Since this particular variant is rarely reported in the literature, a review of multi-trip location and vehicle routing problems methods is performed, which includes approaches related to the multi-trip vehicle routing and the multi-trip location routing. The multiple usage of a vehicle is analyzed in an individual manner and with the additional location routing problem. Several authors [33, 3, 34, 35, 5, 36, 37] demonstrate a relevant interest in the multi-trip vehicle routing problem variant (MVRP) in recent years.

Lin et al. [28] presents a location-routing-loading problem for bill delivery services. The authors consider a system that allows for the distribution of printed bills for customers, since they consider that there is still a large part of the population who prefer to receive paper bills. They consider that this service is important for those who still do not have technology access and a large part of elderly people. A fast delivery of bills enables a faster settlement thereof. A study carried out by the authors has proven that having a own distribution service would be advantageous in terms of cost in relation to the existing postal service. In this particular problem, there are potential locations for depots to be taken into consideration. Vehicles are rented with associated costs, they have load capacity and are limited to certain working hours. The authors consider good routing and scheduling decisions related to capacity constraints of the vehicles, in terms of charge or in working hours. The reuse of a vehicle, which consists in associating to a vehicle more than a single route, is another essential decision. For this purpose, they have developed four heuristic algorithms and a branch and bound exact algorithm. Lin et al. [28] used the SA and Threshold Accepting (TA) meta-heuristics in an initial phase and then combined them to obtain another two different approaches. This combination allows for the escape of local optima in order to improve the final solution. In the presented problem, instances with four potential depots and 27 demand nodes were tested.

Lin and Kwok [29] studied integrated logistic systems that incorporate the LRP. In their approach, the authors consider that the cost of acquiring the vehicles may be more significant than the cost associated to each route. For this reason, they studied the case where different routes are associated to the same vehicle through Tabu Search and Simulated Annealing meta-heuristics. The authors developed two versions of the problem in which they assign the routes to the vehicles in a simultaneous and sequential manner. In logistics, it is necessary to provide services that are executed at the client place. Thus, it is necessary to send specialized vehicles and workers.

In this problem, there are several decisions that have to be made: the selection of the depots location, the scheduling of routes associated to the selected depot and to the clients to serve, and the association of vehicles and workers to the routes. The authors formulated a problem with several goals (minimization of the total cost and vehicle work load) that allow the vehicles to perform more than one route if the total use time of the vehicle is not surpassed. Clients have to be totally satisfied according to the several constraints of the problem. Each client may only be visited once and its demand has to be completely satisfied. The depots are selected from a set of possible locations. A vehicle starts and ends the route at the same depot and visits a set of clients, if its capacity and traveling time are not exceeded. In this problem, the authors use a variant of the LRP where the vehicles may be reused and do more than one route, which is accomplished through two heuristics. Each of these heuristics has two versions. One version tries to solve the multi-objective LRP assigning the resulting routes to the vehicles (sequential approach). The other version takes into consideration the reuse of vehicles for the solution of the multi-objective LRP (simultaneous approach). Lin and Kwok [29] used a Tabu Search meta-heuristic and a Simulated Annealing meta-heuristic in order to solve the two versions of the problem. In the Tabu Search meta-heuristic the minimum number of needed depots is estimated in order to fulfill the total demand. Then, a set of depots that have the largest request density nearby is determined. The initial routes are calculated, associated to the vehicles and the new improved solutions are generated. In the sequential algorithm, the maximum number of routes are assigned to the minimum number of vehicles, then another set of possible initial depots is analyzed and the previously described procedure is performed iteratively. The meta-heuristic ends when it is not possible to find any better solution. The simulated annealing meta-heuristic is similar to this method excluding the criterion for accepting new solutions where a temperature parameter is used. The developed heuristics were tested in real and simulated instances. On the real instances, the heuristic behaviour is affected by the characteristics of the area under analysis. When the demand density is large, the vehicles perform routes with less visited clients, while in smaller demand densities each route has more clients. The simultaneous approach revealed to be more suitable than the sequential one when considering the multi-objective problem.

Olivera and Viera [35] propose a heuristic to solve the Vehicle Routing Problem with Multiple Trips (VRPMT). Each vehicle is able to perform several routes during the same planning period. They consider a homogeneous fleet where vehicles have an associated capacity. The authors use an Adaptive Memory Procedure (AMP) to solve the problem, where components of feasible good solutions are kept. This algorithm allows the periodic construction of new solutions using the good solutions available in the memory and improving them through local search algorithms. The improved solution created is then added to the memory. The authors consider a sorted multi-set of routes as the memory. These components of the memory are improved by a tabu search method and subsequently will become part of the memory. The authors tested their algorithm over 104 benchmark instances from the literature.

In [30], Akca et al. present a graph-based model with three-index decision variables. The model uses two different decision variables. One indicates if a vehicle travels on a specific arc and the other represents the flow that travels through the arc carried by a given vehicle. The aim of the formulation is to minimize the costs associated to the global system, which includes operating costs related to the routes and fixed costs to open a facility or to use a vehicle. The authors use constraints to guarantee the flow conservation of the system and use constraints that limit the travel time of each vehicle. The capacity of the vehicles and depots is also limited by other restrictions. Through the branch-and-price methodology, the authors propose an exact solution to the integrated Location Routing and Scheduling Problem (LRSP).

Akca et al. [13] propose a branch-and-price algorithm for combined location and routing problems under capacity constraints. The authors introduced a variant of the model, which they previously proposed in [30] in order to solve a LRP problem with capacity constraints. The authors address the problem through a column generation model. The original problem is reformulated and the main problem is strengthened with additional constraints. In order to solve the sub-problem, an elementary shortest path problem with resources constraints is used. The authors use a set of exact and heuristic methods to find a good solution to the LRP and consider the multiple usage of a vehicle. The depots and vehicle have limited capacities and a vehicle cannot travel more than a certain time limit.

Azi et al. [3] use an exact method to solve the Vehicle Routing Problem with Time Windows (MVRPTW). The authors consider that a vehicle makes several routes in order to serve a set of customers within a specified time window. In [33], the authors use only a vehicle to solve the problem, and in [3] consider the same problem but allowing the use of more than one vehicle. They use a branch-and-price approach where the problem is divided into several other problems. The main problem is a set-covering problem where a set of routes are assigned to one vehicle and for one planning horizon. Azi et al. [3] use the elementary shortest path problem with resource constraints as sub-problem where the nodes of the graph represent vehicle routes. The aim of the problem is to serve the maximum number of clients minimizing the distance travelled. Azi et al. [3] simplify the problem generating routes a priori. The authors present the computational results that show that their procedure is able to solve different instances with up to 40 clients.

In [34], Azi et al. propose an heuristic method to solve the problem described previously in [3] (MVRPTW). The authors present an adaptive large neighborhood search (ALNS) that uses a ruin-and-recreate principle. This method allows to search for a better solution reconstructing the current solution through the destruction of part of them. For that reason the authors create various destruction operators that are defined at the client, trip and planning horizon level. At each iteration a destruction and a reconstruction operator is randomly selected in order to find a good valid solution. The main concern of the problem is to satisfy the maximum requests of the customers, minimizing the total distance travelled by the vehicles. The authors adapt several instances from the literature and test the algorithm solving instances with up to 1000 clients.

Macedo et al. [32] analyze a location routing problem variant. A vehicle can now make more than a single route during the workday. Two difficult problems are combined by the authors: a vehicle routing problem and a location problem. The authors point out that the conciliation of these two problems can lead to significant savings. The former has as main objective obtaining a set of optimal routes to fulfill the clients needs and the latter selects the depot from which the vehicle will perform the route chosen from a set of available locations. The authors use a pseudo-polynomial network flow model where the nodes of the problem represent time instants. The arcs associated to the network represent vehicle routes that are feasible. In this approach the global minimization of the costs associated to the system is considered. The costs may be fixed such as the usage of a vehicle or opening a depot or variable such as the routes performing. The authors consider a capacitated problem. The depots have capacity and vehicles can only transport a given load. A vehicle also has a travel time limit. In the pseudo-polynomial model proposed, the variables are explicitly generated producing valid vehicle routes. In order to increase the performance of the presented model Macedo et al. [32] only consider routes with potential interest to the optimal solution. They implement the arc reduction that allows to consider less arcs in the global system. The authors conducted a set of tests with instances with five available depots and with up to 25 clients. Other parameters were varied.

Macedo et al. [5] solve a vehicle routing problem through a pseudo-polynomial model. They address a vehicle routing problem with time windows and multiple routes (MVRPTW). The authors describe an exact pseudo-polynomial network flow model. All feasible routes are generated in an initial phase according to the additional duration of the routes. The nodes associated to the problem represent instant times and a workday is composed by a set of paths. They consider that a vehicle can perform more than a single route in the same period. The problem considers only one available depot to fulfill the demands of the clients. All the vehicle routes start and end at the unique depot available. The authors consider a homogeneous fleet and each vehicle has a capacity associated. It may not be possible to visit all clients since the number of vehicles is limited. For that reason, the main objective is to satisfy the maximum number of clients. The authors compare the obtained results with the one proposed by Azi and Gendreau [3] and prove that their algorithm is more efficient.

Mingozzi et al. [36] propose an exact method to solve the multi-trip vehicle routing problem. The authors formulated two set-partitioning-like models. In the former formulation, it is necessary to generate feasible routes and, in the latter, to generate all feasible schedules. They consider that a schedule is associated to a vehicle and is composed by a subset of trips, allowing the concept of multiple routes during its working period. The total duration of a schedule may be less than a workday, and the sum of the single trips costs associated to a schedule determines its total cost. They assume that a vehicle has a capacity and use a fleet of homogeneous vehicles to serve a set of clients. Each customer requires products from the depot where the fleet is located. Mingozzi et al. [36] analyzed the valid lower bounds obtained through the linear relaxation of the presented models that are strengthened with valid inequalities. For that, the authors present four column-and-cut generation procedures. The lower bound values are inserted posteriorly into the exact solution method that helps to create a reduced set of trips to the former formulation and a smaller set of schedules to the latter formulation. With this method, the authors guarantee that any optimal solution of the problem is not discarded. The main objective is to minimize the total cost associated to the selected schedules ensuring that a client is visited exactly once by the routes that compose the schedules. The algorithms are successfully tested in instances available in the literature, and can optimally solve instances involving up to 120 clients. The authors highlight that the resulting reduced problem is directly solved through an integer programing solver.

Cataruzza et al. [37] address the multi trip vehicle routing problem through a hybrid genetic algorithm. The main objective is to serve a set of clients through a fleet of vehicles minimizing the total travel times. The authors take into account temporal and capacity restrictions and each vehicle is able to perform more than a single trip per period horizon. They highlight that this variant of the problem is particularly relevant in the city logistics context. Lower capacity vehicles are generally favored by road and laws restrictions in deliveries. This limitation of capacity leads to trips that do not occupy all the workday. Each vehicle may return to the depot to reload the demand of another service trip. Authors combine moves and swaps between trips to create a new local search operator for this specific problem. The obtained results are compared with other described in the literature.

The special features of the models described by several authors aforementioned are summarized in Table 2.1 enabling an easier identification of the differences between the various approaches.

Table 2.1: Synthesis of the state of the art to the multiple routes variant

Features	Authors	[28] Lin et al. 2002	[29] Lin et al. 2006	[33] Azi et al. 2007	[35] Olivera & Viera 2007	[13] Akca et al. 2009	[3] Azi et al. 2010	[32] Macedo et al. 2011	[5] Macedo et al. 2011	[36] Mingozzi et al. 2013	[34] Azi et al. 2014	[37] Cattaruzza et al. 2014
Problem	VRP			✓	✓		✓		✓	✓	✓	✓
	LRP	✓	✓			✓		✓				
Approach	Exact	✓		✓		✓	✓	✓	✓	✓		
	Heuristic	✓	✓		✓	✓					✓	✓
Variant	Multiple routes	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Multiple vehicles	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓
	Multiple depots	✓	✓			✓		✓				
	Depot capacity[1]	✓	✓			✓		✓				
	Load capacity	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Time horizon	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Time windows			✓			✓		✓		✓	

[1] The absence of the checkmark represents an unlimited capacitated depot

2.3 Integrated scheduling and distribution

During the last years, several authors have defended the integrated planning and management of operations in the industry. It is important that companies ask how to drive the operations strategy and how to structure the process of planning and controlling these operations. With the advent of strategies such as just-in-time, companies seek to reduce inventories in order to gain competitiveness. They use integrated models, which make the connection between jobs scheduling, during the production phase, and delivery of products. Companies make a continuous effort to integrate the resolution of problems in order to minimize the costs associated to the global and main problem. However, despite this effort many of the problems continue to be solved sequentially. The integrated scheduling and distribution problem represents a very comprehensive area of problems as described by Chen in [38]. The classification of integrated scheduling and distribution problems is a difficult task since these problems include several variants in the production, inventory, distribution and routing problems, among other specificities. Chen in [38] proposes a model representation that is divided into five key aspects of the integrated scheduling and distribution problem:

1. **Machine Configuration:** most authors address problems involving a single production plant that deals with different machine configurations. Few authors consider multiple plants where each plant processes a set of orders. The configuration of machines has many peculiarities:

    ▷ **Single-machine configuration:** a single machine processes all the orders of the production plant;

    ▷ **Parallel-machine configuration:** a set of identical machines where each machine processes a specific order;

    ▷ **Flowshop configuration:** a set of heterogeneous machines processes sequentially each order;

- ▷ **Bundling configuration:** a set of dedicated machines processes a set of independent tasks in each the tasks must be bundled up to be delivered;
- ▷ **Two-stage flexible flowshop configuration:** the orders are processed in two stages, where each stage has a set of parallel machines and each order is processed sequentially over the two stages.

2. **Restrictions and constraints on order parameters:** a set of orders have many particularities that have to be considered in the integrated problem:

   - ▷ Different release dates;
   - ▷ Common due dates;
   - ▷ Delivery deadlines;
   - ▷ Time window or a delivery fixed time;
   - ▷ Setup times between orders;
   - ▷ Precedence constraints;
   - ▷ Preemption constraints;
   - ▷ Pick-up, process and deliver of finish orders by the same vehicle;
   - ▷ No-wait between machines in a flowshop configuration;
   - ▷ Machine maintenance constraints;
   - ▷ Delivery threshold times.

3. **Delivery characteristics:** delivery characteristics include specification of delivery methods and characteristics associated with the fleet.

   - ▷ **Type of vehicles:** Most of the authors consider a homogeneous fleet of vehicles.
     - Single delivery vehicle;

  - Multiple vehicles available;
  - Unlimited fleet of vehicles.

- ▷ **Capacity of vehicles:** the vehicles can be limited in terms of the number of orders or units of product that they can deliver:
  - only one order;
  - a limited number of orders;
  - infinite number of orders;
  - a finite number of units.
- ▷ **Delivery methods:** variants of the problem may justify different methods of delivery:
  - Individual and immediate delivery;
  - Batch delivery by direct shipping;
  - Batch delivery with routing;
  - Shipping with fixed delivery departure dates;
  - Splittable delivery.

4. **Number of clients:** there is one or more clients located at distinct coordinates:
   - ▷ Single client;
   - ▷ Multiple clients, identical orders;
   - ▷ Multiple clients, different orders.
5. **Objective function:** the objective function may consider different performance measures which determine the final goal. These performance measures may be used in an individual way or combined. The most frequently used are:
   - ▷ Client service level;

▷ Total cost;

▷ Total revenue;

The contributions described in the literature for variants of the integrated production and distribution scheduling problems are recent. Most of them assume particular constraints, which simplify the problems. In some cases, the authors assume that the processing is carried out on a single machine and all batches have the same dimension [2, 39]. In many cases, these constraints helped in solving the problems in polynomial time [39]. Most of the methods developed for integrated production scheduling and distribution problems are heuristic approaches [7, 9, 10, 11]. Wang et al. [10] describe a set of rules and heuristics for the combined processing and distribution mail problem. Geismar et al. [9] describe a two-phase heuristic based on genetic and memetic algorithms for the production and delivery of products with a short lifetime. Chang and Lee [7] propose three different heuristics for production scenarios with one or two machines in parallel, but consider only one or two clients. The contributions in the field of exact resolution of these problems are rare [2], and once again, most of the algorithms are based on assumptions which simplify its resolution, such as fixed sequences of customers and immediate individual deliveries. Many of the contribution referred to in the literature with respect to planning and integrated production optimization would refer to the last decade. Although there is a wide variety of problems to be addressed, they have particular constraints, simplifying their resolutions. Certain authors describe exact algorithms with constraints that make possible to solve these problems in polynomial time [40, 39].

According to Armstrong et al. [2], in make-to-order business processes, production is only initiated after customers orders are received. When the production and transportation facilities have limited capacities, the coordination of these operations becomes a challenge, especially when the product has a short lifetime. One of the products reported by these authors is an adhesive used in plywood panels, which has a seven-day lifetime. When this period ends the adhesive strength decreases abruptly, which makes it non-viable for delivery to the client. The cost of production is increased and there is an additional cost for the destruction of the material so it will not pollute the environment. The company is doubly penalized. Due to this nature of the product, it is not possible to keep any stock in either consumer or producer. According to the production facilities, there has to be a high level of coordination in the integration of schedules and the customers opportunity window. Mismanagement of this integration may lead to the products expiration before arriving to the customer or they may not be delivered in accordance with the customers needs. Armstrong et al. [2] consider in their study a production facility with just one truck and a fixed sequence of customers. They also assume that the product has a limited lifetime and production rate. The transportation time between customers is not neglected. When a truck arrives at a customer before the opportunity windows, idle time occurs and if it arrives late, the product is rejected.

In daily life, clients fixed sequences exist, for example, when the first customer to perform the request is the first to be served (FIFO - first in first out) or when a truck is associated with a particular route and satisfies the needs of customers in accordance with the following requests. The production must be delivered to the respective customer within the lifetime of the product. Given limited manufacturing resources such as transport, there may be a subset of the initial sequence of customers that will not receive the orders within the specified time window. This results in a new problem: choose a subset of customers to receive orders in time so that the total amount of load is maximized. Constraints related to the customer opportunity window, production capacity, transportation time and product validity should be taken into consideration. Production and Distribution Problems require simultaneous optimization of the sequence of production and transportation route. The zero inventory problems are not frequently discussed.

According to Chen and Vairaktarakis[40], nowadays there are many companies that produce and deliver the items directly to the consumer without maintaining any level of intermediate stock. These authors discuss the particular case of computer and catering companies. In the particular case of computer business systems, there are, for example, numerous possible configurations for a computer. The existence of stock becomes unsustainable even when using forecasts of possible configurations because the stock can easily become obsolete. Thus, the assembly and packaging operations may be performed only after the customers effective request. Such features can also be observed in catering services. In these cases, to maintain the food fresh, the preparation may only be made after customers orders are completely known. There cannot be any intermediate stock and the orders are fulfilled and delivered in a few hours. According to Chen and Vairaktarakis [40], both manufacturing and distribution operations are directly connected to each other.

In make-to-order requests, cost and quality service to customer are the main concerns of the decision maker. Due to lack of intermediate stock or finished products at any instant of time, the cost of ownership is considered negligible. The quality of service to the customer is measured in terms of lead time. The authors consider that for shorter lead times, the level of service quality is higher. However, small lead times require the use of more shipping resources which make the distribution cost higher. Thus, the main objective of the decision maker is to optimize the trade-off between the cost of distribution and the level of customer service. The close connection between production operations and distribution requires a very detailed coordination of schedules.

When a set of orders starts in processing facilities, it is necessary to deliver them directly to the customer. The problem is to find a schedule for the production and distribution such that the objective function takes into consideration both the level of customer service and the distribution costs.

The model studied by Chen and Vairaktarakis [40] integrates the production scheduling with distribution of completed requests. Although these problems have been extensively discussed in the literature, they are rarely analysed simultaneously. Other authors consider similar problems, but define their goal in customer satisfaction. They do not take into account the transportation costs and do not assume that the orders can be delivered instantly to the customer with no shipping delay.

The approach of Chen and Vairaktarakis[40] is innovative because they consider the decisions of distribution routes. The authors have considered multiple orders from different customers who are in different locations of a subjacent transportation network. As stated above, this model is composed by a part of processing and production, and another part of distribution. Different configurations were considered in the papers that refer to production: a single machine or similar machines in parallel. In the first case the machine processes all orders, while in the second a machine processes each order. In distribution problems, there are multiple homogeneous vehicles available, such that each vehicle will only be used at most once. Each vehicle has a limited capacity and is in a depot in an initial time, returning after completing the deliveries.

The authors consider several changes on the model presented which include machine settings on the facilities, number of clients involved and objective function in order to minimize the total cost of distribution and service. It was proved that for every kind of variation performed, an exact solution in polynomial time might be admissible. For intractable problems, heuristics have been proposed and their worst case performance has been analyzed.

Li et al. [39], unlike Armstrong et al. [2] and Chen and Vairaktarakis [40], studied the case where the number of vehicles was fixed (one vehicle), varying only the number of customers and orders. This vehicle may have limited or infinite capacity. The aim of the study was to determine the best sequence for processing orders in production facilities with scheduling deliveries. The goal is to minimize the total time between customers requests and the product delivery. The authors studied various types of models such as the single client and multiple clients. They developed a dynamic programming application to solve the general case of this problem, which is NP-complete. For a higher number of customers, it is required a higher computational complexity, although complexity is polynomial in the number of requests for a fixed number of customers. Research in supply chains intends to assist in developing strategies, but most of the literature focuses on stock control or lot sizing issues. In this context, Wang and Cheng [11] study scheduling problems that consider both the production and delivery of products. Also taking into account the availability of the machines that are part of the manufacturing centre. Only one vehicle is available for delivery in a fixed transportation time for a distribution centre. The main purpose of this study was to minimize the arrival time of the last batch to the distribution centre. The machines availability constraints are incorporated in the model.

Wang and Cheng [11] defined, like other authors, two different production configurations: production on a single machine and production on identical parallel machines. With two identical parallel machines, the authors assume that one machine is always available for production and the other has a similar behaviour to the case of a single machine. The authors consider reasonable to assume that the maintenance is done in a rotative way.

For this case study, the authors consider not only the jobs scheduling on the machine, but also the delivery schedule for transporting finished jobs to the distribution centre. The coordination between these two production stages is essential to achieve a global optimal solution. The authors state that the work should be processed as soon as possible. If there are batches for delivery, the vehicle should start shipping soon. The jobs in a batch are processed consecutively in the machine and the batch that becomes available sooner is the first to be delivered. The authors studied various scenarios and instances of the problem, and proposed an optimal algorithm and two heuristics.

The studies conducted by Chang and Lee [7] focus on problems that include two stages of scheduling, the production phase and then the delivery phase, in an integrated way. The products are delivered in batches and the transportation method is a busy and concurrently resource during delivery. These problems have a capacity constraint which is the total physical space occupied by the products that may be delivered in a single trip and each trip has an associated time. The objective is to minimize the time of delivery to the respective customer. Customers areas are considered when travel times between these clients are not significant compared to the time spent in the production system, or when the products are delivered to a distribution centre.

Three scenarios were discussed. The first features a product processed in a single machine and delivered by a single vehicle to a customer area. The second considers a product that can be processed in one of two machines and delivered by a single vehicle to an area for customers. The third considers a product processed in a single machine and delivered by a single vehicle for two areas of different customers.

Wang et al. in [10] addressed a more complex scenario based on the real case of postal services in the United States. The scenario presented is the processing of incoming mail in a processing and distribution centre to match with a schedule delivery. Mail arrives at the centre locally or remotely and then follows the schedule entry. For each destination, there are scheduled transports with limited capacities. The objective of the problem is to determine the sequence in which the incoming mail should be processed so that the total unused delivery capacity is minimal.

In their study, Wang et al. [10] started by considering the processing and distribution centre as a single machine. The authors formulated the case as an Integer Programming problem, whose solution could not be obtained due to the large number of integer variables. The number of origins and destinations was normally 60 and the number of mailboxes typically 70. Since a direct solution was unpractical, the authors made progress in developing of shipping rules and heuristics for solving the problem. The first shipping rule considers processing mail whose origin has higher proportion and lower processing time for the faster delivery destination. The second rule considers the first mailbox with the highest capacity that remains. These rules focus on the short term. In order to provide a better solution, Wang et al. [10] planned sequencing over the entire time horizon and develop two heuristics, being one an approximation of Linear Programming and the other a modification of the greedy algorithm.

2.4 Production, inventory, distribution and routing

The integrated scheduling and distribution problem includes problems related to production, inventory, distribution and routing, which have had special attention over the last few years. They are known in the literature as Production, Inventory, Distribution and Routing Problem (PIDRP) and integrate the main characteristics of three difficult problems mentioned in the literature, the Production and Distribution Problem (PDP), the Inventory Routing Problem (IRP) and Vehicle Routing Problem (VRP). The PIDRP has several variants that are explored by different authors. Some authors give more importance to production decisions, while others emphasize distribution and routing decisions. On the other hand, there are authors that give particular attention to inventory management decisions. All these authors try to integrate these problems and to see them from a global perspective, and they selected different approaches to solve them according to the complexity of the variant.

Lei et al. [41] consider the existence of several production facilities which manufacture a single product which is distributed to several customers. Each customer has a deterministic demand that must be fulfilled over the planning horizon and a maximum ending inventory capacity and safety stock. A facility also has a maximum ending inventory capacity and safety stock and has a limited production capacity. A fleet of heterogeneous vehicles that has a particular capacity, speed and availability is associated to each facility. The aim of the authors is to minimize the costs of the integrated operations.

To solve the PIDRP with a single product, they propose an exact integer programming model which deals with a large number of variables related to distribution and routing decisions, inventory and production schedules. They highlight that the computational time to exactly solve this problem to optimality can often become excessive. For that reason, the authors presented a two-phase decomposition heuristic to solve the problem. During phase I, the original MIP model is solved without the routing constraints that are limited to direct shipments. This phase determines the quantity to be manufactured, inventoried and carried out by each vehicle within each period. Phase

II uses a Load Consolidation (LC) algorithm to determine the routing decision. The LC algorithm removes from the Phase I the Less-than Transporter Load (LTL) assignments and consolidates these assignments considering the routing constraints. The authors compare the original MIP model with the proposed heuristic. Lastly, they add some constraints to the problem in order to solve real instances with 12 periods, 2 facilities, 13 customers and 3 heterogeneous vehicles.

In [42, 43, 44], Boudia et al. and Boudia and Prins emphasize the importance of integrating production and distribution decisions. Boudia et al. [42] use a weighted and undirected graph to define the problem. They consider that node zero represents a single plant that produces a single product over a planning horizon. The plant possesses a limited fleet of homogeneous vehicles and each vehicle has a load limit. The plant has a periodic production capacity and inventory limit. The remaining nodes represent the clients who have a varying demand per period and a limited storage capacity. A client can be served at most once per period and each vehicle can only make one route per period. The main objective is to minimize the cost of the integrated system determining, for each period, the quantity to be manufactured and the quantity delivered to each client, taking into account the routing and inventory decisions. They propose an integer linear model which is not solved to optimality for large instances due to its inherent complexity and present a Greedy Randomized Adaptive Search Procedure (GRASP) instead of the classical two-phase method. This heuristic is divided into two phases. First, a constructive phase is used to determine the quantity delivered to clients at a specific period, and then the local search phase aims to improve the solution exploiting the defined neighborhoods regarding constraints related to production, routing and inventory management. To test the approach, the authors use 90 instances generated randomly with 50, 100 and 200 clients and 20 periods.

The problem addressed by Boudia et al. [43], in 2008, presents the same details as the problem previously described in 2007 [42]. Boudia et al. [43] address greedy heuristics with the goal to minimize the cost associated to the described PIDRP problem.

The authors propose two greedy heuristics followed by two local search methods. The first heuristic is divided into two sequential phases (uncoupled heuristic), where production decisions such as inventories and quantity produced at the facility are prioritized. The distribution planning is subsequently done according to the production plan. The demand for the period is assuredly fulfilled, and local search is applied in an attempt to improve trips, trying to anticipate requests according to vehicle and customers inventory capacity. The second heuristic determines the production and distribution planning simultaneously. This coupled heuristic is performed in three phases before the application of local search method. In the first phase, the amount delivered for each period is determined through a preliminary production plan. The second phase creates a distribution plan, and the last one determines the definitive production dates. The authors propose two different local search methods after completing the 3 phases, creating two different coupled heuristics.

During this newest research work, the authors highlight that the heuristics proposed can be applied to multiple products since different products can be mixed in the distribution vehicles. The authors test the heuristics with instances generated in a random way with up to 20 periods and 200 clients. The results present substantial savings according to the authors.

In [44], Boudia and Prins address a problem that is similar to the one described in [42, 43], regarding its main characteristics such as facility, vehicle and clients features. They assume that inventory cost at the clients is supported by them, being ignored along the problem. The authors propose an innovative form of meta-heuristic which simultaneously takes into account production and distribution decisions. The aim of this method is to minimize the cost associated to the Integrated Production and Distribution Problem (IPDP).

The authors propose a memetic algorithm with population management, where an initial population is generated. In each iteration, two parents are selected and a crossover operation is applied. Some elements of the population are replaced by the new offspring. Mutations are replaced by diversity control based on a distance measure in a solution space.

Boudia and Prins [44] compare the obtained results with other approaches previously described in their work such as two-phase heuristic and GRASP, and emphasize the substantial savings observed. The authors use 90 randomly generated instances with 50, 100 and 200 clients and 20 periods.

Solyali and Süral [45] study a common problem in industry, the vendor managed systems. In this type of problems, there is a supplier that has to distribute the goods over multiple retailers and control the retailers inventory. Indeed, in these managed systems the supplier is responsible for not allowing the retailer inventory to decrease from an established value. This distribution may occur in different periods and the inventory at a supplier may not go beyond a pre-established minimum. Indeed, the goal of the entire system is to minimize the inventory and routing costs. In their approach, the authors considered a single supplier that must deliver a single product to multiple retailers, considering the intended demand of each. Indeed, the distribution is accomplished by using a homogeneous fleet of vehicles over a finite planning horizon. The level of inventory is taken into account in the distribution decisions, since inventory level has a maximum limit and a pre-set lower bound. Thus, upon delivery, the inventory of the retailer is fully reestablished. This type of inventory policy is commonly denoted as order-up-to level policy.

In their study, the authors developed a mathematical programming based approach in order to solve an inventory routing problem with order-up-to level policy were the goal is to determine the retailers to visit and the corresponding demand, and the distribution routes for each period in order to minimize the routing and inventory costs. To solve this problem, Solyali and Süral [45] proposed a Lagrangian relaxation in which the replenishment and the distribution planning problems are separated from each other.

To test their formulation, the authors used instances from the literature and their computational results shown that their algorithm produced good feasible solutions, considering that in the large instances Mixed Integer Programming (MIP) solvers could not find feasible solutions to the larger instances.

Bard and Nananukul [46] studied a PIDRP that integrates production and distribution decisions. The authors consider a single facility with limited capacity and a finite inventory capacity. The main goal is to serve a set of clients with time varying demand through a finite and discrete planning horizon with a fleet of homogeneous vehicles that also have limitations in terms of capacity. A client demand must be fulfilled from the corresponding period of production or from inventory held at the client. An order can be anticipated, however split deliveries are not allowed. The minimization of the total cost associated to the integrated system is the aim.

The authors present a hybrid methodology which combines exact and heuristic approaches throughout a branch-and-price algorithm in order to solve the MIP model presented in their study. They propose an exact allocation model and highlight that the initial attempts to solve the model exactly were not encouraging due to the high computational complexity of the model. They also propose a branch-and-price algorithm, which also had a high computational complexity. In order to overcome this computational difficulty, Bard and Nananukul [46] propose a methodology based on Tabu Search. They developed a column generation heuristic and a rounding heuristic to update the upper limit of the branch and bound. They emphasize the significant reduction of processing times obtained with the branch-and-price heuristic that starts with tabu search. To evaluate the performance of the proposed models, the authors use a set of instances from the literature adapted from Boudia et al. [42] with 10, 20, 30, 40 and 50 clients and 2, 4, 6, and 8 periods.

Bard and Nananukul [47] explored an IRP in an attempt to analyze and integrate a PIDRP. The authors studied the same problem presented in [46] and proposed a column generation approach for the PIDRP and three two-step heuristics in order to solve the IRP. For the PIDRP, first they determine an estimated distribution plan and then the routing plan through a VRP tabu search heuristic. The main objective is to minimize the cost associated to the inventory, production, distribution and routing problems. The computational results show that it was possible to solve instances up to 50 clients and 8 periods in an hour. The authors note that this level of performance could not be reached through an exact branch-and-price algorithm on CPLEX.

A similar problem proposed by the same authors is addressed in [48]. Bard and Nananukul solve a PIDRP using a two-phase method with a reactive tabu search algorithm. Firstly, the initial solution is obtained through the allocation model where they assume these values for the demand of independent routing problems. In order to find solutions, a subroutine based on tabu search is used. Secondly, a neighborhood search is performed in order to improve the decisions made in the previous phase. To validate the performance of the developed approach the authors use a set of 90 benchmark instances with up to 20 periods and 200 clients.

Ruokokoski et al. [49] addressed the Production-Routing Problem (PDP) in which the production and distribution decisions are considered in a simultaneous way. The coordination of these two sub-problems may lead to better results in terms of cost savings.

The authors considered an uncapacitated depot from which a set of routes for a single uncapacitated vehicle were obtained in order to accomplish the replenishment schedules for multiple customers. The goal was to minimize the total cost of distribution, setups, and inventories by fulfilling the demand of the multiple customers in a finite horizon. The inventories may be kept at the supplier and/or at customers. However, there is a cost associated to it. Ruokokoski et al. [49] also considered the delivery of a single product.

To solve the PDP, the authors present strong formulations, which include a basic mixed integer linear programming formulation and several strong reformulations, and a branch-and-cut algorithm to solve them. The reformulations that strengthen the basic MILP are two families of valid inequalities that were adapted from the literature, 2-matching and generalized comb inequalities. Despite the exact approach, the authors also present a new heuristic separation algorithm for the generalized comb inequalities, and adapted a heuristic algorithm to find high quality integer feasible solutions. To show the performance and the quality of the developed formulations the authors solved several instances to optimality with, for example, 8 time periods and 80 customers; 15 time periods and 40 customers.

Armentano et al. [50] proposed two tabu search variants to solve the IPDP. The authors consider a single facility with capacity constraints that produces multiple items during a finite planning horizon. The items are distributed by a fleet of homogeneous vehicles with capacity limitations. The main objective is to minimize the fixed and variable costs related to production, inventory and distribution.

The authors use a tabu search methodology with a memory-based local search strategy that could overcome local optima by prohibiting certain moves in the solution space. Armentano et al. [50] take advantage of short and long-term memories in the two approaches. In the short-term memory, an attribute list of recently explored solutions is created to prevent these solutions from being revisited. The long-term memory contains a selective history and attributes of solutions. During their study, the authors emphasize the importance of introducing some infeasible solutions in the tabu search and path relinking methods. Armentano et al. [50] generated some instances with up to 10 different items to evaluate the heuristics performance, and used instances from the literature with single item proposed by Boudia et al. [42].

Some authors emphasize the importance of inventory management policies, relaxing the constraints related to the production and distribution, such as production capacity, the use of a single vehicle, among others. Archetti et al. [51] studied two types of replenishment policies. The Order-Up (OU) to level and the Maximum Level (ML) policies. The OU policy occurs when the amount delivered to each client or retailer is such that the level of its inventory reaches the maximum level. On the other hand, the maximum level (ML) policy occurs when the amount shipped to each retailer is such that the inventory is not higher than the maximum level. The authors developed a hybrid heuristic and an exact model that considers only one vehicle and a single retailer, and compare the obtained results to the ML policy. The main goal is to determine the number of items that should be produced in each period, and to create a routing and distribution plan which minimizes the total cost and guarantees that there are no stock outs.

Nananukul [52] uses clustering in his approach to solve a PIDRP. The main objective is to minimize the operating cost associated to the production, inventory and delivery decisions using clusters of clients. The author highlights that the PIDRP computational complexity limits the number of clients considered for different approaches. Using clustering techniques it is possible to group sets of clients that have similar features. The approach carried out by Nananukul considers a single facility that produces a single product. Variable customer demand has to be met through periodic production or inventory at the factory. A fleet of homogeneous vehicles performs the routes defined in the routing and distribution plan. Each cluster of clients is served by a single vehicle.

The author proposes a clustering model using a two-phase reactive tabu search-based algorithm. The first phase determines an initial solution and the second phase tries to improve the current solution throughout a neighborhood search. He also uses different techniques to group the clients instead of processing the original data points. Nananukul tests the performance of the algorithm in instances with up to 200 clients and 20 periods.

Absi et al. [53] address an integrated optimization of production, distribution and inventory problem. They consider that a single facility produces a single item and fulfills the time varying demand associated to the different retailers during a planning horizon. The inventory management is performed through a ML policy and the distribution is made by a homogeneous fleet of capacitated vehicles.

The authors propose a heuristic for the PRP with a ML policy. This iterative approach considers that production planning and routing sub-problems are sequentially solved. During the first phase, the lot-sizing phase, the retailers who need to be served on each period are determined. The second phase, the routing phase, considers routing and distribution decisions where a Traveling Salesman Problem (TSP) is solved for each vehicle. At the end of the iterative process, a diversification mechanism is performed in order to present local optima convergence. The authors emphasized that their best heuristic outperforms existing approaches.

Adulyasak et al. [54] considered a stochastic production routing problem in which the demand of the customers is uncertain. The authors studied a major issue in supply chain management which is getting crucial information for the decision making only in an approximation mode through forecasts. The resolution of deterministic models in these situation may lead to wrong decisions which may increase significantly the overall costs.

In their approach, the authors considered the production and distribution of a single product in a discrete and finite time horizon where the distribution network includes a limited production capacity plan and multiple customers. The routes are performed using capacitated vehicles. Backlogging is not allowed, however there is the possibility of not meeting the clients demand in each period. In this case, there is a unit penalty cost. The goal of the problem was to minimize the production costs, which includes fixed setup and unit costs, holding costs associated to the inventory both at the plant and at the customers, the cost of unmet demand, and the routing costs for the distribution of the products.

To solve the problem, Adulyasak et al. [54] proposed a two-stage and a multistage decision process. In the two-stage process the authors initially define the production setups and the customer visit schedules. Then, in the second stage, the production and delivery quantities are calculated. A branch-and-cut algorithm is used to solve the formulation of the problem. For larger instances, due to the size of the problem, the authors proposed a Benders decomposition approach which is composed by a single branch-and-bound tree and enhanced using lower-bound lifting inequalities, scenario group cuts, and Pareto-optimal cuts. In the multistage decision process, the decisions defined for a given stage did not consider the demand of future periods, which was not known. For this multistage process, the authors developed a rollout heuristic and they obtained good feasible solutions for the problem.

The main characteristics of the presented models by the different authors aforementioned are summarized in Table 2.2, in order to provide an easier identification of the differences between the various approaches.

Table 2.2: Synthesis of the state of the art to the production, inventory, distribution and routing variant

Features	Authors	[41] Lei et al. 2006	[42, 43] Boudia et al. 2007, 2008	[44] Boudia and Prins 2009	[45] Solyali and Süral 2009	[48, 47] Bard and Nananukul 2009	[46] Bard and Nananukul 2010	[49] Ruokokoski et al. 2010	[50] Armentato et al. 2011	[51] Archetti et al. 2011	[52] Nananukul et al. 2013	[53] Absi et al. 2014	[54] Adulyasak et al. 2015
Approach	Exact	✓			✓			✓		✓			✓
	Heuristic	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Production	Multiple facilities	✓											
	Facility capacity[1]	✓	✓	✓		✓	✓		✓		✓	✓	✓
	Multiple product								✓				
	Multiple clients	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Inventory	Facility capacity[1]	✓	✓	✓		✓	✓		✓	✓	✓	✓	✓
	Client capacity[1]	✓	✓		✓	✓	✓		✓	✓	✓	✓	✓
Distribution	Multiple vehicles	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓
	Homogeneous Fleet		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Load capacity	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
	Time horizon	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Time windows												
	Multiple routes	✓											

[1] The absence of the checkmark represents an unlimited capacity

Chapter 3

The multi-trip location routing problem: integer programming models[1]

Outline

[1] The results of this chapter were published in:

[55] R. Macedo, B. Ramos, C. Alves, J. V. de Carvalho, S. Hanafi, and N. Mladenović, "Integer Programming Based Approaches for Multi-Trip Location Routing." Springer International Publishing, 2016, pp. 79–90

[56] B. Ramos, C. Alves, and J. Valério de Carvalho, "Column Generation Based Approaches for Combined Routing and Scheduling," *Electronic Notes in Discrete Mathematics*, vol. 64, pp. 155–164, 2018

3.1 Problem description

The Multi-trip Location Routing Problem (MLRP) is a management science problem that occurs typically in the logistics and transportation field. The MLRP is characterized as an integrated problem which combines two important and difficult optimization problems: the Facility Location Problem (FLP) and the Multi-trip Vehicle Routing Problem (MVRP). In the FLP, one has to determine the set of facilities that can be used to serve the clients. In order to fulfill the clients needs, a set of routes is generated by solving the MVRP. This problem has a specificity since it can associate more than one single-trip to a vehicle during the planning horizon. The integration of these two problems aims to minimize the costs of the entire system. In this integrated solution both the FLP and the MVRP are solved simultaneously. This leads to better solutions than solving them in the independent way. However, this method has a drawback, *i.e.*, it increases the problem complexity since now variables related to the global system are considered.

The MLRP consists in the selection of the depots that should be opened and the single-trips and multi-trips that should be performed to serve the set of clients at minimum cost. The multi-trip variant considers the possibility of a vehicle performing more than one single-trip during the planning period. Hence, it is typically applied to cases in which the routes are performed within a small geographic area, and involve, for example, the transportation of perishable goods, which must be delivered in a short period of time. As a consequence, the inherent complexity of the problem increases since now it is necessary to determine the route that should be assigned to a vehicle.

Definition

A route r is composed by an ordered set of clients to be served. A route is a broader term that may be a single-trip (r_2 in Figure 3.1) or a multi-trip (r_1 in Figure 3.1). Each route is associated to a vehicle that serves a depot. For the sake of clarity, a set of single-trips is denoted as multi-trip. Figure 3.1 presents an example of possible routes that serve seven different clients. The first

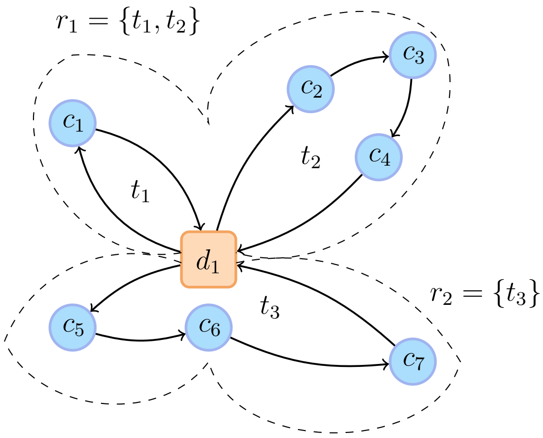

Figure 3.1: Example of possible routes

single-trip (t_1) visits only the client c_1, the second (t_2) satisfies clients c_2, c_3 and c_4 and the last (t_3) serves the other three clients (c_5, c_6, and c_7). As presented in Figure 3.1, the route r_1 satisfies the client c_1, returns to the depot d_1 and then serves the clients contained in the single-trip t_2. The route r_1 uses a multi-trip to serve its assigned clients, while the route r_2 is composed by a single trip.

The problem has some details that are important to describe. For that reason, some peculiarities associated to the system are presented below:

- Each customer can only be visited once;
- Each client is associated with a particular depot that will fulfill his demand;
- Each route must start and end at the same depot regardless of the number and order of visited clients;
- The load of each single-trip must be less than or equal to the capacity of the vehicle;
- A vehicle can perform several single-trips during a planning horizon but the total load of the multi-trip associated to one vehicle cannot exceed the capacity of the depot;

- ▷ A vehicle cannot work more than a workday (the length of the planning horizon); and
- ▷ The demands of the customers associated to one facility cannot exceed its capacity.

The length of the route associated to a vehicle must be less than or equal to the workday W, which represents the maximum length that the vehicle may travel. It must start and end at the same depot regardless of the number and the order of the visited clients. Each vehicle v, from a fleet of homogeneous vehicles V, may perform several single-trips as long as the load of each single-trip does not surpass the capacity Q of the vehicle. However, the total load of the routes associated to all vehicles of a depot d cannot exceed the capacity L_d of that depot. Each customer $i \in N$, $N = 1, \ldots, n$, may only be visited once, being associated to one route that must fulfill his total demand b_i. All client orders must be satisfied.

The cost of the solution considers the fixed costs C_f^d, $d \in D$ if the depot d is open, and the cost of the routes C_r, $r \in R_d$, where R_d is the set of routes associated to the depot d. The cost C_r of each route includes the cost C_v of using a vehicle, and depends on the traveled distance. It is assumed that a distance unit (*e.g.*, one mile) has an associated cost of one monetary unit (*e.g.*, one euro). The goal of the MLRP is to minimize the total costs associated to the entire system.

The MLRP has been previously addressed in [28, 29, 30, 31, 32]. In [28], Lin *et al.* explored the problem using heuristics and branch-and-bound. In [29], Lin and Kwok addressed a multi-objective case combining cost minimization with the minimization of the imbalance among vehicles. In [30], Akca et al. proposed a compact three-index commodity flow formulation, and a branch-and-price algorithm for a column generation reformulation of the problem.

3.2 A three-index commodity flow model

The three-index commodity flow model is a graph-based model addressed by Akca et al. in [30] and Macedo et a. in [55], which considers the variables of the problem explicitly. For that reason, the model can handle a large number of variables. On the other hand, the model presents a simple and well defined structure that allows its decomposition into different less complex problems.

This model is represented by a graph G with a set of nodes associated to the depots and to the clients, and a set of arcs between each pair depot-client and client-client, such that $G = (N \cup D, A)$, with $A = (D \times N) \cup (N \times N) \cup (N \times D)$. The complete set of vehicles is denoted by H, with H_d being the subset of vehicles assigned to a depot d. The travel time between nodes i and j, with $(i, j) \in A$ is denoted by t_{ij}, and the cost associated to a unit of time is denoted by C^o.

The three-index commodity flow model has variables related to the opening of the depots and to the vehicles usage and operation. The binary variables λ_d, $d \in D$, state if a depot is selected. The usage of a vehicle h is represented by the binary variables v_h, $h \in H$. If a vehicle h goes through an arc $(i, k) \in A$, then the corresponding variable x_{ikh} will take the value 1, and 0 otherwise. The load the vehicle h carries through (i, k) is denoted by y_{ikh}.

Figure 3.2 presents an example of a solution for the three-index commodity flow model through a graph that includes a set of six clients (denoted by $c_1, \ldots, c_6$) and a set of three depots (d_1, d_2 and d_3). All arcs between the nodes (which may be depots or clients) are represented by a dotted line. The arcs which have an associated flow are depicted by oriented lines connecting a depot with a client, a client to another client, or a client to the same depot from which the route has started.

The presented solution uses three homogeneous vehicles (v_1, v_2 and v_3) in order to fulfill the demand of the six different clients. As depicted in Figure 3.2, only two of the three available depots are open (λ_1 and λ_2). The first depot d_1 serves a set of three clients (c_4, c_1 and c_3) through the vehicle v_1 and a second vehicle v_2 serves just the customer c_6. Clients c_2 and c_5

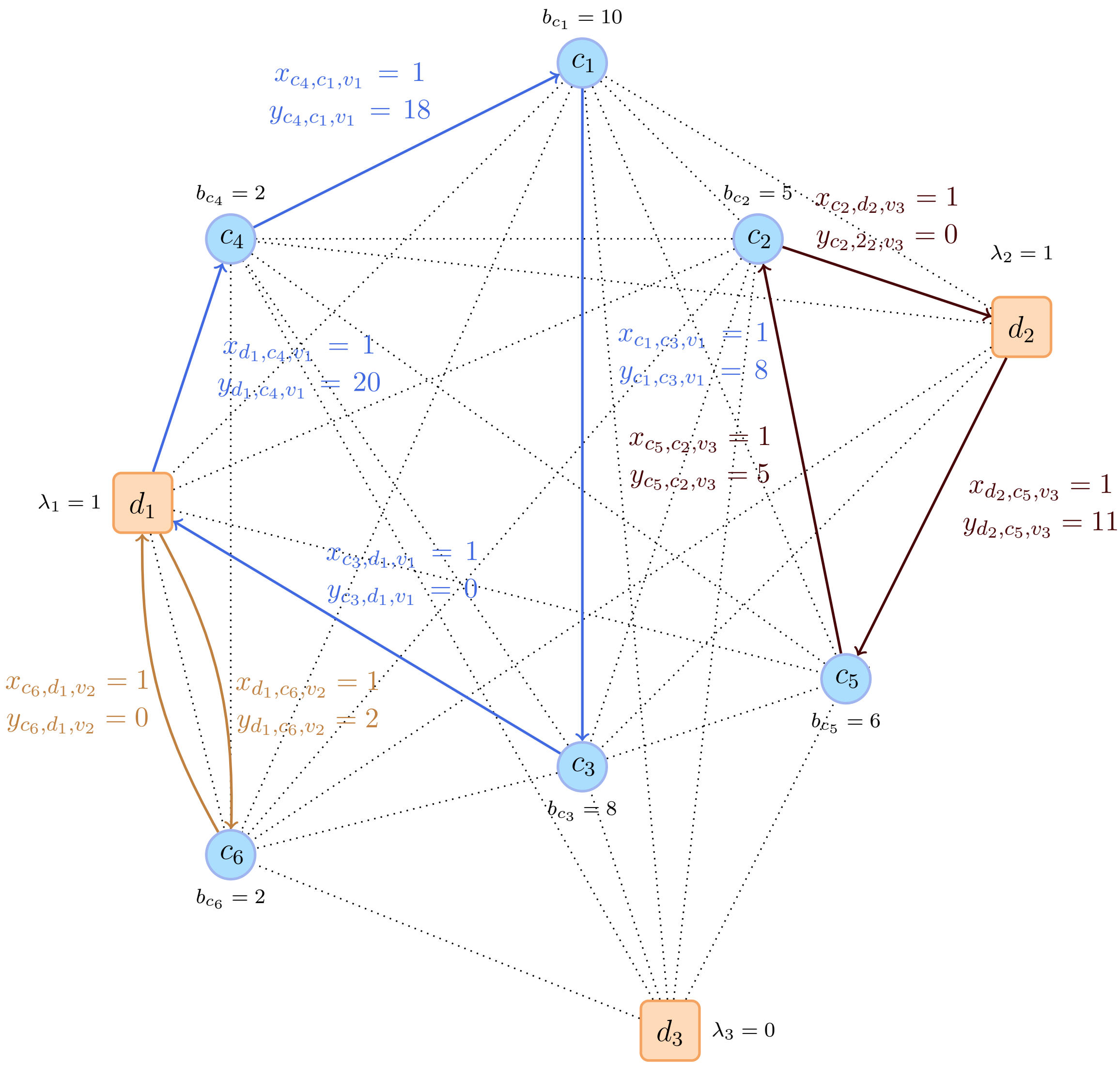

Figure 3.2: Solution example of three-index commodity flow model

have their demand fulfilled by vehicle v_3 being their needs satisfied by the depot d_2. The flow on the arcs is identified by variables y_{ikh}, and it only occurs when an arc is selected, *i.e.*, $x_{ikh} = 1$. For instance, one may identify a flow of eight units between the client c_1 and the client c_3 for the vehicle v_1 (y_{c_1,c_3,v_1}), which confirms that the variable x_{c_1,c_3,v_1} is activated.

For an easier identification of the parameters and decision variables used in the model under analysis, they are summarized next.

Parameters

$C_f^d =$ fixed cost associated to opening a depot $d, \forall d \in D$

$C^o =$ cost per travel time unit associated to operating a vehicle

$C_v =$ cost associated to the use of a vehicle $v, \forall v \in H$

$b_i =$ demand associated to a client $i, \forall i \in N$

$L_d =$ capacity associated to the depot $d, \forall d \in D$

$Q =$ capacity associated to the vehicle

$W =$ length of the plan horizon

$t_{ik} =$ the travel time between i and $k, \forall (i,k) \in A$

Decision Variables

$y_{ikh} =$ load that the vehicle h carries through the arc $(i,k), \forall d \in D\ and\ \forall (i,k) \in A,$

$$x_{ikh} = \begin{cases} 1 \text{ if vehicle } h \text{ goes through the arc } (i,k), \forall h \in H\ and\ \forall (i,k) \in A, \\ 0 \text{ otherwise} \end{cases}$$

$$\lambda_d = \begin{cases} 1 \text{ if the depot } d \text{ is selected}, \forall d \in D, \\ 0 \text{ otherwise} \end{cases}$$

$$v_h = \begin{cases} 1 \text{ if the vehicle } h \text{ is used }, \forall h \in H, \\ 0 \text{ otherwise} \end{cases}$$

The three-index commodity flow model has the main goal of minimizing the cost associated to the entire system (FLP and MVRP), which includes the fixed cost of opening a depot, the fixed cost associated to the vehicle usage and a variable cost associated to the routes performed by the vehicles. The model is defined from Equation (3.1) to (3.12).

Minimize

$$\sum_{d \in D} C_f^d \lambda_d + C_v \sum_{h \in H} v_h + C^o \sum_{h \in H} \sum_{(i,k) \in A} t_{ik} x_{ikh} \tag{3.1}$$

Subject to:

$$\sum_{h \in H} \sum_{k \in (N \cup D)} x_{ikh} = 1, \quad \forall i \in N, \tag{3.2}$$

$$\sum_{k \in (N \cup D)} x_{ikh} - \sum_{k \in (N \cup D)} x_{kih} = 0, \quad \forall i \in N \cup D, \; \forall h \in H, \tag{3.3}$$

$$\sum_{h \in H_d} \sum_{k \in N} y_{dkh} \leq L_d \lambda_d, \quad \forall d \in D, \tag{3.4}$$

$$y_{ikh} \leq Q x_{ikh}, \; \forall (i,k) \in A, \quad \forall h \in H, \tag{3.5}$$

$$\sum_{k \in N} y_{ikh} - \sum_{k \in N} y_{kih} + b_i \sum_{k \in (N \cup D)} x_{ikh} = 0, \quad \forall i \in N, \forall h \in H, \tag{3.6}$$

$$\sum_{(i,k) \in A} t_{ik} x_{ikh} \leq W v_h, \quad \forall h \in H, \tag{3.7}$$

$$x_{dkh} = 0, \quad \forall d \in D, \forall k \in (N \cup D), \forall h \in H_t, \forall t \in D \backslash \{d\}, \tag{3.8}$$

$$x_{ikh} \in \{0,1\}, \quad \forall (i,k) \in A, \forall h \in H, \tag{3.9}$$

$$y_{ikh} \geq 0, \quad \forall (i,k) \in A, \forall h \in H, \tag{3.10}$$

$$\lambda_d \in \{0,1\}, \quad \forall d \in D, \tag{3.11}$$

$$v_h \in \{0,1\}, \quad \forall h \in H. \tag{3.12}$$

The mandatory visit to every client is represented by constraints (3.2) and (3.3). Constraints (3.2) ensures that a vehicle reaches and leaves a node exactly the same number of times, while constraints (3.3) guarantees that a client cannot be visited by more than one vehicle. The capacity constraints of the depot and vehicles are expressed through constraints (3.4) and (3.5)-(3.6), respectively. Constraints (3.6) also allow for the conservation flow at each node. Constraints (3.7) forbid a vehicle to travel more than W units of time, while constraints (3.8) force a vehicle to travel only through the arcs associated to its depot. As mentioned above, the objective function (3.1) denotes the objective of minimizing the total cost.

To clarify the three-index commodity flow model, an example of the Mixed Integer Programming (MIP) table for this model is depicted in Table 3.1. In the table, all constraints and a few example columns are presented. For the sake of clarity, the constraints are grouped according to the order of appearance in the model (Constraints (3.1)-(3.7)).

Table 3.1: MIP structure for three-index commodity flow model

		λ_1	λ_2	...	λ_d	x_{d_1,c_1,v_1}	x_{c_1,d_1,v_1}	x_{c_1,c_2,v_1}	x_{c_2,c_1,v_1}	x_{d_2,c_1,v_2}	x_{c_1,d_2,v_2}	...	$x_{(i,j)\in A,v_h}$	y_{d_1,c_1,v_1}	y_{c_1,d_1,v_1}	y_{c_1,c_2,v_1}	y_{c_2,c_1,v_1}	y_{d_2,c_1,v_2}	y_{c_1,d_2,v_2}	...	$y_{(i,j)\in A,v_h}$	v_1	v_2	...	v_h		
c_1		0	0	...	0	0	1	1	0	0	1	...	...	0	0	0	0	0	0	...	...	0	0	...	0	=	1
c_2		0	0	...	0	0	0	0	1	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	=	1
...		...	...	...	0	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	=	⋮
c_n		0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	=	1
c_1	v_1	0	0	...	0	-1	1	1	-1	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	=	0
	v_2	0	0	...	0	0	0	0	0	-1	1	...	...	0	0	0	0	0	0	...	...	0	0	...	0	=	0
	...	...	...	...	0	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	=	⋮
	v_h	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	=	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	=	⋮
c_n	v_1	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	=	0
	v_2	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	=	0
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	=	⋮
	v_h	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	=	0
λ_1	v_1	0	0	...	0	1	-1	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	=	0
	v_2	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	=	0
	...	...	...	...	0	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	=	⋮
	v_h	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	=	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	=	⋮
λ_d	v_1	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	=	0
	v_2	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	=	0
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	=	⋮
	v_h	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	=	0

(continues on next page)

Table 3.1: MIP structure for three-index commodity flow model (continued)

		λ_1	λ_2	...	λ_d	x_{d_1,c_1,v_1}	x_{c_1,d_1,v_1}	x_{c_1,c_2,v_1}	x_{c_2,c_1,v_1}	x_{d_2,c_1,v_2}	x_{c_1,d_2,v_2}	...	$x_{(i,j)\in A,v_h}$	y_{d_1,c_1,v_1}	y_{c_1,d_1,v_1}	y_{c_1,c_2,v_1}	y_{c_2,c_1,v_1}	y_{d_2,c_1,v_2}	y_{c_1,d_2,v_2}	...	$y_{(i,j)\in A,v_h}$	v_1	v_2	...	v_h		
λ_1		$-L_{d_1}$	0	...	0	0	0	0	0	0	0	...	...	1	0	0	0	0	0	...	...	0	0	...	0	$\leq$	0
λ_2		0	$-L_{d_2}$	...	0	0	0	0	0	0	0	...	...	0	0	0	0	1	0	...	...	0	0	...	0	$\leq$	0
...		0	0	...	0	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	$\leq$	$\vdots$
λ_d		0	0	...	$-L_{d_d}$	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	$\leq$	0
(λ_1, c_1)	v_1	0	0	...	0	$-Q$	0	0	0	0	0	...	...	1	0	0	0	0	0	...	...	0	0	...	0	$\leq$	0
	v_2	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	$\leq$	0
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	$\leq$	$\vdots$
	v_h	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	$\leq$	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	$\leq$	$\vdots$
(λ_d, c_n)	v_1	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	$\leq$	0
	v_2	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	$\leq$	0
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	$\leq$	$\vdots$
	v_h	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	$\leq$	0
(c_1, λ_1)	v_1	0	0	...	0	0	$-Q$	0	0	0	0	...	...	0	1	0	0	0	0	...	...	0	0	...	0	$\leq$	0
	v_2	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	$\leq$	0
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	$\leq$	$\vdots$
	v_h	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	$\leq$	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	$\leq$	$\vdots$
(c_n, λ_d)	v_1	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	$\leq$	0
	v_2	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	$\leq$	0
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	$\leq$	$\vdots$
	v_h	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	$\leq$	0

(continues on next page)

Table 3.1: MIP structure for three-index commodity flow model (continued)

		λ_1	λ_2	...	λ_d	x_{d_1,c_1,v_1}	x_{c_1,d_1,v_1}	x_{c_1,c_2,v_1}	x_{c_2,c_1,v_1}	x_{d_2,c_1,v_2}	x_{c_1,d_2,v_2}	...	$x_{(i,j)\in A,v_h}$	y_{d_1,c_1,v_1}	y_{c_1,d_1,v_1}	y_{c_1,c_2,v_1}	y_{c_2,c_1,v_1}	y_{d_2,c_1,v_2}	y_{c_1,d_2,v_2}	...	$y_{(i,j)\in A,v_h}$	v_1	v_2	...	v_h		
(c_1,c_2)	v_1	0	0	...	0	0	0	$-Q$	0	0	0	...	...	0	0	1	0	0	0	...	...	0	0	...	0	$\leq$	0
	v_2	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	$\leq$	0
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	$\leq$	$\vdots$
	v_h	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	$\leq$	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	$\leq$	$\vdots$
(c_n,c_n)	v_1	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	$\leq$	0
	v_2	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	$\leq$	0
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	$\leq$	$\vdots$
	v_h	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	$\leq$	0
c_1	v_1	0	0	...	0	0	b_{c_1}	b_{c_1}	0	0	0	...	...	-1	1	1	-1	0	0	...	...	0	0	...	0	=	0
	v_2	0	0	...	0	0	0	0	0	0	b_{c_1}	...	...	0	0	0	0	-1	1	...	...	0	0	...	0	=	0
	...	...	...	...	0	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	=	$\vdots$
	v_h	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	=	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	=	$\vdots$
c_n	v_1	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	=	0
	v_2	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	=	0
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	=	$\vdots$
	v_h	0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	0	=	0
v_1		0	0	...	0	t_{ik}	t_{ik}	t_{ik}	t_{ik}	0	0	...	...	0	0	0	0	0	0	...	...	$-W$	0	...	0	$\leq$	0
v_2		0	0	...	0	0	0	0	0	t_{ik}	t_{ik}	...	...	0	0	0	0	0	0	...	...	0	$-W$	...	0	$\leq$	0
...		...	...	...	0	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	$\leq$	$\vdots$
v_h		0	0	...	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	...	...	0	0	...	$-W$	$\leq$	0

3.3 A column generation model

The Column Generation (CG) process is typically used to solve large-scale problems or to improve their efficiency for the smallest instances. Many linear problems have a huge number of variables, and considering them all in an explicit way may be computationally infeasible. In the optimal solution, a large number of variables are equal to zero since most of them are non-basic variables. Therefore, in theory, one does not need to consider all the variables, but simply a subset of them when solving the problems. The main idea of the column generation method is to start with a reduced set of variables and columns to generate only new columns that have the potential to improve the value of the objective function. This is possible by finding variables with negative reduced cost when dealing with minimization problems.

The CG approach, presented by Ramos et al. in [56], is based on two different problems (Figure 3.3): the Restricted Master Problem (RMP) and the set of sub-problems. The former includes the general constraints where only a subset of variables is considered. The latter is a set of sub-problems which group the constraints that have a special structure. These sub-problems are created in order to identify new variables that could be included into the RMP according to the defined criteria.

In this particular case (Figure 3.4), the RMP is a Facility Location Problem where the goal is to determine which depots should be opened. The sub-problem is the Elementary Shortest Path

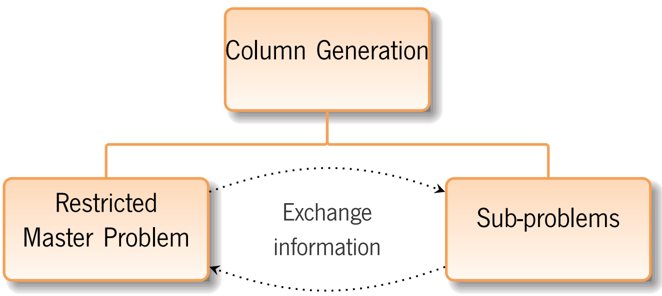

Figure 3.3: Column Generation workflow

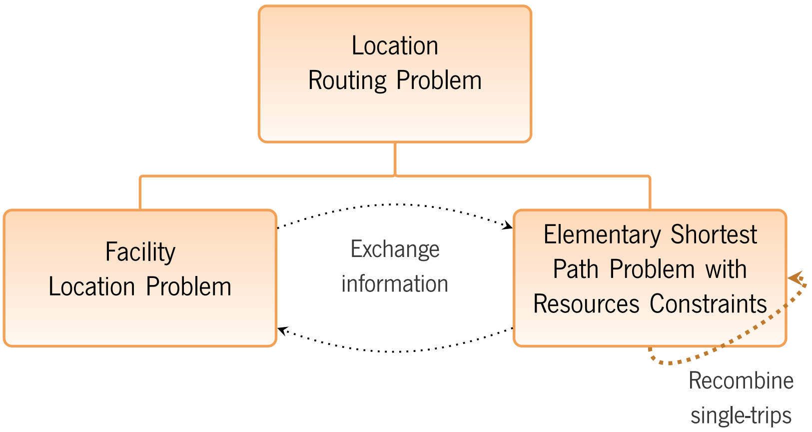

Figure 3.4: Location Routing Problem workflow

Problem with Resource Constraints (ESPPRC) in which its solution outputs the routes that will be assigned to the vehicles. The restricted master problem and the sub-problems will exchange information in order to find the optimal solution of the original problem. While solving the ESPPRC, it is possible to find all valid single-trips and then re-arrange them to create the multi-trips.

The CG is an iterative process as depicted in Figure 3.5. In a first phase, an initial valid solution is generated through a rounding single-trip initialization heuristic in order to have a set of valid columns for the initial RMP. This heuristic creates a valid initial solution by generating several single-trips in which the vehicle leaves a depot, visits just one client and returns to the same depot. Then, this procedure is applied to each depot for each client, which generates $D \times N$ valid columns.

The initial RMP is solved in order to obtain the value of the dual variables for each constraint considered in the RMP. These values are then used for the resolution of the sub-problems, which, in this case, are used to recalculate the traveling costs of a single-trip. The resolution of the sub-problem is a two-step process. In the first step, the Elementary Shortest Path Problem with Resources Constraints (ESPPRC) algorithm proposed in [57] is adapted. In this version, an exact recursive method to calculate the set of valid single-trips is developed, being a single-trip attractive when its reduced cost is negative. This version of the execution of the ESPPRC algorithm applies

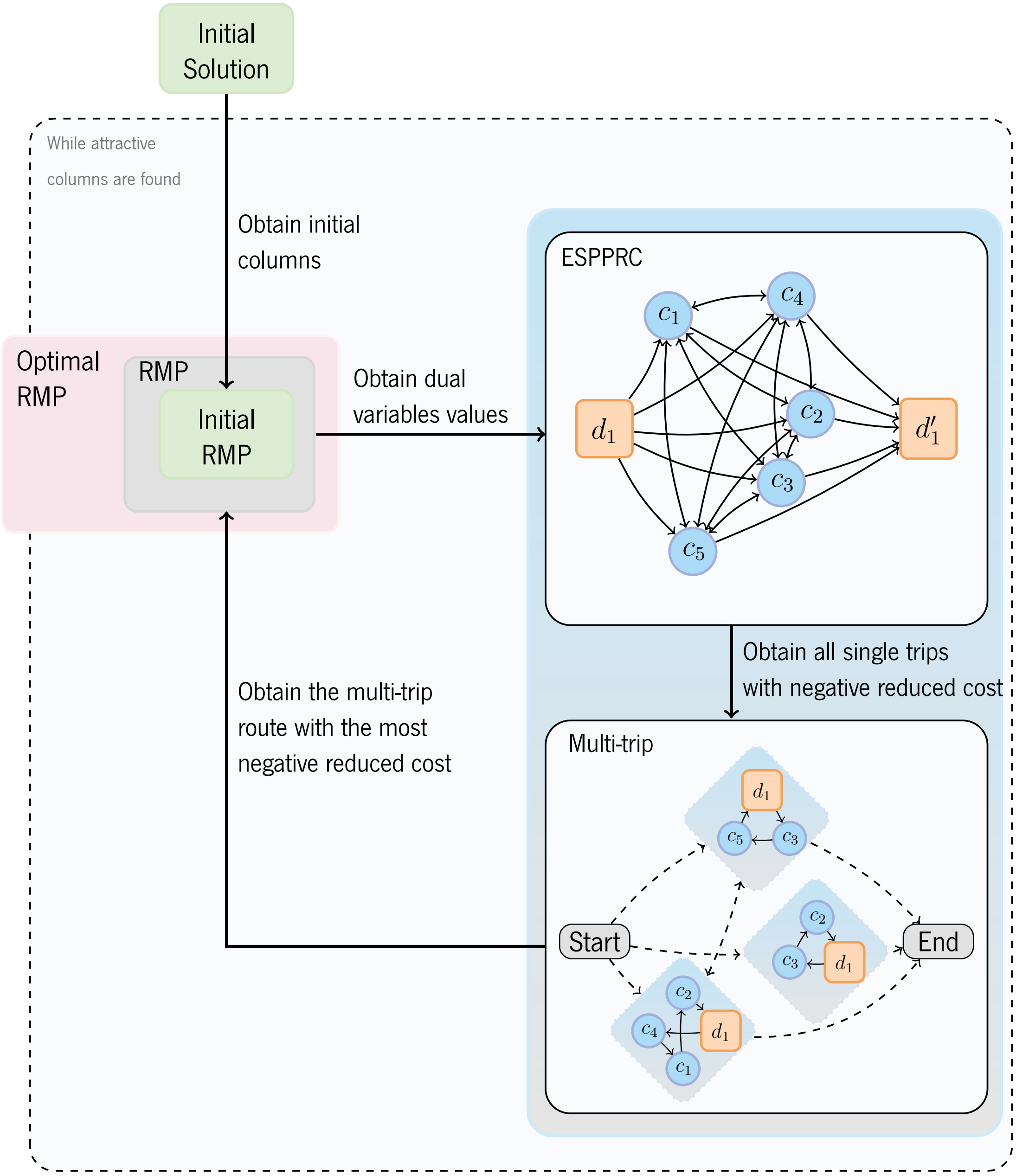

Figure 3.5: Column Generation Process for the Location Routing Problem

the concept of recursion with dynamic programming, in which the problem solution is dependent of solving smaller dimensional problems (opposing to the use of iterations), being applied as many times as needed until the stopping case is reached. Then, in the second phase, all attractive single-trips are recombined with each other in order to obtain multi-trips. These multi-trips are also recombined with other single-trips and multi-trips. For this process, a sequential algorithm which selects a single-trip or multi-trip and attempts to join with a different single-trip or multi-trip is implemented. The execution of this phase ends when no more recombinations are possible. Then, the cost of the vehicle is added to all final valid single-trips or to all final multi-trips. The ones with negative reduced cost are inserted as columns in the RMP. These two phases are explained in more detail throughout Sections 3.3.1 and 3.3.2, respectively.

The iterative process of the CG ends when, after solving the RMP and the corresponding subproblems, it is not possible to find any more routes with negative reduced cost, being assumed that the solution of the RMP is optimal, since it is not possible to find any more attractive routes.

3.3.1 Restricted master problem

Before defining the column generation process, a model for the multi-trip LRP is introduced. This model has an exponential number of columns, each representing a route or a depot.

The parameters and decision variables for the model are defined as follows:

Parameters

$N_i =$ set of clients $i, \forall i \in N$

$L_d =$ capacity L of depot $d, \forall d \in D$

$C_f^d =$ fixed cost to open a depot $d, \forall d \in D$

$b_i =$ client demand $i, \forall i \in N$

$C_r =$ cost of the route r associated to the depot $d, \forall r \in R_d, d \in D$

$$a_{ir} = \begin{cases} 1 \text{ if client } i \text{ is associated to the route } r \text{ of depot } d, \\ 0 \text{ otherwise} \end{cases} \qquad \forall i \in N, r \in R_d, d \in D$$

Decision variables

$$\lambda_d = \begin{cases} 1 \text{ if depot } d \text{ is selected} \\ 0 \text{ otherwise} \end{cases} \quad \forall d \in D,$$

$$\theta_r = \begin{cases} 1 \text{ if the route } r \text{ is selected for depot } d \\ 0 \text{ otherwise} \end{cases} \quad \forall r \in R_d, \forall d \in D,$$

The integer programming model is formulated through constraints (3.13)-(3.20), and it is composed by two sets of binary variables λ_d and θ_r. When λ_d takes the value 1 it means that the depot d is open, being closed when the binary variable takes the value 0. If a route performed by a vehicle associated to a depot d is selected then θ_r takes the value 1, $\forall r \in R_d$. R_d represents the set of routes performed by all vehicles associated to depot d. The RMP takes care of the opening or the closure of the depots, since it is the sub-problem which determines the attractive routes that should be included in the problem. The binary parameter a_{ir} indicates if a client i is served by the route r.

Minimize

$$\sum_{d \in D} C_f^d \lambda_d + \sum_{d \in D} \sum_{r \in R_d} C_r \theta_r \tag{3.13}$$

subject to:

$$\sum_{d \in D} \sum_{r \in R_D} a_{ir} \theta_r = 1 \ \forall i \in N, \tag{3.14}$$

$$\sum_{r \in R_d} \sum_{i \in N} a_{ir} b_i \theta_r - L_d \lambda_d \leq 0 \ \forall d \in D, \tag{3.15}$$

$$\sum_{r \in R_d} a_{ir} \theta_r \leq \lambda_d \ \forall d \in D, \forall i \in N, \tag{3.16}$$

$$\lambda_d \leq 1 \ \forall d \in D, \tag{3.17}$$

$$\sum_{d \in D} \lambda_d \leq U, \tag{3.18}$$

$$\lambda_d \in \{0,1\} \; \forall d \in D, \tag{3.19}$$

$$\theta_r \in \{0,1\} \; \forall r \in R_d. \tag{3.20}$$

where:

$$U = \left\lceil \frac{\sum_{i \in N} b_i}{L_{d_0}} \right\rceil \tag{3.21}$$

The goal of the model is to minimize the total distribution cost, *i.e.*, minimize the cost associated to opening a depot and the cost of the routes that are necessary to do such distribution. Constraints (3.14) ensure that clients are served and constraints (3.15) guarantee that the total demand of the served clients does not surpass the depot capacity. The use of a depot corresponds, at most, to its total capacity (Constraints (3.16) and (3.17)) and the total number of open depots is limited through constraint (3.18). The value of U (constraint (3.21)) indicates the maximum number of depots that may be opened in order to satisfy the total demand of the clients. This value also considers the depots capacity. Finally, constraints (3.19) and (3.20) define the binary variables of the problem.

The RMP is formulated as the linear relaxation of the integer programming model that considers the MLRP with constraints ensuring that each client is visited once and his demand is satisfied without exceeding the depots capacity. In the RMP, it is possible to obtain the dual variables values, which are fundamental to start the column generation process. With these values, the reduced cost is determined and used to find routes that have the potential of improving the value of the objective function.

In order to clarify the structure of the model, Table 3.2 illustrates an example of the RMP from the MLRP in which all the constraints are presented with some examples of columns. A column of the type θ_r represents a route that visits a set of clients. The total demand of that route is

Table 3.2: RMP structure of column generation

			λ_1	λ_2	$\dots$	λ_d	θ_1	θ_2	$\dots$	θ_r		
π		c_1	0	0	$\dots$	0	1	0	$\dots$	0	$=$	1
		c_2	0	0	$\dots$	0	0	1	$\dots$	0	$=$	1
		$\vdots$	$\dots$	$\dots$	$\ddots$	$\dots$	$\dots$	$\dots$	$\ddots$	$\dots$	$\vdots$	$\vdots$
		c_i	0	0	$\dots$	0	1	0	$\dots$	1	$=$	1
μ		λ_1	$-L_{\lambda_1}$	0	$\dots$	0	D_{θ_1}	0	$\dots$	0	$\leq$	0
		λ_2	0	$-L_{\lambda_2}$	$\dots$	0	0	D_{θ_2}	$\dots$	0	$\leq$	0
		$\vdots$	$\dots$	$\dots$	$\ddots$	$\dots$	$\dots$	$\dots$	$\ddots$	$\dots$	$\vdots$	$\vdots$
		λ_d	0	0	$\dots$	$-L_{\lambda_d j}$	0	0	$\dots$	D_{θ_r}	$\leq$	0
σ	λ_1	c_1	-1	0	$\dots$	0	1	0	$\dots$	0	$\leq$	0
		c_2	-1	0	$\dots$	0	0	0	$\dots$	0	$\leq$	0
		$\vdots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\vdots$	$\vdots$
		c_i	-1	0	$\dots$	0	1	0	$\dots$	0	$\leq$	0
	λ_2	c_1	0	-1	$\dots$	0	0	0	$\dots$	0	$\leq$	0
		c_2	0	-1	$\dots$	0	0	1	$\dots$	0	$\leq$	0
		$\vdots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\vdots$	$\vdots$
		c_i	0	-1	$\dots$	0	0	0	$\dots$	0	$\leq$	0
	$\vdots$	c_1	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\vdots$	$\vdots$
		c_2	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\vdots$	$\vdots$
		$\vdots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\vdots$	$\vdots$
		c_i	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\vdots$	$\vdots$
	λ_d	c_1	0	0	$\dots$	-1	1	0	$\dots$	0	$\leq$	0
		c_2	0	0	$\dots$	-1	0	0	$\dots$	0	$\leq$	0
		$\vdots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\vdots$	$\vdots$
		c_i	0	0	$\dots$	-1	0	0	$\dots$	1	$\leq$	0
		c_1	1	0	$\dots$	0	0	0	$\dots$	0	$\leq$	1
		c_2	0	1	$\dots$	0	0	0	$\dots$	0	$\leq$	1
		$\vdots$	$\dots$	$\dots$	$\ddots$	$\dots$	$\dots$	$\dots$	$\ddots$	$\dots$	$\vdots$	$\vdots$
		c_i	0	0	$\dots$	1	0	0	$\dots$	0	$\leq$	1
			1	1	$\dots$	1	0	0	$\dots$	0	$\leq$	U
Objective			$C_f^{\lambda_1}$	$C_f^{\lambda_2}$	$\dots$	$C_f^{\lambda_d}$	C_{θ_1}	C_{θ_2}	$\dots$	C_{θ_r}		

denoted by D_{θ_r}, which is associated to a given depot. The coefficient C_{θ_r} is used to represent the total cost of a route θ_r.

The λ_d columns stand for the opening of a depot d, which has an associated capacity L_d and an opening cost denoted by C_f^d. For each constraint of the primal problem there is an associated dual variable. Thus, for each line of the table, there is a correspondent value from the dual variable. Dual variables for the clients are denoted by π_i, while μ_d are the ones associated to the depots. There are other types of dual variables which include both the depots d and the clients i which are designated by σ_{di}.

After solving the initial RMP and using the value of the dual variables, it is possible to initiate the iterative process of the column generation by calculating the cost of the path in order to solve the ESPPRC. These reduced costs are calculated through Equation (3.22), as follows.

3.3.2 Sub-problem

The goal of the sub-problem is to obtain valid routes considering all constraints. The sub-problem is solved by two different algorithms which are used in a sequential manner (Figure 3.6). The first algorithm is an adapted version of the algorithm proposed by Feillet et al. [57] to solve the ESPPRC, in which the goal is to obtain valid single-trips.

All generated single-trips with negative reduced cost are used as input for the second algorithm. This combines all single-trips in order to obtain valid multi-trips through a route recombination algorithm. Generated multi-trips may also be recombined with other multi-trips or single-trips. The sub-problem ends when it is not possible to generate any more multi-trips. Then, the cost of the vehicle is added to all final valid routes, and those with negative reduced cost are considered valid and used for the next iteration of the RMP.

In order to determine if a column (route) generated by the sub-problem has the potential to improve the value of the objective function, it is necessary to calculate its reduced cost:

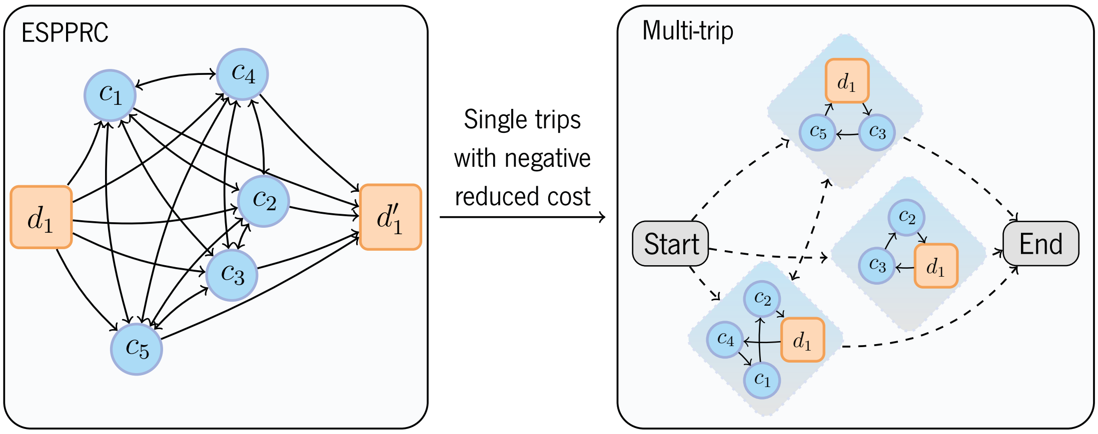

Figure 3.6: Sub-problem workflow example for depot d_1

$$\text{Reduced Cost} = C_r - \sum_{i \in N} a_{ir}\pi_i - \sum_{i \in N} a_{ir} b_i \mu_d - \sum_{i \in N} a_{ir}\sigma_{di} \tag{3.22}$$

The π_i represents the dual variable associated with constraints (3.14) for client i, μ_d denotes the dual variable associated to the constraints (3.15) for depot d, and σ_{di} is the dual variable associated to constraints (3.16) for depot d and client i.

After determining the cost of all arcs, a graph is built where, through an adaptation of the algorithm proposed by Feillet *et al.* [57], the elementary shortest paths are determined. These paths are single-trips and those with negative reduced cost are used in the second phase of the sub-problem resolution. Here, the algorithm tries, in a sequential manner, to combine the previously generated single-trips in order to create new valid multi-trips. All recently created multi-trips are then combined with different single-trips or multi-trips. The second phase of the resolution of the sub-problem ends when no more multi-trips are generated. In order to reduce the number of valid routes available during this process, whenever a new single-trip or multi-trip is generated the dominance rules are verified.

The adapted version of the algorithm proposed by Feillet *et al.* [57] to solve the ESPPRC is presented, in more detail, in Algorithm 3.1. The notation used in the algorithm is the following:

- ▷ E is the set of untreated nodes;
- ▷ $v_i \in V$ represents the current node (under treatment) from which the algorithm is trying to extend to a different node;
- ▷ $v_j \in V$ is the node to which one is trying to extend. A node v_j may only be selected if it is possible to extend the trip from the current node v_i, *i.e.*, if the vehicle capacity or distance constraints are not violated;
- ▷ λ_i denotes the trip in node v_i that is being extended to a new node;
- ▷ Λ_i is the set of non-dominated trips in node v_i.

To start the execution of the algorithm, one needs to use four arguments: the initial starting node v_i, a initialized trip for that node λ_i, the node v_j to which the current trip will be extended, and the set of untreated nodes E.

In each iteration, the model verifies three conditions before extending the selected single-trip to the next selected node. First (representing the stopping case), the model checks if the untreated nodes are not empty (otherwise there is nothing to do and the execution stops). Then, the next node has to exist, *i.e.*, it has to be a valid node, otherwise it means that the selected single-trip has been extended to all successor nodes. In this case, the next single-trip of the current node is selected and the list of successor for that single-trip is erased. If the selected single-trip does not exist, then the execution is in the case where the selected single-trip is the last one from the current node and has been extended to all its successors. Thus, in this case, the current node is considered treated (and removed from the untreated set E), a new current node is selected and the list of its successors is computed.

Passing the above mentioned verifications, the execution checks whether the current single-trip may be extended, *i.e.*, checks the constraint related to the distance travelled by the vehicle and

Algorithm 3.1: $ESPPRC_{rec}$

Input: v_i, λ_i, v_j, E

if $E = \emptyset$ **then**
  end ;
if $v_j = null$ **then**
  $\lambda_i \leftarrow choose\_next(\lambda_i) \in \Lambda_i$;
  $v_j \leftarrow restart\_succ(v_i) \in succ(v_i)$;
  $ESPPRC_{rec}(v_i, \lambda_i, v_j, E)$;
  end ;
if $\lambda_i = null$ **then**
  $E \leftarrow E \backslash \{v_i\}$;
  $v_i \leftarrow choose\_next(v_i) \in E$;
  $\lambda_i \leftarrow choose\_trip(v_i)$;
  $v_j \leftarrow choose\_next\_succ(v_i) \in succ(v_i)$;
  $ESPPRC_{rec}(v_i, \lambda_i, v_j, E)$;
  end ;
if $possible\_to\_extend(\lambda_i, v_j)$ **then**
  $F_{ij} \leftarrow F_{ij} \cup \{Extended(\lambda_i, v_j)\}$;
  $\Lambda_z \leftarrow EFF(F_{ij} \cup \Lambda_z)$;
  if Λ_z *has changed* **then**
    $E \leftarrow E \cup \{v_j\}$;
$v_j \leftarrow choose\_next\_succ(v_i) \in succ(v_i)$;
$ESPPRC_{rec}(v_i, \lambda_i, v_j, E)$;
end ;

the capacity constraint. If it is possible to extend, a new single-trip is created and the dominance rules are tested between the new single-trip and the current single-trips under the v_j node. If the new single-trip is non-dominated, the next node v_j is added to the untreated set of nodes E and a new iteration is started.

Ending the search for single-trips, the execution of the sub-problem starts a new phase in which it tries to merge the generated single-trips with negative reduced cost in order to find multi-trips. The algorithm for this phase is depicted in Algorithm 3.2. Instead of applying a recursive implementation, in the generation of multi-trips, a cycle is used to iterate over all single-trips and

Algorithm 3.2: Multi-trip

Input: set of single-trips E

repeat
    choose $\lambda_i \in E$;
    forall $\lambda_j \in \Lambda$ **do**
        if $possible\_to\_extend(\lambda_i, \lambda_j)$ **then**
            $F \leftarrow F \cup \{Extended(\lambda_i, \lambda_j)\}$;
            $\Lambda \leftarrow EFF(F \cup \Lambda)$;
            if Λ *has changed* **then**
                $E \leftarrow E \cup \{v_z\}$;
    $E \leftarrow E \backslash \{v_w\}$;
until $E = \emptyset$;

all recently generated multi-trip. This last process enables the creation of multi-trips from two or more single-trips.

In a first phase, two different routes are selected, and it is verified if they can be part of the same route. If the routes λ_i and λ_j can be aggregated, then the routes are merged into one new route. It is important to note that λ_i and λ_j can represent a single-trip or a multi-trip. However the new route becomes necessarily a multi-trip route. Before this attempt to create a newer route, there are some conditions that must be verified according to the constraints of the model. For instance, two routes cannot be merged if a client c_w is visited by both routes λ_i and λ_j, or the total distance available for the vehicle is exceeded.

After merging two trips, the dominance rules are verified to check if the newest route is not dominated. When this occurs, the merged multi-trip is considered untreated and is added to the set E. After extending the selected route λ_i with all other routes, it is considered treated and removed from E. Then, the next untreated route is selected and the execution of the algorithm continues until there are no more routes to extend, *i.e.*, $E = \emptyset$.

In order to check whether a route is dominated or not, it is important to define the structure of the route. Formally it is defined as $R_i = \{(T_i^1, T_i^2, \ldots, T_i^N), (D_i^1, D_i^2, \ldots, D_i^N), C_i, RC_i\}$ and composed by:

- ▷ an ordered set of clients (T_i^c), where 1 represents a visited client and 0 otherwise;
- ▷ an ordered set of consumed resources (D_i^c) that is 0 when the client is not visited;
- ▷ the total cost of the trip (C_i);
- ▷ the reduced cost associated to the trip (RC_i).

A route R_i dominates other route R_j ($i \neq j$) if and only if $T_i^n \leq T_j^n$ for $n = 1, \ldots, N$, $D_i^n \leq D_j^n$ for $n = 1, \ldots, N$, and $RC_i \leq RC_j$.

3.4 A network flow formulation

A network flow model has a graph-based structure that is used to solve problems, in which the arcs have an associated flow. The flow associated to an arc must be less than or equal to its capacity being, in this case, the capacity associated with a vehicle. The network nodes represent discrete instants of time, in which the flow conservation must be ensured.

3.4.1 The model

The network flow model, presented by Macedo et al. in [55], is defined on acyclic directed graphs (one per depot) that will be denoted by $\Pi_d = (\Delta, \Psi_d)$, $d \in D$. A path on these graphs corresponds to the workday W of a given vehicle. The vehicle is associated to a depot d. The vertices in Δ represent discrete time instants starting from 0 up to the time limit W. The arcs are associated to the vehicle routes, and additionally to waiting periods at the depot. An arc $(u, v)^r \in \Psi_d$ is related to a route r that starts at time instant u and ends at time instant v.

The set Ψ_d is defined as follows:

$$\Psi_d = \{(u, v)^r : 0 \leq u < v \leq W, r \in R_d\} \cup \{(u, v)^o : 0 \leq u < v \leq W, v = u + 1\},$$

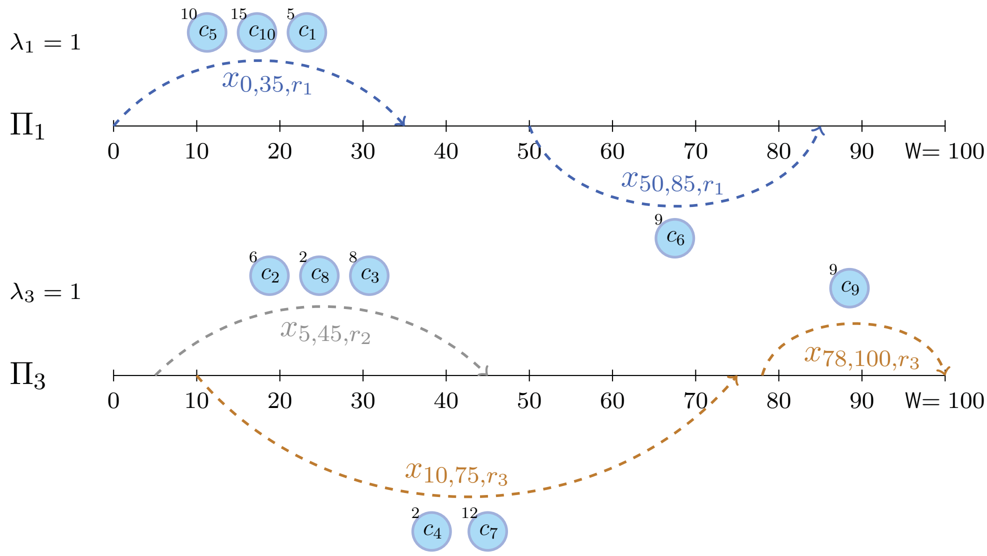

Figure 3.7: Solution example of network flow model

with R_d being the set of all the routes from depot d. The load, duration and cost of a route r will be denoted by l_r, t_r and C_r, respectively. The set of clients visited by a route r will be represented by N_r, with $N_r \subseteq N$. Clearly, a route is feasible only if $l_r \leq Q$ and $t_r \leq W$.

The model is composed by two sets of variables. The binary variable λ_d, $d \in D$, states whether a depot is selected or not, while the binary variable x^d_{uvr} states whether the route r associated to depot d is selected or not. The route r starts and finishes at time instants u and v, respectively.

In Figure 3.7 an example of a network flow for the MLRP that serves a set of ten different clients ($c_1, \ldots, c_{10}$) from three possible depots (λ_1, λ_2 and λ_3) is depicted. The example solution presents the graph Π_1 and the graph Π_3 associated to opening depots d_1 and d_3, respectively (denoted by $\lambda_1 = 1$ and $\lambda_3 = 1$). Depot d_2 is not represented in the figure since it is not opened ($\lambda_2 = 0$). For the sake of simplicity, only the arcs that have an associated flow are represented in the figure.

For the sake of simplicity the decision variable x_{uvr} denotes an arc. The arc $x_{0,35,r_1}$ serves three different clients (c_5, c_{10} and c_1) during the planning horizon that begins at time $u = 0$ and ends at time $v = 35$. This flow is associated to the first route denoted by r_1. Associated to this route r_1, there is also another flow that satisfies a single client (c_6) and goes from time $u = 50$ to instant $v = 85$. This arc is denominated $x_{50,85,r_1}$. Arcs $x_{0,35,r_1}$ and $x_{50,85,r_1}$ may not be merged into a single arc since the demand of this joined arc would surpass the capacity of the vehicle.

Similarly to the arcs $x_{0,35,r_1}$ and $x_{50,85,r_1}$, arcs $x_{10,75,r_3}$ and $x_{78,100,r_3}$ may not be merged due to vehicle capacity constraints. Furthermore, and according to the vehicle workday (W) constraint, it is not possible to use just one vehicle to fulfill the clients demand since adding arcs of route r_2 to route r_3 exceeds the vehicle workday.

The parameters and some important definitions, and decision variables used in the network flow model are listed below.

Parameters and definitions

$\Pi_d =$ acyclic directed graph associated to the depot $d, \forall d \in D$

$\Psi_d =$ set of arcs associated to the depot $d, \forall d \in D$

$\Delta =$ set of vertices,

$(u, v)^r =$ arc that represents a route r that starts at instant of time u and ends at time instant $v, \forall r \in R_d$

$R_d =$ set of all routes associated to the depot $d, \forall d \in D$

$C_r =$ cost associated to perform a route $r, \forall r \in \Psi_d$

$C_v =$ cost associated to the use of a vehicle $v, \forall v \in H$

$C_f^d =$ fixed cost associated to opening a depot $d, \forall d \in D$

$K_d^{max} =$ limits the maximum number of vehicles per depot $d, \forall d \in D$

$W =$ length of the plan horizon

$l_r =$ load associated to a route $r, \forall r \in R_d$

$t_r =$ duration associated to a route$r, \forall r \in R_d$

$b_i =$ demand associated to a client $i, \forall i \in N$

$L_d =$ capacity associated to the depot $d, \forall d \in D$

$Q =$ capacity associated to the vehicle $v, \forall v \in H$

$N_r =$ set of clients visited by route $r, \forall r \in R_d$

Decision variables

$$\lambda_d = \begin{cases} 1 \text{ if the depot } d \text{ is selected ,} \\ 0 \text{ otherwise} \end{cases} \quad \forall d \in D,$$

$$x^d_{uvr} = \begin{cases} 1 \quad \text{if route } r \text{ goes through the arc } (u,v) \text{ for the depot } d, \forall r \in R_d, \\ \quad 0 \leq u \leq W \in \Psi \text{ and } \forall d \in D, \\ 0 \quad \text{otherwise} \end{cases}$$

The main goal of the network flow model is the minimization of the cost associated to all the routes performed by the vehicles, the cost related to the vehicle usage and the cost for opening a depot. The network flow formulation is defined through constraints (3.23) to (3.29).

Minimize

$$\sum_{d \in D} \sum_{(u,v)^r \in \Psi_d} C_r x^d_{uvr} + C_v \sum_{d \in D} \sum_{(0,v)^r \in \Psi_d} x^d_{0vr} + \sum_{d \in D} C^d_f \lambda_d \tag{3.23}$$

subject to:

$$\sum_{d \in D} \sum_{(u,v)^r \in \Psi_d | i \in N_r} x^d_{uvr} = 1, \quad \forall i \in N, \tag{3.24}$$

$$\sum_{(0,v)^r \in \Psi_d} x^d_{0vr} \leq K^{max}_d \lambda_d, \quad \forall d \in D, \tag{3.25}$$

$$-\sum_{(u,v)^r\in\Psi_d} x^d_{uvr} + \sum_{(v,y)^t\in\Psi_d} x^d_{vyt} = \begin{cases} 0, \text{ if } v = 1, ..., W-1, \\ -\sum_{(0,v)^r\in\Psi_d} x^d_{0vr}, \text{ if } v = W, \end{cases} \quad \forall d \in D, \quad (3.26)$$

$$\sum_{(u,v)^r\in\Psi_d} l_r x^d_{uvr} \leq L_d\lambda_d, \quad \forall d \in D, \quad (3.27)$$

$$x^d_{uvr} \geq 0, \quad \text{and integer}, \forall (u,v)^r \in \Psi_d, \ \forall d \in D, \quad (3.28)$$

$$\lambda_d \in \{0,1\}, \quad \forall d \in D. \quad (3.29)$$

Every upper bound on the number of workdays can be used for K^{max}_d (Constraint (3.30)). In the experiments presented in Tables 3.5 and 3.6, the following value is used, assuming the clients are sorted in decreasing order of their demands:

$$K^{max}_d = \max\left\{j : \sum_{i=1}^{j} b_i \leq L_d\right\}, \quad (3.30)$$

for a given depot $d \in D$. Flow conservation is enforced through constraints (3.26).

Constraints (3.24) force the visit to every client. Constraints (3.25) limit the number of vehicles per depot to a maximum of K^{max}_d. Note that x^d_{0vr} is directly related to an independent workday starting at time instant 0 from depot d and finishing at time instant v. If the corresponding depot is not selected, the maximum number of vehicles becomes naturally 0. Constraints (3.27) ensure the capacities of the depots are not exceeded. The objective function is represented through the expression (3.23).

3.4.2 Valid inequalities

To improve the quality of the continuous lower bounds obtained with the network flow model (3.23)-(3.29), the following valid inequalities can be used. The first consists in forcing a minimum number D^{min} of depots to be opened through the constraint (3.31).

$$\sum_{d\in D} \lambda_d \geq D^{min}. \quad (3.31)$$

The depot capacity L_d is equal for all depots. Thus, the problem of determining D^{min} is an one-dimensional bin-packing problem. In order to compute its value, it is resorted to dual-feasible functions [58] which provide a means to obtain high quality lower bounds frequently close to those achieved with column generation models.

The second inequality is similar to the previous one, but applies now to the vehicles. The principle is to enforce a minimum number H^{min} of vehicles to use through the constraint (3.32).

$$\sum_{d \in D} \sum_{(u,v)^r \in \Psi_d} x_{uvr}^d \geq H^{min}. \tag{3.32}$$

Again, H^{min} is a lower bound for the bin-packing problem defined by using the clients demands and the vehicles capacities, and it can be computed using the aforementioned dual-feasible functions.

The last set of inequalities consists in relating the selection of workdays to opening depots. These inequalities state that if a depot is open, there should be at least one workday to be performed from this depot:

$$\sum_{(0,v)^r \in \Psi_d} x_{0vr}^d \geq \lambda_d, \ \forall d \in D. \tag{3.33}$$

In Table 3.3 an example of the MIP for the network flow model is presented. All constraints (described from Equation 3.24 to 3.29) and the above mentioned inequalities are depicted in the table. Due to the number of columns, only a subset of them was considered. Each set of rows represents constraints of the model in the same order as they appear above.

Table 3.3: MIP structure for network flow model

	$x^{d_1}_{0,35,r_1}$	$x^{d_1}_{1,36,r_1}$	$\vdots$	$x^{d_1}_{35,70,r_1}$	$\vdots$	$x^{d_1}_{65,100,r_1}$	$x^{d_1}_{0,40,r_2}$	$x^{d_1}_{1,41,r_2}$	$\dots$	$x^{d_1}_{60,100,r_2}$	$x^{d_2}_{0,35,r_3}$	$x^{d_2}_{1,36,r_3}$	$\vdots$	$x^{d_2}_{35,70,r_3}$	$\vdots$	$x^{d_2}_{65,100,r_3}$	$x^{d_2}_{0,40,r_4}$	$x^{d_2}_{1,41,r_4}$	$\dots$	λ_1	λ_2	λ_3	...	λ_d		
c_1	1	1	...	1	...	1	0	0	...	0	0	0	...	0	...	0	0	0	...	0	0	0	...	0	=	1
c_2	0	0	...	0	...	0	1	1	...	1	0	0	...	0	...	0	0	0	...	0	0	0	...	0	=	1
c_3	0	0	...	0	...	0	1	1	...	1	0	0	...	0	...	0	0	0	...	0	0	0	...	0	=	1
$\vdots$	...	...	...	...	...	...	...	...	$\ddots$	...	...	...	...	...	...	...	...	$\ddots$	...	...	...	...	$\ddots$	...	=	$\vdots$
c_n	1	1	...	1	...	1	0	0	...	0	0	0	...	0	...	0	0	0	...	0	0	0	...	0	=	1
d_1	1	0	...	0	...	0	1	0	...	0	0	0	...	0	...	0	0	0	...	$-K^{max}$	0	0	...	0	$\leq$	0
d_2	0	0	...	0	...	0	0	0	...	0	1	0	...	0	...	0	1	0	...	0	$-K^{max}$	0	...	0	$\leq$	0
d_3	0	0	...	0	...	0	0	0	...	0	0	0	...	0	...	0	0	0	...	0	0	$-K^{max}$	...	0	$\leq$	0
$\vdots$	...	...	...	...	...	...	...	...	$\ddots$	...	...	...	...	...	...	...	...	$\ddots$	...	...	...	...	$\ddots$	...	$\leq$	$\vdots$
d_d	0	0	...	0	...	0	0	0	...	0	0	0	...	0	...	0	0	0	...	0	0	0	...	$-K^{max}$	$\leq$	0
d_1			...		...				...				...		...				...				...		=	
d_2			...		...				...				...		...				...				...		=	
d_3			...		...				...				...		...				...				...		=	
$\vdots$	...	...	...	...	...	...	...	...	$\ddots$	...	...	...	...	...	...	...	...	$\ddots$	...	...	...	...	$\ddots$	...	=	$\vdots$
d_d			...		...				...				...		...				...				...		=	
d_1	l_{r_1}	l_{r_1}	...	l_{r_1}	...	l_{r_1}	l_{r_2}	l_{r_2}	...	l_{r_2}	0	0	...	0	...	0	0	0	...	$-L_1$	0	0	...	0	$\leq$	0
d_2	0	0	...	0	...	0	0	0	...	0	l_{r_3}	l_{r_3}	...	l_{r_3}	...	l_{r_3}	l_{r_4}	l_{r_4}	...	0	$-L_2$	0	...	0	$\leq$	0
d_3	0	0	...	0	...	0	0	0	...	0	0	0	...	0	...	0	0	0	...	0	0	$-L_3$	...	0	$\leq$	0
$\vdots$	...	...	...	...	...	...	...	...	$\ddots$	...	...	...	...	...	...	...	...	$\ddots$	...	...	...	...	$\ddots$	...	$\leq$	$\vdots$
d_d	0	0	...	0	...	0	0	0	...	0	0	0	...	0	...	0	0	0	...	0	0	0	...	$-L_d$	$\leq$	0
	0	0	...	0	...	0	0	0	...	0	0	0	...	0	...	0	0	0	...	1	1	1	...	1	$\geq$	D^{min}
	1	1	...	1	...	1	1	1	...	1	1	1	...	1	...	1	1	1	...	0	0	0	...	0	$\geq$	H^{min}
d_1	1	0	...	0	...	0	1	0	...	0	0	0	...	0	...	0	0	0	...	-1	0	0	...	0	$\geq$	0
d_2	0	0	...	0	...	0	0	0	...	0	1	0	...	0	...	0	1	0	...	0	-1	0	...	0	$\geq$	0
d_3	0	0	...	0	...	0	0	0	...	0	0	0	...	0	...	0	0	0	...	0	0	-1	...	0	$\geq$	0
$\vdots$	...	...	...	...	...	...	...	...	$\ddots$	...	...	...	...	...	...	...	...	$\ddots$	...	...	...	...	$\ddots$	...	$\geq$	$\vdots$
d_d	0	0	...	0	...	0	0	0	...	0	0	0	...	0	...	0	0	0	...	0	0	0	...	-1	$\geq$	0

3.4.3 Arcs generation

The network flow model requires the pre-existence of a set of single-trips and selects those that best serve the clients with the aim of minimizing the total costs associated to the global system concerning the distribution decisions. The model simultaneously considers location decisions, and decides which single-trips should be used to serve a set of clients at the minimum cost. A single-trip is represented through an arc that must start and end at the same depot and fulfill a set of clients. A vehicle is able to perform a set of single-trips that are limited by a workday. The model defines the arcs traversed by a certain vehicle associated to a specific depot. The single-trips performed by a vehicle are named multi-trip.

Algorithm 3.3 allows for the generation of the single-trips used by the network flow model that creates all the possible combinations according to the constraints defined in the model. In order to generate the set of single-trips, it is necessary to note some details such as:

- ▷ A single-trip must start and end at the same depot;
- ▷ The capacity of a vehicle cannot be exceeded;
- ▷ A single-trip cannot travel a distance greater than the workday of the vehicle;
- ▷ A customer can only be visited exactly once in each arc.

The arcs of the network flow model are generated for each depot according to the method defined in Algorithm 3.3. Indeed, the execution of the algorithm for a given depot is independent from the execution for a different depot.

During the execution of the algorithm, there are two different types of arcs: partial arcs and final arcs. The former considers arcs which start in the depot and visit one or more clients, but do not return to the depot. Thus, it is possible to add more clients or close a partial arc back to the depot. The latter are arcs that start and end at the depot and serve one or multiple clients.

Algorithm 3.3: MLRP Arc Generation

```
Input: depot
partial_arcs ← empty_arc ;
while partial_arcs! = ∅ do
    current ← first_element(partial_arcs);
    partial_arcs ← partial_arcs\{current};
    if possible_to_reach_depot(current) then
        new_closed_arc ← close_arc(current);
        final_arcs ← final_arcs ∪ {new_closed_arc};
    foreach client c ∈ N do
        if client_not_present(current, c) then
            if check_distance_client(current, c) then
                if check_demand_client(current, c) then
                    new_partial_arc ← extend_arc(current, c);
                    partial_arcs ← partial_arcs ∪ {new_partial_arc};
```

As depicted in Algorithm 3.3, the execution of the arcs generation starts with an empty arc, *i.e.*, an arc that does not serve any client. After this step, the generation of arcs may start. In order to evaluate if a partial arc can be transformed into a final arc it is necessary to perform two verifications. The first is to test if the partial arc may be closed, *i.e.*, it is possible to return to the depot from the last visited client. When the answer is affirmative, the new closed arc, considered as a final arc, is added to the set of final arcs. The other verification is to check whether it is possible to add more clients considering the distance and capacity constraints. Indeed, to add a new client to the selected partial arc, there are three conditions that must be met:

- ▷ the new client being served is not already in the partial arc;
- ▷ the distance to reach the new client does not surpass the available vehicle workday;
- ▷ the demand of the new client does not exceed the available vehicle capacity.

If and only if these constraints are not violated, a new partial arc is generated and added to the corresponding set. After trying to extend the selected partial arc to all clients, a new iteration starts if there are more partial arcs. Thus, the execution of the algorithm will attempt to close the new selected arc and to add more clients to it. The process ends when it is not possible to extend partial arcs, *i.e.*, there are no more partial arcs in the corresponding set.

3.5 Implementation details

Nowadays, with the constant technological advances or improvements, it becomes important to use available resources in the best possible way. During the implementation of the optimization algorithms, resource management has been taken into account and the most relevant implementation details are discussed throughout this section.

A very important detail with respect to the execution of the MLRP algorithm is the use of parallelization techniques. This parallelization is done by applying the thread concept which is used in the MLRP sub-problems by the column generation. This approach is possible since determining the elementary paths is independent from depot to depot. Thus, sub-problems are solved in parallel for each possible depot. During the execution of the column generation algorithm, all the sub-problems are initiated at the same time, so the algorithm waits for the ESPPRC of the slowest depot instead of waiting for a sequential processing of the ESPPRC for each depot. This parallelization is either used in the generation of single-trips or in the creation of multi-trips.

Table 3.4 presents the results obtained in the series and parallel mode for 64 different instances of 20 and 25 clients. The relevant parameters of the instances are shown in the table. The tests were executed on a PC with an i7 CPU with 3.5 GHz and 32GB of RAM. The total time of each test was limited to 7200 seconds, considering that 900 seconds (from the 7200) were used in the search of the integer solution through the CPLEX 12.6 subroutine. In each iteration for each depot, 400 seconds were used to determine the single routes and 400 seconds to calculate the multi-routes. As depicted in the Table 3.4, in most instances the use of the parallelization is justified since there is an evident computational time reduction. The cases in which the parallel mode presents higher computational times than the series one happens because there is a larger number of iterations leading to smaller value of the objective function. When the algorithm starts an iteration (since there is available time), it has to end it. However, this iteration end may occur after the time limit.

Considering the above mentioned detail and results, it is observed that parallelization mode allows for an improvement of the algorithms execution.

Table 3.4: Comparative analysis for the series and parallel mode for the Column Generation

					Series mode					Parallel mode					Red.
	Inst	n	W	Q	LP	MIP	Gap(%)	Time	It.	LP	MIP	Gap(%)	Time	It.	Time(%)
$C_{20,1}$	1_1	20	140	50	3381,93	4431	0,00	628	2	3381,93	4431	0,00	340	2	45,86
$C_{20,1}$	1_2	20	160	50	3325,50	4430	0,00	1916	2	3325,50	4430	0,00	868	3	54,70
$C_{20,1}$	1_3	20	140	70	3277,98	4384	0,00	2443	2	3277,98	4384	1,08	1832	3	25,01
$C_{20,1}$	1_4	20	160	70	3306,69	4384	0,00	3296	2	3251,05	4374	1,84	3374	5	-2,37
$C_{20,2}$	2_1	20	140	50	3114,42	4681	0,00	135	2	3114,42	4681	0,00	114	2	15,56
$C_{20,2}$	2_2	20	160	50	3031,03	4472	0,00	736	2	3031,03	4472	0,00	415	2	43,61
$C_{20,2}$	2_3	20	140	70	2992,65	4424	0,00	767	2	2992,65	4424	0,00	425	2	44,59
$C_{20,2}$	2_4	20	160	70	2921,72	4424	0,00	2449	2	2921,72	4424	0,00	1635	3	33,24
$C_{20,3}$	3_1	20	140	60	3751,96	4431	0,00	606	2	3751,96	4431	0,00	473	2	21,95
$C_{20,3}$	3_2	20	160	60	3705,33	4426	0,00	1405	2	3674,23	4426	0,00	1472	3	-4,77
$C_{20,3}$	3_3	20	140	80	3610,87	4385	0,00	1610	2	3610,87	4385	0,00	1217	2	24,41
$C_{20,3}$	3_4	20	160	80	3620,68	4385	0,00	2531	2	3620,68	4385	0,00	1733	3	31,53
$C_{20,4}$	4_1	20	140	60	3750,33	4537	0,00	212	2	3750,33	4537	0,00	136	2	35,85
$C_{20,4}$	4_2	20	160	60	3666,42	4471	0,00	1075	2	3666,42	4471	0,00	714	2	33,58
$C_{20,4}$	4_3	20	140	80	3579,67	4409	0,00	950	2	3579,67	4409	0,00	674	2	29,05
$C_{20,4}$	4_4	20	160	80	3589,22	4409	0,00	2645	2	3556,34	4405	0,00	2364	4	10,62
$C_{20,5}$	5_1	20	140	60	3255,44	4674	0,00	266	2	3255,44	4674	0,00	147	2	44,74
$C_{20,5}$	5_2	20	160	60	3175,16	4475	0,00	1294	2	3174,13	4475	0,00	795	4	38,56
$C_{20,5}$	5_3	20	140	80	3122,17	4449	0,00	1614	2	3122,17	4449	0,00	686	2	57,50
$C_{20,5}$	5_4	20	160	80	3044,16	4411	0,00	3404	2	3036,41	4401	0,00	3174	5	6,76
$C_{20,6}$	6_1	20	140	50	2734,17	4420	0,00	975	2	2734,17	4420	0,00	533	2	45,33
$C_{20,6}$	6_2	20	160	50	2726,12	4420	0,00	2420	2	2683,96	4415	31,24	2668	4	-10,25
$C_{20,6}$	6_3	20	140	70	2659,63	4382	0,00	3832	2	2652,23	4383	0,00	4238	5	-10,59
$C_{20,6}$	6_4	20	160	70	2630,13	4401	34,74	4759	2	2586,95	4386	26,92	4918	5	-3,34
$C_{20,7}$	7_1	20	140	50	3306,31	4835	0,00	197	2	3306,31	4835	0,00	134	2	31,98
$C_{20,7}$	7_2	20	160	50	3226,85	4795	0,00	982	2	3226,85	4795	0,00	676	2	31,16
$C_{20,7}$	7_3	20	140	70	3192,57	4748	0,00	1002	2	3192,57	4748	0,00	510	2	49,10
$C_{20,7}$	7_4	20	160	70	3128,56	4540	0,00	2767	2	3128,56	4548	0,00	1733	2	37,37
$C_{20,8}$	8_1	20	140	50	3355,59	4744	0,00	36	2	3355,59	4744	0,00	25	2	30,56
$C_{20,8}$	8_2	20	160	50	3248,33	4656	0,00	112	2	3248,33	4656	0,00	47	2	58,04
$C_{20,8}$	8_3	20	160	50	3169,33	4587	0,00	159	2	3169,33	4587	0,00	69	2	56,60

(continues on next page)

Table 3.4: Comparative analysis (continued)

					Series mode					Parallel mode					Red.
	Inst	n	W	Q	LP	MIP	Gap(%)	Time	It.	LP	MIP	Gap(%)	Time	It.	Time(%)
$C_{20,8}$	8_4	20	140	70	3101,05	4451	0,00	743	2	3101,05	4451	0,00	351	2	52,76
$C_{25,1}$	1_1	25	140	50	4071,70	4751	0,00	2564	2	3992,74	4751	0,00	1638	4	36,12
$C_{25,1}$	1_2	25	160	50	4309,67	4992	2,59	3107	2	3959,84	4760	13,93	3432	6	-10,46
$C_{25,1}$	1_3	25	140	70	3952,11	4679	3,10	4947	2	3951,54	4687	2,99	4117	4	16,78
$C_{25,1}$	1_4	25	160	70	4001,77	4722	3,01	4953	2	3936,78	4690	3,35	4122	4	16,78
$C_{25,2}$	2_1	25	140	50	3660,07	4764	0,00	1425	2	3660,07	4764	0,00	923	3	35,23
$C_{25,2}$	2_2	25	160	50	3626,58	4764	2,53	2890	2	3601,23	4765	2,77	3064	5	-6,02
$C_{25,2}$	2_3	25	140	70	3554,13	4703	1,06	4398	2	3534,76	4706	0,99	4015	4	8,71
$C_{25,2}$	2_4	25	160	70	3557,82	4759	3,82	4776	2	3489,88	4501	0,00	4499	5	5,80
$C_{25,3}$	3_1	25	140	60	4633,33	4943	1,19	1992	2	4611,06	4844	0,00	1457	3	26,86
$C_{25,3}$	3_2	25	160	60	4644,26	5025	3,92	2869	2	4617,36	5037	4,34	2125	3	25,93
$C_{25,3}$	3_3	25	140	80	4460,20	4721	0,00	3279	2	4444,04	4721	0,00	2966	4	9,55
$C_{25,3}$	3_4	25	160	80	4475,90	4769	0,00	4745	2	4444,27	4733	0,00	2934	4	38,17
$C_{25,4}$	4_1	25	140	60	4541,91	4785	0,00	1772	2	4467,40	4767	0,00	1563	3	11,79
$C_{25,4}$	4_2	25	160	60	4595,36	4843	3,45	3021	2	4512,25	4785	0,87	2527	4	16,35
$C_{25,4}$	4_3	25	140	80	4352,44	4710	1,22	3863	2	4350,98	4703	0,72	3171	4	17,91
$C_{25,4}$	4_4	25	160	80	4358,69	4732	1,13	4892	2	4342,26	4690	0,00	2843	4	41,88
$C_{25,5}$	5_1	25	140	60	3975,09	4779	0,00	1798	2	3970,99	4779	0,00	2025	4	-12,63
$C_{25,5}$	5_2	25	160	60	3933,67	4780	0,00	4631	2	3878,90	4779	0,00	3734	6	19,37
$C_{25,5}$	5_3	25	140	80	3913,18	4760	0,00	3319	2	3863,17	4731	0,00	6604	8	-98,98
$C_{25,5}$	5_4	25	160	80	4062,75	4788	0,00	4788	2	3927,54	4775	2,50	4922	5	-2,80
$C_{25,6}$	6_1	25	140	50	3472,93	4755	0,00	2882	2	3413,54	4755	0,00	2511	4	12,87
$C_{25,6}$	6_2	25	160	50	3501,76	4808	25,70	4951	2	3436,72	4766	21,22	3635	5	26,58
$C_{25,6}$	6_3	25	140	70	3370,14	4729	1,87	4586	2	3326,59	4716	0,52	4914	5	-7,15
$C_{25,6}$	6_4	25	160	70	3364,36	4766	23,92	4940	2	3330,36	4580	0,00	3576	4	27,61
$C_{25,7}$	7_1	25	140	50	4104,92	5083	0,00	478	2	4104,92	5083	0,00	303	2	36,61
$C_{25,7}$	7_2	25	160	50	3997,05	4864	0,00	1958	2	3997,05	4864	0,00	911	3	53,47
$C_{25,7}$	7_3	25	140	70	3923,24	4772	0,00	2894	2	3923,24	4772	0,00	1845	4	36,25
$C_{25,7}$	7_4	25	160	70	3850,01	4792	2,84	4934	2	3797,32	4776	2,34	4437	5	10,07
$C_{25,8}$	8_1	25	140	50	3779,30	4889	0,00	2675	2	3733,95	4889	0,00	1997	4	25,35
$C_{25,8}$	8_2	25	160	50	3857,24	4928	16,10	4733	2	3694,81	4882	19,50	3383	5	28,52
$C_{25,8}$	8_3	25	160	50	3673,02	4798	0,00	4230	2	3619,05	4782	0,00	5281	6	-24,85
$C_{25,8}$	8_4	25	140	70	3625,41	4803	3,99	4948	2	3588,76	4732	1,04	6531	7	-31,99

3.6 Computational results

In this section, the computational experiments performed on benchmark instances adapted from Akca *et al.* [30] to evaluate the performance of the models discussed in this chapter both in terms of the quality of their lower bounds (with respect to the model (3.1)-(3.12) and (3.23)-(3.29), Section 3.6.1) and in their ability to drive efficiently the search for good quality integer solutions are reported. Each benchmark instance has five possible depots. The tests were run on a PC with an i7 CPU with 3.5 GHz and 32 GB of RAM. The optimization subroutines rely on CPLEX 12.5.

The different tests are based on various benchmark instances from the literature and relevant parameters are presented in the results tables. In the tables, the *Inst* column represents the name associated to the instance and the *n* column provides the number of clients. The W and Q columns represent the length of the workday and the vehicle capacity, respectively.

3.6.1 Solving the compact models exactly

For the experiments related to the model (3.1)-(3.12) and (3.23)-(3.29), a set of 40 benchmark instances from the literature was used whose relevant parameters are given in Table 3.5 and 3.6.

The tests are divided in two parts. First, the quality of the continuous lower bounds of the models (3.1)-(3.12) and (3.23)-(3.29) was compared without enforcing any other valid inequality. The results of these tests are listed in the Table 3.5. The columns z_{RL} and t_{RL} denote respectively the value of the lower bound and the computing time (in seconds) required for the solution of the linear programming relaxation of the corresponding model by CPLEX. Column lb represents the best lower bound obtained when the corresponding model is solved by CPLEX up to integrality using a maximum of 900 seconds of computation. The columns ub and t_{UB} denote respectively the value of the best incumbent found within this time limit, and the total execution time in seconds (which is smaller than 900 seconds only if a proven optimal solution has been found within the time limit). The column gap provides the value in percentage of the optimality gap reached at the end of the solution procedure. A "—" entry denotes the fact that no feasible integer solution was found. Finally, column t_g gives the total computing time required to generate the routes for the model (3.23)-(3.29). The second set of experiments is reported in Table 3.6. The same tests as above were repeated enforcing now the valid inequalities described in Section 3.4.2. For a fair evaluation, and since the first and second cut described in this section can also be enforced in model (3.1)-(3.12), this model was solved again using these two cuts.

Table 3.5: Results of 3-index model vs network flow model

					Model (3.1)-(3.12)						**Model (3.23)-(3.29)**						
	$Inst.$	n	W	Q	z_{RL}	t_{RL}	lb	ub	t_{UB}	gap	z_{RL}	t_g	t_{RL}	lb	ub	t_{UB}	gap
$C_{20,1}$	1_1	20	140	50	3151,24	0,62	3359,31	4671	900,15	28,08	3254,44	1,04	0,11	4431,00	4431	29,73	0,00
$C_{20,1}$	1_2	20	160	50	3095,01	0,59	3218,96	4656	900,20	30,86	3192,87	1,19	0,21	4332,58	4430	900,87	2,20
$C_{20,1}$	1_3	20	140	70	3071,06	0,64	3174,38	4618	900,17	31,26	3181,82	51,04	0,24	4335,69	4384	901,11	1,10
$C_{20,1}$	1_4	20	160	70	3017,67	0,64	3069,76	4399	902,73	30,22	3122,97	58,96	0,56	4187,46	4378	901,96	4,35
$C_{20,2}$	2_1	20	140	50	2856,71	0,61	2903,55	4764	900,20	39,05	3059,33	2,22	0,07	4681,00	4681	701,16	0,00
$C_{20,2}$	2_2	20	160	50	2790,07	0,64	2882,60	4780	902,99	39,69	2949,29	2,58	0,13	4472,00	4472	64,06	0,00
$C_{20,2}$	2_3	20	140	70	2760,71	0,64	2800,63	4674	902,24	40,08	2956,06	156,70	0,18	4424,00	4424	188,28	0,00
$C_{20,2}$	2_4	20	160	70	2698,69	0,62	2736,22	4697	900,18	41,75	2873,88	181,02	0,35	4406,04	4424	900,68	0,41
$C_{20,3}$	3_1	20	140	60	3471,53	0,66	3654,39	4684	900,17	21,98	3598,47	0,10	0,10	4431,00	4431	33,00	0,00
$C_{20,3}$	3_2	20	160	60	3417,93	0,59	3479,33	4909	900,22	29,12	3526,57	0,11	0,17	4354,51	4426	900,88	1,62
$C_{20,3}$	3_3	20	140	80	3411,59	0,66	3540,66	4633	900,20	23,58	3516,09	5,86	0,17	4305,53	4385	900,89	1,81
$C_{20,3}$	3_4	20	160	80	3357,59	0,66	3521,45	4445	900,17	20,78	3442,63	6,78	0,42	4198,45	4389	901,72	4,34
$C_{20,4}$	4_1	20	140	60	3431,68	0,59	3567,30	4740	906,10	24,74	3566,01	0,08	0,07	4507,57	4668	900,38	3,44
$C_{20,4}$	4_2	20	160	60	3379,11	0,61	3497,41	4894	900,22	28,54	3500,70	0,09	0,11	4431,08	4471	900,48	0,89
$C_{20,4}$	4_3	20	140	80	3350,75	0,62	3417,21	4638	902,77	26,32	3457,59	4,73	0,26	4409,00	4409	629,40	0,00
$C_{20,4}$	4_4	20	160	80	3299,21	0,64	3475,06	4437	900,15	21,68	3406,97	5,46	0,29	4229,27	4402	901,28	3,92
$C_{20,5}$	5_1	20	140	60	2929,15	0,64	3049,45	4951	900,15	38,41	3096,93	68,87	0,13	4674,00	4674	160,79	0,00
$C_{20,5}$	5_2	20	160	60	2867,78	0,61	2941,52	4727	900,17	37,77	3038,31	80,98	0,24	4475,00	4475	227,71	0,00
$C_{20,6}$	6_1	20	140	50	2508,09	0,61	2629,81	4590	902,04	42,71	2652,43	21,02	0,16	4420,00	4420	103,99	0,00
$C_{20,6}$	6_2	20	160	50	2455,01	0,61	2574,96	4797	901,00	46,32	2578,59	24,46	0,28	4312,04	4414	900,29	2,31
$C_{20,7}$	7_1	20	140	50	3020,74	0,66	3245,80	5137	900,18	36,82	3241,31	6,43	0,07	4835,00	4835	38,94	0,00
$C_{20,7}$	7_2	20	160	50	2949,96	0,62	3134,84	4969	901,71	36,91	3161,96	7,63	0,13	4675,55	4795	900,31	2,49
$C_{20,8}$	8_1	20	140	50	3058,84	0,58	3130,66	5146	901,45	39,16	3265,44	26,26	0,04	4744,00	4744	43,26	0,00
$C_{20,8}$	8_2	20	160	50	2984,49	0,59	3075,07	4882	900,18	37,01	3186,08	31,27	0,07	4656,00	4656	67,14	0,00
$C_{25,1}$	1_1	25	140	50	3778,16	1,05	3792,85	-	909,58	-	3903,70	10,87	0,24	4682,11	4751	901,15	1,45
$C_{25,1}$	1_2	25	160	50	3710,46	1,00	3722,81	6712	900,26	44,54	3833,08	12,49	0,38	4554,87	4646	900,81	1,96
$C_{25,2}$	2_1	25	140	50	3378,47	1,03	3389,69	5506	909,45	38,44	3615,80	16,32	0,17	4764,00	4764	116,82	0,00
$C_{25,2}$	2_2	25	160	50	3305,20	0,97	3334,35	-	913,82	-	3507,23	19,12	0,57	4598,88	4764	901,47	3,47
$C_{25,3}$	3_1	25	140	60	4283,93	1,01	4322,40	7384	900,43	41,46	4456,56	1,35	0,15	4817,00	4817	46,43	0,00
$C_{25,3}$	3_2	25	160	60	4213,33	1,01	4276,19	6876	907,18	37,81	4376,88	1,56	0,45	4683,61	4803	901,29	2,49
$C_{25,4}$	4_1	25	140	60	4150,87	1,03	4178,40	5409	900,26	22,75	4297,64	0,79	0,19	4669,18	4767	901,04	2,05
$C_{25,4}$	4_2	25	160	60	4085,55	1,03	4092,00	7110	900,19	42,45	4214,16	0,91	0,33	4527,01	4767	901,20	5,03
$C_{25,5}$	5_1	25	140	60	3654,99	1,01	3691,40	7585	905,24	51,33	3877,60	384,26	0,25	4779,00	4779	560,22	0,00
$C_{25,5}$	5_2	25	160	60	3577,31	1,00	3615,25	7420	905,71	51,28	3773,96	441,71	0,64	4643,14	4774	900,23	2,74
$C_{25,6}$	6_1	25	140	50	3159,18	1,01	3166,00	6910	900,24	54,18	3354,05	344,90	0,33	4670,22	4755	900,15	1,78
$C_{25,6}$	6_2	25	160	50	3092,86	0,99	3131,67	6633	913,87	52,79	3253,57	396,97	0,76	3503,94	4777	901,90	26,65
$C_{25,7}$	7_1	25	140	50	3766,16	0,98	3780,75	-	900,20	-	4036,95	70,72	0,17	5083,00	5083	269,30	0,00
$C_{25,7}$	7_2	25	160	50	3682,22	0,97	3709,32	-	904,97	-	3929,26	82,49	0,20	4864,00	4864	139,04	0,00
$C_{25,8}$	8_1	25	140	50	3422,25	1,01	3528,78	7501	900,25	52,96	3623,33	244,34	0,29	4889,00	4889	439,84	0,00
$C_{25,8}$	8_2	25	160	50	3355,86	1,03	3527,48	6688	901,62	47,26	3516,24	282,99	0,59	3898,86	4970	901,52	21,55

Table 3.6: Results of 3-index model vs network flow model with additional valid inequalities described in Section 3.4.2

					Model (3.1)-(3.12)						**Model (3.23)-(3.29)**						
	$Inst.$	n	W	Q	z_{RL}	t_{RL}	lb	ub	t_{UB}	gap	z_{RL}	t_g	t_{RL}	lb	ub	t_{UB}	gap
$C_{20,1}$	1_1	20	140	50	4102,90	0,75	4243,68	4821	900,19	11,98	4220,60	1,10	0,14	4431,00	4431	21,48	0,00
$C_{20,1}$	1_2	20	160	50	4048,67	0,72	4180,86	4684	900,17	10,74	4177,21	1,17	0,25	4320,23	4430	900,50	2,48
$C_{20,1}$	1_3	20	140	70	4041,16	0,64	4134,01	4676	900,17	11,59	4161,58	50,60	0,35	4327,69	4384	900,91	1,28
$C_{20,1}$	1_4	20	160	70	3990,65	0,59	4067,83	4621	900,15	11,97	4126,90	58,23	0,63	4187,73	4374	901,76	4,26
$C_{20,2}$	2_1	20	140	50	4254,64	0,62	4310,68	4788	902,31	9,97	4452,00	2,20	0,07	4681,00	4681	400,36	0,00
$C_{20,2}$	2_2	20	160	50	4189,73	0,59	4246,68	4479	900,25	5,19	4347,02	2,56	0,16	4472,00	4472	22,15	0,00
$C_{20,2}$	2_3	20	140	70	4177,13	0,61	4210,62	4753	900,18	11,41	4346,06	155,02	0,24	4424,00	4424	159,18	0,00
$C_{20,2}$	2_4	20	160	70	4115,92	0,66	4150,27	4714	900,18	11,96	4269,82	180,26	0,38	4372,46	4424	894,55	1,17
$C_{20,3}$	3_1	20	140	60	4091,69	0,58	4193,51	4716	900,22	11,08	4236,77	0,10	0,11	4431,00	4431	8,28	0,00
$C_{20,3}$	3_2	20	160	60	4038,11	0,61	4125,63	4718	900,22	12,56	4178,71	0,11	0,19	4364,48	4426	900,57	1,39
$C_{20,3}$	3_3	20	140	80	4042,85	0,64	4137,37	4418	900,14	6,35	4163,41	5,82	0,22	4276,15	4385	900,15	2,48
$C_{20,3}$	3_4	20	160	80	3992,55	0,69	4072,23	4405	900,25	7,55	4134,48	6,76	0,33	4196,36	4387	900,09	4,35
$C_{20,4}$	4_1	20	140	60	4080,59	0,59	4228,23	5054	905,02	16,34	4231,15	0,08	0,09	4537,00	4537	55,04	0,00
$C_{20,4}$	4_2	20	160	60	4029,93	0,59	4176,39	4711	903,01	11,35	4188,32	0,09	0,16	4440,73	4471	900,47	0,68
$C_{20,4}$	4_3	20	140	80	4014,69	0,69	4105,45	4633	903,46	11,39	4152,15	4,63	0,20	4330,24	4409	900,12	1,79
$C_{20,4}$	4_4	20	160	80	3967,37	0,64	4056,60	4503	905,08	9,91	4138,81	5,40	0,36	4234,47	4402	900,89	3,81
$C_{20,5}$	5_1	20	140	60	4160,48	0,64	4255,33	5062	900,18	15,94	4333,73	68,61	0,10	4674,00	4674	135,92	0,00
$C_{20,5}$	5_2	20	160	60	4103,24	0,80	4183,77	5091	900,18	17,82	4284,75	80,26	0,24	4475,00	4475	87,88	0,00
$C_{20,6}$	6_1	20	140	50	4080,34	0,72	4203,90	4719	908,30	10,92	4226,93	21,07	0,18	4420,00	4420	32,62	0,00
$C_{20,6}$	6_2	20	160	50	4027,86	0,66	4125,87	5028	900,22	17,94	4177,64	24,48	0,37	4347,47	4414	901,06	1,51
$C_{20,7}$	7_1	20	140	50	4328,32	0,62	4424,90	5050	907,35	12,38	4568,72	6,47	0,08	4835,00	4835	32,54	0,00
$C_{20,7}$	7_2	20	160	50	4257,48	0,64	4432,56	4897	900,17	9,48	4485,41	7,63	0,17	4670,36	4795	900,65	2,60
$C_{20,8}$	8_1	20	140	50	4327,32	0,62	4389,74	5073	900,20	13,47	4514,76	25,94	0,05	4744,00	4744	27,60	0,00
$C_{20,8}$	8_2	20	160	50	4254,49	0,64	4373,42	4656	901,26	6,07	4439,88	30,69	0,09	4656,00	4656	48,13	0,00
$C_{25,1}$	1_1	25	140	50	4269,12	1,11	4275,27	-	902,31	-	4398,97	10,73	0,27	4675,25	4751	900,65	1,59
$C_{25,1}$	1_2	25	160	50	4201,42	1,09	4203,27	-	900,20	-	4329,62	12,35	0,49	4552,01	4646	900,12	2,02
$C_{25,2}$	2_1	25	140	50	4357,11	0,98	4447,17	5516	901,73	19,38	4580,81	16,27	0,20	4764,00	4764	89,45	0,00
$C_{25,2}$	2_2	25	160	50	4285,83	1,03	4369,81	-	903,68	-	4478,98	18,92	0,80	4599,30	4759	901,32	3,36
$C_{25,3}$	3_1	25	140	60	4347,52	1,00	4407,50	7265	900,28	39,33	4514,70	1,35	0,17	4817,00	4817	458,31	0,00
$C_{25,3}$	3_2	25	160	60	4276,64	0,97	4360,26	-	900,22	-	4435,38	1,56	0,32	4671,01	4800	901,01	2,69
$C_{25,4}$	4_1	25	140	60	4271,49	1,03	4328,35	-	901,35	-	4420,88	0,79	0,21	4654,18	4767	900,61	2,37
$C_{25,4}$	4_2	25	160	60	4206,29	1,01	4230,27	-	912,18	-	4351,79	0,90	0,39	4526,00	4767	901,09	5,06
$C_{25,5}$	5_1	25	140	60	4374,35	1,06	4403,43	5767	900,28	23,64	4581,74	382,16	0,29	4779,00	4779	387,52	0,00
$C_{25,5}$	5_2	25	160	60	4300,78	1,12	4377,00	7535	900,25	41,91	4493,35	443,39	1,16	4703,91	4774	901,54	1,47
$C_{25,6}$	6_1	25	140	50	4263,53	1,01	4278,66	9817	900,20	56,42	4458,98	355,09	0,36	4655,75	4755	900,06	2,09
$C_{25,6}$	6_2	25	160	50	4198,13	1,05	4229,79	-	900,22	-	4373,88	403,45	0,81	4557,76	4606	900,11	1,05
$C_{25,7}$	7_1	25	140	50	4492,47	1,01	4667,72	-	900,23	-	4754,01	71,59	0,12	5083,00	5083	338,08	0,00
$C_{25,7}$	7_2	25	160	50	4408,97	1,05	4463,85	-	909,89	-	4646,35	83,54	0,26	4864,00	4864	149,90	0,00
$C_{25,8}$	8_1	25	140	50	4346,90	1,01	4587,28	-	900,22	-	4562,43	247,52	0,36	4889,00	4889	302,53	0,00
$C_{25,8}$	8_2	25	160	50	4280,69	1,00	4385,18	5451	902,48	19,55	4476,18	285,20	0,66	4711,41	4880	900,25	3,45

3.6.2 Solving the column generation model

The results presented in Tables 3.7 and 3.8 were obtained through 64 benchmarks instances from the literature for the column generation model presented in Section 3.3. The results in Table 3.7 refer to instances with 20 clients and the Table 3.8 presents the results for the instances with 25 clients.

In order to evaluate the model, three different tests were performed (*run #1*, *run #2* and *run #3*). The total time of *run #1* was limited to 900 seconds, considering that 300 seconds

Table 3.7: Column generation results for instances with 20 clients

					run #1					run #2					run #3				
	$Inst.$	n	W	Q	z_{RL}	MIP	gap	t_{total}	$it.$	z_{RL}	MIP	gap	t_{total}	$it.$	z_{RL}	MIP	gap	t_{total}	$it.$
$C_{20,1}$	1_1	20	140	50	3381,93	4431	0,00	239	3	3381,93	4431	0,00	341	2	3381,93	4431	0,00	340	2
$C_{20,1}$	1_2	20	160	50	3325,50	4430	0,00	463	4	3325,50	4430	0,21	910	6	3325,50	4430	0,00	868	3
$C_{20,1}$	1_3	20	140	70	3281,94	4384	0,09	818	4	3277,98	4384	0,00	1018	3	3277,98	4384	0,01	1832	3
$C_{20,1}$	1_4	20	160	70	3237,30	4374	0,02	929	5	3234,44	4374	0,24	1122	2	3251,05	4374	0,02	3374	5
$C_{20,2}$	2_1	20	140	50	3114,42	4681	0,00	120	2	3114,42	4681	0,00	114	2	3114,42	4681	0,00	114	2
$C_{20,2}$	2_2	20	160	50	3031,03	4472	0,00	272	4	3031,03	4472	0,00	399	2	3031,03	4472	0,00	415	2
$C_{20,2}$	2_3	20	140	70	2992,65	4424	0,00	388	2	2992,65	4424	0,00	491	2	2992,65	4424	0,00	425	2
$C_{20,2}$	2_4	20	160	70	2921,72	4428	0,00	902	4	2921,72	4431	0,30	956	1	2921,72	4424	0,00	1635	3
$C_{20,3}$	3_1	20	140	60	3789,91	4651	0,03	424	2	3751,96	4431	0,00	475	2	3751,96	4431	0,00	473	2
$C_{20,3}$	3_2	20	160	60	3767,75	4426	0,00	531	6	3674,60	4426	0,00	1037	3	3674,23	4426	0,00	1472	3
$C_{20,3}$	3_3	20	140	80	3629,78	4385	0,00	431	3	3610,87	4385	0,01	913	2	3610,87	4385	0,00	1217	2
$C_{20,3}$	3_4	20	160	80	3620,68	4391	0,00	583	4	3620,68	4401	0,14	926	2	3620,68	4385	0,00	1733	3
$C_{20,4}$	4_1	20	140	60	3750,33	4537	0,00	139	2	3750,33	4537	0,00	135	2	3750,33	4537	0,00	136	2
$C_{20,4}$	4_2	20	160	60	3765,94	4493	0,00	289	3	3666,42	4471	0,00	709	3	3666,42	4471	0,00	714	2
$C_{20,4}$	4_3	20	140	80	3600,64	4409	0,00	479	5	3579,67	4409	0,00	852	4	3579,67	4409	0,00	674	2
$C_{20,4}$	4_4	20	160	80	3605,77	4409	0,00	495	4	3556,41	4405	0,16	903	2	3556,34	4405	0,00	2364	4
$C_{20,5}$	5_1	20	140	60	3255,44	4674	0,00	151	2	3255,44	4674	0,00	145	2	3255,44	4674	0,00	147	2
$C_{20,5}$	5_2	20	160	60	3174,13	4475	0,00	433	4	3174,13	4475	0,00	815	3	3174,13	4475	0,00	795	4
$C_{20,5}$	5_3	20	140	80	3122,17	4454	0,00	601	4	3122,59	4452	0,00	698	2	3122,17	4449	0,00	686	2
$C_{20,5}$	5_4	20	160	80	3043,76	4413	0,00	850	4	3091,88	4438	0,00	892	1	3036,41	4401	0,00	3174	5
$C_{20,6}$	6_1	20	140	50	2734,17	4420	0,00	468	4	2734,17	4420	0,00	534	2	2734,17	4420	0,00	533	2
$C_{20,6}$	6_2	20	160	50	2696,97	4424	0,27	778	5	2726,12	4421	0,30	946	2	2683,96	4415	0,31	2668	4
$C_{20,6}$	6_3	20	140	70	2682,20	4402	0,00	848	4	2668,29	4383	0,00	887	1	2652,23	4383	0,00	4238	5
$C_{20,6}$	6_4	20	160	70	2623,34	4398	0,03	1029	4	2651,02	4439	0,39	956	1	2586,95	4386	0,27	4918	5
$C_{20,7}$	7_1	20	140	50	3306,31	4835	0,00	133	2	3306,31	4835	0,00	130	2	3306,31	4835	0,00	134	2
$C_{20,7}$	7_2	20	160	50	3226,85	4795	0,00	301	4	3226,85	4795	0,00	548	2	3226,85	4795	0,00	676	2
$C_{20,7}$	7_3	20	140	70	3192,57	4751	0,00	501	3	3192,57	4748	0,00	509	2	3192,57	4748	0,00	510	2
$C_{20,7}$	7_4	20	160	70	3128,78	4589	0,00	962	4	3128,56	4577	0,24	958	1	3128,56	4548	0,00	1733	2
$C_{20,8}$	8_1	20	140	50	3355,59	4744	0,00	24	2	3355,59	4744	0,00	18	2	3355,59	4744	0,00	25	2
$C_{20,8}$	8_2	20	160	50	3248,33	4656	0,00	50	2	3248,33	4656	0,00	47	2	3248,33	4656	0,00	47	2
$C_{20,8}$	8_3	20	160	50	3169,33	4587	0,00	73	2	3169,33	4587	0,00	68	2	3169,33	4587	0,00	69	2
$C_{20,8}$	8_4	20	140	70	3101,05	4451	0,00	268	3	3101,05	4451	0,00	354	2	3101,05	4451	0,00	351	2

(from the 900) were reserved for the search of the integer solution. In each iteration and for each depot, 120 seconds were used to determine the single-trips and 60 to the multi-trips. Regarding the *run #2* the total time was also 900 seconds being 350 for the integer solution. In each iteration, for each depot, the single-trips and multi-trips routines were limited to 350 seconds. Finally, *run #3* had a total execution time of 7200 seconds with 900 designated for the search of the integer solution and the search of single-trips and multi-trips was limited to 400 seconds.

The column z_{RL} denotes the value of the linear relaxation of the column generation model and the MIP column includes the values of the best integer solution obtained during the allowed maximum time for each run. The column gap provides the value in percentage of the optimality gap reached when the solution is found. Column t_{total} is related to the total time of the algorithm and it represents the total number of iterations.

Table 3.8: Column generation results for instances with 25 clients

					run #1					run #2					run #3				
	Inst.	n	W	Q	z_{RL}	MIP	gap	t_{total}	$it.$	z_{RL}	MIP	gap	t_{total}	$it.$	z_{RL}	MIP	gap	t_{total}	$it.$
$C_{25,1}$	1_1	25	140	50	4047,63	4751	0,00	927	8	3992,74	4751	0,02	902	2	3992,74	4751	0,00	1638	4
$C_{25,1}$	1_2	25	160	50	4062,54	4769	0,01	643	4	4024,48	4758	0,13	1095	3	3959,84	4760	0,14	3432	6
$C_{25,1}$	1_3	25	140	70	3965,77	4694	0,03	1030	4	3993,49	4703	0,14	958	1	3951,54	4687	0,03	4117	4
$C_{25,1}$	1_4	25	160	70	3971,46	4709	0,03	1030	4	4017,17	4734	0,14	960	1	3936,78	4690	0,03	4122	4
$C_{25,2}$	2_1	25	140	50	3660,07	4764	0,00	447	3	3660,07	4764	0,00	653	2	3660,07	4764	0,00	923	3
$C_{25,2}$	2_2	25	160	50	3626,52	4764	0,02	865	6	3619,53	4765	0,19	973	2	3601,23	4765	0,03	3064	5
$C_{25,2}$	2_3	25	140	70	3602,74	4749	0,00	934	4	3585,85	4718	0,02	957	1	3534,76	4706	0,01	4015	4
$C_{25,2}$	2_4	25	160	70	3573,38	4770	0,03	1029	4	3597,06	4772	0,23	958	1	3489,88	4501	0,00	4499	5
$C_{25,3}$	3_1	25	140	60	4649,71	5038	0,04	564	4	4633,33	4944	0,02	1123	3	4611,06	4844	0,00	1457	3
$C_{25,3}$	3_2	25	160	60	4771,34	5275	0,06	568	4	4617,36	5036	0,05	1126	3	4617,36	5037	0,04	2125	3
$C_{25,3}$	3_3	25	140	80	4461,20	4729	0,00	856	4	4460,29	4726	0,00	880	1	4444,04	4721	0,00	2966	4
$C_{25,3}$	3_4	25	160	80	4444,41	4745	0,00	1000	4	4483,39	4770	0,00	962	1	4444,27	4733	0,00	2934	4
$C_{25,4}$	4_1	25	140	60	4631,66	5015	0,03	495	3	4508,58	4790	0,03	1118	3	4467,40	4767	0,00	1563	3
$C_{25,4}$	4_2	25	160	60	4673,82	5023	0,05	624	5	4595,35	4842	0,03	1121	3	4512,25	4785	0,01	2527	4
$C_{25,4}$	4_3	25	140	80	4351,66	4694	0,00	879	4	4356,81	4731	0,04	961	1	4350,98	4703	0,01	3171	4
$C_{25,4}$	4_4	25	160	80	4342,26	4701	0,00	911	4	4372,61	4745	0,03	960	1	4342,26	4690	0,00	2843	4
$C_{25,5}$	5_1	25	140	60	3975,09	4779	0,00	721	5	3971,18	4779	0,00	1076	2	3970,99	4779	0,00	2025	4
$C_{25,5}$	5_2	25	160	60	3914,68	4795	0,00	854	4	3981,08	4826	0,12	971	1	3878,90	4779	0,00	3734	6
$C_{25,5}$	5_3	25	140	80	3919,15	4786	0,00	846	4	3940,38	4772	0,00	723	1	3863,17	4731	0,00	6604	8
$C_{25,5}$	5_4	25	160	80	3970,38	4792	0,02	1030	4	4097,89	4835	0,02	958	1	3927,54	4775	0,03	4922	5
$C_{25,6}$	6_1	25	140	50	3472,93	4755	0,00	756	4	3472,93	4755	0,01	935	1	3413,54	4755	0,00	2511	4
$C_{25,6}$	6_2	25	160	50	3450,01	4771	0,02	934	4	3536,20	4828	0,24	958	1	3436,72	4766	0,21	3635	5
$C_{25,6}$	6_3	25	140	70	3392,74	4753	0,00	935	4	3407,58	4746	0,02	955	1	3326,59	4716	0,01	4914	5
$C_{25,6}$	6_4	25	160	70	3367,24	4585	0,00	1015	4	3413,04	4780	0,04	957	1	3330,36	4580	0,00	3576	4
$C_{25,7}$	7_1	25	140	50	4104,92	5083	0,00	253	2	4104,92	5083	0,00	303	2	4104,92	5083	0,00	303	2
$C_{25,7}$	7_2	25	160	50	3997,05	4864	0,00	523	4	3997,05	4864	0,00	883	4	3997,05	4864	0,00	911	3
$C_{25,7}$	7_3	25	140	70	3923,87	4780	0,00	815	4	3931,65	4774	0,00	749	1	3923,24	4772	0,00	1845	4
$C_{25,7}$	7_4	25	160	70	3851,48	4811	0,02	1029	4	3892,52	4822	0,18	957	1	3797,32	4776	0,02	4437	5
$C_{25,8}$	8_1	25	140	50	3733,95	4913	0,00	918	6	3758,66	4889	0,18	1026	2	3733,95	4889	0,00	1997	4
$C_{25,8}$	8_2	25	160	50	3738,51	4897	0,17	1037	5	3954,45	5139	0,21	959	1	3694,81	4882	0,19	3383	5
$C_{25,8}$	8_3	25	160	50	3637,73	4786	0,00	919	4	3694,40	4808	0,00	921	1	3619,05	4782	0,00	5281	6
$C_{25,8}$	8_4	25	140	70	3605,41	4788	0,04	1031	4	3660,67	4798	0,02	960	1	3588,76	4732	0,01	6531	7

3.6.3 Comparative analysis

From the results obtained for the two compact models, it is clear that the continuous lower bound of (3.23)-(3.29) is better than the bound of (3.1)-(3.12), both with and without enforcing additional inequalities. Furthermore, the network flow formulation (3.23)-(3.29) proved to be much more effective than (3.1)-(3.12) in finding good quality integer solutions. In some cases, the model of Akca *et al.* [30] fails even in finding a feasible solution (represented by symbol '—' in Table 3.9), when (3.23)-(3.29) provides an optimal integer solution.

The comparisons are performed between the different executions of the column generation model and the network flow model, since the network flow model (3.23)-(3.29) proved to be more efficient than the three-index commodity flow model (3.1)-(3.12) in finding good integer solutions. Nevertheless, the results regarding the model (3.1)-(3.12) will remain on Table 3.9 in order to demonstrate that the integer solutions presented by this model are of lesser quality than those presented by the other models under analysis.

In order to perform this comparison, three different symbols are used. The symbol * means the solution is an optimal integer solution. When a value has the symbol $^\downarrow$ associated, it means

Table 3.9: Comparative analysis of the presented models

					CG: run #1		CG: run #2		CG: run #3		Model (3.1)-(3.12)		Model (3.23)-(3.29)	
	$Inst.$	n	W	Q	MIP	gap	MIP	gap	MIP	gap	MIP	gap	MIP	gap
$C_{20,1}$	1_1	20	140	50	4431*	0,00	4431*	0,00	4431*	0,00	4671	28,08	4431*	0,00
$C_{20,1}$	1_2	20	160	50	4430	0,00	4430	0,21	4430	0,00	4656	30,86	4430	2,20
$C_{20,1}$	1_3	20	140	70	4384	0,09	4384	0,00	4384	0,01	4618	31,26	4384	1,10
$C_{20,1}$	1_4	20	160	70	4374$^\downarrow$	0,02	4374$^\downarrow$	0,24	4374$^\downarrow$	0,02	4399	30,22	4378	4,35
$C_{20,2}$	2_1	20	140	50	4681*	0,00	4681*	0,00	4681*	0,00	4764	39,05	4681*	0,00
$C_{20,2}$	2_2	20	160	50	4472*	0,00	4472*	0,00	4472*	0,00	4780	39,69	4472*	0,00
$C_{20,2}$	2_3	20	140	70	4424*	0,00	4424*	0,00	4424*	0,00	4674	40,08	4424*	0,00
$C_{20,2}$	2_4	20	160	70	4428$^\uparrow$	0,00	4431$^\uparrow$	0,30	4424	0,00	4697	41,75	4424	0,41
$C_{20,3}$	3_1	20	140	60	4651$^\uparrow$	0,03	4431*	0,00	4431*	0,00	4684	21,98	4431*	0,00
$C_{20,3}$	3_2	20	160	60	4426	0,00	4426	0,00	4426	0,00	4909	29,12	4426	1,62
$C_{20,3}$	3_3	20	140	80	4385	0,00	4385	0,01	4385	0,00	4633	23,58	4385	1,81
$C_{20,3}$	3_4	20	160	80	4391$^\uparrow$	0,00	4401$^\uparrow$	0,14	4385$^\downarrow$	0,00	4445	20,78	4389	4,34
$C_{20,4}$	4_1	20	140	60	4537$^\downarrow$	0,00	4537$^\downarrow$	0,00	4537$^\downarrow$	0,00	4740	24,74	4668	3,44
$C_{20,4}$	4_2	20	160	60	4493$^\uparrow$	0,00	4471	0,00	4471	0,00	4894	28,54	4471	0,89
$C_{20,4}$	4_3	20	140	80	4409*	0,00	4409*	0,00	4409*	0,00	4638	26,32	4409*	0,00
$C_{20,4}$	4_4	20	160	80	4409$^\uparrow$	0,00	4405$^\uparrow$	0,16	4405$^\uparrow$	0,00	4437	21,68	4402	3,92
$C_{20,5}$	5_1	20	140	60	4674*	0,00	4674*	0,00	4674*	0,00	4951	38,41	4674*	0,00
$C_{20,5}$	5_2	20	160	60	4475*	0,00	4475*	0,00	4475*	0,00	4727	37,77	4475*	0,00
$C_{20,6}$	6_1	20	140	50	4420*	0,00	4420*	0,00	4420*	0,00	4590	42,71	4420*	0,00
$C_{20,6}$	6_2	20	160	50	4424$^\uparrow$	0,27	4421$^\uparrow$	0,30	4415$^\uparrow$	0,31	4797	46,32	4414	2,31
$C_{20,7}$	7_1	20	140	50	4835*	0,00	4835*	0,00	4835*	0,00	5137	36,82	4835*	0,00
$C_{20,7}$	7_2	20	160	50	4795	0,00	4795	0,00	4795	0,00	4969	36,91	4795	2,49
$C_{20,8}$	8_1	20	140	50	4744*	0,00	4744*	0,00	4744*	0,00	5146	39,16	4744*	0,00
$C_{20,8}$	8_2	20	160	50	4656*	0,00	4656*	0,00	4656*	0,00	4882	37,01	4656*	0,00
$C_{25,1}$	1_1	25	140	50	4751	0,00	4751	0,02	4751	0,00	-	-	4751	1,45
$C_{25,1}$	1_2	25	160	50	4769$^\uparrow$	0,01	4758$^\uparrow$	0,13	4760$^\uparrow$	0,14	6712	44,54	4646	1,96
$C_{25,2}$	2_1	25	140	50	4764*	0,00	4764*	0,00	4764*	0,00	5506	38,44	4764*	0,00
$C_{25,2}$	2_2	25	160	50	4764$^\uparrow$	0,02	4765$^\uparrow$	0,19	4765$^\uparrow$	0,03	-	-	4764	3,47
$C_{25,3}$	3_1	25	140	60	5038$^\uparrow$	0,04	4944$^\uparrow$	0,02	4844$^\uparrow$	0,00	7384	41,46	4817*	0,00
$C_{25,3}$	3_2	25	160	60	5275$^\uparrow$	0,06	5036$^\uparrow$	0,05	5037$^\uparrow$	0,04	6876	37,81	4803	2,49
$C_{25,4}$	4_1	25	140	60	5015$^\uparrow$	0,03	4790$^\uparrow$	0,03	4767	0,00	5409	22,75	4767	2,05
$C_{25,4}$	4_2	25	160	60	5023$^\uparrow$	0,05	4842$^\uparrow$	0,03	4785$^\uparrow$	0,01	7110	42,45	4767	5,03
$C_{25,5}$	5_1	25	140	60	4779*	0,00	4779*	0,00	4779*	0,00	7585	51,33	4779*	0,00
$C_{25,5}$	5_2	25	160	60	4795$^\uparrow$	0,00	4826$^\uparrow$	0,12	4779$^\uparrow$	0,00	7420	51,28	4774	2,74
$C_{25,6}$	6_1	25	140	50	4755	0,00	4755	0,01	4755	0,00	6910	54,18	4755	1,78
$C_{25,6}$	6_2	25	160	50	4771$^\uparrow$	0,02	4828$^\uparrow$	0,24	4766$^\uparrow$	0,21	6633	52,79	4777	26,65
$C_{25,7}$	7_1	25	140	50	5083*	0,00	5083*	0,00	5083*	0,00	-	-	5083*	0,00
$C_{25,7}$	7_2	25	160	50	4864*	0,00	4864*	0,00	4864*	0,00	-	-	4864*	0,00
$C_{25,8}$	8_1	25	140	50	4913$^\uparrow$	0,00	4889*	0,18	4889*	0,00	7501	52,96	4889*	0,00
$C_{25,8}$	8_2	25	160	50	4897$^\uparrow$	0,17	5139$^\uparrow$	0,21	4882$^\uparrow$	0,19	6688	47,26	4970	21,55

this value is lower than the one presented by the model (3.23)-(3.29). Since this problem is a minimization one, the symbol $^{\downarrow}$ is used when a better integer solution is found. On the other hand, when the symbol $^{\uparrow}$ is displayed it means that the integer solution found is worse than the one presented by the model (3.23)-(3.29).

For 18 instances, model (3.23)-(3.29) provides the optimal solution within the time limit, while for the other cases solutions with very small optimality gaps are given. The column generation model is able to find always a valid solution, even during executions with smaller time limits than those presented in Section 3.6.2. For the first execution (*run #1*), the column generation model was able to find the optimal solution for 15 instances. There are also 2 better and 16 worse integer solutions during the 900 seconds allowed. If one considers the execution named *run #2*, the column generation model provides the optimal solution for 17 instances within the 900 seconds also used in the compact models. As presented in Table 3.9, in the execution *run #2* there are also 2 other better solutions comparing to the compact models and 4 worse ones for the instances with 20 clients when compared with the more effective model, the network flow model. For the 25 clients instances, only 9 results are worse than those presented by the same model. For the last execution (*run #3*), which allowed higher execution times (7200 seconds), there are 3 better solutions found, although 10 are worse. This execution was able to find 17 optimal integer solutions.

The column generation model is competitive with the network flow model for instances with 20 clients. However, for larger instances it is no longer a good approach due to the complexity inherent to the sub-problem.

3.7 Conclusions

The MLRP is a management science problem which can be solved using different approaches. This type of problem commonly occurs in the logistics and transportation fields and combines two different optimization problems. In the first problem, the FLP, the set of facilities that can be used to serve the clients is determined. In the second problem, the MVRP, a set of valid routes to fulfill the clients demands is obtained. The integration of these two difficult problems is important and leads to better solutions than solving them independently, since it considers the global system. This problem presents a particular issue since a vehicle can now perform more than a single route in the planning horizon.

MLRP consists in the selection of a set of depots to be opened and the determination of a set of routes used to serve all clients. These routes should be assigned to a vehicle and an opened depot. The main goal is to minimize the costs associated to the entire system. In this chapter, three integer programming models to solve the MLRP were presented.

The three-index commodity flow model is a graph-based model which includes a higher number of variables, since it allows for the model to take into account the variables related to the vehicles usage explicitly. The graph is defined in an explicitly way and there are variables associated to the usage of an arc and vehicle, and others that represents the flow through the arc for a specific vehicle. The column generation approach is divided in the RMP that includes the general constraints associated to the FLP and the sub-problem that groups the constraints which have a special structure concerning the ESPPRC. In the RMP, only a subset of variables is considered, being determined by the sub-problem. These two different type of problems exchange information in order to find the global optimal solution. The ESPPRC finds valid single-trips and then new multi-trips are obtained through the recombination of the valid single-trips and the new multi-trips. Network flow model is also a graph-based structure but instead of having nodes associated to clients or depots, the vertices represent time instants. This technique is less intuitive but proved to be efficient when compared with other approaches.

Several computational experiments were conducted based on a set of benchmark instances from the literature. The column generation model and the network flow model proved to be more efficient than the three-index commodity flow model, leading to better integer solutions in shorter computational times. The three-index commodity flow model sometimes fails when attempting to obtain integer solutions, even in finding a feasible one. The column generation model is competitive with the network flow model for instances with less than 20 clients. However, instances with more customers significantly increase the complexity of the sub-problem. This complexity is highlighted due to the higher number of sub-problem nodes. The search for elementary paths becomes more exhaustive.

Chapter 4

The multi-trip location routing problem: heuristic and hybrid approaches[1]

Outline

[1] The results of this chapter were published in:

[59] R. Macedo, C. Alves, S. Hanafi, B. Jarboui, N. Mladenović, B. Ramos, and J. Valério de Carvalho, "Skewed general variable neighborhood search for the location routing scheduling problem," *Computers & Operations Research*, vol. 61, pp. 143–152, 2015

[55] R. Macedo, B. Ramos, C. Alves, J. V. de Carvalho, S. Hanafi, and N. Mladenović, "Integer Programming Based Approaches for Multi-Trip Location Routing." Springer International Publishing, 2016, pp. 79–90

4.1 Introduction

The LRP is a complex problem which has deserved special attention during the last years. Several authors defined different approaches according to the peculiarities of the variant they considered. These characteristics have great influence on the solution method that should be used. Approaches using exact methods, such as those presented in the previous chapter (Chapter 3), ensure the optimal solution of the problem is found. However, when solving large scale instances, the computational time may be too long for real supply chain applications. In practice, it becomes unpractical to wait for a long time for the result of some exact method, since there is a frequent need to recalculate new distribution and collection routes. Such variations often occur weekly or even daily, according to the type of the product. For instance, hard planning methods may not be adequate for the management of perishable goods. In order to suppress this requirement, heuristics methods are used.

Thus, researchers have been studying heuristic methods which can usually get solutions very close to the optimal, since these can find a valid solution in an acceptable computational time. It is important to consider the trade-off between obtaining the optimal solution in higher computational times or getting a good valid solution in lower computational times. It is also possible to consider a hybridization of these two approaches where only a part of the problem is solved to optimality, and heuristic methods are used to solve the other part of the problem. Indeed, some heuristic methods are based on exact methods. Several authors limit the computational time and change some parameters to accelerate the problem convergence. The LRP includes the calculation of routes. However, this process is frequently simplified.

Many authors use clustering approaches, while others apply methods that simplify the generation of routes such as combining the routes in ascending order of customer indices such as Akca *et al.* [30]. For instance, Barreto *et al.* [21] present a cluster analysis based on a sequential heuristic for the LRP. The authors group the clients and then determine the most adequate route in the defined cluster. Then, they improve the route through a local search method and finally associate it to a depot. Like the rounding heuristic presented in this chapter, some authors define heuristics where the values of the linear relaxation variables are refined. Local search heuristics are also frequently used. These exploit a given neighborhood with the aim of finding better solutions than the current one. In [60], Macedo *et al.* proposed a meta-heuristic, the Variable Neighborhood Search for the LRP, where different neighborhoods are exploited until a good solution is found. The neighborhood is defined according to the peculiarities of the problem.

Other authors use heuristic methods that are based on behaviors of the nature. Derbel *et al.* [61] select a Genetic Algorithm with a local search method to solve the LRP. Their algorithm is based on analogies with the human genetic process. The authors defined genetic operators for their specific problem so the new population may inherit characteristics of their progenitors and still have its own singularity.

All of these approaches and others are valid, but some lead to better results concerning the execution times and the value obtained in the objective function. The main objective is to have a good trade-off between the objective function value and the computational time needed to solve the problem. Thus, it is necessary to analyze various solution methods and optimization techniques.

4.2 An iterative rounding heuristic

Rounding heuristics, such the one proposed by Macedo et al. in [55], are often used since they are simple and efficient methods in finding integer solutions given a linear relaxation solution. These heuristics use the fractional value of the variables and transform them into integer variables through simple rounding techniques. These heuristics seem to be very simple but there are some peculiarities that deserve special attention. One must define simple and valid criteria to select the variables that are going to be submitted to these rounding techniques. The techniques and rounding parameters should also be carefully selected. Inefficient management of these details may lead to unfeasible solutions. When the heuristic leads to unfeasible solutions, the parameters should be readjusted or complementary techniques should be used to ensure feasibility.

The computational results presented in Section 3.6.1 proved that the network flow model is very effective for deriving good incumbents for the problem. Consequently, the linear relaxation solution will be used as a starting point for the iterative rounding heuristic. A fast procedure for obtaining good quality solutions consists in the following rounding heuristic which relies on the iterative solution of the linear relaxation of (3.23)-(3.29). In order to set the rounding heuristic, it is important to clearly define:

- ▷ A model with good incumbents to determine the variables values of the linear relaxation to initiate the heuristic;
- ▷ The parameters and rounding techniques;
- ▷ The method used to determine if the heuristic conducts to a valid solution;
- ▷ An alternative method to find an integer solution when the heuristic fails.

The Algorithm 4.1 presents the implemented iterative rounding heuristic in detail. For a better technical understanding, some important parameters are presented below and the meaning of the decision variables is also remembered.

Parameters and definitions

$\Lambda =$ set of λ_d sorted in decreasing order of decision variable, $\forall d \in D$,

$\Omega =$ set of x^d_{uvr} sorted in decreasing order of decision variable, $\forall r \in R_d$,

$0 \leq u \leq W \in \Psi$ and $\forall d \in D$,

$\alpha =$ paremeter that determine if λ_d should be rounded,

$\beta =$ parameter that determine if x^d_{uvr} should be rounded.

Decision Variables

$$\lambda_d = \begin{cases} 1, & \text{if the depot } d \text{ is selected to be open }, \\ 0, & \text{otherwise} \end{cases} \quad \forall d \in D,$$

$$x^d_{uvr} = \begin{cases} 1, & \text{if route } r \text{ goes through the arc } (u,v) \text{ for the depot } d, \forall r \in R_d, \\ & 0 \leq u \leq W \in \Psi \text{ and } \forall d \in D, \\ 0, & \text{otherwise} \end{cases}$$

Algorithm 4.1: Iterative rounding heuristic

Solve the linear relaxation;
repeat
    $\Lambda \leftarrow$ list λ_d in decreasing order ;
    forall $\lambda_d \in \Lambda$ **do**
        if $\lambda_d > \alpha$ **then**
            $\lambda_d \leftarrow \lceil \lambda_d \rceil$;
            $\Omega \leftarrow$ list x^d_{0vr} in decreasing order ;
            forall $x^d_{0vr} \in \Omega$ **do**
                if $x^d_{0vr} > \beta$ **then**
                    $x^d_{0vr} \leftarrow \lceil x^d_{0vr} \rceil$;
                    repeat
                        if $x^d_{uv'r'} > \beta$ **then**
                            pair $(u,v')^{r'} \in \Psi_d, u \geq v$ in increasing order of u with previous one ;
                    until *there are routes given* W;
until *no more variables are fixed*;
if *unavailable solution for the original problem* **then**
    Solve the model up to integrality for the remaining instance;

As previously mentioned, the heuristic starts with the solution of the linear relaxation of (3.23)-(3.29). It follows with an attempt to fix the different variables of the model. Primarily, one attempt to fix the variables related to the opening of depots, the λ_d variables, and then the fixed variables concerning to the flows associated to the workday instants, in other words the x^d_{0vr} and the x^d_{uvr} with $u > 0$ in that order and repeatedly.

The principle is to sequentially force the opening of the depots whose corresponding λ_d is above a given parameter α. The iterative rounding heuristic starts by the depot with the largest λ_d and continues to select the variables associated to the depots in their decreasing order. Then, it builds workdays for the selected depot d first by rounding up a x^d_{0vr} variable whose value is above a given parameter β. The x^d_{uvr} variables are once again selected in decreasing order of their value, then by selecting further routes $(u, v')^{r'} \in \Psi_d$ (with $u \geq v$, and in increasing order of u) to pair with the previous one such that $x^d_{uv'r'} > \beta$ and until there remain routes given the time limit W.

When there are no more variables to set, the linear relaxation of (3.23)-(3.29) is solved again for the remaining instance, and the process is repeated until it cannot fix any more variables. At this stage, if a solution for the original problem is not already available, the model (3.23)-(3.29) is solved up to integrality for the remaining instance. In that case, a limit on the computing time can be enforced and the best incumbent found until this time limit can be used as a solution.

4.3 A variable neighborhood search approach

The Variable Neighborhood Search (VNS) is a higher-level heuristic proposed by Mladenović and Hansen [62]. This metaheuristic allows one to create a heuristic which explores, in a systematic way, several neighborhood structures providing a good solution with less computational effort. This attribute is important when dealing with optimization problems with hard instances such as location routing problem. VNS is a strategy which conducts the search process by changing the neighborhood structures, allowing to explore distant neighborhoods of the current incumbent solution. The local search ends when it finds a local optimum. Then, it restarts with a different neighborhood structure until all the defined neighborhood space has been exploited.

This metaheuristic has been used to find approximate solutions of various combinatorial optimization problems. Some of these applications are described in different surveys [63, 64]. According to Hansen and Mladenović [65], there are some details that must be taken into account. It is important to be careful when choosing the set of neighborhood structures that will be used in the local search. The stopping criterium is also an important condition in the VNS algorithm. In [65], Hansen and Mladenović present seven properties that help to evaluate the VNS: simplicity, coherence, efficiency, effectiveness, robustness, user-friendliness and innovation. These properties are essential when dealing with integrated optimization problems such as MLRP (Multi-trip Location Routing Problem).

In [59], Macedo *et al.* propose a VNS metaheuristic for the LRP where vehicles and depots have limited capacities and the vehicles can perform several routes in the same planning period.

During the search conducted by the VNS a set of possible solutions is explored according to the defined neighborhood. This set of possible solutions is called the search space. To evaluate the VNS it is important to define the search space and the evaluation function. A four-dimensional matrix S_{dvrc} is used to represent the solution in a simpler and unique way. Other important parameters are presented below.

Parameters

$$
\begin{aligned}
S &= \text{Solution} \\
n_d &= \text{number of depots } d, \forall d \in D \\
m(S) &= \text{Number of open depots } d \\
v(d) &= \text{Number of workdays associated to the depot } d, \forall d \in m(S) \\
r(d,v) &= \text{Number of routes allocated to vehicle } v \text{ associated to depot } d, \\
&\quad \forall v \in v(d) \text{ and } d \in m(S) \\
c(d,v,r) &= \text{Number of clients on route } r \text{ allocated to vehicle } v \text{ of depot } d, \\
&\quad \forall r \in r(d,v), v \in v(d) \text{ and } d \in m(S)
\end{aligned}
$$

S_{dvrc} = Solution from the search space where customer c $(c = 1, ..c(d, v, r))$ is served by route r $(r = 1, ..., r(d, v))$ of vehicle v $(v = 1, ..., v(d))$ from depot d $(d = 1, ..., m(S)), \forall c \in c(d, v, r), r \in r(d, v), v \in v(d)$ and $d \in m(S)$

C_f^d = Fixed cost associated to the opening of a depot $d, \forall d \in m(S)$

C_v = Fixed cost that represents the use of a vehicle $v, \forall v \in v(d)$

L_d = Capacity of depot $d, \forall d \in m(S)$

b_s = The load of the solution S

Q = Capacity of a vehicle

t_{π_i,π_j} = Cost of traveling from π_i to π_j.

There are two different types of unfeasible solutions that may be reached through the exploration of the search space. One set of unfeasible solutions allows to exceed the depots capacity, while the other set allows to exceed the vehicles capacity. Thus, the evaluation function should take into account the violations aforementioned through the introduction of penalizations. A solution S is evaluated for its fixed and variable cost through the evaluation function which takes into account the penalties associated to the violation of the constraints. The penalty of surpassing the vehicle capacity is represented by β while α denotes the penalty for exceeding the depots capacity.

Evaluate

$$f(S) = Cost_d + Cost_v + Cost_r, \tag{4.1}$$

where

$$Cost_d = \sum_{d=1}^{m(S)} \left(C_f^d + \alpha\, max \left\{ 0, \sum_{v=1}^{v(d)} \sum_{r=1}^{r(d,v)} \sum_{c=1}^{c(d,v,r)} b_{s_{dvrc}} - L_d \right\} \right) \tag{4.2}$$

$$Cost_v = C_v \sum_{d=1}^{m(S)} v(d) \tag{4.3}$$

$$Cost_r = \sum_{d=1}^{m(S)} \sum_{v=1}^{v(d)} \sum_{r=1}^{r(d,v)} \left(\sum_{c=0}^{c(d,v,r)} t_{\pi_c,\pi_{c+1}} + \beta\, max \left\{ 0, \sum_{c=1}^{c(d,v,r)} b_{s_{dvrc}} - Q \right\} \right) \tag{4.4}$$

Note that, if a solution S is feasible, then

$$\max \left\{ 0, \sum_{v=1}^{v(d)} \sum_{r=1}^{r(d,v)} \sum_{c=1}^{c(d,v,r)} b_{s_{dvrc}} - L_d \right\} = 0, \forall d \in 1, \ldots, m(S),$$

$$\max \left\{ 0, \sum_{c=1}^{c(d,v,r)} b_{s_{dvrc}} - Q \right\} = 0, \forall d \in \{1, \ldots, m(S)\}, v \in \{1, \ldots, v(d)\},$$

$r \in \{1, \ldots, r(d,v)\}$.

Parameters α and β are dynamically updated being increased or decreased in order to allow the intensification or diversification of the search, respectively. Thus, when the current solution violates a constraint, the parameter is increased so the current search is intensified, guiding the search to a feasible solution space. Whenever the current solution is feasible, *i.e.* respects the constraints, the corresponding parameter is decreased allowing for the violation of the capacity constraint in order to enable a broader search. The process to update the α and β parameters is described in Algorithm 4.2 using an iteration with the current solution S.

Algorithm 4.2: Dynamic update for α and β

Input: $S, \alpha, \beta, \varepsilon_\alpha, \varepsilon_\beta, (\alpha, \beta > 0; \varepsilon_\alpha, \varepsilon_\beta \in]0,1[)$

if $\sum_{v=1}^{v(d)} \sum_{r=1}^{r(d,v)} \sum_{c=1}^{c(d,v,r)} b_{s_{dvrc}} \leq L_d, \quad \forall d \in \{1, \ldots, m(S)\}$ **then**

  $\alpha = \alpha(1 - \varepsilon_\alpha)$;

else

  $\alpha = \alpha(1 + \varepsilon_\alpha)$;

if $\sum_{c=1}^{c(d,v,r)} b_{s_{dvrc}} \leq Q, \quad \forall d \in \{1, \ldots, m(S)\}, \forall v \in \{1, \ldots, v(d)\}, \forall i \in \{1, \ldots, r(d,v)\}$ **then**

  $\beta = \beta(1 - \varepsilon_\beta)$;

else

  $\beta = \beta(1 + \varepsilon_\beta)$;

return α, β

4.3.1 Neighborhood structures

The selection of the neighborhood structures must be performed according to the problem particularities and within the defined solution space $\mathcal{N}$. For this problem, six distinct neighborhood structures ($\mathcal{N} \in \{\mathcal{N}_1, \ldots, \mathcal{N}_6\}$) were defined. These different structures are used for distinct purposes such as the improvement of the costs and the diversification of the search. Neighborhoods $\mathcal{N}_1$ and $\mathcal{N}_2$ are used in order to improve the cost associated to the routes whereas neighborhoods $\mathcal{N}_3$ and $\mathcal{N}_4$ have as main objective the improvement of the costs related to the workdays. For the location problem, two neighborhoods $\mathcal{N}_5$ and $\mathcal{N}_6$ are used, allowing for the search diversification since they permit the opening of new depots. Neighborhoods $\mathcal{N}_1$, $\mathcal{N}_5$ and $\mathcal{N}_6$ are combined in order to reduce the perturbations caused by the last two neighborhoods ($\mathcal{N}_5$ and $\mathcal{N}_6$), since the diversification carried out by these two neighborhoods strongly disturbs the good local optima found during the search.

The first two neighborhoods, $\mathcal{N}_1$ and $\mathcal{N}_2$, are routing neighborhoods. A new element of the first neighborhood $\mathcal{N}_1$, named “move client”, is achieved by changing a client of its position. This modification can be made between two different routes, which may or may not be associated with the same depot, or in the same route. The customer removed is inserted into a certain

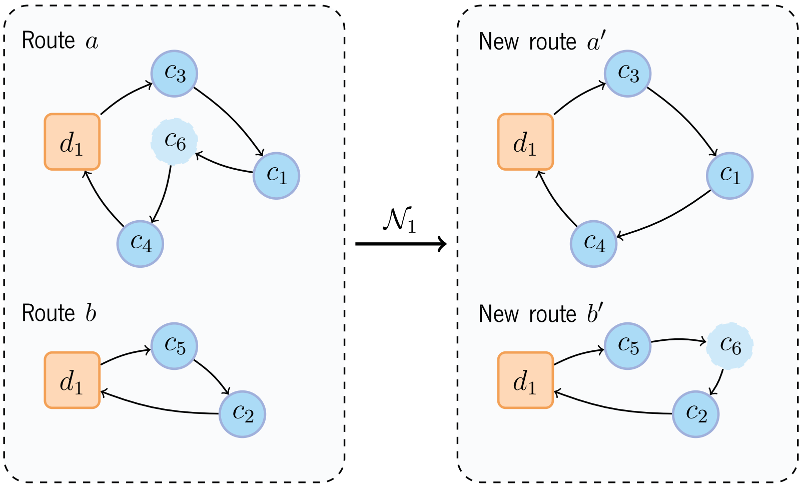

Figure 4.1: Example of a move in neighborhood $\mathcal{N}_1$

position of another selected route. Once again, this modification can be performed with identical routes which may or may not be associated to the same depot. Accordingly, neighboring routes are derived from those that have been selected to perform the shift move of a customer. In the example depicted in Figure 4.1 two routes associated with the same depot (d_1) are selected. The client c_6 is removed from the route a which visits clients c_3, c_1, c_6 and c_4 yielding a new route a' which no longer visits the customer c_6. In the route b (which satisfies clients c_5 and c_2), the client removed from the route a is inserted resulting in a new route b' that now visits the client c_6 between clients c_5 and c_2.

The "swap two clients" is the designation for the neighborhood $\mathcal{N}_2$. Through this neighborhood, a customer may be exchanged between different routes or within the same route. This shift can be made in routes associated with the same depot or linked to a different depot. Figure 4.2 presents the swap of clients c_6 and c_2 between routes a and b. Despite this exchange, the routes are not modified in relation to the number of visited clients, although the cost of the route may be completely different. Route a visits four clients (c_3, c_1, c_6 and c_4) and the new route a' also satisfies four customers (c_3, c_1, c_2 and c_4). Similarly, routes b and b' satisfy the same number of clients, but the former visits client c_2 and the latter route includes client c_6 instead of c_2. It should

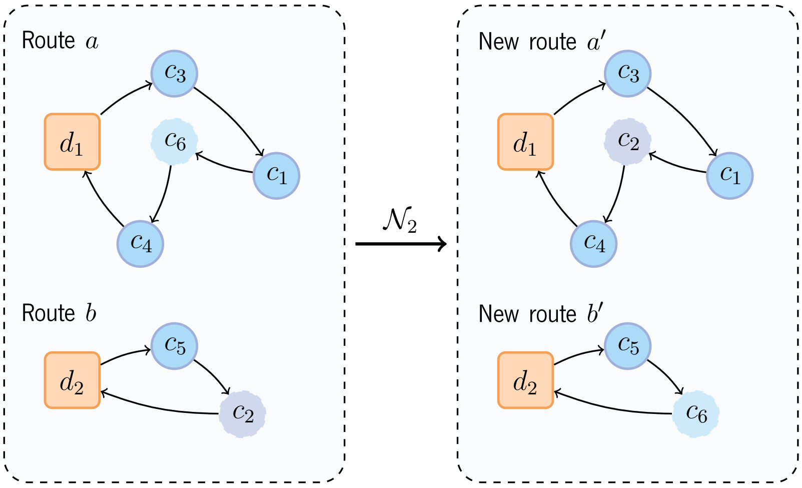

Figure 4.2: Example of a move in neighborhood $\mathcal{N}_2$

be noted that the selected routes in the example are associated with different depots which does not invalidate the use of the neighborhood involved. The customer position in Figure 4.2 is not changed in order to clarify the process of the neighborhood generation.

Neighborhoods $\mathcal{N}_3$ and $\mathcal{N}_4$ are workday neighborhoods. The workday of a vehicle can be composed by more than one route. In order to generate neighbors in the neighborhood type $\mathcal{N}_3$ a route of a vehicle is removed from its workday. The removed route is then associated to a different workday of a vehicle. This change can be performed in workdays associated to the same or different depots. Neighborhood $\mathcal{N}_3$ is termed "shift move of a route". As presented in Figure 4.3, route a is composed by two different routes (r_1 and r_2) that visit clients c_1 and c_2, c_3 and c_4, respectively. These routes are performed by vehicle v_1. The new route a' ceases to conduct the route r_1, no longer visiting the customer c_1. The route b that just fulfilled clients c_5, c_6 and c_7, now visits the client c_1 through the route r_1' yielding new route b'. It becomes necessary to highlight that the route r_1' is now the rearrangement of route r_1. The customers visit order of route r_1 can be handled in order to build a most efficient route for the new associated depot.

The neighborhood "swap two routes" ($\mathcal{N}_4$) works similarly to the neighborhood $\mathcal{N}_3$, however the removed route is replaced by a route of another workday. As previously mentioned a neighbor

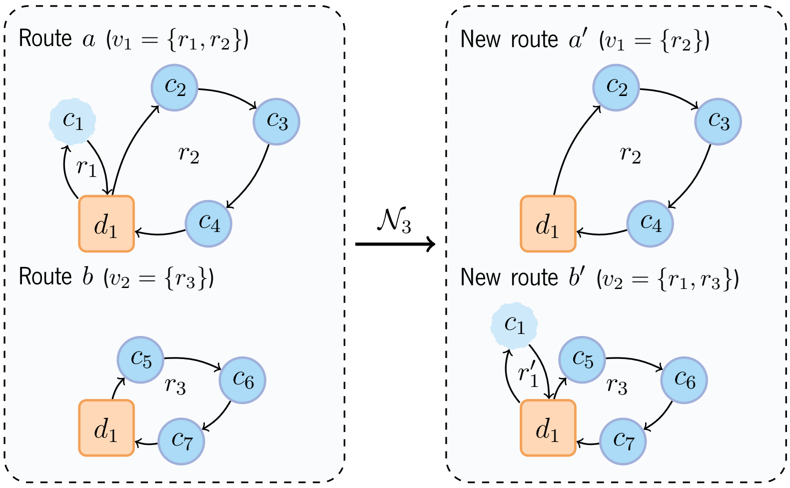

Figure 4.3: Example of a move in neighborhood $\mathcal{N}_3$

of neighborhood $\mathcal{N}_4$ is obtained by swapping two routes from the workday (Figure 4.4). The two vehicles used in these workdays may be associated to the same or different depots. Once more, a rearrangement of the routes is made in order to make them more efficient for the new depot that is associated. Route a uses the vehicle v_1 to satisfy clients c_1, through route r_1, and c_2, c_3 and c_4, using route r_2. On the other hand, route b uses vehicle v_2 to perform the route r_3 that fulfills clients c_5, c_6 and c_7. New route a' performs route r_1 and r'_3. Note that the order of visited clients of the route r_3 is rearranged, since the depot associated to the route is different from the original one. The route r_3 is now associated to the depot d_1 instead of d_2 resulting in route r'_3. The same situation occurs with the route r_2 that is optimized to route r'_2.

Location neighborhoods is how the neighborhoods $\mathcal{N}_5$ and $\mathcal{N}_6$ are denoted. In order to generate neighbors in the neighborhood $\mathcal{N}_5$ (“move workday”), a slightly different concept is used. A new neighbor is created by opening a closed depot. A workday of a vehicle is removed from an open depot and then inserted in a closed depot. The main feature of this neighborhood $\mathcal{N}_5$ is to enforce the opening of closed depots. Figure 4.5 presents a set of three routes r_1, r_2 and r_3 associated to the depot d_1 and a closed depot d_3. Route a and b represent different workdays associated to the vehicle v_1 and v_2, respectively. Neighborhood $\mathcal{N}_5$ forces opening the depot d_3

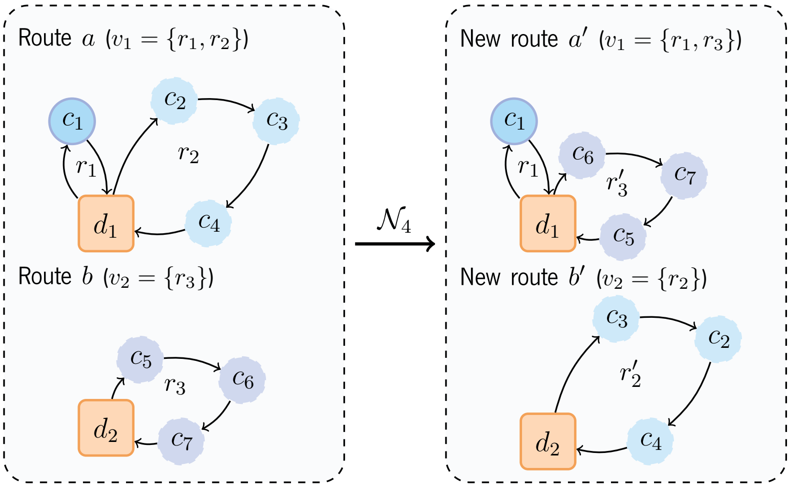

Figure 4.4: Example of a move in neighborhood $\mathcal{N}_4$

by associating to it route r'_3 which serves client c_5 through the new route b'. The depot d_1 still has two routes, r_1 and r_2, associated that serve clients c_1, c_2, c_3 and c_4 throughout the workday related to the vehicle v_1. The routes associated to the new opened depot are also optimized.

The neighborhood $\mathcal{N}_6$ named "move depot" generates new neighbor routes by opening a new depot and closing an used one. All workdays of the different vehicles are inserted in the closed depot. The depot previously open is closed. In Figure 4.6 clients c_5 and c_1, c_2, c_3 and c_4 are

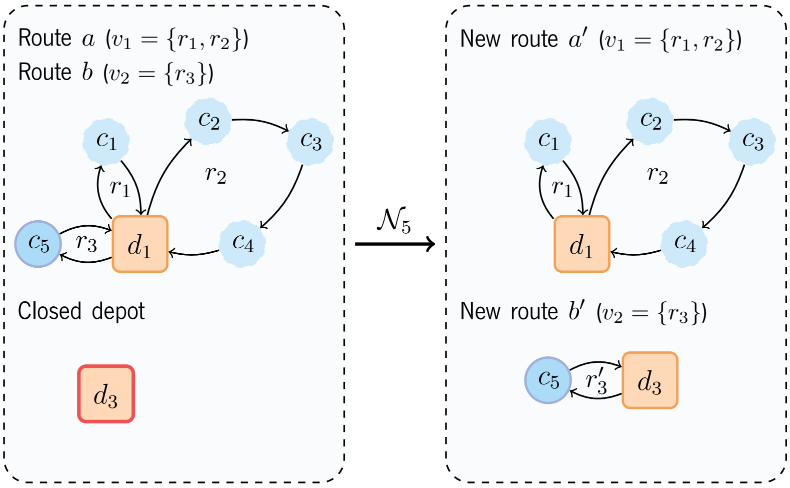

Figure 4.5: Example of a move in neighborhood $\mathcal{N}_5$

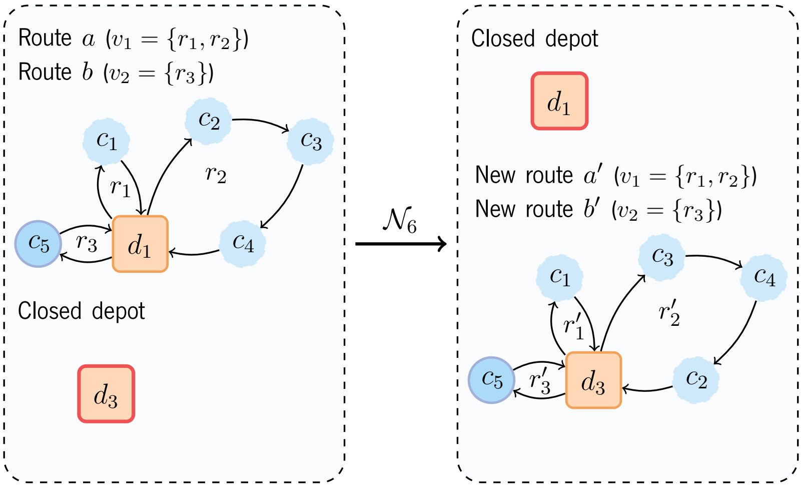

Figure 4.6: Example of a move in neighborhood $\mathcal{N}_6$

served by two different vehicles v_1 and v_2 (which represent two different workdays), respectively. All workdays associated to the vehicle v_1 and v_2 are removed from the depot d_1 and inserted in depot d_3. Again, the routes concerning the new open depot are rearranged.

4.3.2 A skewed variable neighborhood search algorithm

As previously mentioned the Variable Neighborhood Search is a high-level heuristic whose goal is to find integer solutions by exploring, in a systematic way, a set of neighborhood structures, being in this case applied to the location routing problem.

The skewed variable neighborhood search (SVNS) heuristic, presented by Macedo et al. in [59], is similar to the VNS method. However it is a broader strategy since it allows for a more comprehensive exploration. Thus, distant neighbors of the current incumbent solution may be considered in order to find better local optima. However, this algorithm deals with higher perturbations that are bounded by a specific distance function, which defines the solution space of the neighborhood.

This method, summarized in Algorithm 4.3, starts with an initial solution generated through a simple greedy heuristic. Through a Variable Neighborhood Descendent (VND) the current solution is improved by using neighborhoods $\mathcal{N}_1$, $\mathcal{N}_2$, $\mathcal{N}_3$ and $\mathcal{N}_4$ in a sequential way. This process is a General VNS (GVNS) algorithm.

After the initialization step the SVNS algorithm goes through a shaking phase (denoted by Shaking) using neighborhoods $\mathcal{N}_{i\in\{1,5,6\}}$. Neighborhoods $\mathcal{N}_5$ and $\mathcal{N}_6$ may strongly perturb the value obtained in the search phase since the opening of a new depot allows for a diversification of the search. Thus, neighborhood $\mathcal{N}_1$ is combined in order to diminish the generated perturbation. The number of consecutive moves (using the previously selected neighborhoods) to perturb the current solution is defined by k. On each iteration, one of the three neighborhoods may be selected according to the probability $P_{\mathcal{N}_{i\in\{1,5,6\}}}$.

Algorithm 4.3: SGVNS algorithm

Input: Set of neighborhood structures $\mathcal{N} = \{\mathcal{N}_{i\in\{1,\ldots,6\}}\}, P_{\mathcal{N}_{i\in\{1,5,6\}}}, k_{max}, \rho, \eta$

Initialization: find an initial solution S with a greedy randomized heuristic;
$S^* \leftarrow S$;
repeat
  $k \leftarrow 1$;
  while $k \leq k_{max}$ **do**
    $S' \leftarrow$ Shaking$(S, \mathcal{N}_{i\in\{1,5,6\}}, P_{\mathcal{N}_{i\in\{1,5,6\}}}, k)$;
    $S'' \leftarrow$ VND$(S', \mathcal{N}_{i\in\{1,\ldots,4\}})$;
    if $f(S'') < f(S^*)$ *and* S'' *is feasible* **then**
      $S^* \leftarrow S''$;
    if $f(S'') < (1 + \eta\rho(S, S''))f(S)$ **then**
      $S \leftarrow S''$;
      $k \leftarrow 1$;
    else
      $k \leftarrow k + 1$;
    Dynamic update for α and β;
until *a termination condition is met*;
return S^*

In order to explore solutions which are at higher distances from the incumbent, the SGVNS is applied, allowing for the visit of a worse solution (comparing to the current incumbent). This step may only occur if the solution that is going to be visited and the incumbent are considered different if the distance function ρ, defined below, takes a value above a given threshold.

The distance function ρ (Equation 4.5) calculates the structural difference between solution S and S'', taking into account their open depots. Let the variable π ($\overline{\pi}$ for S'') be 1 if the depot i is opened in S (or in S''), being 0 otherwise.

$$\rho(S, S'') = \frac{\sum_{t=1}^{n_d} |\pi_i - \overline{\pi_i}|}{n_d} \tag{4.5}$$

4.4 Computational results

This section demonstrates the computational results for the two presented heuristics. Various benchmark instances from the literature are used and the relevant parameters are presented in the tables that include the computational results. As in the previous chapter, and for the sake of clarity, in the tables, the *Inst* column represents the name associated to the instance and the *n* column provides the number of clients. Columns W and Q represent the length of the workday and the vehicle capacity, respectively. In order to evaluate the performance of the two heuristics, their ability to efficiently drive the search for good quality integer solutions is analyzed. The tests concerning the iterative rounding heuristic were run on a PC with an i7 CPU with 3.5 GHz and 32 GB of RAM. The tests related to the skewed general variable neighborhood search heuristic were run on a Pentium 4 with 3.6 GHz and 2GB of RAM. In both heuristics the optimization subroutines rely on CPLEX 12.5.

4.4.1 Iterative rounding heuristic

In order to prove the ability of the iterative rounding heuristic to efficiently drive good integer solutions, a set of 24 instances with 20 clients and another set of 16 instances with 25 clients from the literature were used. The experiments were performed in order to evaluate the iterative rounding heuristic described in Section 4.2 considering the model (3.23)-(3.29) both with and without cuts.

The results are given in Table 4.1. Column $best\ ub$ gives the value of the best known upper bound for the corresponding instance, column ub_h represents the value of the solution found by the heuristic and t_{ub_h} the total computing time in seconds. In these experiments, the total time spent in solving the remaining instance up to integrality was limited to 30 seconds. The parameters α and β were both set initially to $0,9$. If the process fails in fixing variables, then it is repeated with smaller values of α ($0,75$ and $0,5$) and β ($0,5$ and $0,25$). If it still fails, after using these values, the exact solution procedure for the remaining instances as described in Section 3.4 is used. The results given in Table 4.1 further illustrate the fact that model (3.23)-(3.29) can be used to efficiently generate good incumbents for the initialization of the problem.

Table 4.1: Iterative rounding heuristic results

						Model (3.23)-(3.29)			
						without cuts		*with cuts*	
	$Inst.$	n	W	Q	$best\ ub$	ub_h	t_{ub_h}	ub_h	t_{ub_h}
$C_{20,1}$	1_1	20	140	50	4431	4679	8,85	4435	5,62
$C_{20,1}$	1_2	20	160	50	4430	4750	10,90	4436	11,13
$C_{20,1}$	1_3	20	140	70	4384	4437	60,85	4599	52,90
$C_{20,1}$	1_4	20	160	70	4374	4430	72,75	4492	61,59
$C_{20,2}$	2_1	20	140	50	4681	4753	4,08	4745	7,19
$C_{20,2}$	2_2	20	160	50	4472	4785	10,32	4722	4,11
$C_{20,2}$	2_3	20	140	70	4424	4732	162,40	4437	159,39
$C_{20,2}$	2_4	20	160	70	4424	4766	195,47	4652	201,34
$C_{20,3}$	3_1	20	140	60	4431	4694	8,59	4692	0,63
$C_{20,3}$	3_2	20	160	60	4426	4692	30,60	4439	13,86
$C_{20,3}$	3_3	20	140	80	4385	4431	6,58	4644	19,74
$C_{20,3}$	3_4	20	160	80	4387	4407	37,57	4445	14,38
$C_{20,4}$	4_1	20	140	60	4537	4726	2,98	4714	31,49
$C_{20,4}$	4_2	20	160	60	4471	4741	4,93	4483	6,84
$C_{20,4}$	4_3	20	140	80	4409	4457	10,16	4719	5,46
$C_{20,4}$	4_4	20	160	80	4402	4412	35,88	4419	6,86
$C_{20,5}$	5_1	20	140	60	4674	4684	72,67	4684	70,88
$C_{20,5}$	5_2	20	160	60	4475	4475	111,28	4787	83,98
$C_{20,6}$	6_1	20	140	50	4420	4434	51,58	4421	22,34
$C_{20,6}$	6_2	20	160	50	4414	4448	55,13	4663	25,95
$C_{20,7}$	7_1	20	140	50	4835	4835	17,94	5020	8,04
$C_{20,7}$	7_2	20	160	50	4795	4823	38,42	4807	8,58
$C_{20,8}$	8_1	20	140	50	4744	5048	27,87	5044	26,69
$C_{20,8}$	8_2	20	160	50	4656	4809	34,21	4744	31,63
$C_{25,1}$	1_1	25	140	50	4751	4807	12,73	5022	12,65
$C_{25,1}$	1_2	25	160	50	4646	4796	20,77	4791	14,68
$C_{25,2}$	2_1	25	140	50	4764	5023	47,17	4797	17,32
$C_{25,2}$	2_2	25	160	50	4759	4815	50,53	4783	21,39
$C_{25,3}$	3_1	25	140	60	4817	5131	6,09	5131	7,11
$C_{25,3}$	3_2	25	160	60	4800	5115	32,25	5091	32,20
$C_{25,4}$	4_1	25	140	60	4767	5063	31,88	5069	31,47
$C_{25,4}$	4_2	25	160	60	4767	4922	11,37	5032	31,82
$C_{25,5}$	5_1	25	140	60	4779	5382	388,27	5028	418,68
$C_{25,5}$	5_2	25	160	60	4774	4966	474,46	4842	452,50
$C_{25,6}$	6_1	25	140	50	4755	4792	381,08	4796	351,51
$C_{25,6}$	6_2	25	160	50	4606	4796	437,96	4782	438,14
$C_{25,7}$	7_1	25	140	50	5083	5409	81,02	5329	80,45
$C_{25,7}$	7_2	25	160	50	4864	5067	114,17	4952	106,13
$C_{25,8}$	8_1	25	140	50	4889	5157	279,87	5085	248,76
$C_{25,8}$	8_2	25	160	50	4880	5172	318,70	4904	292,49

4.4.2 Skewed general variable neighborhood search heuristic

To validate the skewed general variable neighborhood search heuristic, two sets of 32 instances for 25 and 40 clients and three sets of 16 instances for 50, 75 and 100 clients were used. The results associated to the referred instances are presented in Table 4.2 and Table 4.3. All instances consider the availability of five different depots. The distance between the various depots available and the clients to serve was rounded to the nearest smaller integer. The performance of this heuristic is measured through a comparative analysis with the solution values obtain with the best lower bound of the network flow model (3.23)-(3.29), with the additional inequalities defined in 3.4.2, and the three-index commodity flow model (3.1)-(3.12).

In order to evaluate the SGVNS algorithm, 5 different runs of the heuristic are performed. The best and the average results are presented in Table 4.2 and Table 4.3. The algorithm is limited through a CPU time $(n \times n_d)$ for the 5 runs which correspond to 125, 200, 250, 375 and 500 seconds, respectively. The parameter k_{max} is set to 10 and the probability of a move in a particular neighborhood is defined by $P_{\mathcal{N}_1} = 0.7$, $P_{\mathcal{N}_5} = 0.05$, $P_{\mathcal{N}_6} = 0.25$. The parameters related to the penalization of the evaluation function are set to $\alpha = 0.1$, $\varepsilon_\alpha = 0.001$, $\beta = 0.1$, $\varepsilon_\beta = 0.001$ and $\eta = 0.1$. The values of the lower bounds that have the symbol * are obtained with the model defined by Acka [30] (3.1)-(3.12). In Table 4.2, columns lb and ub represent the best lower bound and the best upper bound acquired with the network flow model. The column gap gives the value of the gap (Equation 4.6):

$$gap = \frac{ub - lb}{lb} \tag{4.6}$$

Table 4.3 also has a column lb that represents the best lower bound found with the model (3.1)-(3.12) proposed by Acka [30]. The SGVNS model and the three-index commodity flow model were run over 3600 seconds. The best and the average solution obtained in the five runs are presented by columns z^b and z^{av}, respectively.

The tables also include columns for the CPU time needed to find the z^b (denoted by t^b) and the average time for finding the finals solutions (t^{av}). Columns gap^b and gap^{av} represent the gap between z^b and z^{av} and the best corresponding lower bound. The gap is computed according to Equation 4.7:

$$gap = \frac{z - lb}{lb} \tag{4.7}$$

Table 4.2: SGVNS results for 25 and 40 customers

					Model (3.23)-(3.29)			SGVNS					
	$Inst.$	n	W	Q	lb	ub	gap	z^b	t^b	gap^b	z^{av}	t^{av}	gap^{av}
$C_{25,1}$	1_1	25	140	50	4698	4751	1,10	4751	7,20	1,10	4751	8,10	1,10
$C_{25,1}$	1_2	25	160	50	4562	4646	1,80	4646	35,80	1,80	4646	30,80	1,80
$C_{25,1}$	1_3	25	140	70	4439	4590	3,40	4580	4,50	3,20	4580	14,60	3,20
$C_{25,1}$	1_4	25	160	70	4372	4460	2,00	4460	21,70	2,00	4460	33,50	2,00
$C_{25,2}$	2_1	25	140	50	4764	4764	0,00	4764	41,50	0,00	4764	32,40	0,00
$C_{25,2}$	2_2	25	160	50	4608	4759	3,30	4759	25,70	3,30	4759,2	38,50	3,30
$C_{25,2}$	2_3	25	140	70	4553	4699	3,20	4699	22,00	3,20	4700,6	33,20	3,20
$C_{25,2}$	2_4	25	160	70	4474	4474	0,00	4474	53,70	0,00	4477	40,40	0,10
$C_{25,3}$	3_1	25	140	60	4806	4806	0,00	4817	32,80	0,20	4817	18,90	0,20
$C_{25,3}$	3_2	25	160	60	4699	4800	2,10	4800	18,30	2,10	4800	26,40	2,10
$C_{25,3}$	3_3	25	140	80	4607	4721	2,50	4721	13,70	2,50	4721	23,20	2,50
$C_{25,3}$	3_4	25	160	80	4467	4722	5,70	4721	12,90	5,70	4721	53,30	5,70
$C_{25,4}$	4_1	25	140	60	4677	4767	1,90	4767	0,40	1,90	4767	6,00	1,90
$C_{25,4}$	4_2	25	160	60	4539	4767	5,00	4767	0,30	5,00	4767	6,10	5,00
$C_{25,4}$	4_3	25	140	80	4446	4687	5,40	4684	0,20	5,40	4684	11,40	5,40
$C_{25,4}$	4_4	25	160	80	4319	4684	8,50	4511	32,40	4,40	4516,6	53,60	4,60
$C_{25,5}$	5_1	25	140	60	4779	4779	0,00	4779	6,70	0,00	4779	22,20	0,00
$C_{25,5}$	5_2	25	160	60	4709	4774	1,40	4774	10,80	1,40	4774	13,20	1,40
$C_{25,5}$	5_3	25	140	80	4645	4715	1,50	4715	44,40	1,50	4716,4	19,50	1,50
$C_{25,5}$	5_4	25	160	80	4509	4509	0,00	4509	0,80	0,00	4633,2	18,20	2,80
$C_{25,6}$	6_1	25	140	50	4687	4755	1,50	4755	9,40	1,50	4755	7,20	1,50
$C_{25,6}$	6_2	25	160	50	4540	4540	0,00	4606	3,30	1,50	4647,4	42,80	2,40
$C_{25,6}$	6_3	25	140	70	4480	4480	0,00	4480	68,40	0,00	4480,4	68,10	0,00
$C_{25,6}$	6_4	25	160	70	4403	4474	1,60	4477	8,80	1,70	4477	20,80	1,70
$C_{25,7}$	7_1	25	140	50	5083	5083	0,00	5083	47,20	0,00	5083	24,70	0,00
$C_{25,7}$	7_2	25	160	50	4864	4864	0,00	4864	3,50	0,00	4864	15,30	0,00
$C_{25,7}$	7_3	25	140	70	4772	4772	0,00	4772	43,30	0,00	4772	23,10	0,00
$C_{25,7}$	7_4	25	160	70	4648	4771	2,60	4771	46,50	2,60	4771	30,50	2,60
$C_{25,8}$	8_1	25	140	50	4889	4889	0,00	4889	1,60	0,00	4889	22,60	0,00
$C_{25,8}$	8_2	25	160	50	4723	4860	2,90	4860	58,50	2,90	4860	27,10	2,90
$C_{25,8}$	8_3	25	140	70	4604	4742	3,00	4742	122,30	3,00	4743,2	73,70	3,00

(continues on next page)

Table 4.2: Results for 25 and 40 customers (continued)

					Model (3.23)-(3.29)			SGVNS					
	$Inst.$	n	W	Q	lb	ub	gap	z^b	t^b	gap^b	z^{av}	t^{av}	gap^{av}
$C_{25,8}$	8_4	25	160	70	4540	4540	0,00	4540	80,50	0,00	4540	25,80	0,00
$C_{40,1}$	9_1	40	140	50	6848	6918	1,00	6920	173,70	1,10	6980,8	87,70	1,90
$C_{40,1}$	9_2	40	160	50	6682	6910	3,40	6920	151,00	3,60	6927,8	112,10	3,70
$C_{40,1}$	9_3	40	140	70	6530	6815	4,40	6813	146,40	4,30	6813	109,20	4,30
$C_{40,1}$	9_4	40	160	70	6426	6815	6,10	6615	154,80	2,90	6620,8	93,10	3,00
$C_{40,2}$	10_1	40	140	50	6880	6930	0,70	6956	28,10	1,10	7113,8	89,00	3,40
$C_{40,2}$	10_2	40	160	50	6708	6924	3,20	6924	126,80	3,20	6931,6	108,60	3,30
$C_{40,2}$	10_3	40	140	70	6525	6818	4,50	6813	178,10	4,40	6814,6	127,80	4,40
$C_{40,2}$	10_4	40	160	70	6433	6832	6,20	6614	182,00	2,80	6639,6	68,90	3,20
$C_{40,3}$	11_1	40	140	60	8546	8711	1,90	8715	33,10	2,00	8855	91,80	3,60
$C_{40,3}$	11_2	40	160	60	8490	8490	0,00	8713	179,00	2,60	8730,6	149,60	2,80
$C_{40,3}$	11_3	40	140	80	8397	8397	0,00	8411	11,70	0,20	8547,6	43,00	1,80
$C_{40,3}$	11_4	40	160	80	8392	8392	0,00	8403	187,50	0,10	8413,6	67,50	0,30
$C_{40,4}$	12_1	40	140	60	6993	7250	3,70	7239	196,30	3,50	7244,8	147,70	3,60
$C_{40,4}$	12_2	40	160	60	6831	7289	6,70	7092	64,00	3,80	7178,2	129,30	5,10
$C_{40,4}$	12_3	40	140	80	6727	6969	3,60	6959	56,50	3,40	6970,8	107,60	3,60
$C_{40,4}$	12_4	40	160	80	6621	7011	5,90	6912	103,70	4,40	6941,8	128,70	4,80
$C_{40,5}$	13_1	40	140	60	6908	7243	4,80	7207	50,00	4,30	7210	109,90	4,40
$C_{40,5}$	13_2	40	160	60	6786	7065	4,10	6976	41,70	2,80	6995,6	94,00	3,10
$C_{40,5}$	13_3	40	140	80	6616	6973	5,40	6877	124,20	3,90	6888,2	72,50	4,10
$C_{40,5}$	13_4	40	160	80	6552	6865	4,80	6730	136,30	2,70	6832,2	95,20	4,30
$C_{40,6}$	14_1	40	140	50	7022	7218	2,80	7246	114,30	3,20	7261,6	92,60	3,40
$C_{40,6}$	14_2	40	160	50	6831	7039	3,00	7042	46,10	3,10	7064,2	75,10	3,40
$C_{40,6}$	14_3	40	140	70	6404*	-	-	6908	32,70	7,90	6919,6	74,40	8,10
$C_{40,6}$	14_4	40	160	70	6310*	-	-	6762	75,60	7,20	6858	91,30	8,70
$C_{40,7}$	15_1	40	140	50	7237	7316	1,10	7333	104,30	1,30	7351,4	139,40	1,60
$C_{40,7}$	15_2	40	160	50	7076	7104	0,40	7244	77,00	2,40	7293,6	100,60	3,10
$C_{40,7}$	15_3	40	140	70	6454*	-	-	6982	30,70	8,20	7004,6	121,30	8,50
$C_{40,7}$	15_4	40	160	70	6355*	-	-	6957	102,10	9,50	6962,2	72,10	9,60
$C_{40,8}$	16_1	40	140	50	6844	7025	2,60	7223	80,40	5,50	7236	137,50	5,70
$C_{40,8}$	16_2	40	160	50	6768	7005	3,50	7015	196,50	3,60	7022,6	109,30	3,80
$C_{40,8}$	16_3	40	140	70	6335*	-	-	6857	34,20	8,20	6867,8	109,60	8,40
$C_{40,8}$	16_4	40	160	70	6226*	-	-	6645	183,00	6,70	6962,2	72,10	11,80

Table 4.3: Results for 50, 75 and 100 customers

					Model (3.1)-(3.12)	SGVNS					
	Inst.	n	W	Q	lb	z^b	t^b	gap^b	z^{av}	t^{av}	gap^{av}
$C_{50,1}$	17_1	50	140	50	4715	5172	34,40	9,70	5276,60	108,90	11,90
$C_{50,1}$	17_2	50	160	50	4613	4890	118,60	6,00	4981,20	97,60	8,00
$C_{50,1}$	17_3	50	140	70	4581	5041	132,40	10,00	5045,60	70,10	10,10
$C_{50,1}$	17_4	50	160	70	4488	4818	40,80	7,40	4818,80	82,90	7,40
$C_{50,2}$	18_1	50	140	50	6546	7016	27,20	7,20	7097,60	121,60	8,40
$C_{50,2}$	18_2	50	160	50	6444	6991	70,30	8,50	7003,00	112,00	8,70
$C_{50,2}$	18_3	50	140	70	6303	6759	137,00	7,20	6796,20	148,90	7,80
$C_{50,2}$	18_4	50	160	70	6217	6593	33,30	6,00	6618,00	104,60	6,50
$C_{50,3}$	19_1	50	140	60	4914	5151	185,50	4,80	5151,80	132,60	4,80
$C_{50,3}$	19_2	50	160	60	4801	5138	47,30	7,00	5142,80	80,70	7,10
$C_{50,3}$	19_3	50	140	80	4713	5023	84,60	6,60	5024,60	111,20	6,60
$C_{50,3}$	19_4	50	160	80	4621	4798	201,00	3,80	4802,80	134,70	3,90
$C_{50,4}$	20_1	50	140	60	4751	5479	144,50	15,30	5483,00	117,00	15,40
$C_{50,4}$	20_2	50	160	60	4650	5237	232,10	12,60	5336,40	140,60	14,80
$C_{50,4}$	20_3	50	140	80	4511	5114	208,70	13,40	5114,00	164,60	13,40
$C_{50,4}$	20_4	50	160	80	4433	4877	188,10	10,00	5012,20	140,10	13,10
$C_{75,1}$	21_1	75	140	50	5358	6141	272,70	14,60	6145,40	214,90	14,70
$C_{75,1}$	21_2	75	160	50	5215	5912	242,00	13,40	5921,40	217,90	13,50
$C_{75,1}$	21_3	75	140	70	5000	5438	101,80	8,80	5443,00	208,50	8,90
$C_{75,1}$	21_4	75	160	70	4876	5236	181,90	7,40	5286,60	155,40	8,40
$C_{75,2}$	22_1	75	140	50	5298	6181	122,30	16,70	6195,00	171,00	16,90
$C_{75,2}$	22_2	75	160	50	5158	5946	8,00	15,30	5956,40	190,10	15,50
$C_{75,2}$	22_3	75	140	70	4936	5463	274,00	10,70	5470,60	230,10	10,80
$C_{75,2}$	22_4	75	160	70	4816	5244	251,80	8,90	5339,40	198,70	10,90
$C_{75,3}$	23_1	75	140	60	5270	5788	325,00	9,80	5792,80	249,20	9,90
$C_{75,3}$	23_2	75	160	60	5128	5568	207,00	8,60	5622,80	230,80	9,60
$C_{75,3}$	23_3	75	140	80	5002	5183	179,10	3,60	5355,80	116,00	7,10
$C_{75,3}$	23_4	75	160	80	4878	5171	58,40	6,00	5178,00	227,90	6,20
$C_{75,4}$	24_1	75	140	60	5135	5888	273,40	14,70	6022,00	263,10	17,30
$C_{75,4}$	24_2	75	160	60	5001	5874	276,20	17,50	5879,60	250,10	17,60
$C_{75,4}$	24_3	75	140	80	4857	5486	277,00	13,00	5493,80	137,70	13,10
$C_{75,4}$	24_4	75	160	80	4743	5256	195,10	10,80	5427,20	165,40	14,40
$C_{100,1}$	25_1	100	140	50	5433	6178	91,30	13,70	6268,60	225,10	15,40
$C_{100,1}$	25_2	100	160	50	5276	5925	240,40	12,30	6042,00	396,90	14,50
$C_{100,1}$	25_3	100	140	70	5147	5524	146,50	7,30	5709,00	331,50	10,90
$C_{100,1}$	25_4	100	160	70	5009	5526	110,60	10,30	5543,20	246,30	10,70
$C_{100,2}$	26_1	100	140	50	5466	6121	118,20	12,00	6139,80	322,80	12,30
$C_{100,2}$	26_2	100	160	50	5306	5867	145,50	10,60	5918,20	260,70	11,50
$C_{100,2}$	26_3	100	140	70	5152	5468	138,90	6,10	5488,60	318,30	6,50
$C_{100,2}$	26_4	100	160	70	5013	5453	197,10	8,80	5480,20	280,30	9,30
$C_{100,3}$	27_1	100	140	60	5325	6154	364,50	15,60	6176,80	229,70	16,00
$C_{100,3}$	27_2	100	160	60	5179	5916	108,80	14,20	5980,60	308,70	15,50
$C_{100,3}$	27_3	100	140	80	4955	5462	464,40	10,20	5532,00	316,90	11,60

(continues on next page)

Table 4.3: Results for 50, 75 and 100 customers (continued)

					Model (3.1)-(3.12)	SGVNS					
	$Inst.$	n	W	Q	lb	z^b	t^b	gap^b	z^{av}	t^{av}	gap^{av}
$C_{100,3}$	27_4	100	160	80	4832	5461	467,30	13,00	5490,00	382,00	13,60
$C_{100,4}$	28_1	100	140	60	5339	6165	455,70	15,50	6184,80	314,80	15,80
$C_{100,4}$	28_2	100	160	60	5195	5923	144,00	14,00	5982,00	168,90	15,10
$C_{100,4}$	28_3	100	140	80	4964	5537	487,40	11,50	5668,00	359,40	14,20
$C_{100,4}$	28_4	100	160	80	4840	5517	386,70	14,00	5529,40	354,50	14,20

4.4.3 Comparative discussion

As depicted in Table 4.4, it is clear that the average values of the integer solutions obtained by the SGVNS over the 5 different runs is better than those presented by the iterative rounding heuristic. This statement is valid whether the incumbent solution used as a starting point for the iterative rounding heuristic is determined with or without cuts. If one does not consider the different performances of the used computers for the different heuristics, it may be stated that for all presented instances only two and four computational times concerning the iterative rounding heuristic with and without cuts, respectively, are better when compared with the SGVNS heuristic. For that reason, and considering that the computer used in the SGVNS execution tests has an older processor (Pentium 4 with 3,5GHz), after comparing it to the iterative rounding heuristic (i7 with 3,6GHz) it is clearly noted that the results obtained by the SGVNS are much more promising. In order to carry out an equitable comparative evaluation, the column t_f^{av} represents the average time with a factor, demonstrated by Equation 4.8:

$$t_f^{av} = tav \times \frac{PassMark_{\text{Pentium 4 660}}}{PassMark_{\text{Core i7 3770K}}} \tag{4.8}$$

From the data available on the CPUBOSS database[1], the PassMark score of the Intel Pentium 4 660 with 3,5GHz clock speed released in 2005 is $820,9$. On the other hand, the Intel i7 3770K with 3,6GHz clock speed released in 2013 has a PassMark score of $6731,8$. Comparing both values one must notice that the newer processor is about 8,2 times faster. Applying this factor to the computational time of the SGVNS heuristic it becomes clear that better integer solutions in much shorter execution times are found.

Table 4.4: Comparative analysis of the heuristic models

					Model (3.23)-(3.29)						
					without cuts		*with cuts*		SGVNS		
	$Inst.$	n	W	Q	ub_h	t_{ub_h}	ub_h	t_{ub_h}	z^{av}	t^{av}	t_f^{av}
$C_{25,1}$	1_1	25	140	50	4807	12,73	5022	12,65	4751,00	8,1	0,99
$C_{25,1}$	1_2	25	160	50	4796	20,77	4791	14,68	4646,00	30,8	3,76
$C_{25,2}$	2_1	25	140	50	5023	47,17	4797	17,32	4764,00	32,4	3,95
$C_{25,2}$	2_2	25	160	50	4815	50,53	4783	21,39	4759,20	38,5	4,70
$C_{25,3}$	3_1	25	140	60	5131	6,09	5131	7,11	4817,00	18,9	2,30
$C_{25,3}$	3_2	25	160	60	5115	32,25	5091	32,20	4800,00	26,4	3,22
$C_{25,4}$	4_1	25	140	60	5063	31,88	5069	31,47	4767,00	6,0	0,73
$C_{25,4}$	4_2	25	160	60	4922	11,37	5032	31,82	4767,00	6,1	0,74
$C_{25,5}$	5_1	25	140	60	5382	388,27	5028	418,68	4779,00	22,2	2,71
$C_{25,5}$	5_2	25	160	60	4966	474,46	4842	452,50	4774,00	13,2	1,61
$C_{25,6}$	6_1	25	140	50	4792	381,08	4796	351,51	4755,00	7,2	0,88
$C_{25,6}$	6_2	25	160	50	4796	437,96	4782	438,14	4647,40	42,8	5,22
$C_{25,7}$	7_1	25	140	50	5409	81,02	5329	80,45	5083,00	24,7	3,01
$C_{25,7}$	7_2	25	160	50	5067	114,17	4952	106,13	4864,00	15,3	1,87
$C_{25,8}$	8_1	25	140	50	5157	279,87	5085	248,76	4889,00	22,6	2,76
$C_{25,8}$	8_2	25	160	50	5172	318,70	4904	292,49	4860,00	27,1	3,30

4.5 Conclusions

The LRP is a complex problem that has received much attention in the literature. Different authors use various approaches in accordance with the particularities of the system and the resolution methods used by them are strongly influenced by these characteristics. Thus, exact methods usually require longer computational times, so other approaches have been studied. Heuristic

[1] `http://cpuboss.com/cpus/Intel-Pentium-4-660-vs-Intel-Core-i7-3770K#performance`, available on June 10th, 2016

methods have proven their efficiency since they usually get solutions very close to the optimal one without compromising the different business activities. These heuristics can find a valid solution in an acceptable computational time. The trade-off between longer computational times and good valid solutions in shorter computational times is another significant decision. In order to accelerate the problem convergence, some researchers limit the computational time of their algorithms. In the LRP, it is frequent to simplify the process for generating the trips in order to reduce the calculation time. Although all these approaches have proved to be valid, some lead to better results than others. Thus, it is necessary to select the best approach to the problem under analysis. The choice of good resolution methods and optimization techniques is essential.

This chapter covers two distinct heuristics applied to the LRP: an iterative rounding heuristic and a skewed variable neighborhood search heuristic. The iterative rounding heuristic is initialized with the value of the decision variables obtained from the good incumbents generated by the model (3.23)-(3.29) presented in Section 3.6.1. The fractional value of the decision variable is rounded obeying to certain parameters and simple rounding techniques. However, inefficient management of rounding techniques and parameters may lead to unfeasible solutions. For this reason, it is important to define an alternative method to find a valid integer solution. This type of heuristics is simple and efficient in the search for integer solutions given a linear relaxation solution. The skewed variable neighborhood search heuristic is a high-level heuristic which explores, in a systematic way, a set of neighborhood structures considering the problem under analysis. The aim is to find good integer solutions to the location routing problem in acceptable computational times. This heuristic is similar to the VNS heuristic. Nevertheless, it permits a more comprehensive neighborhood exploration in order to find better local optima. In order to conduct a better management of the higher perturbations created by the algorithm, a distance function that bounds the solution space of the neighborhoods is used.

The two different heuristics are evaluated based on a set of benchmark instances from the literature. In order to perform a comparative discussion it was necessary to calculate a factor to compare the different CPU processors.The Intel i7 processor used to run the iterative rounding heuristic was 8,2 times faster than the Intel Pentium 4 used to execute the SGVNS heuristic, according to the CPUBOSS database. According to this factor, it is clear that the SGVNS heuristic is more promising since it may find better integer solutions in much shorter execution times.

Chapter 5

The multi-trip production, inventory, distribution and routing problem with time windows: exact solution approaches

Outline

5.1 Problem description

The Multi-trip Production, Inventory, Distribution and Routing Problem (MPIDRP) is considered an integrated problem as it combines important management science problems such as the Production and Distribution Problem (PDP), the Multi-trip Vehicle Routing Problem (MVRP) and the Inventory Routing Problem (IRP). The mentioned problems typically occur in the logistics and transportation fields, being related to the inventory, distribution and production management. In the PDP, the clients are served according to their periodic needs and decisions related to the distribution and the production for each period are taken into account. In this type of problem, the customers may receive the needs for future periods. However, constraints concerning the inventory are not considered. The IRP addresses the lack of inventory management of the PDP disregarding production management. This inventory management may occur in the clients, in the warehouse or in both. During the last years, the integration of these two important problems has been particularly studied through the Production, Inventory, Distribution and Routing Problem (PIDRP). The PIDRP considers simultaneously restrictions related to the management of production and the management of inventory. The PDP, IRP and PIDRP also may consider constraints concerning routing and distribution of goods. The MVRP determines a set of routes to fulfill the clients needs having the particularity that a vehicle can perform more than a single-trip during the planning horizon. The main objective of the integration of these problems is the cost minimization considering the entire system. Solving these management problems in an integrated manner leads to better solutions from the global perspective. However, the size of the problems increases significantly. When clients have time windows for the distribution of their orders, the problem is called Multi-trip Production, Inventory, Distribution and Routing Problem with Time Windows (MPIDRPTW).

The MPIDRPTW includes a single production facility that may fulfill a set of clients that have a time varying demand during a finite planning horizon and each client has a specific time window to deliver their orders. This problem considers that a fleet of homogeneous vehicles performs a set of routes in order to distribute all goods. The multi-trip variant allows for each homogeneous vehicle to make more than a single-trip during the planning horizon. The demand of a client may be satisfied through inventory held at the facility or from periodic production. Customers may receive the demand associated to future periods considering the global minimization of the costs associated to the distribution, the production and the inventory process, however split delivery is not allowed. The inventory holding costs occur at the facility when there is overproduction and at the client when the demand associated to future periods is stored. The facility incurs in a setup cost when a production period is scheduled and the capacity of the facility is limited.

The multi-trip variant is commonly used in the transportation of perishable goods, which must be delivered in a short planning horizon, and when the routes are bounded to a small geographic area.

Definition

A route r serves an ordered set of clients and may be considered a single-trip (r_2 in Figure 5.1) or a multi-trip (r_1 in Figure 5.1). A single-trip visits a set of clients and then returns to the facility, whereas a multi-trip returns and leaves a facility at least twice. A set of single-trips executed by the same vehicle is called a multi-trip and is associated only to one vehicle. Each vehicle performs a single-trip or a multi-trip according to its time availability, which is determined by a workday. In Figure 5.1, an example of two different routes that serve a set of seven clients for the first planning period (t_1) is provided. Although the customers may receive the needs for future periods, one assumes that the demand of the current period must be necessarily fulfilled. The route r_1 is composed by two single-trips (st_1 and st_2). The first single-trip (st_1) only visits the client c_1, but the vehicle (v_1) that performs the route is loaded with the entire demand associated to the first and second periods (t_1 and t_2, respectively). The second single-trip st_2, which is associated to the same vehicle v_1, satisfies three different clients (c_2,c_3 and c_4) and it only delivers the demand associated to the period t_1. In route r_1 the single-trip st_1 serves client c_1 and then returns to

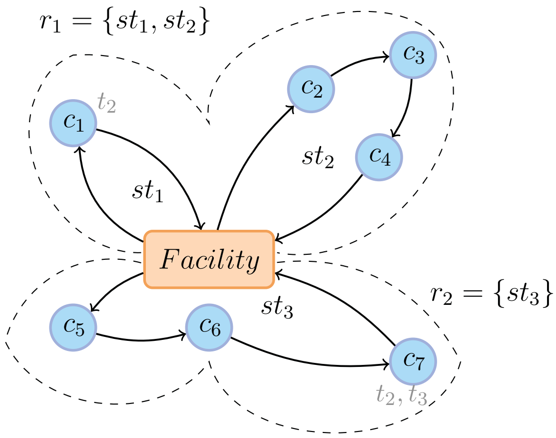

Figure 5.1: Example of possible routes for MPIDRPTW

the facility to reload the vehicle and perform the single-trip st_2. The second route r_2 is a single-trip st_3 that serves client c_5, c_6 and c_7. It is important to highlight that the vehicle (v_2), which performs route r_2, loads the future demand associated to period t_2 and t_3 for client c_7 in addition to the demands related to period t_1 for all the subset of clients present in route r_2. Note that the distribution plan must take into account the time windows of each client.

The MPIDRPTW incorporates some particularities which are important to be clearly described. These peculiarities are presented below:

- ▷ Each customer may at most be visited once per period;
- ▷ Each client is associated to a single facility;
- ▷ Each route must start and end at the facility regardless of the number and order of visited clients;
- ▷ The load of each single-trip must not exceed the capacity of the vehicle;
- ▷ A vehicle can perform several single-trips during a planning horizon but the total load of the multi-trip associated to one vehicle cannot exceed the production capacity of the facility;

- ▷ A vehicle cannot work more than the length of the planning horizon;
- ▷ Each client has a time-varying demand and may receive the needs for future periods;
- ▷ Split deliveries are not allowed, which means that a demand for a future period is entirely delivered; and
- ▷ The inventory at the facility and at the clients is allowed and limited.

The length of a route r cannot be greater than the maximum length a vehicle may travel. Thus, the length of a route is limited by a workday W. A route must start and end at the facility regardless of the order and the number of visited clients. The MPIDRPTW has an available fleet F of homogeneous vehicles v where each vehicle may perform more than one single-trip. A single-trip cannot exceed the vehicle capacity Q, and the total load of a route cannot exceed the facility production capacity C_t for period t. The demand d_i^t of a client $i \in N$ must be fulfilled through a production plan that is distributed over a set of periods $t \in T$. All clients must be served for all periods. This problem allows for the management of limited inventories at the facility ($0 \leq I^P \leq I^P_{max}$) and at the clients ($0 \leq I^C_i \leq I^C_{max,i}$).

The solution of the problem includes both the fixed cost C_v when a new vehicle of a fleet is used and the fixed setup production cost f_t for each period in which the facility is active. The additional costs incorporate the distribution cost C_r associated to the usage of a route, the facility holding cost h^P and the clients inventory cost C_{h_i}. The main objective of the problem is to minimize the global costs associated to the entire system.

5.2 An arc flow formulation

A network flow model for the MPIDRPTW is presented in this section. In this model the nodes do not represent clients or demand periods, but discrete instants of time. For the MPIDRPTW, this approach defines a set of graphs Π_t that have a set of Δ vertices that represent discrete time instants for each period of production t. The arcs associated to the manufacturing period t are grouped in a set of arcs represented by Ψ_t.

5.2.1 The model

The network flow model allows for the definition of an acyclic directed graph per distribution period $\Pi_t = (\Delta, \Psi_t), \forall t \in T$. The Δ vertices represent instants of time that vary from 0 up to the time limit W, that represents a workday of a given vehicle. The facility uses a fleet of homogeneous vehicles that perform the routes associated to the arcs. An arc $(u, v)^r \in \Psi_t$ represents a route r that starts and end at instant time u and v, respectively.

The set of arcs Ψ_t has a particular definition presented below:

$$\Psi_t = \{(u, v)^r : 0 \leq u < v \leq W, r \in R_t\} \cup \{(u, v)^o : 0 \leq u < v \leq W, v = u + 1\},$$

where R_t represents the set of all the routes associated to the period t. A route r has an associated load D_r^t, duration t_r and cost C_r and visits a set of clients N_r, with $N_r \subseteq N$. The route is valid only if the conditions $D_r^t \leq Q$ and $t_r \leq W$ occur.

The model is composed by two sets of binary variables x_{uvr}^t and z_t related to distribution and production decision, respectively. The variables x_{uvr}^t state whether the route r that starts at instant u, ends at instant v, and is associated to period t is selected or not. For the sake of simplicity the decision variable x_{uvr} denotes an arc. Variables z_t determine the periods t in which there is production. The integer variables are identified by three different sets: the p_t variables which represent the quantity produced in period t, and variables I_t^P and I_{it}^C, which represent the

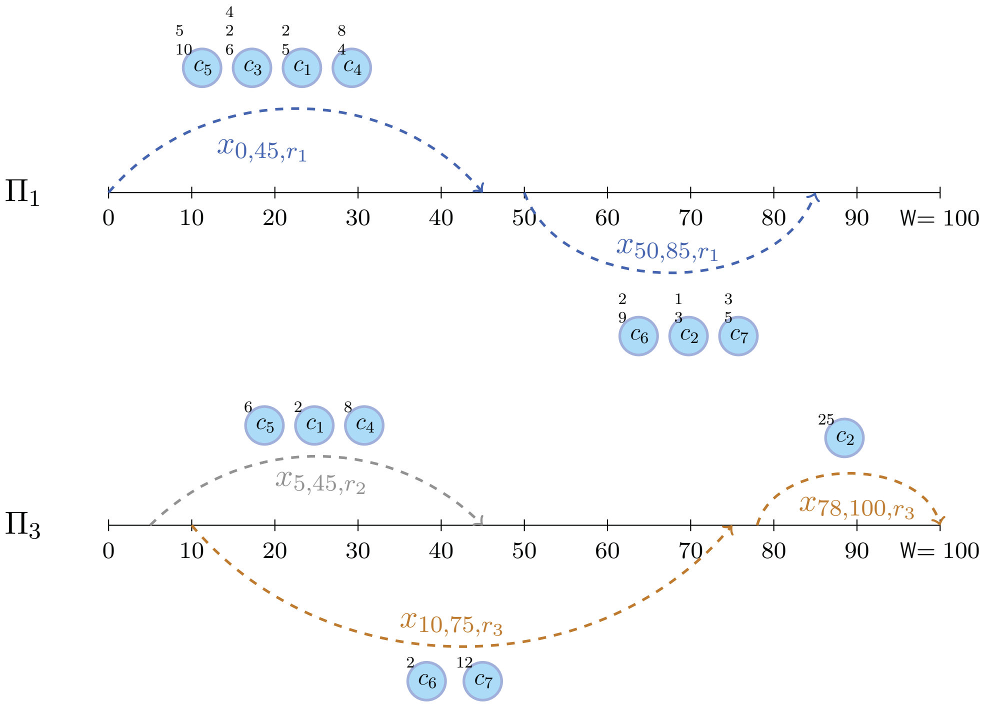

Figure 5.2: Solution example of network flow model for the MPIDRP

inventory quantity for period t at the facility or the client, respectively.

Figure 5.2 presents a solution example of a network flow model for the MPIDRP that fulfills seven clients ($c_1, \ldots, c_7$) for three periods ($t_1, \ldots, t_3$). As depicted in Figure 5.2, the distribution of goods is performed over two different periods (t_1 and t_3) represented by Π_1 and Π_3. It is important to note that the load carried by a route can be obtained from the quantity stored in the facility or from production. Distribution during a period does not mean that there is production in the same period. For the sake of simplicity, in Figure 5.2, only arcs that have a flow associated are represented, and production and inventory decisions are not presented.

The clients c_5, c_3, c_1 and c_4 are fulfilled through the arc $x_{0,45,r_1}$ that starts at the instant of time 0 and ends at instant 45. This flow is associated to the route r_1. The clients c_5, c_1 and c_4 are served for the period $t = 1$ and $t = 2$. Client c_3 receives the demand related to period $t = \{1, \ldots, 3\}$. This occurs since there is capacity to carry this amount in the vehicle for future periods. For the route r_1 there is another arc $x_{50,85,r_1}$ that serves clients c_6, c_2 and c_7, which begins at time 50 and terminates at time 85. The two mentioned arcs ($x_{0,35,r_1}$ and $x_{50,85,r_1}$) may not be combined into a single arc due to vehicle capacity constraints. In the graph Π_3, the routes r_2 and r_3 are performed through two different vehicles, since just one vehicle may not visit two different clients at the same time. Capacity constraints do not allow to merge arcs $x_{10,75,r_3}$ and $x_{78,100,r_3}$. For that reason, the vehicle that performs route r_3 serves clients C_6 and c_7 ($x_{10,75,r_3}$), returns to the facility to load the vehicle and then fulfills the other client c_2 in the route r_3 through the arc $x_{78,100,r_3}$. Route r_2 serves clients c_5, c_1 and c_4 for the third period ($t = 3$).

All parameters, definitions, and decision variables used in the arc flow model are listed bellow.

Parameters and definitions

$D =$ single facility

$N =\{1, \dots, n\}$ set of clients $i, \forall i \in N$

$T =\{1, \dots, \tau\}$ set of periods t associated to the distribution, $\forall t \in T$

$T_0 =\{0, \dots, \tau\}$ set of periods t associated to the production and inventory, $\forall t \in T$

$F =$ fleet of homogeneous vehicles $v, \forall v \in F$

$W =$ length of the planning horizon

$Q =$ capacity associated to the vehicle

$\Pi_t =$ acyclic directed graph associated to the period $t, \forall t \in T$

$\Psi_t =$ set of arcs associated to the period $t, \forall t \in T$

$\Delta =$ set of vertices,

$(u, v)^r =$ arc that represents a route r that starts at time u and ends at time $v, \forall r \in R_t$

$R_t =$ set of all routes associated to the period $t, \forall t \in T$

$C_r =$ cost associated to perform a route $r, \forall r \in R_t$

C_v = cost associated to the use of a vehicle $v, \forall v \in F$

h^P = unitary holding cost at the facility

C_{h_i} = unitary holding cost at the client i during a period of the planning horizon, $\forall i \in N$

$C_{H_r^t}$ = holding cost associated to the route r and period t , $\forall r \in R_t$ and $\forall t \in T$

f_t = setup cost associated to production period $t, \forall t \in T$

N_r = set of clients visited by route $r, \forall r \in R_t$

D_r^t = load associated to a route $r, \forall r \in R_t$

D_{ir}^t = load associated to a route r for the client $i, \forall r \in R_t, \forall t \in T$ and $\forall i \in N$

t_r = duration associated to a route $r, \forall r \in R_t$

t_r^i = total waiting time at client i for route $r, \forall r \in R_t$

d_i^t = demand associated to a client i associated to the period $t, \forall i \in N$ and $\forall t \in T$

C = capacity associated to the unique facility

I_{max}^P = maximum inventory allowed at the facility

$I_{max,i}^C$ = maximum inventory allowed at the client $i, \forall i \in N$

$$\alpha_{irt}^{t'} = \begin{cases} 1 & \text{if the route } r \text{ serves client } i \text{ for period } t \text{ , } \forall i \in N_r, \forall r \in R_t \text{ and } \forall t \in T, \\ & \text{where } t \leq t', \\ 0 & \text{otherwise} \end{cases}$$

Decision variables

$$z_t = \begin{cases} 1 \text{ if there is production in period } t, \forall t \in T, \\ 0 \text{ otherwise} \end{cases}$$

$$x^t_{uvr} = \begin{cases} 1 & \text{if route } r \text{ uses arc } (u,v) \text{ for the period } t, \forall r \in R_t, \\ & 0 \leq u \leq W \in \Psi_t \text{ and } \forall t \in T, \\ 0 & \text{otherwise} \end{cases}$$

$p_t =$ amount produced for the period $t, \forall t \in T$

$I^P_t =$ inventory at the facility at the end of the period $t, \forall t \in T$

$I^C_{it} =$ inventory at client i at the end of the period $t, \forall i \in N$ and $t \in T$

The integer programming arc flow model for the MPIDRPTW is defined through (5.1) to (5.19). The main objective of the network flow model is to minimize the total costs associated to the entire problem that includes the cost of performing the distribution routes and the vehicles usage. When there is production during a period, there is a setup cost associated to the facility. The problem considers inventory costs at the facility and at the clients. The latter occurs when clients orders for future periods are anticipated.

Minimize

$$\sum_{t\in T}\sum_{(u,v)^r\in\Psi_t} C_r x^t_{uvr} + C_v \sum_{t\in T}\sum_{(0,v)^r\in\Psi_t} x^t_{0vr} + \sum_{t\in T}\sum_{(u,v)^r\in\Psi_t} C_{H^t_r} x^t_{uvr} + \sum_{t\in T_0} f_t z_t + \sum_{t\in T_0} h^P I^P_t \tag{5.1}$$

subject to:

$$\sum_{t\in T, t\leq t'}\ \sum_{(u,v)^r\in\Psi_t | i\in N_r} \alpha^{t'}_{irt} x^t_{uvr} = 1, \quad \forall i \in N, t' \in T, \tag{5.2}$$

$$\sum_{(0,v)^r\in\Psi_t} x^t_{0vr} \leq F, \quad \forall t \in T, \tag{5.3}$$

$$-\sum_{(u,v)^r\in\Psi_t} x^t_{uvr} + \sum_{(v,y)^s\in\Psi_t} x^t_{vys} = \begin{cases} 0, \text{ if } v = 1, ..., W-1, \\ -\sum_{(0,v)^r\in\Psi_t} x^t_{0vr}, \text{ if } v = W, \end{cases} \quad \forall t \in T, \tag{5.4}$$

$$p_t \leq Cz_t, \forall t \in T_0 \backslash \{\tau\} \tag{5.5}$$

$$p_0 \geq \sum_{i \in N} (d_{i1} - I_{i0}^C) - I_0^P, \tag{5.6}$$

$$p_\tau = 0, \tag{5.7}$$

$$I_\tau^P = 0 \tag{5.8}$$

$$I_t^P = I_{t-1}^P + p_t - \sum_{(u,v)r \in \Psi_t} D_r^t x_{uvr}^t \quad \forall t \in T \backslash \{\tau\} \tag{5.9}$$

$$\sum_{(u,v)r \in \Psi_t} D_r^t x_{uvr}^t \leq I_{t-1}^P, \quad \forall t \in T \tag{5.10}$$

$$I_{i\tau}^C = 0 \quad \forall i \in N \tag{5.11}$$

$$I_{it}^C = I_{it-1}^C + \sum_{(u,v)^r \in \Psi_t} D_{ir}^t x_{uvr}^t - d_{it} \quad \forall i \in N, t \in T \backslash \{\tau\} \tag{5.12}$$

$$0 \leq I_t^P \leq I_{max}^P, \forall t \in T_0 \tag{5.13}$$

$$0 \leq I_{it}^C \leq I_{max,i}^C, \forall i \in N, t \in T_0 \tag{5.14}$$

$$x_{uvr}^t \in \{0,1\}, \forall (u,v)^r \in \Psi_t, \ \forall t \in T \tag{5.15}$$

$$z_t \in \{0,1\}, \forall t \in T_0. \tag{5.16}$$

where

$$D_r^t = \sum_{t \leq t', i \in N} d_i^t \alpha_{irt}^{t'}, \qquad \forall (u,v)^r \in \Psi_t \tag{5.17}$$

$$D_{ir}^t = \sum_{t \leq t'} d_i^{t'} \alpha_{irt}^{t'}, \qquad \forall (u,v)^r \in \Psi_t, i \in N \tag{5.18}$$

$$C_{H_r^t} = \sum_{i \in N_r} C_{h_i} t_r^i \tag{5.19}$$

The objective function is represented by (5.1), and it consists in the minimization of all cost associated to the MPIDRPTW problem. Constraints (5.2) guarantee that every client is visited. Constraints (5.3) limit the number of vehicles used in the distribution phase and (5.4) ensure the flow conservation in the model. The capacity of the production is limited per period through (5.5). Production in the first period is enforced through (5.6), and production in the last period is avoided through (5.7). In the last period, the inventories at the facilities and at the clients are set to zero, to avoid excessive production (Constraints (5.8) and (5.11)). Constraints (5.9) and (5.12) guarantee the management of inventory at the facility and at the clients, respectively. Constraints (5.10) ensure that the distribution phase is done according to the production and inventory phase.

5.2.2 Arcs generation

The arc flow model uses a set of predefined arcs and selects those that best serve customers in an attempt to minimize costs of the overall system related to the distribution of goods. The model also considers production and inventory decisions, as well as it decides which arcs should be used to serve a set of customers at the lowest possible cost. The demand for a client varies by period and can be anticipated. An arc represents a single-trip that, after leaving the facility, visits a set of clients and then returns to the same facility. The model also determines that a vehicle can make a set of single-trips that is limited by the respective workday. This set of arcs is called multi-trip and is performed by a specific vehicle.

For the arc flow model, a set of arcs are created through Algorithm 5.1 that allows for the generation of all possible combinations of arcs according to the constraints defined in the model. The algorithm for creating arcs pays a particular attention to details such as:

- ▷ An arc must start and end at the facility;
- ▷ The capacity of a vehicle cannot be exceeded;
- ▷ Clients time windows must be met;
- ▷ The distance of an arc cannot exceed the workday of the vehicle;

▷ An arc can only visit a client exactly once, either for serving the current demand or the demand of future periods.

The algorithm to generate the arcs is defined in Algorithm 5.1. The period t is an input to the algorithm. Thus, the execution for a given period is independent from a different period. There are two types of arcs which differ in the last visited location. If it is a client, then the arc is not closed, *i.e.*, it does not return to the depot. In this case, the arc is considered partial. When the arc starts and ends at the depot it is considered a final arc. From a final arc, it is not possible to add new clients to serve, however this does not apply to a partial arc where one may add a new client to serve or close the arc back to the depot.

Algorithm 5.1: MPIDRPTW Arc Generation

```
Input: period
partial_arcs ← empty_arc ;
while partial_arcs! = ∅ do
    current ← first_element(partial_arcs);
    partial_arcs ← partial_arcs\{current}
    if possible_to_reach_depot(current) then
        new_closed_arc ← close_arc(current);
        final_arcs ← final_arcs ∪ {new_closed_arc}
    foreach period p ∈ T, where p ≥ period do
        foreach client c ∈ N do
            if client_not_present(current, c) then
                if client_valid_time_window(current, c) then
                    if check_distance_client(current, c) then
                        if check_demand_client(current, c) then
                            new_partial_arc ← extend_arc(current, c);
                            partial_arcs ← partial_arcs ∪ {new_partial_arc};
```

As shown in Algorithm 5.1, the generation of arcs for the MPIDRPTW starts with an empty partial arc. This arc starts at the depot but does not serve any client. This arc represents the starting point to serve clients, *i.e.*, new clients will be added. After generating a temporarily empty partial arc, the generation of more partial and final routes may start. For each partial arc in the set of partial arcs, the first partial arc is removed for analysis. Then, the arc is tested in order to evaluate if the partial arc may return back to the depot considering the distance constraints. In the case of a positive answer, a new final arc is generated and added to the set of final arcs. Then, the selected partial arc is tested over different constraints to check whether it is possible to add new clients to serve. This process is done for each client and for the current and future periods, allowing for the anticipation of demands for a client. Indeed, the arc may only be extended if the new client is not already being served by the arc, the distance necessary to reach the client does not exceed the available workday length, and the new demand does not violate the available vehicle capacity. When all these constraints are met, it is possible to add a new client to the current partial arc. Indeed, a new partial arc is created and added to the corresponding set of partial arcs. After trying to extend the current arc to each client of the current and future periods, the current iteration ends and a new one is started if the set of partial arcs is not empty. The algorithm tries to close and extend all existing arcs in the set of partial arcs. The arc generation process ends when it is not possible to add new clients to the last partial arc.

5.3 Implementation details

Despite the constant technological evolution, it is important to make an adequate management of resources. Only in this way, the search for new strategies, models and approaches is justified for solving difficult problems. When dealing with hard problems each minor implementation detail can represent large savings in computational time. When these details are used in scenarios of real complexity they become even more valued, since they can enable companies to achieve higher levels of competitiveness.

The complexity of a problem is higher with the increasing number of variables. The MPIDRPTW integrates different hard problems and has many details which make it even harder to solve, such as the multiple usage of a vehicle and the orders anticipation. The generation of arcs has a high computational complexity since, for the creation and validation of the arcs, several computational calculations are necessary. This phase is very important since the arc flow model uses the pre-generated routes to determine the best production, inventory and distribution planning. With the increasing number of clients and periods in the benchmark instances the creation of arcs becomes increasingly difficult, thus it becomes imperative to get answers in real time, being necessary to use faster and more efficient methods.

The arc generation algorithm takes advantage of parallelization techniques. The thread concept is applied, since the generation of arcs per period is independent. Pre-orders are taken into account, but are not dependent on arcs generated for future periods. This approach allows for a better use of computational processing, without influencing the obtained results. In this way, the generation of arcs is done in parallel for each period of the planning horizon.

For all instances, the arc generation of each period is executed at the same time. Thus, in the parallel mode, the model waits for the arc generation of the slowest period instead of waiting for a sequential processing of each period. In the sequential mode, the arc generation of a new period is started only after all arcs of the current period are generated. The relevant parameters of the instances are shown in the table. The tests were executed on a PC with an Intel Xeon CPU ES-1620 v3 with 3.5GHz and 64GB of RAM.

The columns of the table are defined as follows:

$Inst$: instance name Disti_t where i represent the number of clients and t the number of periods

ub : value of the best known upper bound

lb : value of the lower bound

$\#Arcs$: total number of arcs generated for all the periods

T_{series} : total time to generate arcs in series mode

$T_{parallel}$: total time to generate arcs in parallel mode

$T_{imp}(\%)$: percentage time improvement with parallelization techniques for the arc generation

Table 5.1 shows that the parallelization techniques become important when the arc generation process has higher computational times, *i.e.*, the number of generated arcs is very high - 5000000, for example. Indeed, the use of the parallelization is considered an asset to the generation of arcs. To show the time improvement when applying the parallelization technique, several instances from all sets were selected.

Table 5.1: Comparative analysis for the series and parallel mode for the Arc Generation

	Inst	ub	lb	#Arcs	T_{series}	$T_{parallel}$	T_{imp} (%)
Set 1	Dist10_2	131101	131101	35676	0	0	0,00
	Dist10_4	238963	238963	78601	0	1	0,00
	Dist10_6	323088	323056	265417	0	1	0,00
	Dist10_8	442963	442963	386959	0	0	0,00
	Dist20_2	134640	129026	6237937	668	546	18,26
Set 2	Dist10_2	172161	172161	9102	0	0	0,00
	Dist10_4	272563	272563	25067	0	0	0,00
	Dist10_6	377051	377051	35031	0	0	0,00
	Dist10_8	469139	469093	51075	0	0	0,00
	Dist20_2	278830	278830	31893	0	0	0,00
	Dist20_4	457218	455611	86698	0	0	0,00
	Dist20_6	651148	651083	104547	0	0	0,00
	Dist20_8	794620	773394	773555	2	1	50,00
	Dist30_2	325000	325000	335511	0	1	0,00
	Dist30_4	565230	560255	995404	2	1	50,00
	Dist30_6	940558	939903	884508	2	1	50,00
	Dist30_8	1347960	1327370	1620705	8	6	25,00
	Dist10_2	159265	159265	6272	0	0	0,00
	Dist10_4	295403	295403	18293	0	0	0,00
	Dist10_6	383228	383228	29558	0	0	0,00
	Dist10_8	495255	495255	58349	1	0	100,00
	Dist20_2	231778	231633	126387	0	0	0,00

Continues on next page

Table 5.1 – continued from previous page

	Inst	ub	lb	#Arcs	T_{series}	$T_{parallel}$	T_{imp} (%)
Set 3	Dist20_4	414401	409884	161351	0	0	0,00
	Dist20_6	809436	809356	68213	0	0	0,00
	Dist20_8	958297	935540	630682	5	5	0,00
	Dist30_2	346908	345819	236719	1	0	100,00
	Dist30_4	623887	614118	494888	1	0	100,00
	Dist30_6	935402	904505	1272022	2	1	50,00
	Dist30_8	1191260	1064930	2039617	5	3	40,00
	Dist40_2	358135	349558	8882464	1281	1099	14,21
Set 4	Dist10_2	189192	189192	18590	0	0	0,00
	Dist10_4	229833	229810	432276	0	1	0,00
	Dist10_6	314347	314335	220160	0	0	0,00
	Dist10_8	368200	357270	284669	0	0	0,00
	Dist20_2	237142	236745	400087	1	1	0,00
	Dist20_4	323946	316339	5519716	453	434	4,19
	Dist20_6	503080	485100	2835399	16	11	31,25
Set 5	Dist10_2	205395	205395	39532	0	0	0,00
	Dist10_4	293166	293166	67063	0	0	0,00
	Dist10_6	290025	289999	185198	0	0	0,00
	Dist10_8	347438	334572	726288	0	1	0,00
	Dist20_2	226903	225346	503444	1	1	0,00
	Dist20_4	341116	340621	1078899	5	4	20,00
	Dist20_6	514038	491315	4947212	73	45	38,36
	Dist20_8	638242	589456	8057169	150	118	21,33
	Dist30_2	279766	279094	1375993	7	4	42,86
	Dist30_4	459172	425261	5014887	56	42	25,00
	Dist30_6	-	633267	17349130	6551	6290	3,98

5.4 Computational results

In this section, the computational results performed on benchmark instances adapted from Bard and Nananukul in [46] are presented. The benchmark instances presented by Bard and Nananukul are divided into 5 different sets. Each set contains 20 instances that vary in the number of clients (10, 20, 30, 40 and 50) and in the number of planning periods (2, 4, 6 and 8). The different instances also present parameters related to the distribution problem, such as the number and capacity of the vehicles and also inventory and production parameters, such as capacity of the facility, storage cost, inventory capacity, among others.

The mentioned instances were adapted to fit the characteristics of the PIDRP variant. For the multi-trip variant, the capacity of the vehicles must be small and there is a cost associated with the use of a vehicle to emphasize the importance of the multiple usage of a vehicle. The cost associated with the acquisition or rental of a vehicle should not be neglected in integrated problems and for this reason this parameter was added to the mentioned instances. The cost associated to the usage of a vehicle was set to 10000 ($C_v = 10000$). Vehicle capacity was set to 500 ($Q = 500$) in all instances. The variant presented foresees that the clients have temporary windows for the delivery of the requests. In this way, it was necessary to introduce these parameters in the definition of the instances. Time windows were randomly generated with values between 10% and 70% of the workday value for each client. The workday value was set to 500 ($W = 500$). The only facility is located at coordinates $(0, 0)$. The distance between the single facility available and the clients to serve was rounded to the nearest smaller integer. These were the only adaptations made to the instances presented by the authors in [46]. For all the benchmark instances the facility unit holding cost is set to 1 ($h^P = 1$) and the clients holding cost is set to 0 for all periods ($C_{h_i} = 0$). The production capacity of the single facility (C) and the production setup cost (f_t) vary with the different sets according to Table 5.2, and is the same for all periods t.

Table 5.2: Parameters according the different Sets

Set	C	f_t
1	50000	50000
2	120000	70000
3	120000	70000
4	240000	120000
5	240000	120000

For the sake of clarity, the notation of the computational results columns is presented below:

$Inst$: instance name Disti_t where i represent the number of clients and t the number of periods

ub : value of the best known upper bound

lb : value of the lower bound

T_{total} : total time used to solve the instance that includes the time to generate arcs and the time to solve the problem using cplex

T_{arcs} : total time to generate arcs

T_{ub} : total cplex time to solve the problem

$\#Arcs$: total number of arcs generated for all the periods

$\#C^{Arc}_{AVG}$: average number of clients per arc of all generated arcs

$\#C^{Arc}_{MAX}$: maximum number of clients per arc of all generated arcs

$\#L^{Arc}_{AVG}$: average number of loads per arc of all generated arcs, considering client demand anticipation

$TW_{AVG}(\%)$: average of time windows at clients for a set of clients

$\#A$: number of used arcs

$\#F$: number of used vehicles

$gap(\%)$: provides the value in percentage of the optimality gap reached at the end of the solution procedure

The computational tests were performed on a computer with a Intel Xeon CPU ES-1620 v3 with 3.5GHz and 64GB of RAM. The optimization subroutines were executed on CPLEX 12.6.

5.4.1 Solving the model exactly

For the exact resolution of the problem instances with 10, 20, 30 and 40 clients and 2 to 8 periods (2, 4, 6 and 8) were used. For instances of different sets with higher computational times, the results are not presented. The results are shown in Tables 5.3, 5.4 and 5.5.

The results are only shown for instances that take less than 7200 seconds to generate routes. The CPLEX routines were also limited to 7200 seconds. The '—' symbol is used whenever CPLEX could not find a valid solution within the established time limit.

Indeed, for Set 1 the highest instance has 20 clients and 2 periods; for Set 2 it has 30 clients and 8 periods; in Set 3 the biggest instance has 40 clients and 2 periods; Set 4 was executed to the one with 20 clients and 6 periods; and in Set 5 the biggest instance is of 30 clients and 6 periods.

Table 5.3: Results of Arc Flow Model - Set 1 and 2

	Inst	ub	lb	T_{total}	T_{arcs}	T_{ub}	#Arcs	$\#C_{AVG}^{Arc}$	$\#C_{MAX}^{Arc}$	$\#L_{AVG}^{Arc}$	TW_{AVG} (%)	#A	#F	gap (%)
Set 1	Dist10_2	131101	131101	3	0	3	35676	2,31	4	3,99	39,44	10	5	0,00
	Dist10_4	238963	238963	25	1	24	78601	2,50	4	6,97	30,56	19	12	0,00
	Dist10_6	323088	323056	1497	1	1496	265417	2,82	5	11,19	35,58	29	16	0,01
	Dist10_8	442963	442963	232	0	232	386959	2,76	5	11,63	43,24	38	23	0,00
	Dist20_2	134640	129026	7774	546	7228	6237937	4,47	7	7,46	42,01	14	5	4,17
Set 2	Dist10_2	172161	172161	1	0	1	9102	1,57	3	2,55	35,06	14	6	0,00
	Dist10_4	272563	272563	22	0	22	25067	1,69	3	4,39	43,12	27	11	0,00
	Dist10_6	377051	377051	21	0	21	35031	1,62	3	5,07	45,48	43	16	0,00
	Dist10_8	469139	469093	4930	0	4930	51075	1,81	4	10,74	40,68	55	19	0,01
	Dist20_2	278830	278830	3	0	3	31893	2,11	3	3,14	34,80	28	12	0,00
	Dist20_4	457218	455611	7203	0	7203	86698	2,14	4	6,11	41,69	54	20	0,35
	Dist20_6	651148	651083	3135	0	3135	104547	2,15	4	9,69	35,39	87	29	0,01
	Dist20_8	794620	773394	7207	1	7206	773555	3,16	5	22,37	44,51	98	33	2,67
	Dist30_2	325000	325000	6658	1	6657	335511	2,99	4	4,12	46,18	39	14	0,00
	Dist30_4	565230	560255	7201	1	7200	995404	2,97	5	6,20	37,31	70	26	0,88
	Dist30_6	940558	939903	7210	1	7209	884508	3,01	5	7,99	35,71	119	45	0,07
	Dist30_8	1347960	1327370	7214	6	7208	1620705	3,12	5	16,81	40,07	158	64	1,53

Table 5.4: Results of Arc Flow Model - Set 3 and 4

	Inst	ub	lb	T_{total}	T_{arcs}	T_{ub}	#Arcs	$\#C^{Arc}_{AVG}$	$\#C^{Arc}_{MAX}$	$\#L^{Arc}_{AVG}$	TW_{AVG} (%)	#A	#F	gap (%)
Set 3	Dist10_2	159265	159265	0	0	0	6272	1,36	2	2,02	38,74	16	5	0,00
	Dist10_4	295403	295403	2	0	2	18293	1,53	3	3,76	42,88	31	12	0,00
	Dist10_6	383228	383228	23	0	23	29558	1,62	3	5,96	38,88	42	17	0,00
	Dist10_8	495255	495255	36	0	36	58349	2,57	5	17,98	30,74	53	22	0,00
	Dist20_2	231778	231633	7265	0	7265	126387	2,68	4	3,79	45,22	25	9	0,06
	Dist20_4	414401	409884	7204	0	7204	161351	2,47	4	4,80	39,94	50	18	1,09
	Dist20_6	809436	809356	1044	0	1044	68213	1,87	4	7,01	35,41	92	38	0,01
	Dist20_8	958297	935540	7210	5	7205	630682	3,32	5	25,35	38,16	112	43	2,37
	Dist30_2	346908	345819	7201	0	7201	236719	2,75	4	4,58	43,91	41	15	0,31
	Dist30_4	623887	614118	7204	0	7204	494888	2,81	5	7,63	40,39	77	29	1,57
	Dist30_6	935402	904505	7202	1	7201	1272022	3,04	5	8,74	42,17	115	43	3,30
	Dist30_8	1191260	1064930	7205	3	7202	2039617	3,14	5	15,08	41,95	149	55	10,60
	Dist40_2	358135	349558	8301	1099	7202	8882464	4,30	7	7,12	40,93	43	16	2,39
Set 4	Dist10_2	189192	189192	7	0	7	18590	1,77	3	2,61	40,94	11	4	0,00
	Dist10_4	229833	229810	1009	1	1008	432276	3,40	5	10,59	52,80	16	6	0,01
	Dist10_6	314347	314335	546	0	546	220160	2,73	5	9,34	37,86	26	10	0,00
	Dist10_8	368200	357270	7204	0	7204	284669	2,82	5	15,59	30,10	29	13	2,97
	Dist20_2	237142	236745	7370	1	7369	400087	3,10	5	3,96	37,75	17	7	0,17
	Dist20_4	323946	316339	7636	434	7202	5519716	3,84	7	7,87	39,44	31	11	2,35
	Dist20_6	503080	485100	7213	11	7202	2835399	3,49	6	10,44	39,34	49	20	3,57

Table 5.5: Results of Arc Flow Model - Set 5

	Inst	ub	lb	T_{total}	T_{arcs}	T_{ub}	#Arcs	$\#C_{AVG}^{Arc}$	$\#C_{MAX}^{Arc}$	$\#L_{AVG}^{Arc}$	TW_{AVG} (%)	#A	#F	gap (%)
Set 5	Dist10_2	205395	205395	1	0	1	39532	2,26	3	3,77	48,54	9	5	0,00
	Dist10_4	293166	293166	21	0	21	67063	2,19	4	5,78	37,26	19	10	0,00
	Dist10_6	290025	289999	248	0	248	185198	2,72	5	10,69	33,96	24	9	0,01
	Dist10_8	347438	334572	7207	1	7206	726288	3,17	5	17,15	46,76	32	11	3,70
	Dist20_2	226903	225346	7201	1	7200	503444	3,17	5	5,46	42,93	17	6	0,69
	Dist20_4	341116	340621	7205	4	7201	1078899	3,42	6	10,85	36,62	33	12	0,15
	Dist20_6	514038	491315	7247	45	7202	4947212	3,80	6	15,63	40,84	47	21	4,42
	Dist20_8	638242	589456	7321	118	7203	8057169	3,87	6	21,47	39,40	60	26	7,64
	Dist30_2	279766	279094	7209	4	7205	1375993	3,30	5	4,94	39,01	25	9	0,24
	Dist30_4	459172	425261	7251	42	7209	5014887	3,55	5	10,81	44,30	50	18	7,39
	Dist30_6	-	633267	13500	6290	7210	17349130	3,97	7	8,48	42,65	0	0	-

In order to evaluate the quality of the model, the convergence of the GAP values, obtained during the execution of the CPLEX routines, was analyzed. This analysis was only performed for instances in which the CPLEX time limit was reached. Figures 5.3 to 5.7 show the evolution of the GAP values for the presented instances of the different sets.

Figure 5.3 shows the evolution of the gap given by CPLEX for instance 20_2 of set 1, which reached the established time limit. Indeed, after 1300 seconds of execution the gap was less than 5%. As presented in Figure 5.4, in 60% of the instances of set 2 the gap value is less than 5% after 1000 seconds. For the instances of Set 3, Figure 5.5, 75% had a gap less than 5% in 700 seconds. For instance 30_8 the CPLEX routine found the largest gap, of 10.6%, within 7200 seconds. In Set 4, two of the instances (Figure 5.6) had the gap less than 5% in 1500 seconds. For the remaining instances, CPLEX found a solution with a gap smaller than 5% in 2500 seconds. Finally, Figure 5.7 presents the gap evolution for Set 5. Here, 57% of the instances had a gap less than 5% after 2000 seconds, and 28% of the instances had a first valid solution after 4000 seconds. In the instances analyzed in the Figures 5.3 to 5.7, 84% achieve a gap value smaller than 5% in 3600 seconds.

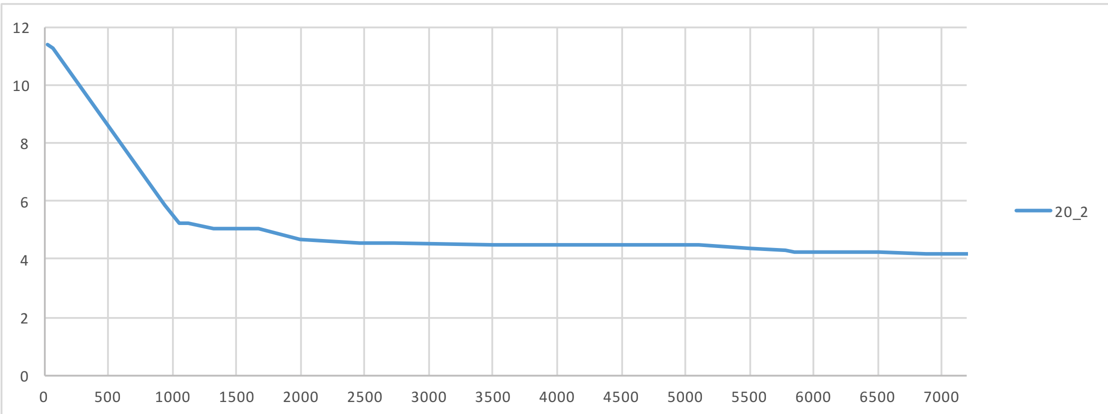

Figure 5.3: Gap evolution (%) - Set 1

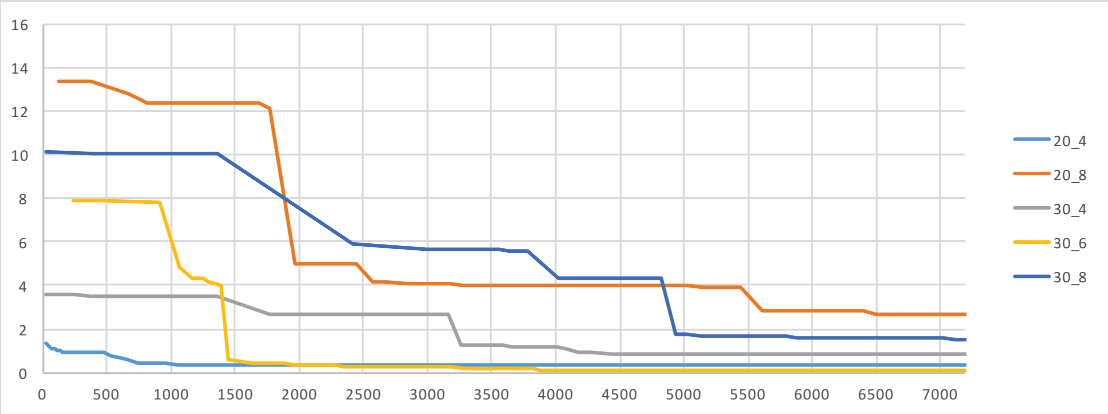

Figure 5.4: Gap evolution (%) - Set 2

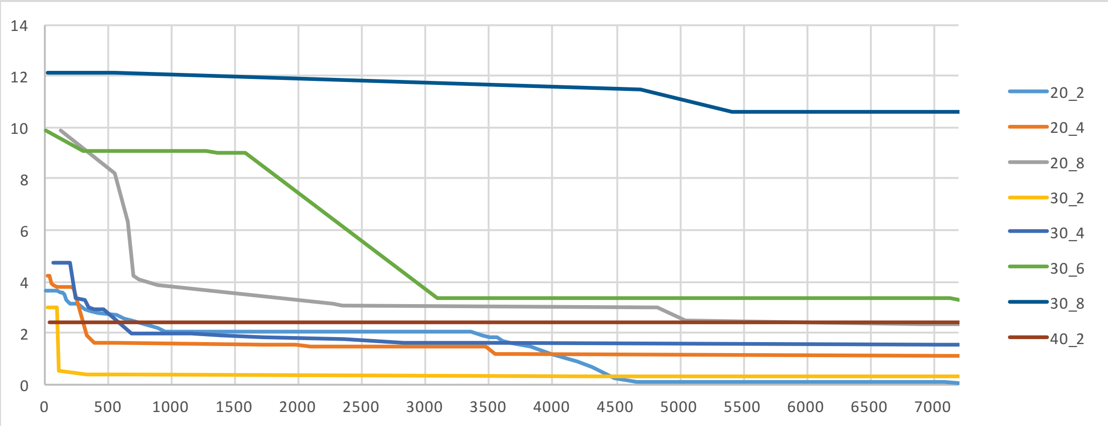

Figure 5.5: Gap evolution (%) - Set 3

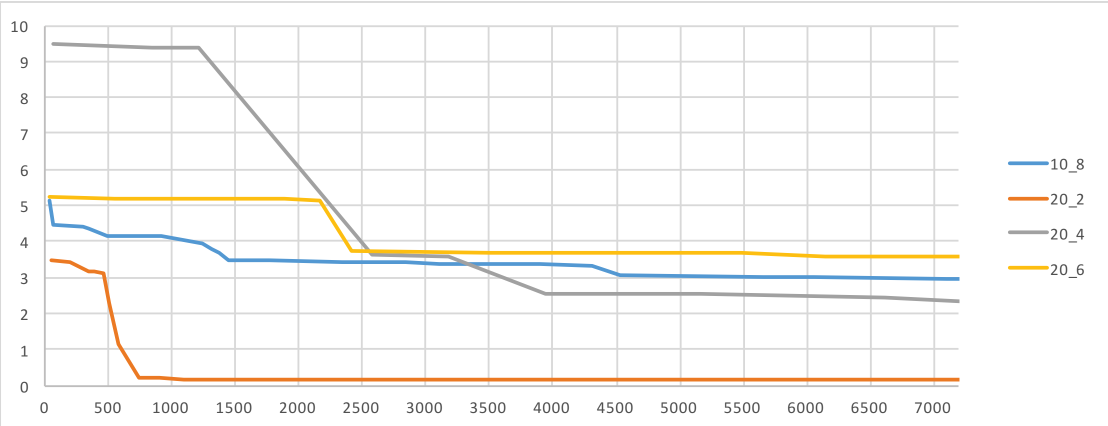

Figure 5.6: Gap evolution (%) - Set 4

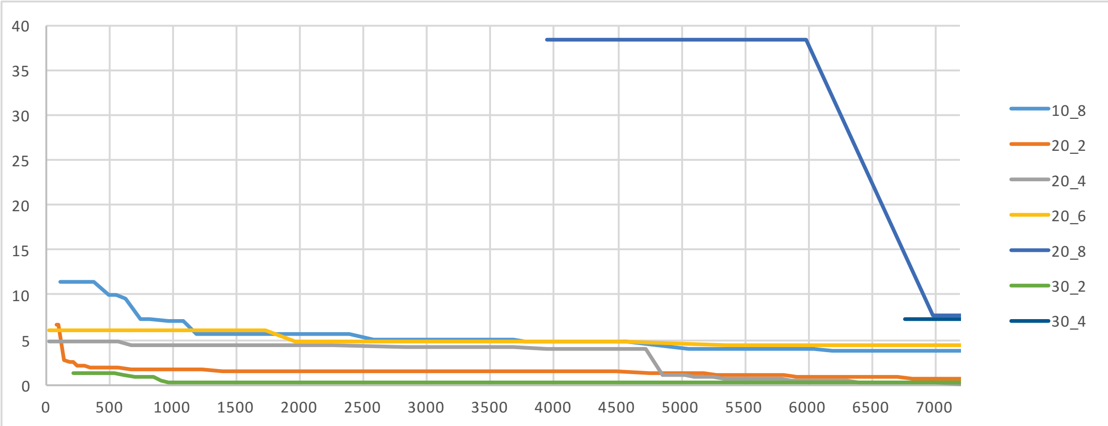

Figure 5.7: Gap evolution (%) - Set 5

5.4.2 Solving the model exactly with arcs limitations

Computational tests were performed for all instances adapted from Bard and Nananukul [46] with a limit on the number of arcs generated per period.

Since the computational times to solve the instances to the optimality are large, an analysis was made in order to understand where most of the computing time was used. Through the observation of the computational times presented in the previous section (Section 5.4.1), the model proved to be efficient. However the generation of single-trips took a lot of processing time. Although arc processing seem a simple client recombination algorithm, it requires a larger number of verifications and replications.

The generation of arcs is a complex process, since the problem allows for the demand of a customer to be anticipated. In this way, arc generation must take into account customer satisfaction for the current period and for future periods, which allows for many more combinations. Another factor which justifies the long computational time is the replications and validations of arcs. When an arc is created, restrictions regarding the capacity of vehicles, time windows and workday limit must be taken into account. For each arc, a start time and an end time are created, which allows for the single-trip to initialize and terminate at different times considering time windows and workday constraints. The time is discrete and unitary, therefore more arcs may be generated.

Although the previously described factors already justify long computational times, there is also the post-processing of the generated arcs in order to eliminate single-trips that are dominated by others, allowing for the model to process much less arcs. One arc dominates another when it visits exactly the same customers for the same periods at a lower cost, respecting all the problem constraints.

In order to run all instances in acceptable computational times, the number of arcs generated was limited per period. Since the generation of arcs occurs sequentially, no limit was imposed to the total number of arcs, to avoid reaching the limit with the first clients and to not generate any valid single-trip for the latest clients, invalidating any solution.

After the validation of an arc, it is replicated according to the possible time interval throughout the workday, along its possible minimum and maximum begin times. Since each arc has a distinct time interval, the number of replications is variable. In order to limit the number of replications an upper bound of replications was defined. When the upper limit is not reached, then all replications are considered. If the replication limit is reached, then the replication interval is readjusted so that the number of replications is distributed evenly over the time interval.

A formula has been created to calculate the number of routes by period and customer. This formula is presented in (5.20) and provides a balanced distribution of arcs per periods. This considers that in the last period it is not possible to anticipate any requests, but in the first one there is the possibility of all being anticipated.

$$\#arcs^{t}_{c_{t'}} = \frac{\#arcs}{(N \times (T - t))}, \forall t \in T \qquad and \qquad \forall c_{t'} \in N \times (T - t) \qquad and \qquad t' \geq t \tag{5.20}$$

The $\#arcs^{t}_{c_{t'}}$ represents the maximum number of arcs that may be generated for period t and client $c_{t'}$. The $\#arcs$ represents the maximum number of total arcs that may be created. Parameter T represents the planning horizon periods and each individual period is denoted by t.

The arc limit is applied per customer and period and takes into account if the demand is or is not anticipated for future periods. Thus, there are generated as many arcs as $\#arcs^{t}_{c_{t'}}$ for the client $c_{t'}$, in period t that satisfy the demand for t', where t' is equal or greater than t.

It should be noted that the limit of arcs is defined by period and that not all periods reach this limit. Since these situations occur frequently, it is expected that the final number of generated arcs for all periods is smaller than the established limit $\#arcs^{t}_{c_{t'}} \times T$. However, when this situation occurs, the number of arcs that are not used is distributed over the remaining periods.

For the resolution of the limited-arc problem, instances with 10, 20, 30, 40 and 50 clients with 2 to 8 periods (2, 4, 6 and 8) were used. The performed computational tests to evaluate the proposed limits are presented through Table 5.6 to Table 5.10. The CPLEX routines were limited to 3600 seconds and the arc generation process was limited to 5000 arcs per period, and 25 replications by arc.

Table 5.6: Results of Arc Flow Model with arc limitation - Set 1

Inst	ub	lb	T_{total}	T_{arcs}	T_{ub}	#Arcs	$\#C^{Arc}_{AVG}$	$\#C^{Arc}_{MAX}$	$\#L^{Arc}_{AVG}$	TW_{AVG} (%)	#A	#F	gap (%)
Dist10_2	131101	131101	4	0	4	9666	2,55	4	4,52	39,44	10	5	0,00
Dist10_4	240733	240733	2	0	2	29774	2,70	4	7,52	30,56	20	12	0,00
Dist10_6	324957	324957	17	0	17	96144	3,04	5	12,29	35,58	28	16	0,00
Dist10_8	442963	442941	15	0	15	117243	3,01	5	12,83	43,24	38	23	0,00
Dist20_2	134735	134735	49	2	47	151613	3,48	7	5,68	42,01	14	5	0,00
Dist20_4	216124	216103	323	20	303	368116	3,61	8	10,50	43,27	27	9	0,01
Dist20_6	322936	321681	3622	21	3601	556792	3,38	7	13,63	39,22	42	14	0,39
Dist20_8	437588	390346	3661	60	3601	768125	3,30	7	17,03	41,48	56	19	10,80
Dist30_2	223932	223910	213	14	199	161808	3,30	6	5,33	37,23	23	10	0,01
Dist30_4	360682	357677	3646	45	3601	393283	3,22	7	9,31	38,43	42	16	0,83
Dist30_6	540298	516785	4014	414	3600	601194	3,12	8	11,96	43,25	66	25	4,35
Dist30_8	756757	715215	3653	52	3601	776040	3,06	7	15,22	34,22	90	35	5,49
Dist40_2	263094	256696	3735	134	3601	194796	3,28	7	5,30	44,66	33	12	2,43
Dist40_4	498430	498154	3893	293	3600	403238	3,07	7	8,41	41,89	70	22	0,06
Dist40_6	760747	699345	4765	1165	3600	627607	3,01	7	11,41	42,32	111	34	8,07
Dist40_8	1058650	944168	5075	1474	3601	873357	2,99	6	14,80	42,94	144	51	10,81
Dist50_2	334740	334707	1912	1681	231	199539	3,21	7	5,32	43,24	44	16	0,01
Dist50_4	676024	665150	3784	184	3600	441379	2,97	6	8,18	41,27	92	33	1,61
Dist50_6	1169340	1143950	3648	47	3601	615546	2,82	6	10,39	34,64	142	61	2,17
Dist50_8	1376180	1218520	4797	1196	3601	866250	2,85	7	13,85	42,43	183	68	11,46

Table 5.7: Results of Arc Flow Model with arc limitation - Set 2

Inst	ub	lb	T_{total}	T_{arcs}	T_{ub}	#Arcs	$\#C^{Arc}_{AVG}$	$\#C^{Arc}_{MAX}$	$\#L^{Arc}_{AVG}$	TW_{AVG} (%)	#A	#F	gap (%)
Dist10_2	172326	172326	0	0	0	1830	1,73	3	2,89	35,06	14	6	0,00
Dist10_4	282161	282161	7	0	7	4777	1,90	3	5,14	43,12	26	12	0,00
Dist10_6	379122	379098	82	0	82	5879	1,85	3	5,75	45,48	45	16	0,01
Dist10_8	469303	469303	20	0	20	11804	2,12	4	13,23	40,68	56	19	0,00
Dist20_2	288969	288948	2	0	2	9274	2,32	3	3,41	34,80	28	13	0,01
Dist20_4	468541	468513	13	0	13	24499	2,43	4	7,24	41,69	54	21	0,01
Dist20_6	662933	662867	193	0	193	28489	2,41	4	10,98	35,39	87	29	0,01
Dist20_8	800132	799183	3604	0	3604	164885	3,25	5	22,72	44,51	99	34	0,12
Dist30_2	325420	325389	17	0	17	85071	3,13	4	4,19	46,18	39	14	0,01
Dist30_4	576052	575995	484	1	483	269462	3,02	5	6,67	37,31	71	27	0,01
Dist30_6	946532	946437	1144	0	1144	262435	3,07	5	7,86	35,71	121	45	0,01
Dist30_8	1360550	1350900	3608	1	3607	319037	3,07	5	13,60	40,07	158	65	0,71
Dist40_2	322876	322848	98	13	85	193603	3,30	6	5,48	45,34	39	14	0,01
Dist40_4	723183	722180	3640	33	3607	428683	3,26	6	8,98	38,04	84	35	0,14
Dist40_6	1110870	1092760	3646	45	3601	416856	2,99	6	8,77	42,31	135	53	1,63
Dist40_8	1447650	1432550	3616	14	3602	595827	3,02	7	12,33	36,26	178	69	1,04
Dist50_2	434180	434137	911	6	905	195433	3,06	6	5,05	42,06	51	20	0,01
Dist50_4	689275	682918	4320	720	3600	434424	3,20	8	9,12	38,92	93	33	0,92
Dist50_6	1314040	1281930	3989	388	3601	507427	2,94	6	9,39	40,28	163	63	2,44
Dist50_8	1860500	1745510	3811	211	3600	599142	2,88	6	11,57	36,73	225	94	6,18

Table 5.8: Results of Arc Flow Model with arc limitation - Set 3

Inst	ub	lb	T_{total}	T_{arcs}	T_{ub}	#Arcs	$\#C_{AVG}^{Arc}$	$\#C_{MAX}^{Arc}$	$\#L_{AVG}^{Arc}$	TW_{AVG} (%)	#A	#F	gap (%)
Dist10_2	169265	169265	0	0	0	1128	1,47	2	2,16	38,74	16	6	0,00
Dist10_4	304641	304641	5	1	4	3140	1,71	3	4,18	42,88	30	13	0,00
Dist10_6	384846	384846	6	0	6	6781	1,88	3	6,73	38,88	43	17	0,00
Dist10_8	496056	496056	20	0	20	24261	3,10	5	22,81	30,74	53	22	0,00
Dist20_2	231883	231868	13	0	13	29326	2,84	4	4,10	45,22	25	9	0,01
Dist20_4	422390	422353	272	0	272	42163	2,65	4	5,03	39,94	49	19	0,01
Dist20_6	835711	835628	22	0	22	18972	2,13	4	7,67	35,41	91	40	0,01
Dist20_8	971500	971403	168	1	167	101836	3,14	5	23,09	38,16	114	43	0,01
Dist30_2	356019	355993	29	0	29	55615	2,90	4	4,90	43,91	41	16	0,01
Dist30_4	627953	627891	264	1	263	162331	3,04	5	8,51	40,39	77	29	0,01
Dist30_6	933302	932295	3609	1	3608	342497	3,19	5	9,06	42,17	114	43	0,11
Dist30_8	1187290	1169000	3601	1	3600	443925	3,16	5	12,80	41,95	147	55	1,54
Dist40_2	364662	364651	126	3	123	207264	3,42	7	5,57	40,93	44	16	0,00
Dist40_4	672640	663028	3619	11	3608	396758	3,16	6	8,63	38,71	86	32	1,43
Dist40_6	1155420	1139600	3619	18	3601	377745	2,90	7	9,07	35,78	137	57	1,37
Dist40_8	1482520	1375020	3624	23	3601	528759	2,99	7	12,51	39,98	180	72	7,25
Dist50_2	433007	431907	3624	20	3604	209469	3,17	6	5,24	36,68	49	21	0,25
Dist50_4	803880	790757	3648	48	3600	446719	2,97	6	8,14	40,79	103	39	1,63
Dist50_6	1254230	1220280	3681	80	3601	481560	2,92	6	8,74	37,88	157	61	2,71
Dist50_8	1652740	1530240	3677	76	3601	639381	2,90	6	12,45	41,78	213	79	7,41

Table 5.10: Results of Arc Flow Model with arc limitation - Set 5

Inst	ub	lb	T_{total}	T_{arcs}	T_{ub}	#Arcs	$\#C^{Arc}_{AVG}$	$\#C^{Arc}_{MAX}$	$\#L^{Arc}_{AVG}$	TW_{AVG} (%)	#A	#F	gap (%)
Dist10_2	206698	206698	0	0	0	7051	2,43	3	3,82	48,54	10	5	0,00
Dist10_4	293166	293166	5	0	5	18680	2,42	4	6,47	37,26	19	10	0,00
Dist10_6	299376	299347	48	0	45	79667	3,03	5	11,84	33,96	23	10	0,01
Dist10_8	347700	346449	3605	0	3600	203665	3,36	5	18,72	46,76	31	11	0,36
Dist20_2	226903	226903	63	1	60	133736	3,26	5	5,55	42,93	17	6	0,00
Dist20_4	350880	344555	3607	1	3601	232176	3,24	5	8,96	36,62	33	13	1,80
Dist20_6	514067	501527	3629	1	3608	527098	3,41	6	12,54	40,84	48	21	2,44
Dist20_8	629659	608559	3636	2	3601	739668	3,33	6	15,71	39,40	59	25	3,35
Dist30_2	288760	288373	3609	0	3603	205438	3,13	5	4,89	39,01	24	9	0,13
Dist30_4	446089	433473	3616	1	3601	389652	3,08	5	8,14	44,30	49	17	2,83
Dist30_6	676189	650326	3642	10	3601	596574	3,13	7	11,55	42,65	72	28	3,82
Dist30_8	879662	828515	3666	14	3600	801687	3,16	6	14,82	45,53	94	36	5,81
Dist40_2	363871	363222	3614	7	3600	197598	3,06	5	4,79	40,58	34	14	0,18
Dist40_4	603742	587203	3633	12	3601	420220	3,03	5	8,07	38,20	65	26	2,74
Dist40_6	830717	795661	3730	89	3601	612355	3,00	6	11,04	42,15	93	36	4,22
Dist40_8	1144300	971973	3721	52	3601	849536	3,00	7	14,43	40,42	126	51	15,06
Dist50_2	384999	384961	2048	59	1980	215185	3,15	6	5,17	40,58	40	14	0,01
Dist50_4	674616	663433											

5.4.3 Comparative discussion

This section presents comparative results for the arc flow model with and without limits in terms of generated arcs. This comparison is made in terms of the total arcs processed by the model, and the total processing time. The quality of the solution is evaluated through the value of the upper bound. Note that in the exact method all possible routes are generated and processed. The arc flow model is solved through CPLEX to optimality. The limited arc flow model has a higher bound, and the number of generated routes was set to 5000 for each period and a maximum of 25 replications per arc. The CPLEX routines ended after 3600 seconds, considering the current incumbent solution.

Table 5.11 shows the comparative results between the model with limits and the exact approach. The last three columns represent the improvements of the limited model relative to the exact model in terms of percentage.

Table 5.11: Comparative analysis of the arc flow model with and without limit on arc generation

	Inst	ub		#Arcs		T_{total}		Comparison (%)		
		Exact	Limited	Exact	Limited	Exact	Limited	ub	#Arcs	T_{total}
Set 1	Dist10_2	131101	131101	35676	9666	3	4	0,00	72,91	-33,33
	Dist10_4	238963	240733	78601	29774	25	2	-0,74	62,12	92,00
	Dist10_6	323088	324957	265417	96144	1497	17	-0,58	63,78	98,86
	Dist10_8	442963	442963	386959	117243	232	15	0,00	69,70	93,53
	Dist20_2	134640	134735	6237937	151613	7774	49	-0,07	97,57	99,37
Set 2	Dist10_2	172161	172326	9102	1830	1	0	-0,10	79,89	100,00
	Dist10_4	272563	282161	25067	4777	22	7	-3,52	80,94	68,18
	Dist10_6	377051	379122	35031	5879	21	82	-0,55	83,22	-290,48
	Dist10_8	469139	469303	51075	11804	4930	20	-0,03	76,89	99,59
	Dist20_2	278830	288969	31893	9274	3	2	-3,64	70,92	33,33
	Dist20_4	457218	468541	86698	24499	7203	13	-2,48	71,74	99,82
	Dist20_6	651148	662933	104547	28489	3135	193	-1,81	72,75	93,84
	Dist20_8	794620	800132	773555	164885	7207	3604	-0,69	78,68	49,99
	Dist30_2	325000	325420	335511	85071	6658	17	-0,13	74,64	99,74
	Dist30_4	565230	576052	995404	269462	7201	484	-1,91	72,93	93,28
	Dist30_6	940558	946532	884508	262435	7210	1144	-0,64	70,33	84,13
	Dist30_8	1347960	1360550	1620705	319037	7214	3608	-0,93	80,31	49,99
	Dist10_2	159265	169265	6272	1128	0	0	-6,28	82,02	0,00

Continues on next page

Table 5.11 – continued from previous page

	Inst	ub		#Arcs		T_{total}		Comparison (%)		
		Exact	Limited	Exact	Limited	Exact	Limited	ub	#Arcs	T_{total}
Set 3	Dist10_4	295403	304641	18293	3140	2	5	-3,13	82,83	-150,00
	Dist10_6	383228	384846	29558	6781	23	6	-0,42	77,06	73,91
	Dist10_8	495255	496056	58349	24261	36	20	-0,16	58,42	44,44
	Dist20_2	231778	231883	126387	29326	7265	13	-0,05	76,80	99,82
	Dist20_4	414401	422390	161351	42163	7204	272	-1,93	73,87	96,22
	Dist20_6	809436	835711	68213	18972	1044	22	-3,25	72,19	97,89
	Dist20_8	958297	971500	630682	101836	7210	168	-1,38	83,85	97,67
	Dist30_2	346908	356019	236719	55615	7201	29	-2,63	76,51	99,60
	Dist30_4	623887	627953	494888	162331	7204	264	-0,65	67,20	96,34
	Dist30_6	935402	933302	1272022	342497	7202	3609	0,22	73,07	49,89
	Dist30_8	1191260	1187290	2039617	443925	7205	3601	0,33	78,23	50,02
	Dist40_2	358135	364662	8882464	207264	8301	126	-1,82	97,67	98,48
Set 4	Dist10_2	189192	189192	18590	3652	7	1	0,00	80,36	85,71
	Dist10_4	229833	230255	432276	111598	1009	142	-0,18	74,18	85,93
	Dist10_6	314347	315073	220160	68274	546	70	-0,23	68,99	87,18
	Dist10_8	368200	376568	284669	119594	7204	985	-2,27	57,99	86,33
	Dist20_2	237142	237172	400087	122241	7370	84	-0,01	69,45	98,86
	Dist20_4	323946	333044	5519716	367283	7636	3602	-2,81	93,35	52,83
	Dist20_6	503080	500727	2835399	465238	7213	3602	0,47	83,59	50,06
Set 5	Dist10_2	205395	206698	39532	7051	1	0	-0,63	82,16	100,00
	Dist10_4	293166	293166	67063	18680	21	5	0,00	72,15	76,19
	Dist10_6	290025	299376	185198	79667	248	48	-3,22	56,98	80,65
	Dist10_8	347438	347700	726288	203665	7207	3605	-0,08	71,96	49,98
	Dist20_2	226903	226903	503444	133736	7201	63	0,00	73,44	99,13
	Dist20_4	341116	350880	1078899	232176	7205	3607	-2,86	78,48	49,94
	Dist20_6	514038	514067	4947212	527098	7247	3629	-0,01	89,35	49,92
	Dist20_8	638242	629659	8057169	739668	7321	3636	1,34	90,82	50,33
	Dist30_2	279766	288760	1375993	205438	7209	3609	-3,21	85,07	49,94
	Dist30_4	459172	446089	5014887	389652	7251	3616	2,85	92,23	50,13

As depicted in Table 5.11 the number of generated arcs was substantially reduced. However, in some instances the value of the upper bound in the limited approach was better or equal to the value obtained in the exact approach; this occurs when the gap of the exact model is greater than or equal to zero, respectively.

Table 5.12 presents an overall analysis for the different sets. For Set 1, for example, although the total time has improved by 70.09% and that 73.21% of the arcs have been generated, these values are reflected in the cost of the solution which had a penalty of 0.28%.

Table 5.12: Comparative percentage analysis with the average improvement of the arc flow model

	ub	#Arcs	T_{total}
Set 1	-0,28	73,21	70,09
Set 2	-1,37	76,11	48,45
Set 3	-1,63	76,90	58,02
Set 4	-0,72	75,41	78,13
Set 5	-0,58	79,26	65,62
Total	-0,91	76,18	64,06

For all the analyzed instances, on average, less 76,18% of arcs were generated and the total time was decreased by 64,06%. The total cost suffered an average penalty of 0,91%. These values are promising since there is no relevant penalty at the upper limit of the arc flow model and there is a clear reduction in the number of routes and processing times.

5.5 Conclusions

The MPIDRPTW is a management science problem that integrates important problems such as PDP, MVRP and IRP. This type of problems consider production, inventory, distribution and routing decisions simultaneously and frequently occurs in the logistics and transportation fields.

The PDP is primarily concerned with production and distribution decisions. Orders vary by period and can be anticipated. However, there is no concern with inventory management. IRP, despite not dealing with production decisions, is concerned with routing and inventory management, either at the facility or at the customer. The PIDRP is a problem which has received special attention in recent years and explores the integration of these important problems. The PIDRP also takes into account routing decisions commonly present in MVRP problems. This problem determines a set of routes to satisfy customers taking into account the possibility of multiple use of a vehicle during the planning period. The MPIDRPTW variant also includes decisions of multiple usage of the vehicles as well as the compliance of time windows in the distribution of goods to different customers. The multiple usage of a vehicle is frequently used when clients are distributed within small geographic areas.

The aim of this integration is the cost minimization according to all the decisions associated with the entire system. Although the resolution of integrated problem leads to better solutions from the global perspective, the problem grows in size, which increases its complexity.

The main objective of the MPIDRPTW is to serve a set of clients at minimum cost during the planning horizon. The clients have a time varying demand that must be fulfilled within a specific time window. This problem deals with production and inventory decisions at a single facility and uses a fleet of homogeneous vehicles to perform the distribution of goods. Each vehicle can make more than a single-trip during the planning horizon. A client may receive a future period demand, and split deliveries are not allowed. An order of a client may be completed from periodic production or through inventory held at the facility. The facility has a capacity and incurs in a setup cost each time a production period is scheduled.

An innovative method of arc flow formulation was proposed to solve the MPIDRPTW which have a distinct graph-based structure from the commodity flow models. In this method the nodes represent times instead of clients and an arc denotes a single-trip. This optimization technique is less intuitive and requires a set of pre-existing single-trips or arcs. However, this technique is more efficient than other methods.

The model was tested through exhaustive computational tests performed on a set of benchmark instances from the literature. The model proved to be efficient, however, the arc generation is computationally expensive. The complexity inherent to the arcs generation grows exponentially with the increase of the number of clients and time periods, due to the many possible combinations between clients allowing for the anticipation of the demand fulfillment.

Chapter 6

The multi-trip production, inventory, distribution and routing problem with time windows: heuristic and hybrid approaches

Outline

6.1 Introduction

The PIDRP is an integrated problem that has had particular attention during the last few years. Several authors analyze specificities of the problems by focusing on different variants of PIDRP. Some authors make the problems less restrictive, assuming various unlimited capacities such as the load carried by the vehicles. Despite the variants, the authors have a common goal. This main objective is the integration of production planning, inventory management and routing and distribution planning of goods. The peculiarities of the different variants may have a significant influence on the resolution method that should be used.

When exact methods are explored, as presented in the previous chapter (Chapter 5), the optimal solution of the problem is ensured. However, high computational times may be impractical in the supply chain. Sometimes, in supply chains, it may not be acceptable to wait too long, since the plans may have to be recalculated according to management or even clients impositions.

Heuristic methods are frequently used to overcome such difficulties. Although these do not guarantee the optimal solution, they may present good solutions in acceptable computational times. Several authors [46, 50, 49, 53, 54] have studied several heuristic approaches. Most of them choose two-phase algorithms and the exploration of solution space neighborhoods, getting good quality solutions.

It is possible to take advantage of the two approaches, exact methods and heuristics, and create heuristics based on exact models. The heuristic method can be used to solve more complex problems, and the exact models can provide optimal solutions for the sub-problems, providing a final solution of better quality.

There are several valid approaches for solving hard problems, but some lead to better results regarding objective function values and concerning computational times. The main objective of the various authors is to define a positive trade-off between the expected execution times and the objective function value.

6.2 Matheuristic approaches

The effort of cooperation between exact and heuristic methods in solving integrated problems is denominated by matheuristics. A matheuristic is an optimization algorithm which conciliates mathematical programming techniques with heuristic methods in the search of good quality solutions. This type of algorithm has motivated the interest of several authors who try to apply this approach to the most varied type of problems.

The use of matheuristics can occur in two different ways. Using mathematical programming techniques to improve the results obtained by the heuristics or, on the contrary, improving the results obtained through model relaxation with the use of heuristics. The first approach is the most frequent in the literature. Often, mathematical models are combined with local search techniques exploring the solution space defined by neighbor solutions. A heuristic that can be integrated with a mathematical programming model is, for example, the VNS, which is an elaborate heuristic, and so often denoted as meta-heuristic. This meta-heuristic explores, in a systematic way, a set of neighborhoods in order to find good solutions in acceptable computational times. This approach was proposed by Mladenović and Hansen [62]. In this chapter, two different matheuristics will be presented for the MPIDRPTW problem described in the previous chapter (Chapter 5). The matheuristics exchange information between the exact arc flow model proposed in Section 5.2 and the procedure of local search, in this case, the VNS.

6.2.1 Neighborhood structures

Neighborhoods $\mathcal{N}_0$, $\mathcal{N}_1$ and $\mathcal{N}_2$ allow a modification in the set of visited clients that is performed through a given arc. Therefore, they are called routing neighborhoods, since they only change the structure of a single-trip.

The first two neighborhoods $\mathcal{N}_0$ and $\mathcal{N}_1$ are denominated “move client”, since a client changes his position. This change can be made within the same period, through $\mathcal{N}_0$ or for different periods, through $\mathcal{N}_1$. An aperiodic change occurs when this change occurs in the same period ($\mathcal{N}_0$). This modification can occur within the same arc, changing only the client position in the single-trip or can occur between different arcs, changing the constitution of two different single-trips. In the aperiodic change, no demand is anticipated for any customer. If the change occurs between different periods ($\mathcal{N}_1$), then the anticipation of customer requests is allowed, but backlogging never occurs. An anticipation may be made in the current period or may be delayed for a future period, provided that the customers request is not delivered later than the period when the demand is needed. Periodic modification occurs when there are changes between different arcs that do not belong to the same distribution period. Note that the order in which a customer is visited can be changed, since an arc is rearranged in order to represent a more attractive single-trip.

A simple example of a move in the $\mathcal{N}_0$ is presented through Figure 6.1. When client c_6 is removed from route a the new route a' only visits the client c_3, c_1 and c_4. On the other hand, the route b' fulfills three client (c_5, c_6 and c_2) instead of the two served by route b (c_5 and c_2). This example of move is performed in the same period t_1. Note that the client position in the figure is not modified in order to clarify the neighborhood "move client".

Figure 6.2 represents a move in the $\mathcal{N}_1$, which is performed between routes serving clients

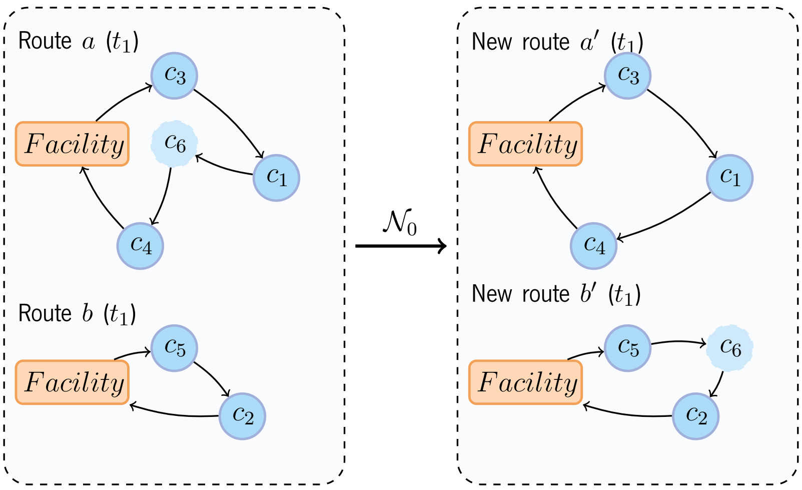

Figure 6.1: Example of a move in neighborhood $\mathcal{N}_0$

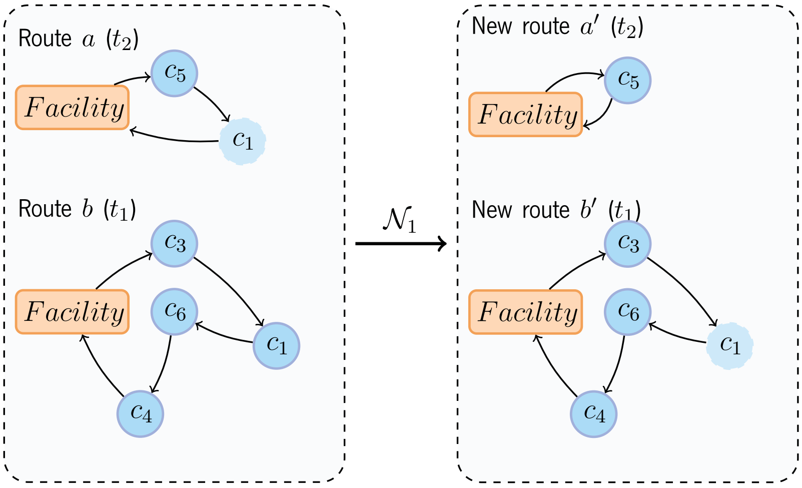

Figure 6.2: Example of a move in neighborhood $\mathcal{N}_1$

in different periods of the planning horizon. Client c_1 is removed from route a that serves client c_5 and c_1 during period t_2. Since route b distributes during period t_1 and also fulfills client c_1, it is possible to anticipate the demand associated to this client. This anticipation can be observed through route b'. Route a' visits only one client.

Neighborhood $\mathcal{N}_2$, the "swap two clients", allows for the permutation of two different clients. This neighborhood allows the exchange of two clients in the same route or in different routes. When the exchange occurs between two clients in the same route, then only one arc is modified. When the permutation occurs between two different single-trips, these two arcs change. This exchange is only permitted for the same distribution period, not allowing the anticipation of future periods if they are not already considered in the customers being exchanged.

Figure 6.3, shows an example of a swap between two clients representing a move in neighborhood $\mathcal{N}_2$. The permutation represented in the figure occurs for period t_1. Client c_6 in route a is replaced by client c_2 in route b, creating a neighbor solution with routes a' and b'. Since the permutation occurs between two different arcs of the same period, two single-trips are modified. Despite this modification the number of clients associated with each arc remains the same. Route a' serves now clients c_3, c_1, c_2 and c_4, and route b' fulfills the demand associated to clients c_5

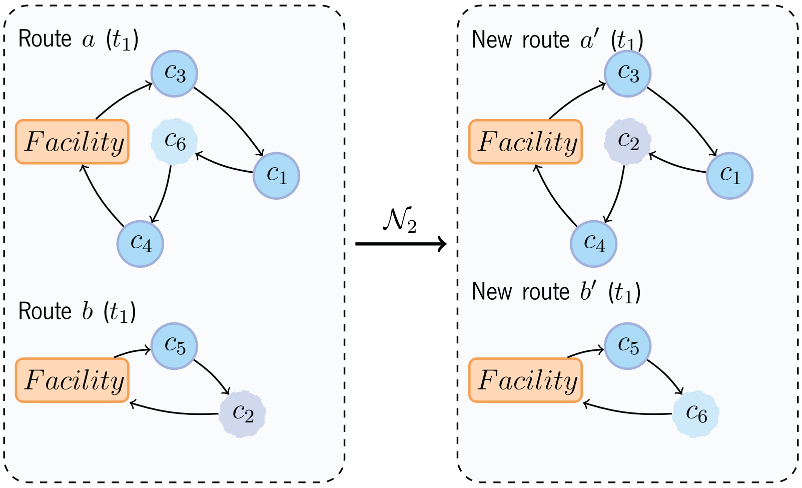

Figure 6.3: Example of a move in neighborhood $\mathcal{N}_2$

and c_6. In the figure the position of the clients exchanged is the same to clarify the swap.

Each vehicle has an associated workday for distributing orders to clients. During each workday, a vehicle can perform more than one single-trip. The neighborhoods $\mathcal{N}_3$ and $\mathcal{N}_4$ are considered workday neighborhoods since they consider changes related to a complete arc. Neighborhood $\mathcal{N}_3$ is namely "move arc" and considers the modification of the position associated to a single-trip. This shift of an arc is only allowed within the same distribution period. It is expected that the exchange occurs between routes associated with different vehicles, however the neighbors for an arc within the same workday and the same vehicle are also considered valid.

As shown in Figure 6.4, routes a and b are performed by vehicles v_1 and v_2, respectively. Vehicles v_1 and v_2 work during the first distribution period (t_1). The single-trip r_1 performed by vehicle v_1 is removed from route a and then inserted in route b made by vehicle v_2, originating route b'. Route a' performs now only a single trip r_2, while route b' visits clients associated to two different single trips (r_1' and r_3).

To create a neighbor of neighborhood $\mathcal{N}_4$, known as "swap arcs", it is necessary to make the exchange between two single-trips. Neighborhood $\mathcal{N}_4$ creates new valid solutions exchanging an arc from two different workdays. A valid arcs swap can only be made for the same distribution

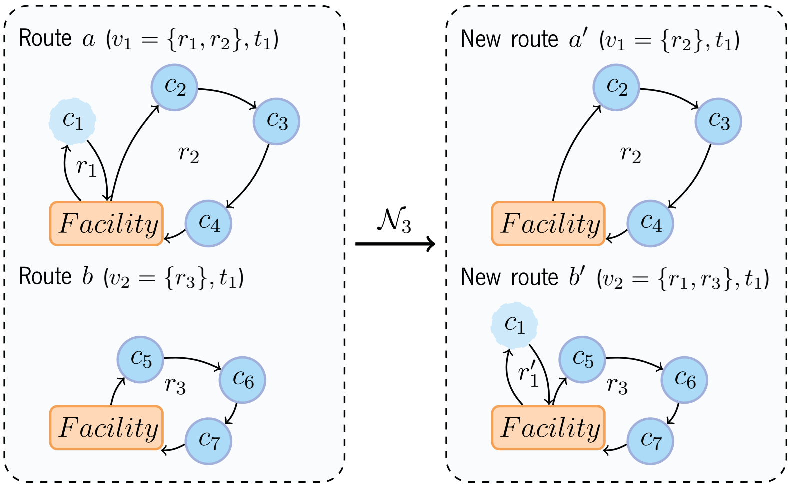

Figure 6.4: Example of a move in neighborhood $\mathcal{N}_3$

period. Although it is allowed to exchange two arcs associated with different vehicles, this exchange can occur for the same vehicle, being considered as valid neighbor.

Figure 6.5 presents an arc exchange between routes a and b for vehicles v_1 and v_2, respectively. Route a satisfies client c_1 through the arc r_1 and clients c_2, c_3 and c_4 using the single-trip r_2. Route a is performed by vehicle v_1, while route b is done by vehicle v_2 that performs a single-trip. Clients c_5, c_6 and c_7 are fulfilled over the route b. Route a' performs now the single-trip r_1

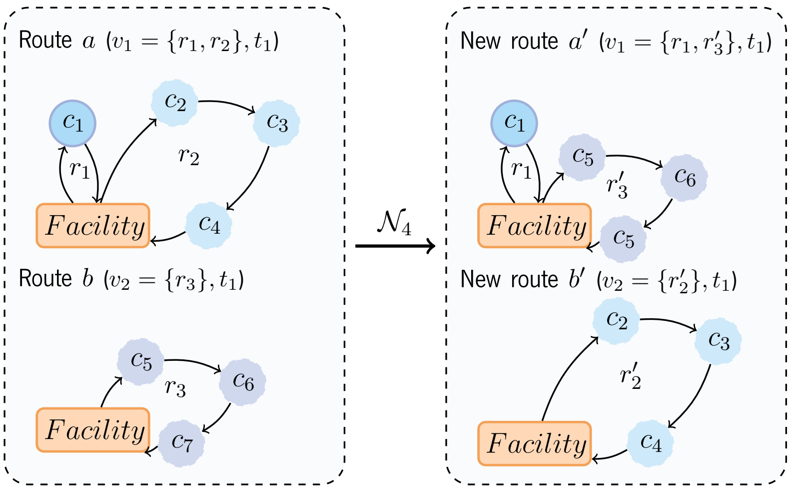

Figure 6.5: Example of a move in neighborhood $\mathcal{N}_4$

and the single-trip r'_3 instead of arc r_2 through vehicle v_1. Route b' is performed by vehicle v_2 and visits only the set of clients presented in the single-trip r'_2. This arc exchange is only possible since these vehicles operate at the same distribution period t_1.

Neighborhoods $\mathcal{N}_5$ and $\mathcal{N}_6$ are called vehicle neighborhoods, since they force the use of new vehicles. The neighborhood $\mathcal{N}_5$ is called "use new vehicles". This neighborhood considers that an arc of a route is removed from the used vehicle and then assigned to a new one. The single-trip attributed to the new vehicle has to be performed in the same period as the original route.

Figure 6.6 shows that in route a the vehicle v_1 performed three single trips (r_1, r_2 and r_3). By exploration of neighborhood $\mathcal{N}_5$ these single-trips were split over three different vehicles, and two new vehicles were used. Route a' satisfies customer c_1 through vehicle v_1, route a'' meets the needs of clients c_2, c_3 and c_4 through vehicle v_2, and finally route a''' delivers customer c_5 needs using vehicle v_3.

The neighborhood $\mathcal{N}_6$ is called "client round trip", since the number of new vehicles used is equal to the number of round single trips generated. Each new vehicle is assigned to a single client. An arc is selected and each client in the single-trip is visited by a different vehicle. A new vehicle leaves the facility, fulfills the single client and then returns to the facility. The new vehicles

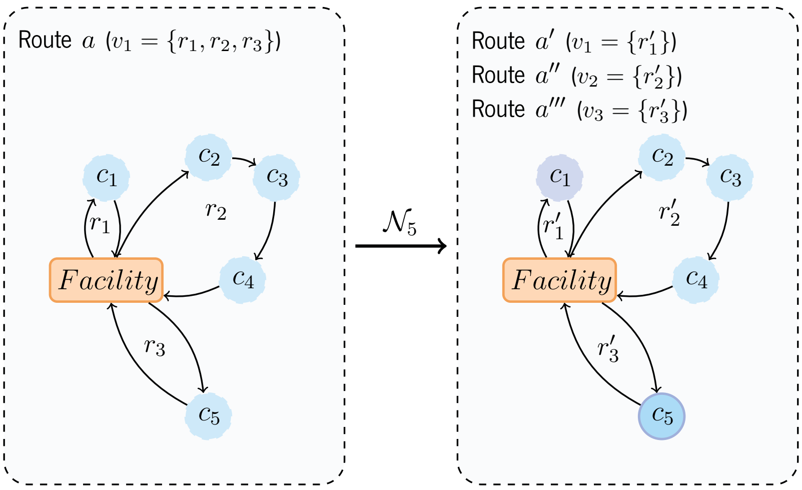

Figure 6.6: Example of a move in neighborhood $\mathcal{N}_5$

guarantee the distribution in the same period of the vehicle associated with the reassigned arc.

As shown in Figure 6.7 the arc r_2 is divided into two different round trip arcs r'_2 and r'_3 that are performed by vehicle v_2 and v_3, respectively. Vehicles v_2 and v_3 fulfill the demand associated to only one client per arc dividing the demand in route a'' for client c_1 and a''' for client c_2. The arc r_1 is not modified.

Neighborhood $\mathcal{N}_7$ adjusts the start time of an arc during a period of time, being called "arc time adjust". An arc is selected and adjusted according to its minimum and maximum starting times. The time u of the selected arc is adjusted in order to maximize the available time between other existing arcs in the same route, when the arc is valid. When the arc has an associated penalty, this adjustment is made taking into account the decrease of that penalty. The time v of the arc is updated according to the duration of the arc.

Figure 6.8 shows two different arcs: $x_{10,45,r_1}$ and $x_{60,100,r_1}$. For the sake of simplicity the decision variable x_{uvr} denotes an arc. Arc $x_{10,45,r_1}$ is valid and instant u can be adjusted between the time interval $[0, 20]$. In order to maximize the interval, time u is set to 0, creating a new arc $x_{0,35,r_1}$.

The neighborhood $\mathcal{N}_8$ is named "anticipate client", since it anticipates the demand of a partic-

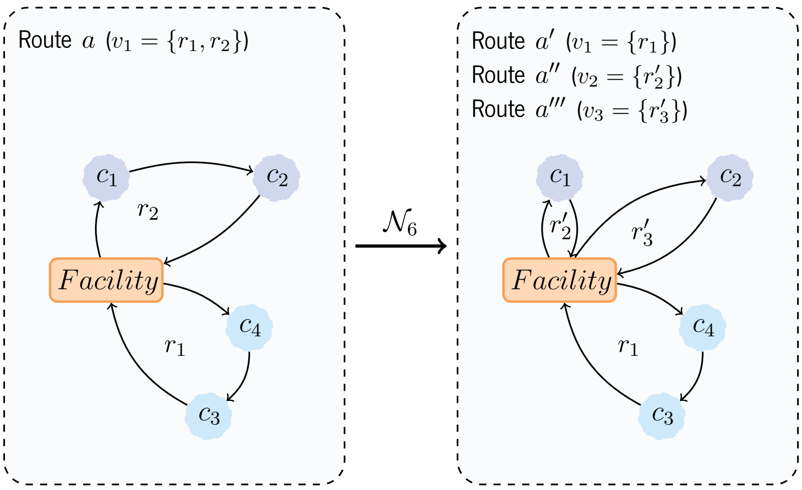

Figure 6.7: Example of a move in neighborhood $\mathcal{N}_6$

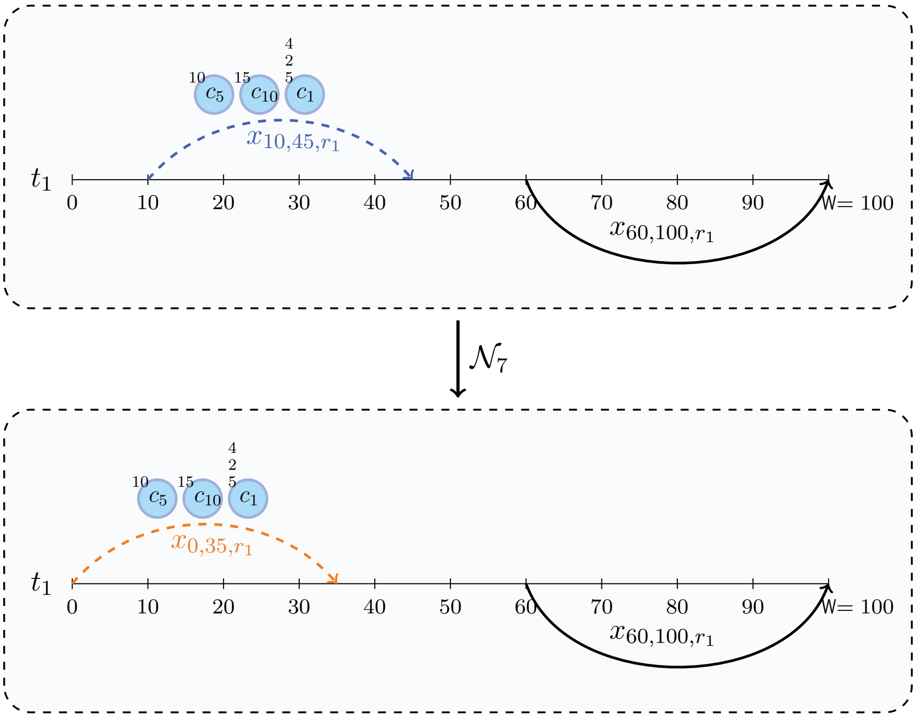

Figure 6.8: Example of a move in neighborhood $\mathcal{N}_7$

ular customer, ensuring the vehicle capacity is not exceeded. This load availability is guaranteed by forcing a particular customer to be removed from the arc and then inserted into another arc which satisfies the same period. This situation occurs only when the vehicle capacity is exceeded. Otherwise, the customer demand is only anticipated.

Figure 6.9 shows the particular case where the demand of customer c_1 is anticipated and the demand of customer c_2 must be removed from the arc in order to ensure that the vehicle capacity is not exceeded. Route a fulfills clients c_5 and c_1 only for period t_2. Route b satisfies the demand of clients c_3, c_1, c_6 and c_4 associated to period t_1 and route c distributes only the demand for the client c_2 for period t_1. In order to anticipate customer c_1 demand and not to exceed the capacity of the vehicle, it is necessary to remove client c_6 from route b and insert it into route c, originating route c'. Route a' serves only one client (c_5) and route b' fulfills the client c_1 for period t_1 and t_2; and the clients c_3 and c_4 for period t_1.

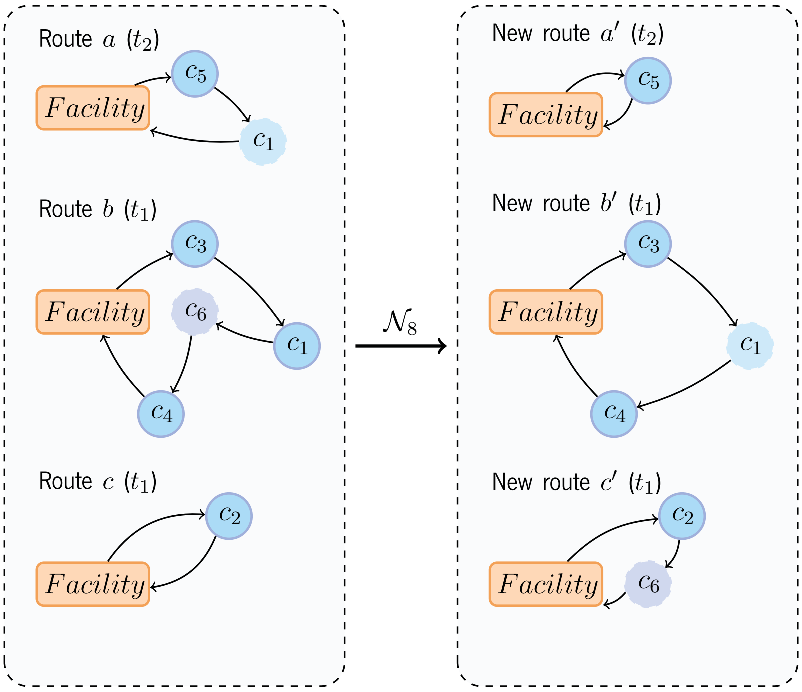

Figure 6.9: Example of a move in neighborhood $\mathcal{N}_8$

6.2.2 Evaluation function

At each iteration of the VNS process a set of solutions is determined, which depends on the explored neighborhood. When a mathematical model assists in the decision making of the VNS process, the set of solutions is changed, and a different solution space is defined. The search space includes a set of solutions that has different costs associated, thus it becomes important to assess the quality of the solution. To evaluate a solution, it is essential to clearly define an evaluation function. The relevant parameters and costs associated to the evaluation function are presented below:

Parameters

$$S = \text{solution}$$
$$v(S) = \text{number of vehicles of the solution } S$$

$inv^P(t) =$ quantity of inventory at the facility at period $t, \forall t \in t^P(S)$

$t^D(S) =$ set of distribution periods of the solution S

$t^P(S) =$ set of production periods of the solution S

$r(t) =$ set of single-trips of period t, $\forall t \in t^D(S)$

$h^P =$ unitary holding cost at the facility

$f_t =$ setup cost of manufacturing period $t, \forall t \in t^P(S)$

$C_v =$ fixed cost that represents the use of a vehicle v

$C_H(r,t) =$ holding cost associated to the route r and period t , $\forall r \in r(t)$ and $\forall t \in t^D(S)$

$R(r,t) =$ cost of a route $r, \forall r \in r(t)$ and $\forall t \in t^D(S)$

$W =$ length of the planning horizon

$Q =$ capacity associated to the vehicle

$$pen_W(r,t) = \begin{cases} 1 & \text{if route } r \text{ exceeds the workday length } W \text{ in period } t, \forall r \in \text{ and } t \in t^D(S), \\ 0 & \text{otherwise} \end{cases}$$

$$pen_Q(r,t) = \begin{cases} 1 & \text{if route } r \text{ exceeds the capacity of vehicle } Q \text{ in period } t, \forall r \in \text{ and } t \in t^D(S), \\ 0 & \text{otherwise} \end{cases}$$

$pen_{TW}(r,t) =$number of time window violations $\forall r \in$ and $t \in t^D(S)$,

$C_{pen} =$penalty cost

Evaluation function

$$f(S) = Cost_r^D + Cost_v^D + Cost_p^P + Cost_{inv}^P + Cost_{pen}^D \ , \tag{6.1}$$

where

$$Cost_r^D = \sum_{t \in T^P(S)} \sum_{r \in r(t)} R(r,t) + C_H(r,t) \tag{6.2}$$

$$Cost_v^D = \sum_{v=1}^{v(S)} C_v \tag{6.3}$$

$$Cost_p^P = \sum_{p \in t^P(S)} f_t \tag{6.4}$$

$$Cost_{inv}^P = h^P \sum_{t \in t^P(S)} inv^P(t) \tag{6.5}$$

$$Cost_{pen}^D = \sum_{t \in T^P(S)} \sum_{r \in r(t)} C_{pen} pen_W(r,t) + C_{pen} pen_Q(r,t) + C_{pen} pen_{TW}(r,t) \tag{6.6}$$

There are three different types of unfeasible solutions during the exploration of the search space. One permits to exceed the vehicle capacity. Another allows for the violation of the customer time window. The last allows to exceed the workday length. The aforementioned violations are considered in the evaluation function through the introduction of penalties. The solution S is evaluated for its fixed and variable cost, considering the penalties associated to violated constraints.

6.2.3 A two-phase model-based variable neighborhood search algorithm

When problems present a high computational complexity, it is often necessary to decompose them and use different methodologies to manage the exchange of information between the subproblems.

During this section, a local search algorithm based on a mathematical model is described, called a two-phase model-based variable neighborhood search algorithm (2MVNS). This method is a meta-heuristic which explores, in a systematic way, a set of neighborhood structures in order to find integer solutions of good quality.

The 2MVNS algorithm, presented in Figure 6.10, starts with a valid initial solution, and a variable local search process is applied, through the successive exploration of a set of neighborhoods. This phase is called Model-Based Variable Neighborhood Descent (MVND) and is referred to as Phase I. During this phase, the 2MVNS algorithm finds several distribution and routing decisions that are evaluated through the mathematical model. Through this approach, it is possible to find good quality solutions, considering all the constraints associated to the MPIDRPTW, not neglecting the inventory and production decisions at the client and at the facility. When a better valid solution is found, the incumbent solution is updated. In order to avoid local optima, the solution is submitted to a disturbance phase, creating significant changes in the current solution. This second phase (Phase II) is called Shaking Phase and aims to strengthen the robustness of the

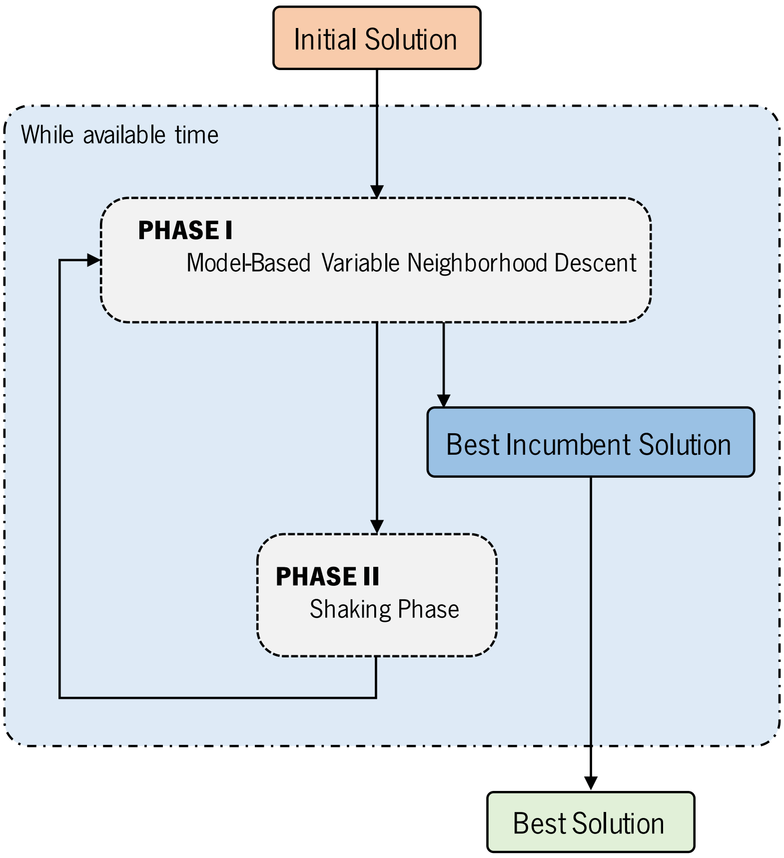

Figure 6.10: The two-phase model-based variable neighborhood search approach

search process. The 2MVNS algorithm provides an iterative search during the available time. At the end of the approach, the current best incumbent solution is considered the final best solution.

The mathematical model that exchanges information with the VND algorithm is described in detail in Section 5.2. Each step of the MVND algorithm considers routing and distribution decisions, creating new possible routes to satisfy the clients demand. That information, in each step, is used by the mathematical model to improve production and inventory decisions at the facility and at the clients.

The 2MVNS algorithm is presented in Algorithm 6.1. The algorithm explores a set of neighborhoods, considered as input, and starts with an initial solution. This solution is created through a greedy randomized heuristic, which randomly serves clients, assigning customers to routes according to the capacity constraints of the workday and the vehicles.

The algorithm tries to improve the current solution through a model-based variable neighborhood descent. The MVND uses neighborhoods $\mathcal{N}_3, \mathcal{N}_2, \mathcal{N}_5, \mathcal{N}_6, \mathcal{N}_1, \mathcal{N}_4$ and $\mathcal{N}_0$ in a sequential way. After MVND, when a better solution is found, the solution is saved, replacing the best current solution S^*. The MVND explores a set of neighborhoods in a sequential way. For all neighborhood, there is a defined number of neighbors ($n_{Neighbors}$) per client to explore, meaning that the complete exploration of a neighborhood is not made. A permutation in the neighborhood is randomly defined for each client. When a first improvement is identified in a given neighborhood, this sequence of neighborhoods is explored from the beginning until a further improvement is found. When no improvement is found, the iterative exploration process of Phase I is terminated.

The shaking phase is performed after the MVND process, allowing for the diversification of the search, since this phase may strongly perturb the objective value obtained in the search space. This perturbation occurs due to the exploration of neighborhoods $\mathcal{N}_3$, $\mathcal{N}_5$, $\mathcal{N}_6$, $\mathcal{N}_1$. Neighborhoods $\mathcal{N}_5$ and $\mathcal{N}_6$ force the use of new vehicles while, on the other hand, $\mathcal{N}_1$ forces an anticipated order to be served in a later period. Neighborhood $\mathcal{N}_3$ is used in order to reduce the perturbation generated. A neighborhood is selected according to the probability $P_{\mathcal{N}_{i\in\{3,5,6,0\}}}$ and

Algorithm 6.1: The two-phase model-based variable neighborhood search algorithm

Input: Set of neighborhood structures $\mathcal{N} = \{\mathcal{N}_{i\in\{0,\ldots,8\}}\}, P_{\mathcal{N}_{i\in\{1,3,5,6\}}}, k_{max}, n_{Neighbors}$

Initialization: find an initial solution S with a greedy randomized heuristic;
$S^* \leftarrow S$;
$S' \leftarrow S$;
repeat
  $k \leftarrow 1$;
  while $k \leq k_{max}$ **do**
    $S'' \leftarrow$ MVND($S', \mathcal{N}_{i\in\{3,2,5,6,1,4,0\}}, n_{Neighbors}$) ;
    $S' \leftarrow$ Shaking($S, \mathcal{N}_{i\in\{3,5,6,1\}}, P_{\mathcal{N}_{i\in\{1,3,5,6\}}}, k$) ;
    if $f(S'') < f(S^*)$ *and* S'' *is feasible* **then**
      $S^* \leftarrow S''$;
until *a termination condition is met*;
return S^*

k represents the number of consecutive moves used to perturb the current solution. When the termination condition is reached, the best current solution is returned.

It is important to highlight that the 2MVNS approach is concerned with the quality and validity of the solution in each iteration, since this search must find a good and valid solution for the MPIDRPTW during the search process.

6.2.4 A three-phase model-based variable neighborhood search algorithm

The three-phase model-based variable neighborhood search (3MVNS) heuristic decomposes the MPIDRPTW in three distinct phases. Once again, the VNS is the local search method selected to determine arcs that have the potential to improve the global solution of the problem, exploring a set of neighborhoods in a systematic approach. The VNS finds routes, taking into account distribution and routing decisions, relaxing factory-related decisions. The production decisions are then made through the exact solution of the integer programming model specified in detail in Section 5.2, which uses the arcs generated by the VNS to decide the production periods, managing the inventory stored at the factory.

In Figure 6.11 a global scheme demonstrating how the 3MVNS algorithm works is depicted. The 3MVNS model-based starts with an initial solution and tries to find better quality arcs over a finite number of iterations. Each iteration is performed in two distinct phases: Model-based Variable Neighborhood Descent and Shaking Phase. Phase I explores the neighborhoods, looking for good local optima. During this phase, the generated arcs with potential to improve the global solution are memorized in a global list of arcs. Phase II creates higher perturbations in the solution, trying to escape from local optima. Lastly, during Phase III, the list of all generated arcs is evaluated to find the best possible solution within a pre-established time limit.

Algorithm 6.2 presents the 3MVNS heuristic, where a pre-established set of neighborhoods is sequentially explored. The algorithm starts with a valid solution obtained through a greedy randomized heuristic. The model-based variable neighborhood descent tries to find good quality

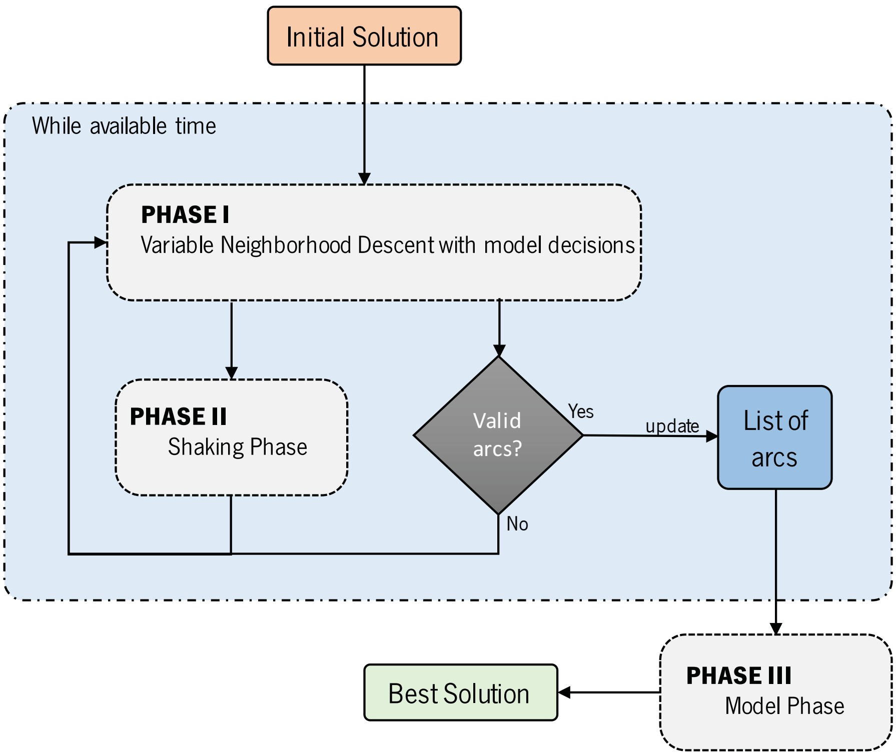

Figure 6.11: The three-phase model-based variable neighborhood search approach

neighbor solutions, exploring the solution space defined through neighborhoods $\mathcal{N}_3$, $\mathcal{N}_2$, $\mathcal{N}_5$, $\mathcal{N}_6$, $\mathcal{N}_1$, $\mathcal{N}_4$ and $\mathcal{N}_0$. During this phase, the 3MVNS tries to adjust routing and distribution decisions and the arc flow model evaluates the global solution, taking into account production and inventory decisions. The MVND does not make the exhaustive exploration of the solution space for each neighborhood. The exploration of a neighborhood is controlled by the parameter $n_{Neighbors}$ which defines the maximum number of random permutations allowed per client for each different neighborhood. The exploration of the solution space of a neighborhood stops when a first improvement is identified. The set of neighborhoods initially defined is explored from the beginning until a new improvement is reached. The iterative exploration process ends when no

Algorithm 6.2: A three-phase model-based variable neighborhood search algorithm

Input: Set of neighborhood structures
$\mathcal{N} = \{\mathcal{N}_{i\in\{0,\ldots,8\}}\}, P_{\mathcal{N}_{i\in\{1,3,5,6\}}}, k_{max}, t_{limit}, n_{Neighbors}$

Initialization: find an initial solution S with a greedy randomized heuristic;
$S^* \leftarrow S$;
$S' \leftarrow S$;
$A \leftarrow get\_arcs(S)$;
repeat
  $k \leftarrow 1$;
  while $k \leq k_{max}$ **do**
    $S'' \leftarrow$ MVND$(S', \mathcal{N}_{i\in\{3,2,5,6,1,4,0\}}, n_{Neighbors})$;
    $S' \leftarrow$ Shaking$(S, \mathcal{N}_{i\in\{3,5,6,0\}}, P_{\mathcal{N}_{i\in\{1,3,5,6\}}}, k)$;
    $new\_A \leftarrow get\_arcs(S'') \cup get\_arcs(S')$;
    $a \leftarrow 0$;
    while $new\_A \neq \emptyset$ **do**
      if $new\_A(a)$ *is valid* **then**
        $A \leftarrow A \cup new\_A(a)$;
        $new\_A \leftarrow new\_A \backslash \{new\_A(a)\}$;
    if $f(S'') < f(S^*)$ *and* S'' *is feasible* **then**
      $S^* \leftarrow S''$;
until *a termination condition is met*;
$S^* \leftarrow model(A, t_{limit})$;
return S^*

more improvements are found.

The diversification of the search is allowed through the shaking phase, creating significant perturbation in the current solution. The application of neighborhoods $\mathcal{N}_5$, $\mathcal{N}_6$, $\mathcal{N}_1$ strongly perturbs the solution, since neighborhoods $\mathcal{N}_5$ and $\mathcal{N}_6$ force the use of new vehicles, and neighborhood $\mathcal{N}_1$ forces an anticipated order to be fulfilled in a posterior period. Neighborhood $\mathcal{N}_3$ reduces the perturbation generated. The probability $P_{\mathcal{N}_{i\in\{3,5,6,0\}}}$ selects a neighborhood and the parameter k identifies the consecutive moves used to perturb the current solution during the shaking phase.

In each iteration, all the valid arcs, obtained through the exploration of the different neighborhoods, are stored in a list of arcs pre-initialized with the arcs belonging to the initial solution. When the termination condition is reached, the list of arcs generated during the MVND phase is evaluated by the mathematical model, returning a valid solution within the time limit t_{limit}.

It is important to note that 3MVNS is not so concerned with the quality of the solution found in each iteration, but with the validity and diversity of the arcs generated in Phase I. Phase III finds a solution through the mathematical model described in Section 5.2. The existence of a valid solution is guaranteed, since the initial solution represents a valid solution and all the arcs that are part of the solution are added to the list of arcs processed by the model. The arc flow model can also make some adjustments to the routing and distribution decisions, since it has access to all the arcs, being able to evaluate them from a global perspective.

6.3 Computational results

In order to evaluate the proposed heuristic, benchmark instances of literature, adapted from Bard and Nananukul in [46], are used. These instances are divided into 5 sets. The sets differ in terms of the location of the clients and the value of the parameters concerning production, inventory and distribution decisions. Each set has 20 instances, where each instance varies the number of clients (10, 20, 30, 40 and 50) and the number of periods (2, 4, 6 and 8) in the planning horizon.

As in the previous chapter, the Bard and Nananukul [46] instances were adapted to fit the specificities of the MPIDRPTW. The multi-trip variant must consider a fixed vehicle costs and a smaller vehicle capacity. These parameters were introduced in the definition of the instances, being set to 10000 ($C_v = 10000$) and 500 ($Q = 500$), respectively, for all instances. Client time windows interval were also introduced with randomly generated with values between 10% and 70% of the workday value. The workday value was set to 500 ($W = 500$). The coordinates of the facility are in the geographical point $(0, 0)$ and the euclidian distance between the facility and the clients was rounded to the nearest smaller integer. These were the only adaptations performed in the Bard and Nananukul [46] instances. The facility unit holding cost is 1 ($h^P = 1$) and the clients holding cost is 0 for all periods ($C_{h_i} = 0$). The facility production capacity (C) and the production setup cost (f_t) vary with the different sets according to Table 5.2 presented in the previous chapter. In the following subsections, tables with computational results are presented for the different algorithms. The common column notation is presented below:

$Inst$: instance name Disti_t where i is the number of clients and t the number of periods

Rep_i : repetition i of the heuristic

ub : value of the best known upper bound

S_{AVG} : average value of the solution repetition

S^* : value of the best known solution

S^- : value of the worst known solution

$\#C^{Arc}_{AVG}$: average number of clients per arc of all generated arcs

$\#C^{Arc}_{MAX}$: maximum number of clients per arc of all generated arcs

$\#L^{Arc}_{AVG}$: average number of loads per arc of all generated arcs, considering client demand anticipation

$TW_{AVG}(\%)$: average length of client time windows for all clients

T_{2MVNS} : time to execute the 2MVNS algorithm

T_{3MVNS} : time to execute the 3MVNS algorithm

$T_{3MVNS_{P_{I/II}}}$: time to execute phases I and II with 3MVNS algorithm

$T_{3MVN_{P_{III}}}$: time to execute phase II with 3MVNS algorithm

T_{total} : total time

$\#Arcs$: total number of arcs generated for all the periods

$\#A$: number of used arcs

$\#F$: number of used vehicles

$\#it$: number of iterations

E : exact model (Section 5.4.1)

L : exact model with arc limitation (Section 5.4.2)

The computational tests were performed on a computer with a processor Intel Xeon CPU ES-1620 v3 with 3.5GHz and 64GB of RAM; the optimization subroutines were executed on CPLEX 12.6.

6.3.1 A two-phase model-based variable neighborhood search algorithm

All adapted instances from Bard and Nananukul [46] were used to evaluate the 2MVNS algorithm and 5 different repetitions of the heuristic were performed for each instance of all sets. The 2MVNS algorithm is limited to a CPU time of 300 seconds for each of the 5 runs. The k_{max} parameter was set to 40% of the number of arcs initially generated by the greedy heuristic. The maximum number of random permutations allowed for each client in each neighborhood was set to $n_{Neighbors} = 5$. To adapt the penalty values to each instance, their values were set to 20% of the initial objective value. The probability of a move in a given neighbor, in the shaking phase, is defined by $P_{\mathcal{N}_3} = 0.1$, $P_{\mathcal{N}_5} = 0.3$, $P_{\mathcal{N}_6} = 0.3$, and $P_{\mathcal{N}_1} = 0.3$.

Tables 6.1 to 6.5 show the best solution value, the worst solution value and the average solution value of the different runs. The results show information regarding the best solution from the 5 repetitions (from column $Inst$ to column #it), whereas the last 7 columns present the worst, average and the objective value for each repetition.

Table 6.1: Results of two-phase model-based variable neighborhood search - Set 1

Inst	S^*	T_{2MVNS}	#Arcs	$\#C^{Arc}_{AVG}$	$\#C^{Arc}_{MAX}$	$\#L^{Arc}_{AVG}$	TW_{AVG} (%)	#A	#F	$\#it$	S^-	S_{AVG}	Rep 1	2	3	4	5
Dist10_2	131101	300	10	1,40	2	2,40	39,44	10	5	4390	131101	131101,00	131101	131101	131101	131101	131101
Dist10_4	248775	300	19	1,47	2	4,11	30,56	19	13	2205	259603	253276,20	249821	259603	258186	249996	248775
Dist10_6	325839	300	29	1,45	3	5,52	35,58	29	16	900	344015	333091,80	335033	325839	326207	334365	344015
Dist10_8	459903	300	39	1,41	3	6,59	43,24	39	24	347	473039	467294,40	459903	461233	473039	472319	469978
Dist20_2	145259	300	14	2,07	3	3,50	42,01	14	6	1383	146470	145871,60	145259	146470	145900	146399	145330
Dist20_4	241458	300	28	1,75	4	4,93	43,27	28	11	351	253949	250189,20	241458	253949	251250	251188	253101
Dist20_6	381165	301	42	1,88	4	6,98	39,22	42	19	138	396099	390127,00	381165	391235	387692	396099	394444
Dist20_8	531699	300	60	1,73	4	8,28	41,48	60	26	64	553760	541809,20	537593	531699	553760	541156	544838
Dist30_2	238550	300	22	1,91	2	3,27	37,23	22	12	522	251050	242315,40	239564	239931	238550	251050	242482
Dist30_4	411180	301	44	1,95	4	5,23	38,43	44	21	122	429025	421762,40	424203	411180	429025	425288	419116
Dist30_6	632316	304	66	2,12	3	7,76	43,25	66	32	43	666886	648083,20	635732	666886	648797	632316	656685
Dist30_8	943146	315	92	1,98	3	9,07	34,22	92	52	23	977453	956553,00	977453	968648	950046	943146	943472
Dist40_2	275884	300	33	2,18	4	3,39	44,66	33	13	221	295793	284165,00	286117	286913	295793	275884	276118
Dist40_4	613363	300	72	1,88	3	4,79	41,89	72	33	48	635049	622295,20	613363	615131	635049	620980	626953
Dist40_6	924747	310	108	1,95	4	6,94	42,32	108	49	18	967095	946636,40	935573	965476	967095	924747	940291
Dist40_8	1264510	305	147	1,97	3	8,98	42,94	147	68	11	1338180	1304324,00	1305580	1314440	1264510	1338180	1298910
Dist50_2	369552	300	46	1,91	3	3,00	43,24	46	19	112	388832	381274,80	380312	379336	388832	369552	388342
Dist50_4	802605	323	93	1,99	3	5,06	41,27	93	44	23	821473	812533,80	819148	821473	808786	802605	810657
Dist50_6	1426070	314	144	1,92	3	6,77	34,64	144	84	10	1466260	1446282,00	1446300	1466260	1449640	1443140	1426070
Dist50_8	1693710	311	183	2,07	3	9,32	42,43	183	95	4	1725720	1705040,00	1701370	1725720	1699420	1704980	1693710

Table 6.2: Results of two-phase model-based variable neighborhood search - Set 2

Inst	S^*	T_{2MVNS}	#Arcs	$\#C_{AVG}^{Arc}$	$\#C_{MAX}^{Arc}$	$\#L_{AVG}^{Arc}$	TW_{AVG} (%)	#A	#F	$\#it$	S^-	S_{AVG}	Rep 1	2	3	4	5
Dist10_2	181044	300	14	1,36	3	2,07	35,06	14	7	5667	181044	181044,00	181044	181044	181044	181044	181044
Dist10_4	282318	300	27	1,30	3	3,33	43,12	27	12	1788	292421	286522,60	283059	282318	282437	292421	292378
Dist10_6	388306	300	44	1,23	4	4,41	45,48	44	17	609	398038	393817,00	388408	397174	398038	388306	397159
Dist10_8	498989	301	55	1,22	5	5,44	40,68	55	22	341	511412	507935,60	511412	508526	498989	509988	510763
Dist20_2	287691	300	28	1,21	3	1,93	34,80	28	13	1263	298890	292279,40	288010	298890	297691	289115	287691
Dist20_4	502445	300	55	1,25	3	3,16	41,69	55	24	262	512235	507155,80	503021	510919	502445	507159	512235
Dist20_6	738076	303	89	1,27	4	4,46	35,39	89	37	100	757349	751363,80	756483	748754	757349	738076	756157
Dist20_8	910038	310	103	1,37	5	6,18	44,51	103	44	56	933537	926604,80	927241	910038	930640	931568	933537
Dist30_2	346212	300	38	1,47	3	2,26	46,18	38	16	471	359677	351702,40	356387	346212	348155	348081	359677
Dist30_4	653887	303	73	1,47	3	3,73	37,31	73	34	98	678646	664151,20	678646	653887	654326	665290	668607
Dist30_6	1104700	306	122	1,34	3	4,76	35,71	122	60	33	1126530	1110050,00	1106740	1107300	1104980	1126530	1104700
Dist30_8	1570000	304	164	1,35	3	6,15	40,07	164	84	22	1615800	1583504,00	1570000	1578220	1576080	1577420	1615800
Dist40_2	345262	301	39	1,77	4	2,79	45,34	39	16	216	357151	354242,20	355950	355929	345262	356919	357151
Dist40_4	831681	300	85	1,78	3	4,46	38,04	85	45	47	853172	841118,00	832531	845841	831681	853172	842365
Dist40_6	1296290	312	142	1,56	3	5,53	42,31	142	70	15	1328840	1310642,00	1299720	1311420	1316940	1296290	1328840
Dist40_8	1754100	326	187	1,59	3	7,18	36,26	187	97	12	1780400	1763928,00	1769110	1780400	1755800	1754100	1760230
Dist50_2	473357	302	50	1,82	3	2,82	42,06	50	24	122	487221	481543,20	485340	487221	484524	473357	477274
Dist50_4	806941	306	93	1,88	4	4,78	38,92	93	43	24	863510	846543,20	806941	861013	861694	863510	839558
Dist50_6	1529250	303	163	1,76	3	6,22	40,28	163	82	12	1577870	1563524,00	1577870	1558320	1529250	1575060	1577120
Dist50_8	2234920	314	224	1,69	3	7,75	36,73	224	129	2	2278110	2256114,00	2234920	2278110	2270960	2235760	2260820

Table 6.3: Results of two-phase model-based variable neighborhood search - Set 3

Inst	S^*	T_{2MVNS}	#Arcs	$\#C^{Arc}_{AVG}$	$\#C^{Arc}_{MAX}$	$\#L^{Arc}_{AVG}$	TW_{AVG} (%)	#A	#F	$\#it$	S^-	S_{AVG}	Rep 1	2	3	4	5
Dist10_2	159265	300	16	1,13	4	1,75	38,74	16	5	6503	169265	161265,00	159265	169265	159265	159265	159265
Dist10_4	303724	300	30	1,17	3	2,97	42,88	30	13	1741	313724	306426,60	303724	313724	304641	304641	305403
Dist10_6	392808	300	42	1,17	5	4,24	38,88	42	18	657	403409	399123,60	392808	403409	393304	403231	402866
Dist10_8	506511	301	55	1,18	4	5,44	30,74	55	23	305	539345	528359,60	506511	528820	529560	539345	537562
Dist20_2	242291	300	25	1,44	3	2,24	45,22	25	10	1329	243909	242976,60	242751	242291	242323	243609	243909
Dist20_4	458069	300	51	1,33	4	3,37	39,94	51	22	270	470410	462882,00	470410	458069	459765	459410	466756
Dist20_6	882200	300	92	1,21	3	4,29	35,41	92	45	95	913001	894696,40	896710	889939	882200	913001	891632
Dist20_8	1090990	303	116	1,22	3	5,49	38,16	116	55	48	1124200	1110132,00	1090990	1104230	1124200	1121950	1109290
Dist30_2	367187	300	41	1,34	4	2,07	43,91	41	17	490	388175	378093,40	367187	378012	377956	388175	379137
Dist30_4	707448	301	80	1,41	4	3,59	40,39	80	36	83	747464	729830,60	747464	738580	707448	718943	736718
Dist30_6	1089880	314	118	1,44	3	5,05	42,17	118	57	34	1117910	1102976,00	1106660	1109800	1117910	1090630	1089880
Dist30_8	1374070	305	153	1,46	4	6,69	41,95	153	72	17	1409850	1384472,00	1378430	1374070	1375460	1384550	1409850
Dist40_2	403121	301	45	1,58	4	2,47	40,93	45	20	202	414501	406751,00	403121	406520	403752	414501	405861
Dist40_4	767269	309	89	1,55	4	3,97	38,71	89	40	43	801714	787165,00	779406	801714	767269	791629	795807
Dist40_6	1321510	300	143	1,56	4	5,55	35,78	143	71	21	1385020	1350386,00	1345270	1321510	1347580	1352550	1385020
Dist40_8	1713680	365	183	1,64	3	7,38	39,98	183	93	11	1753730	1732722,00	1722390	1751710	1753730	1713680	1722100
Dist50_2	473479	302	50	1,78	3	2,78	36,68	50	25	117	486456	478575,80	484728	474201	474015	473479	486456
Dist50_4	952126	307	103	1,81	3	4,62	40,79	103	52	21	974337	963317,60	967099	952126	965181	974337	957845
Dist50_6	1465720	304	164	1,73	4	6,15	37,88	164	80	11	1557240	1520416,00	1520340	1502430	1465720	1556350	1557240
Dist50_8	1959220	311	218	1,75	3	7,87	41,78	218	107	3	2011570	1985140,00	2011570	1992720	1968580	1993610	1959220

Table 6.4: Results of two-phase model-based variable neighborhood search - Set 4

Inst	S^*	T_{2MVNS}	#Arcs	$\#C^{Arc}_{AVG}$	$\#C^{Arc}_{MAX}$	$\#L^{Arc}_{AVG}$	TW_{AVG} (%)	#A	#F	$\#it$	S^-	S_{AVG}	Rep 1	2	3	4	5
Dist10_2	189192	300	11	1,27	3	2,18	40,94	11	4	7592	199015	191156,60	199015	189192	189192	189192	189192
Dist10_4	230868	300	17	1,53	4	4,00	52,80	17	6	1980	242152	235606,60	241542	230868	231532	242152	231939
Dist10_6	325270	300	26	1,42	3	5,50	37,86	26	11	955	356146	338413,80	343722	325270	356146	333717	333214
Dist10_8	394179	300	31	1,48	3	7,32	30,10	31	15	486	419643	407868,20	419643	394179	415686	414498	395335
Dist20_2	237278	300	16	1,88	3	3,13	37,75	16	7	1724	248875	243750,00	237539	247657	247401	248875	237278
Dist20_4	368600	300	31	1,71	3	4,90	39,44	31	15	404	375273	372569,40	368600	372356	374469	372149	375273
Dist20_6	577066	300	51	1,53	3	5,73	39,34	51	26	152	588668	580721,40	579774	577066	579027	588668	579072
Dist20_8	712334	301	67	1,63	3	7,42	39,21	67	32	63	750273	739807,80	737645	749285	712334	749502	750273
Dist30_2	314966	300	25	1,92	3	3,12	43,56	25	12	482	317483	316171,00	316210	316893	317483	314966	315303
Dist30_4	495029	301	47	2,11	4	5,43	40,40	47	22	91	512243	504636,20	503240	505592	495029	512243	507077
Dist30_6	840055	301	75	2,03	3	7,41	33,59	75	42	36	855141	848464,60	853239	855141	849060	844828	840055
Dist30_8	1021850	312	96	1,96	3	9,21	41,96	96	48	22	1079190	1051838,00	1021850	1079190	1074790	1046700	1036660
Dist40_2	372504	300	31	2,29	3	3,58	41,94	31	15	223	385167	379421,00	383629	372890	372504	385167	382915
Dist40_4	650844	302	65	2,02	3	5,28	41,33	65	31	43	676335	665030,40	650844	662112	676335	662719	673142
Dist40_6	975549	321	97	2,06	3	7,48	39,98	97	48	15	1022200	999428,40	1000930	1022200	995163	975549	1003300
Dist40_8	1312530	337	126	2,08	3	9,46	41,40	126	65	10	1375990	1348572,00	1345510	1370770	1375990	1338060	1312530
Dist50_2	407137	304	38	2,34	4	3,66	42,56	38	17	109	428248	421121,20	407137	425050	418255	426916	428248
Dist50_4	825556	313	85	2,12	3	5,36	37,47	85	41	21	859663	842474,40	845528	829096	859663	852529	825556
Dist50_6	1229370	344	118	2,25	3	8,06	36,81	118	65	11	1300420	1260054,00	1259030	1229370	1277060	1300420	1234390
Dist50_8	1613570	301	144	2,53	3	11,52	41,80	144	85	5	1646750	1626408,00	1625420	1646750	1620180	1626120	1613570

Table 6.5: Results of two-phase model-based variable neighborhood search - Set 5

Inst	S^*	T_{2MVNS}	#Arcs	$\#C^{Arc}_{AVG}$	$\#C^{Arc}_{MAX}$	$\#L^{Arc}_{AVG}$	TW_{AVG} (%)	#A	#F	$\#it$	S^-	S_{AVG}	Rep 1	2	3	4	5
Dist10_2	205395	300	9	1,89	2	3,00	48,54	9	5	4727	215271	207972,80	206250	205395	206250	215271	206698
Dist10_4	293166	300	19	1,42	2	3,89	37,26	19	10	1615	305015	301975,60	305015	303468	303824	304405	293166
Dist10_6	305312	300	26	1,19	3	4,69	33,96	26	10	1078	316510	312759,00	315349	316510	314211	312413	305312
Dist10_8	386042	300	35	1,20	3	5,97	46,76	35	14	324	403857	393718,00	392565	393787	386042	403857	392339
Dist20_2	236903	300	17	1,53	4	2,71	42,93	17	7	1546	239367	238460,20	238469	236903	239367	238856	238706
Dist20_4	263796	300	33	1,73	3	4,70	36,62	33	17	349	391720	363120,80	263796	382429	391720	390714	386945
Dist20_6	575718	300	49	1,69	3	6,10	40,84	49	26	143	603561	588269,60	576066	603561	575718	597129	588874
Dist20_8	729008	303	65	1,68	3	7,92	39,40	65	33	73	772229	746885,00	745052	739162	748974	772229	729008
Dist30_2	310974	300	25	2,00	5	3,20	39,01	25	12	490	323745	315285,60	323745	314287	310974	313738	313684
Dist30_4	521966	303	52	1,90	3	4,92	44,30	52	23	80	534578	526907,80	531735	534578	523643	522617	521966
Dist30_6	792288	300	76	1,80	3	6,71	42,65	76	37	41	879713	829161,80	792288	879713	816829	824898	832081
Dist30_8	1080310	302	100	2,04	3	9,34	45,53	100	52	18	1106790	1093576,00	1086870	1098720	1095190	1106790	1080310
Dist40_2	388067	301	33	2,27	3	3,48	40,58	33	16	197	400436	396529,20	400436	388067	397774	397396	398973
Dist40_4	696541	304	68	1,82	3	4,78	38,20	68	34	45	737324	711860,20	719748	708711	737324	696977	696541
Dist40_6	1017350	301	99	2,10	3	7,51	42,15	99	51	16	1029540	1024670,00	1017350	1028430	1023610	1029540	1024420
Dist40_8	1357200	304	123	2,28	3	10,20	40,42	123	69	10	1399010	1380372,00	1357200	1399010	1373370	1387590	1384690
Dist50_2	430410	300	40	2,15	3	3,40	40,58	40	19	125	443567	436584,40	442124	432812	434009	443567	430410
Dist50_4	795523	305	82	2,20	3	5,54	43,68	82	39	23	817244	807648,20	807966	806313	817244	795523	811195
Dist50_6	1210500	317	114	2,27	3	7,92	40,20	114	64	10	1239290	1223412,00	1239290	1228110	1210500	1211730	1227430
Dist50_8	1624110	325	148	2,49	3	11,14	38,58	148	85	6	1671230	1640370,00	1635580	1626190	1671230	1644740	1624110

6.3.2 A three-phase model-based variable neighborhood search algorithm

To evaluate the 3MVNS algorithm all adapted instances from Bard and Nananukul [46] were used. For each instance of all sets, 5 different repetitions of the heuristic were performed. The 3MVNS algorithm is limited by a CPU time of 300 seconds for each of the 5 runs and the CPLEX execution time was limited to 300 seconds. Parameter k_{max} was set to 40% of the number of arcs initially generated by the greedy heuristic. For each client in each neighborhood, the maximum number of random permutations allowed was set to $n_{Neighbors} = 5$. To set a penalty value that is better adapted for each instance, their values were set to 20% of the initial objective value. For the shaking phase the probability of a move in a given neighbor was defined by $P_{\mathcal{N}_3} = 0.1$, $P_{\mathcal{N}_5} = 0.3$, $P_{\mathcal{N}_6} = 0.3$, and $P_{\mathcal{N}_1} = 0.3$.

Tables 6.6 to 6.10 show the best solution value, the worst solution value and the average solution value of the different runs. The tables consider the results obtained regarding the best solution from the 5 repetitions (from column $Inst$ to column #it), whereas the last 7 columns present the worst, average and the objective value for each repetition.

Table 6.6: Results of three-phase model-based variable neighborhood search - Set 1

Inst	S^*	T_{3MVNS}	$T_{3MVNS_{P_{I/II}}}$	$T_{3MVNS_{P_{III}}}$	#Arcs	$\#C^{Arc}_{AVG}$	$\#C^{Arc}_{MAX}$	$\#L^{Arc}_{AVG}$	TW_{AVG} (%)	#A	#F	$\#it$	S^-	S_{AVG}	Rep				
															1	2	3	4	5
Dist10_2	131101	302	300	2	3542	1,79	4	3,10	39,44	10	5	4390	131101	131101,00	131101	131101	131101	131101	131101
Dist10_4	238963	301	300	1	3513	1,77	4	5,23	30,56	19	12	2205	239627	239192,00	238963	239053	238963	239627	239354
Dist10_6	324286	306	300	6	4280	1,79	4	7,46	35,58	30	16	900	325111	324675,00	324846	324286	324431	324701	325111
Dist10_8	445121	301	300	1	6521	1,53	4	7,59	43,24	37	23	347	447405	446595,00	445121	447006	447405	446816	446627
Dist20_2	134579	601	300	301	55893	2,55	6	4,06	42,01	14	5	1383	134849	134633,00	134849	134579	134579	134579	134579
Dist20_4	219063	601	300	301	48885	2,36	7	6,78	43,27	27	9	351	229636	222578,60	221316	229636	222349	220529	219063
Dist20_6	327079	602	301	301	33705	2,10	6	7,88	39,22	41	14	138	341069	335803,20	327079	337526	341069	335560	337782
Dist20_8	450024	601	300	301	31839	2,07	5	9,31	41,48	56	19	64	469021	458655,80	450519	450024	469021	460950	462765
Dist30_2	216744	601	300	301	34390	2,46	5	4,00	37,23	22	10	522	217384	217112,60	217370	217384	217250	216815	216744
Dist30_4	362024	602	301	301	39635	2,39	6	6,52	38,43	41	17	122	364941	363609,80	362024	363946	363808	364941	363330
Dist30_6	554609	605	304	301	36654	2,41	6	8,74	43,25	66	25	43	563286	559065,20	554609	563158	557292	556981	563286
Dist30_8	774689	615	315	300	20224	2,17	6	9,76	34,22	93	37	23	790410	785127,40	774689	788344	790410	785635	786559
Dist40_2	251707	602	300	302	69106	2,35	6	4,06	44,66	33	11	221	252507	252198,20	252268	251707	252507	252058	252451
Dist40_4	503883	486	300	186	42596	2,06	5	5,40	41,89	71	23	48	529209	514597,00	503883	506093	516625	517175	529209
Dist40_6	784418	611	310	301	30706	2,10	5	7,18	42,32	109	37	18	812234	797842,80	800017	795586	812234	796959	784418
Dist40_8	1090680	606	305	301	25798	2,17	6	9,16	42,94	144	54	11	1112860	1103784,00	1112320	1095710	1112860	1107350	1090680
Dist50_2	324380	602	300	302	54870	2,16	6	3,28	43,24	44	15	112	335002	327095,00	324380	325236	325129	325728	335002
Dist50_4	692256	624	323	301	37885	2,12	5	5,73	41,27	90	34	23	695491	693518,60	695491	693615	692256	692696	693535
Dist50_6	1211510	615	314	301	24654	2,27	6	7,58	34,64	143	64	10	1232310	1221624,00	1211510	1218290	1215470	1232310	1230540
Dist50_8	1443200	612	311	301	26817	2,30	5	11,00	42,43	182	72	4	1483430	1453856,00	1445340	1483430	1443200	1450210	1447100

Table 6.7: Results of three-phase model-based variable neighborhood search - Set 2

Inst	S^*	T_{3MVNS}	$T_{3MVNS_{P_{I/II}}}$	$T_{3MVNS_{P_{III}}}$	#Arcs	$\#C^{Arc}_{AVG}$	$\#C^{Arc}_{MAX}$	$\#L^{Arc}_{AVG}$	TW_{AVG} (%)	#A	#F	$\#it$	S^-	S_{AVG}	Rep 1	2	3	4	5
Dist10_2	172161	300	300	0	2142	1,43	3	2,22	35,06	14	6	5667	172161	172161,00	172161	172161	172161	172161	172161
Dist10_4	272563	307	300	7	4506	1,41	3	3,43	43,12	27	11	1788	272563	272563,00	272563	272563	272563	272563	272563
Dist10_6	377051	308	300	8	6282	1,31	3	4,17	45,48	43	16	609	377613	377163,40	377051	377051	377051	377613	377051
Dist10_8	469139	369	301	68	5241	1,34	4	7,13	40,68	56	19	341	469518	469323,40	469518	469518	469303	469139	469139
Dist20_2	278830	300	300	0	4343	1,47	3	2,35	34,80	28	12	1263	278830	278830,00	278830	278830	278830	278830	278830
Dist20_4	457218	437	300	137	7623	1,30	4	3,33	41,69	54	20	262	458229	457810,80	458229	457218	458160	458229	457218
Dist20_6	651148	399	303	96	9964	1,35	4	5,09	35,39	87	29	100	652803	651855,00	651148	651857	652803	651664	651803
Dist20_8	793800	611	310	301	15198	1,52	4	7,72	44,51	100	34	56	802883	799674,40	793800	802883	801635	800417	799637
Dist30_2	325000	601	300	301	19957	1,69	4	2,57	46,18	39	14	471	325000	325000,00	325000	325000	325000	325000	325000
Dist30_4	568533	603	303	300	17552	1,73	4	4,13	37,31	69	27	98	576544	574108,60	576544	574179	576091	568533	575196
Dist30_6	947228	342	306	36	12845	1,48	4	4,54	35,71	121	45	33	956513	952269,20	951702	956513	951045	954858	947228
Dist30_8	1360270	541	304	237	9935	1,58	4	8,12	40,07	158	66	22	1371730	1365598,00	1368650	1360270	1371730	1365250	1362090
Dist40_2	310261	603	301	302	55778	2,21	5	3,29	45,34	37	13	216	311405	310876,40	310845	310784	310261	311405	311087
Dist40_4	712232	400	300	100	25026	1,90	5	4,64	38,04	81	34	47	721672	715634,40	716709	714972	712232	712587	721672
Dist40_6	1112510	612	312	300	21044	1,90	5	5,90	42,31	136	54	15	1134290	1121412,00	1112510	1134290	1114870	1124690	1120700
Dist40_8	1480430	628	326	302	15398	1,78	4	8,25	36,26	180	71	12	1503180	1493036,00	1490730	1503180	1490910	1480430	1499930
Dist50_2	420434	603	302	301	42085	2,07	5	3,23	42,06	50	19	122	423024	421769,60	421995	420434	422062	421333	423024
Dist50_4	696954	607	306	301	36478	2,05	6	5,50	38,92	91	33	24	712199	708239,40	696954	712199	711261	710271	710512
Dist50_6	1320260	604	303	301	22227	2,08	5	6,47	40,28	159	64	12	1334140	1327894,00	1334140	1320260	1332300	1329630	1323140
Dist50_8	1892590	616	314	302	18327	2,07	6	7,93	36,73	223	95	2	1914670	1909438,00	1892590	1913260	1913400	1913270	1914670

Table 6.8: Results of three-phase model-based variable neighborhood search - Set 3

Inst	S^*	T_{3MVNS}	$T_{3MVNS_{P_{I/II}}}$	$T_{3MVNS_{P_{III}}}$	#Arcs	$\#C^{Arc}_{AVG}$	$\#C^{Arc}_{MAX}$	$\#L^{Arc}_{AVG}$	TW_{AVG} (%)	#A	#F	$\#it$	S^-	S_{AVG}	Rep 1	2	3	4	5
Dist10_2	159265	300	300	0	2396	1,35	2	2,19	38,74	16	5	6503	159265	159265,00	159265	159265	159265	159265	159265
Dist10_4	295403	300	300	0	2228	1,22	3	3,15	42,88	31	12	1741	295403	295403,00	295403	295403	295403	295403	295403
Dist10_6	383228	306	300	6	4585	1,24	3	4,90	38,88	42	16	657	383228	383228,00	383228	383228	383228	383228	383228
Dist10_8	495255	307	301	6	3819	1,37	4	7,35	30,74	53	22	305	496511	495769,60	495255	495621	496511	496206	495255
Dist20_2	231778	601	300	301	16459	1,88	4	2,62	45,22	25	9	1329	231778	231778,00	231778	231778	231778	231778	231778
Dist20_4	414401	407	300	107	11637	1,55	4	3,63	39,94	50	18	270	415725	414875,40	414446	414401	414401	415404	415725
Dist20_6	809541	342	300	42	5141	1,28	3	4,66	35,41	90	39	95	816962	812541,00	809863	816962	816242	809541	810097
Dist20_8	965928	605	303	302	7966	1,53	4	8,58	38,16	111	44	48	970577	968379,80	965928	969835	970577	967743	967816
Dist30_2	346908	390	300	90	14452	1,75	4	2,74	43,91	41	15	490	347030	346932,40	346908	346908	346908	346908	347030
Dist30_4	626430	601	301	300	14563	1,54	4	4,34	40,39	77	29	83	627954	627147,20	627213	626430	626773	627366	627954
Dist30_6	926754	614	314	300	16497	1,52	4	5,07	42,17	115	42	34	936538	933889,80	935145	934574	936538	936438	926754
Dist30_8	1182520	612	305	307	13611	1,63	4	8,15	41,95	151	55	17	1193850	1188420,00	1190940	1193850	1189870	1184920	1182520
Dist40_2	358165	602	301	301	32887	1,99	5	2,91	40,93	43	16	202	358472	358326,00	358472	358348	358165	358435	358210
Dist40_4	661402	609	309	300	23248	1,90	5	4,80	38,71	84	31	43	676010	669292,80	676010	661402	669091	673490	666471
Dist40_6	1154510	600	300	300	15901	1,88	5	5,74	35,78	136	57	21	1169570	1160786,00	1158870	1163980	1154510	1157000	1169570
Dist40_8	1467730	665	365	300	19026	1,84	5	7,93	39,98	178	70	11	1485320	1478430,00	1477110	1485320	1482380	1467730	1479610
Dist50_2	411379	603	302	301	39872	2,33	5	3,51	36,68	50	19	117	412410	412087,60	412410	411379	411889	412391	412369
Dist50_4	811600	608	307	301	30226	2,06	4	5,20	40,79	101	39	21	816452	814471,00	813491	815456	816452	811600	815356
Dist50_6	1264760	605	304	301	23450	2,09	5	6,45	37,88	158	62	11	1289260	1277434,00	1284680	1272670	1275800	1289260	1264760
Dist50_8	1684030	612	311	301	24423	1,97	5	8,44	41,78	215	81	3	1707450	1695418,00	1700160	1707450	1684030	1695680	1689770

Table 6.9: Results of three-phase model-based variable neighborhood search - Set 4

Inst	S^*	T_{3MVNS}	$T_{3MVNS_{P_{I/II}}}$	$T_{3MVNS_{P_{III}}}$	#Arcs	$\#C^{Arc}_{AVG}$	$\#C^{Arc}_{MAX}$	$\#L^{Arc}_{AVG}$	TW_{AVG} (%)	#A	#F	$\#it$	S^-	S_{AVG}	Rep 1	2	3	4	5
Dist10_2	189192	303	300	3	4923	1,58	3	2,43	40,94	11	4	7592	189192	189192,00	189192	189192	189192	189192	189192
Dist10_4	229833	317	300	17	16353	1,77	4	4,76	52,80	16	6	1980	230255	229917,40	230255	229833	229833	229833	229833
Dist10_6	314548	315	300	15	6929	1,72	4	6,81	37,86	25	11	955	321225	319691,00	314548	321048	321225	320964	320670
Dist10_8	371640	317	300	17	4419	1,87	5	9,97	30,10	30	14	486	381136	375869,80	381136	380393	371640	373919	372261
Dist20_2	237142	376	300	76	21554	2,01	4	3,14	37,75	17	7	1724	237142	237142,00	237142	237142	237142	237142	237142
Dist20_4	327620	601	300	301	29077	2,18	5	6,02	39,44	31	12	404	337167	334057,00	334883	334564	337167	327620	336051
Dist20_6	503110	600	300	300	17736	1,75	5	6,10	39,34	49	20	152	507933	505510,60	505971	505079	507933	503110	505460
Dist20_8	623769	601	301	300	21998	2,02	6	10,54	39,21	64	24	63	632840	627402,60	623769	630650	625129	632840	624625
Dist30_2	284080	389	300	89	57024	2,15	5	3,22	43,56	25	9	482	293210	286328,20	285067	284642	293210	284080	284642
Dist30_4	425358	602	301	301	48520	2,21	5	5,51	40,40	47	17	91	435487	432134,20	431679	435487	434445	425358	433702
Dist30_6	711478	602	301	301	23887	2,09	5	7,92	33,59	72	32	36	724568	716702,40	713225	711478	721721	712520	724568
Dist30_8	900469	613	312	301	29153	2,14	5	10,99	41,96	98	38	22	917557	912187,20	914515	913561	917557	900469	914834
Dist40_2	337258	423	300	123	59409	2,47	6	4,07	41,94	31	12	223	340082	338032,40	337263	337258	337963	340082	337596
Dist40_4	559043	603	302	301	49542	2,09	5	5,39	41,33	64	24	43	571488	567135,80	568344	568546	571488	559043	568258
Dist40_6	836691	622	321	301	38002	2,24	6	8,52	39,98	93	37	15	854138	848026,20	853971	836691	854138	847274	848057
Dist40_8	1142670	638	337	301	30328	2,28	5	11,61	41,40	123	52	10	1198430	1173652,00	1190950	1142670	1198430	1161720	1174490
Dist50_2	370891	607	304	303	94556	2,38	5	3,94	42,56	38	14	109	371783	371390,00	371076	370891	371451	371749	371783
Dist50_4	708868	615	313	302	44820	2,20	5	5,23	37,47	83	31	21	722867	715284,40	721832	710157	712698	722867	708868
Dist50_6	1062730	645	344	301	34702	2,36	6	8,15	36,81	120	51	11	1099800	1078870,00	1066510	1080070	1085240	1099800	1062730
Dist50_8	1412190	603	301	302	31605	2,66	7	13,14	41,80	148	66	5	1443190	1425912,00	1416960	1443190	1430060	1412190	1427160

Table 6.10: Results of three-phase model-based variable neighborhood search - Set 5

Inst	S^*	T_{3MVNS}	$T_{3MVNS_{P_{I/II}}}$	$T_{3MVNS_{P_{III}}}$	#Arcs	$\#C^{Arc}_{AVG}$	$\#C^{Arc}_{MAX}$	$\#L^{Arc}_{AVG}$	TW_{AVG} (%)	#A	#F	$\#it$	S^-	S_{AVG}	Rep 1	2	3	4	5
Dist10_2	205395	300	300	0	5755	1,81	3	3,15	48,54	9	5	4727	205395	205395,00	205395	205395	205395	205395	205395
Dist10_4	293166	300	300	0	4205	1,65	3	4,84	37,26	19	10	1615	294128	293550,80	294128	293166	293166	294128	293166
Dist10_6	290025	319	300	19	7526	1,62	4	5,65	33,96	24	9	1078	301970	296021,80	301970	295016	293494	290025	299604
Dist10_8	357723	601	300	301	9440	1,74	5	8,04	46,76	31	12	324	360213	359010,60	360213	357723	358302	358848	359967
Dist20_2	226903	587	300	287	32459	2,13	4	3,68	42,93	17	6	1546	227619	227052,20	226918	226903	227619	226903	226918
Dist20_4	342838	600	300	300	26087	1,99	5	5,82	36,62	32	13	349	353516	350899,40	353358	342838	353129	353516	351656
Dist20_6	516482	600	300	300	20027	2,01	5	7,61	40,84	47	20	143	518562	517666,20	517625	518562	516482	518003	517659
Dist20_8	640730	603	303	300	17467	2,05	5	10,31	39,40	60	26	73	645613	644337,40	640730	644666	645125	645613	645553
Dist30_2	279782	372	300	72	48890	2,18	5	3,47	39,01	25	9	490	280190	279989,20	279793	280190	280190	279991	279782
Dist30_4	439229	604	303	301	51159	2,01	5	5,17	44,30	49	16	80	451064	448010,40	449061	451064	449747	450951	439229
Dist30_6	691003	601	300	301	29587	1,94	5	7,41	42,65	75	29	41	703207	696826,60	700333	698093	691497	691003	703207
Dist30_8	919971	603	302	301	29787	2,24	5	10,87	45,53	96	38	18	955663	938335,00	951662	929239	955663	935140	919971
Dist40_2	354014	602	301	301	62585	2,33	5	3,96	40,58	33	13	197	355967	354913,00	355407	354169	355967	354014	355008
Dist40_4	598317	605	304	301	37016	2,13	5	5,98	38,20	65	26	45	609722	603744,40	598317	603076	600119	609722	607488
Dist40_6	859789	602	301	301	37878	2,25	6	7,35	42,15	95	37	16	870848	865287,00	861827	867307	870848	859789	866664
Dist40_8	1158820	605	304	301	29622	2,41	5	11,79	40,42	126	50	10	1187150	1171146,00	1158820	1187150	1167480	1179190	1163090
Dist50_2	384219	602	300	302	72068	2,28	5	3,78	40,58	39	15	125	385300	384893,80	385223	384515	385212	385300	384219
Dist50_4	695088	607	305	302	59419	2,23	5	5,64	43,68	81	30	23	712833	701777,40	695974	708619	695088	696373	712833
Dist50_6	1042700	619	317	302	36548	2,40	5	8,90	40,20	115	49	10	1057810	1052208,00	1042700	1053770	1056320	1050440	1057810
Dist50_8	1423160	627	325	302	35501	2,65	5	13,47	38,58	151	65	6	1454150	1440050,00	1454150	1441180	1448970	1432790	1423160

6.3.3 Comparative discussion

In order to validate the two proposed matheuristics, a comparative analysis is performed, considering the solution values with the best lower bound of the network flow model ((5.1)-(5.19)), presented in Section 5.2.

For an exhaustive analysis, Tables 6.11 to 6.15 present values of the best value obtained in the exact model with and without any limitation in the arc generation as well as the best values obtained by the matheuristics. The number of generated arcs and the gap values are also compared according to the approaches that are under comparison. As an example, the column E - $2MVNS$ presents the gap values that the $2MVNS$ retrieves when comparing to the exact execution of the model. Whenever there is no solution for a given set, the entry '–' is used.

Table 6.11: Comparative analysis of the proposed methods - Set 1

Inst	ub				$\#A$				gap (%)				
	E	L	$2MVNS$	$3MVNS$	E	L	$2MVNS$	$3MVNS$	E - $2MVNS$	E - $3MVNS$	L - $2MVNS$	L - $3MVNS$	$2MVNS$ - $3MVNS$
Dist10_2	131101	131101	131101	131101	35676	9666	10	3542	0,00	0,00	0,00	0,00	0,00
Dist10_4	238963	240733	248775	238963	78601	29774	19	3513	3,94	0,00	3,23	-0,74	-4,11
Dist10_6	323088	324957	325839	324286	265417	96144	29	4280	0,84	0,37	0,27	-0,21	-0,48
Dist10_8	442963	442963	459903	445121	386959	117243	39	6521	3,68	0,48	3,68	0,48	-3,32
Dist20_2	134640	134735	145259	134579	6237937	151613	14	55893	7,31	-0,05	7,24	-0,12	-7,94
Dist20_4	-	216124	241458	219063	-	368116	28	48885	-	-	10,49	1,34	-10,22
Dist20_6	-	322936	381165	327079	-	556792	42	33705	-	-	15,28	1,27	-16,54
Dist20_8	-	437588	531699	450024	-	768125	60	31839	-	-	17,70	2,76	-18,15
Dist30_2	-	223932	238550	216744	-	161808	22	34390	-	-	6,13	-3,32	-10,06
Dist30_4	-	360682	411180	362024	-	393283	44	39635	-	-	12,28	0,37	-13,58
Dist30_6	-	540298	632316	554609	-	601194	66	36654	-	-	14,55	2,58	-14,01
Dist30_8	-	756757	943146	774689	-	776040	92	20224	-	-	19,76	2,31	-21,75
Dist40_2	-	263094	275884	251707	-	194796	33	69106	-	-	4,64	-4,52	-9,61
Dist40_4	-	498430	613363	503883	-	403238	72	42596	-	-	18,74	1,08	-21,73
Dist40_6	-	760747	924747	784418	-	627607	108	30706	-	-	17,73	3,02	-17,89
Dist40_8	-	1058650	1264510	1090680	-	873357	147	25798	-	-	16,28	2,94	-15,94
Dist50_2	-	334740	369552	324380	-	199539	46	54870	-	-	9,42	-3,19	-13,93
Dist50_4	-	676024	802605	692256	-	441379	93	37885	-	-	15,77	2,34	-15,94
Dist50_6	-	1169340	1426070	1211510	-	615546	144	24654	-	-	18,00	3,48	-17,71
Dist50_8	-	1376180	1693710	1443200	-	866250	183	26817	-	-	18,75	4,64	-17,36

Table 6.12: Comparative analysis of the proposed methods - Set 2

Inst	ub				#A				gap (%)				
	E	L	$2MVNS$	$3MVNS$	E	L	$2MVNS$	$3MVNS$	E - $2MVNS$	E - $3MVNS$	L - $2MVNS$	L - $3MVNS$	$2MVNS$ - $3MVNS$
Dist10_2	172161	172326	181044	172161	9102	1830	14	2142	4,91	0,00	4,82	-0,10	-5,16
Dist10_4	272563	282161	282318	272563	25067	4777	27	4506	3,46	0,00	0,06	-3,52	-3,58
Dist10_6	377051	379122	388306	377051	35031	5879	44	6282	2,90	0,00	2,37	-0,55	-2,99
Dist10_8	469139	469303	498989	469139	51075	11804	55	5241	5,98	0,00	5,95	-0,03	-6,36
Dist20_2	278830	288969	287691	278830	31893	9274	28	4343	3,08	0,00	-0,44	-3,64	-3,18
Dist20_4	457218	468541	502445	457218	86698	24499	55	7623	9,00	0,00	6,75	-2,48	-9,89
Dist20_6	651148	662933	738076	651148	104547	28489	89	9964	11,78	0,00	10,18	-1,81	-13,35
Dist20_8	794620	800132	910038	793800	773555	164885	103	15198	12,68	-0,10	12,08	-0,80	-14,64
Dist30_2	325000	325420	346212	325000	335511	85071	38	19957	6,13	0,00	6,01	-0,13	-6,53
Dist30_4	565230	576052	653887	568533	995404	269462	73	17552	13,56	0,58	11,90	-1,32	-15,01
Dist30_6	940558	946532	1104700	947228	884508	262435	122	12845	14,86	0,70	14,32	0,07	-16,62
Dist30_8	1347960	1360550	1570000	1360270	1620705	319037	164	9935	14,14	0,90	13,34	-0,02	-15,42
Dist40_2	-	322876	345262	310261	-	193603	39	55778	-	-	6,48	-4,07	-11,28
Dist40_4	-	723183	831681	712232	-	428683	85	25026	-	-	13,05	-1,54	-16,77
Dist40_6	-	1110870	1296290	1112510	-	416856	142	21044	-	-	14,30	0,15	-16,52
Dist40_8	-	1447650	1754100	1480430	-	595827	187	15398	-	-	17,47	2,21	-18,49
Dist50_2	-	434180	473357	420434	-	195433	50	42085	-	-	8,28	-3,27	-12,59
Dist50_4	-	689275	806941	696954	-	434424	93	36478	-	-	14,58	1,10	-15,78
Dist50_6	-	1314040	1529250	1320260	-	507427	163	22227	-	-	14,07	0,47	-15,83
Dist50_8	-	1860500	2234920	1892590	-	599142	224	18327	-	-	16,75	1,70	-18,09

Table 6.13: Comparative analysis of the proposed methods - Set 3

Inst	ub				#A				gap (%)				
	E	L	$2MVNS$	$3MVNS$	E	L	$2MVNS$	$3MVNS$	E - $2MVNS$	E - $3MVNS$	L - $2MVNS$	L - $3MVNS$	$2MVNS$ - $3MVNS$
Dist10_2	159265	169265	159265	159265	6272	1128	16	2396	0,00	0,00	-6,28	-6,28	0,00
Dist10_4	295403	304641	303724	295403	18293	3140	30	2228	2,74	0,00	-0,30	-3,13	-2,82
Dist10_6	383228	384846	392808	383228	29558	6781	42	4585	2,44	0,00	2,03	-0,42	-2,50
Dist10_8	495255	496056	506511	495255	58349	24261	55	3819	2,22	0,00	2,06	-0,16	-2,27
Dist20_2	231778	231883	242291	231778	126387	29326	25	16459	4,34	0,00	4,30	-0,05	-4,54
Dist20_4	414401	422390	458069	414401	161351	42163	51	11637	9,53	0,00	7,79	-1,93	-10,54
Dist20_6	809436	835711	882200	809541	68213	18972	92	5141	8,25	0,01	5,27	-3,23	-8,98
Dist20_8	958297	971500	1090990	965928	630682	101836	116	7966	12,16	0,79	10,95	-0,58	-12,95
Dist30_2	346908	356019	367187	346908	236719	55615	41	14452	5,52	0,00	3,04	-2,63	-5,85
Dist30_4	623887	627953	707448	626430	494888	162331	80	14563	11,81	0,41	11,24	-0,24	-12,93
Dist30_6	935402	933302	1089880	926754	1272022	342497	118	16497	14,17	-0,93	14,37	-0,71	-17,60
Dist30_8	1191260	1187290	1374070	1182520	2039617	443925	153	13611	13,30	-0,74	13,59	-0,40	-16,20
Dist40_2	358135	364662	403121	358165	8882464	207264	45	32887	11,16	0,01	9,54	-1,81	-12,55
Dist40_4	-	672640	767269	661402	-	396758	89	23248	-	-	12,33	-1,70	-16,01
Dist40_6	-	1155420	1321510	1154510	-	377745	143	15901	-	-	12,57	-0,08	-14,47
Dist40_8	-	1482520	1713680	1467730	-	528759	183	19026	-	-	13,49	-1,01	-16,76
Dist50_2	-	433007	473479	411379	-	209469	50	39872	-	-	8,55	-5,26	-15,10
Dist50_4	-	803880	952126	811600	-	446719	103	30226	-	-	15,57	0,95	-17,31
Dist50_6	-	1254230	1465720	1264760	-	481560	164	23450	-	-	14,43	0,83	-15,89
Dist50_8	-	1652740	1959220	1684030	-	639381	218	24423	-	-	15,64	1,86	-16,34

Table 6.14: Comparative analysis of the proposed methods - Set 4

Inst	ub				#A				gap (%)				
	E	L	$2MVNS$	$3MVNS$	E	L	$2MVNS$	$3MVNS$	E - $2MVNS$	E - $3MVNS$	L - $2MVNS$	L - $3MVNS$	$2MVNS$ - $3MVNS$
Dist10_2	189192	189192	189192	189192	18590	3652	11	4923	0,00	0,00	0,00	0,00	0,00
Dist10_4	229833	230255	230868	229833	432276	111598	17	16353	0,45	0,00	0,27	-0,18	-0,45
Dist10_6	314347	315073	325270	314548	220160	68274	26	6929	3,36	0,06	3,13	-0,17	-3,41
Dist10_8	368200	376568	394179	371640	284669	119594	31	4419	6,59	0,93	4,47	-1,33	-6,06
Dist20_2	237142	237172	237278	237142	400087	122241	16	21554	0,06	0,00	0,04	-0,01	-0,06
Dist20_4	323946	333044	368600	327620	5519716	367283	31	29077	12,11	1,12	9,65	-1,66	-12,51
Dist20_6	503080	500727	577066	503110	2835399	465238	51	17736	12,82	0,01	13,23	0,47	-14,70
Dist20_8	-	617448	712334	623769	-	562460	67	21998	-	-	13,32	1,01	-14,20
Dist30_2	-	292829	314966	284080	-	186762	25	57024	-	-	7,03	-3,08	-10,87
Dist30_4	-	429931	495029	425358	-	413262	47	48520	-	-	13,15	-1,08	-16,38
Dist30_6	-	703681	840055	711478	-	564949	75	23887	-	-	16,23	1,10	-18,07
Dist30_8	-	891709	1021850	900469	-	797419	96	29153	-	-	12,74	0,97	-13,48
Dist40_2	-	347007	372504	337258	-	194299	31	59409	-	-	6,84	-2,89	-10,45
Dist40_4	-	551809	650844	559043	-	448547	65	49542	-	-	15,22	1,29	-16,42
Dist40_6	-	795240	975549	836691	-	620872	97	38002	-	-	18,48	4,95	-16,60
Dist40_8	-	1111500	1312530	1142670	-	777197	126	30328	-	-	15,32	2,73	-14,87
Dist50_2	-	372586	407137	370891	-	225540	38	94556	-	-	8,49	-0,46	-9,77
Dist50_4	-	696879	825556	708868	-	426704	85	44820	-	-	15,59	1,69	-16,46
Dist50_6	-	1014860	1229370	1062730	-	607357	118	34702	-	-	17,45	4,50	-15,68
Dist50_8	-	1273270	1613570	1412190	-	908095	144	31605	-	-	21,09	9,84	-14,26

Table 6.15: Comparative analysis of the proposed methods - Set 5

Inst	ub				#A				gap (%)				
	E	L	$2MVNS$	$3MVNS$	E	L	$2MVNS$	$3MVNS$	E - $2MVNS$	E - $3MVNS$	L - $2MVNS$	L - $3MVNS$	$2MVNS$ - $3MVNS$
Dist10_2	205395	206698	205395	205395	39532	7051	9	5755	0,00	0,00	-0,63	-0,63	0,00
Dist10_4	293166	293166	293166	293166	67063	18680	19	4205	0,00	0,00	0,00	0,00	0,00
Dist10_6	290025	299376	305312	290025	185198	79667	26	7526	5,01	0,00	1,94	-3,22	-5,27
Dist10_8	347438	347700	386042	357723	726288	203665	35	9440	10,00	2,88	9,93	2,80	-7,92
Dist20_2	226903	226903	236903	226903	503444	133736	17	32459	4,22	0,00	4,22	0,00	-4,41
Dist20_4	341116	350880	263796	342838	1078899	232176	33	26087	-29,31	0,50	-33,01	-2,35	23,06
Dist20_6	514038	514067	575718	516482	4947212	527098	49	20027	10,71	0,47	10,71	0,47	-11,47
Dist20_8	638242	629659	729008	640730	8057169	739668	65	17467	12,45	0,39	13,63	1,73	-13,78
Dist30_2	279766	288760	310974	279782	1375993	205438	25	48890	10,04	0,01	7,14	-3,21	-11,15
Dist30_4	459172	446089	521966	439229	5014887	389652	52	51159	12,03	-4,54	14,54	-1,56	-18,84
Dist30_6	-	676189	792288	691003	17349130	596574	76	29587	-	-	14,65	2,14	-14,66
Dist30_8	-	879662	1080310	919971	-	801687	100	29787	-	-	18,57	4,38	-17,43
Dist40_2	-	363871	388067	354014	-	197598	33	62585	-	-	6,24	-2,78	-9,62
Dist40_4	-	603742	696541	598317	-	420220	68	37016	-	-	13,32	-0,91	-16,42
Dist40_6	-	830717	1017350	859789	-	612355	99	37878	-	-	18,35	3,38	-18,33
Dist40_8	-	1144300	1357200	1158820	-	849536	123	29622	-	-	15,69	1,25	-17,12
Dist50_2	-	384999	430410	384219	-	215185	40	72068	-	-	10,55	-0,20	-12,02
Dist50_4	-	674616	795523	695088	-	432247	82	59419	-	-	15,20	2,95	-14,45
Dist50_6	-	978577	1210500	1042700	-	666758	114	36548	-	-	19,16	6,15	-16,09
Dist50_8	-	1265550	1624110	1423160	-	850545	148	35501	-	-	22,08	11,07	-14,12

Tables 6.16 to 6.18 synthesize the comparative analysis of these approaches regarding three main elements: the value of the objective function, the number of generated arcs, and the total execution time. For each table and each column, a negative value represents an improvement whereas a positive one shows a worse value. As an example, in Table 6.16 in the column E - $2MVNS$ the positive values indicate that the matheuristc presented an average result 5.56% worse than the exact model. For the same comparison, in Table 6.17, the number of generated arcs has an average reduction of $99,97\%$. The total time, for the same comparison, was also reduced by $1294,94\%$, in average (see Table 6.18).

The improvement is computed according to Equation 6.7, where A_1 and A_2 refer to the compared values of the approaches. Thus, in Table 6.16 the improvement for the first column considers the objective values of the exact model (A_1) and the 2MVNS (A_2) approach. This process is similar in Tables 6.17 and 6.18 taking in consideration the number of arcs and the total time, respectively.

$$imp_{A_1-A_2} = \frac{A_1 - A_2}{A_2} \tag{6.7}$$

Table 6.16: Comparative analysis between the model and matheuristcs - objective value (%)

$Inst$	E - $2MVNS$	E - $3MVNS$	L - $2MVNS$	L - $3MVNS$	$2MVNS$ - $3MVNS$
Set 1	3,16	0,16	11,50	0,83	-12,51
Set 2	8,54	0,17	9,62	-0,88	-11,90
Set 3	7,51	-0,03	8,51	-1,30	-11,08
Set 4	5,06	0,30	10,59	0,89	-11,23
Set 5	3,51	-0,03	9,11	1,07	-10,00
Total	5,56	0,11	9,86	0,12	-11,35

Through the analysis of the data presented in Tables 6.16 to 6.18 the limited and matheuristics approaches are able to retrieve good quality solutions in acceptable computational times, since these approaches deal with a lower number of arcs. Disregarding the exact model, the remaining approaches were able to reduce both the computational time and the number of generated arcs.

Table 6.17: Comparative analysis between the model and matheuristcs - $\#Arcs$ (%)

$Inst$	E - $2MVNS$	E - $3MVNS$	L - $2MVNS$	L - $3MVNS$	$2MVNS$ - $3MVNS$
Set 1	-99,99	-96,28	-99,98	-87,20	99,70
Set 2	-99,93	-90,55	-99,81	-69,88	99,31
Set 3	-99,93	-91,32	-99,78	-70,39	99,23
Set 4	-99,99	-94,07	-99,97	-82,00	99,76
Set 5	-99,99	-96,33	-99,97	-84,37	99,76
Total	-99,97	-93,71	-99,90	-78,77	99,55

Table 6.18: Comparative analysis between the model and matheuristcs - T_{total} (%)

$Inst$	E - $2MVNS$	E - $3MVNS$	L - $2MVNS$	L - $3MVNS$	$2MVNS$ - $3MVNS$
Set 1	-535,40	-273,82	-728,97	-326,18	39,30
Set 2	-1294,93	-792,38	-438,05	-200,02	32,64
Set 3	-1421,40	-766,92	-439,46	-175,72	35,36
Set 4	-1375,48	-928,47	-795,47	-382,31	36,45
Set 5	-1847,48	-947,13	-861,13	-402,12	41,01
Total	-1294,94	-741,74	-652,62	-297,27	36,95

However, the $3MVNS$ approach presents the best performance considering the obtained gap values, the reduced number of generated arcs and the time needed for its execution.

6.4 Conclusions

In this section, an integrated management science problem that occurs in the logistics and transportation fields is addressed. The MPIDRPTW considers production, inventory, distribution and routing decision at the same time. The PDP deals with production and distribution decisions, but disregards inventory restrictions. The IRP considers inventory and routing decisions, but it does not take into account production constraints. The PIDRP integrates these two problems and also considers routing decisions, such as the MVRP. Thus, this integrated problem has received special attention in the last years. The MVRP allows the multiple usage of a vehicle during the planning horizon, taking into account this specificity when the routes to serve the clients are determined. The variant MPIDRPTW includes the PIDRP concerns, allowing for the multiple usage of a vehicle during the planning horizon. The MPIDRPTW also takes into account the allowed client delivery time intervals, known as time window.

The aim of the MPIDRPTW problem is to minimize the cost of distribution, routing, production and inventory decisions during the planning horizon. The time varying demand client orders must be completely delivered within the time windows. There is a fleet of homogeneous vehicles to fulfill the clients needs and each vehicle can make more than a single-trip during the planning horizon. The problem has a single facility, where production and inventory decisions occur. This facility has a capacity production and setup cost. Split deliveries are not allowed, but future deliveries can be anticipated. A demand may be fulfilled from production or inventory held at the facility.

Two different metaheuristics were proposed to solve the MPIDRPTW. Through these approaches, the VNS explores a set of neighborhoods in order to find good routing and distribution decisions; on the other hand, the arc flow model uses the arcs generated to optimize the production and inventory decision at the facility.

A set of computational tests was performed, on a set of benchmark instances, to prove that the two matheuristics can provide good valid solutions within a short computational time, reducing the difficulty associated with the complete enumeration of all valid arcs. A comparative analysis between the exact arc flow model, limited arc flow model and the two matheuristics was presented to prove the quality of the solutions.

Chapter 7

Conclusions

Outline

7.1 Contributions

Nowadays, companies are becoming more and more competitive. For this reason, integrated planning of operations has become an important support for companies, since in addition to costs it can help to improve the times in the execution of better planning in the different areas of the industrial companies. This integration is essential for companies to achieve higher levels of competitiveness. Despite significant progress in this area, current approaches have significant limitations. In most cases, there is no concern about the strong integration between the various functions of the enterprise and the problems are solved in an independent way. This lack of integration leads to solutions that are sub-optimal from the global perspective of companies.

In this thesis, efficient integrated optimization models for two classes of important problems were proposed and implemented: the facility location and vehicle routing problem; and production scheduling and distribution problems. The research focused on innovative techniques of Integer Programming based on original reformulations and on heuristic methods which explore the neighborhoods or the relaxation value of the original decision variables. Different variants of the problems were analyzed and explored in order to develop a set of models and algorithms to increase the efficiency of the facility location and vehicle routing problem; and production scheduling and distribution problems in transportation and supply chain management.

Applying research to real cases is a prime practical goal. However, it is necessary to develop a theoretical and practical work before using it in complex real scenarios. This thesis aims to have a relevant scientific and practical contribution, although it focuses on the analysis and development of methods and algorithms of integrated optimization in theoretical contexts.

The problems addressed in this thesis explore a particular variant of integrated problems consisting in the multiple uses of a vehicle from a fleet of homogeneous vehicles. This variant is called multi-trip as it allows for a vehicle to make more than one simple route during its working horizon, returning to the facility whenever necessary. This variant is rarely discussed in the literature, although it has significant relevance when the geographical configuration of the network is small and dense, since the use of multiple vehicles in the same temporal window may congest the network. The multi-trip variant is still important for the perishable goods transportation and when resources are limited, both in terms of vehicles and drivers.

According to the exhaustive literature review, this research work appears to be the first to conciliate the variant of multi-trip with both the location routing problem and the production, inventory, distribution and routing problem with time windows.

In Chapter 3 and Chapter 4, innovative exact methods and heuristic methods were presented to solve the multi-trip location routing problem, while in Chapter 5 and Chapter 6 an innovative exact method and two heuristic methods to solve the multi-trip production, inventory, distribution an routing problem with time windows were proposed, respectively. A comparative analysis of results is performed in order to perceive the quality of the solutions obtained and the performance of the proposed models and algorithms. This analysis revealed good results and confirmed the development of efficient algorithms and models.

7.2 Future work

There are several integrated optimization issues that need to be addressed, so opting for one in particular is not an easy decision. Thus there is always future work to be done.

Some fresh ideas emerged concerning the analysis of new variants of the problems addressed or even new approaches, such as the development and comparison of new heuristics methods for the sake of scientific curiosity.

One important idea will be to develop heuristics that will aid in the generation of multi-trips in acceptable computational times. The biggest challenge is to build several routes of very good quality.

It also would be interesting to address the location routing problem introducing time windows constraints; or even to introduce the concept of multi-product in the two main problems explored in this thesis. Another interesting idea would be to manage the load of vehicles through algorithms of 3D bin packing.

The most attractive future work is perhaps the application of the proposed models in a real industrial case and the development of an appropriate interface to manage the entire production process. In this way, it would be possible to rigorously test the algorithms in an industrial context and to enrich them with new emerging challenges.

References

[1] C. Alves and J. M. Valério De Carvalho, "New integer programming formulations and an exact algorithm for the ordered cutting stock problem," *Journal of the Operational Research Society*, vol. 59, no. 11, pp. 1520–1531, 2008.

[2] R. Armstrong, S. Gao, and L. Lei, "A zero-inventory production and distribution problem with a fixed customer sequence," *Annals of Operations Research*, vol. 159, pp. 395–414, 2008.

[3] N. Azi, M. Gendreau, and J.-Y. Potvin, "An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles," *European Journal of Operational Research*, vol. 202, no. 3, pp. 756–763, 2010.

[4] R. Macedo, C. Alves, and J. V. de Carvalho, "Arc-flow model for the two-dimensional guillotine cutting stock problem," *Computers & Operations Research*, vol. 37, no. 6, pp. 991–1001, 2010.

[5] R. Macedo, C. Alves, J. Valério de Carvalho, F. Clautiaux, and S. Hanafi, "Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model," *European Journal of Operational Research*, vol. 214, no. 3, pp. 536–545, Nov. 2011.

[6] A. Bixby, B. Downs, and M. Self, "A Scheduling and Capable-to-Promise Application for Swift & Company," *Interfaces*, vol. 36, no. 1, pp. 69–86, 2006.

[7] Y.-C. Chang and C.-Y. Lee, “Machine scheduling with job delivery coordination,” *European Journal Of Operational Research*, vol. 158, no. 2, pp. 470–487, 2004.

[8] M. Durbin and K. Hoffman, “The Dance of the Thirty-Ton Trucks: Dispatching and Scheduling in a Dynamic Environment,” *Operations Research*, vol. 56, no. 1, pp. 3–19, 2008.

[9] H. N. Geismar, G. Laporte, L. Lei, and C. Sriskandarajah, “The Integrated Production and Transportation Scheduling Problem for a Product with a Short Lifespan,” *Informs Journal on Computing*, vol. 20, no. 1, pp. 21–33, 2008.

[10] Q. Wang, R. Batta, and R. Szczerba, “Sequencing the processing of incoming mail to match an outbound truck delivery schedule,” *Computers & Operations Research*, vol. 32, no. 7, pp. 1777–1791, 2005.

[11] X. Wang and T. Cheng, “Machine scheduling with an availability constraint and job delivery coordination,” *Naval Research Logistics*, vol. 54, no. 1, pp. 11–20, 2007.

[12] G. Nagy and S. Salhi, “Location-routing: Issues, models and methods,” *European Journal of Operational Research*, vol. 177, no. 2, pp. 649–672, 2007.

[13] Z. Akca, R. T. Berger, and T. K. Ralphs, “A Branch-and-Price Algorithm for Combined Location and Routing Problems Under Capacity Restrictions,” *Operations Research*, vol. 47, pp. 1–19, 2009.

[14] G. Nagy and S. Salhi, “Nested heuristic methods for the location-routing problem,” *European Journal of Operational Research*, vol. 47, no. 9, pp. 1166–1174, 1996.

[15] D. Tuzun and L. Burke, “A two-phase tabu search approach to the location routing problem,” *European Journal of Operational Research*, vol. 116, no. 1, pp. 87–99, 1999.

[16] C. Prodhon and C. Prins, “A survey of recent research on location-routing problems,” *European Journal of Operational Research*, vol. 238, no. 1, pp. 1–17, 2014.

[17] M. Drexl and M. Schneider, "A survey of variants and extensions of the location-routing problem," *European Journal of Operational Research*, vol. 241, no. 2, pp. 283–308, 2015.

[18] J.-M. Belenguer, E. Benavent, C. Prins, C. Prodhon, and R. Calvo, "A branch-and-cut method for the capacitated location-routing problem," *Computers & Operations Research*, vol. 38, no. 6, pp. 931–941, 2011.

[19] R. T. Berger, "Location-routing problems with distance constraints," *Transportation Science*, vol. 41, no. 1, pp. 29–43, 2007.

[20] M. Albareda-Sambola, J. A. Días, and E. Fernández, "A compact model and tight bounds for a combined location-routing problem," *Computers & Operations Research*, vol. 32, no. 3, pp. 407–428, 2005.

[21] S. Barreto, C. Ferreira, J. Paixão, and B. S. Santos, "Using clustering analysis in a capacitated location-routing problem," *European Journal of Operational Research*, vol. 179, no. 3, pp. 968–977, 2007.

[22] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Prentice Hall, 1988.

[23] R. B. Lopes, S. Barreto, C. Ferreira, and B. S. Santos, "A decision-support tool for a capacitated location-routing problem," *Decision Support Systems*, vol. 46, no. 1, pp. 366–375, 2008.

[24] G. Laporte, Y. Nobert, and D. Arpin, "An exact algorithm for solving a capacitated location-routing problem," *Annals of Operations Research*, vol. 6, no. 9, pp. 291–310, 1986.

[25] M. Albareda-Sambola, E. Fernández, and G. Laporte, "A computational comparison of several models for the exact solution of the capacity and distance constrained plant location problem," *Computers & Operations Research*, vol. 38, no. 8, pp. 1109–1116, 2011.

[26] J. W. Escobar, R. Linfati, and P. Toth, “A two-phase hybrid heuristic algorithm for the capacitated location-routing problem,” *Computers & Operations Research*, vol. 40, no. 1, pp. 70–79, 2013.

[27] S. Doulabi and A. Seifi, “Lower and upper bounds for location-arc routing problems with vehicle capacity constraints,” *European Journal of Operational Research*, vol. 224, no. 1, pp. 189–208, jan 2013.

[28] C. K. Y. Lin, C. K. Chow, and A. Chen, “A location-routing-loading problem for bill delivery services,” *Computers and Industrial Engineering*, vol. 43, no. 1-2, pp. 5–25, 2002.

[29] C. Lin and R. Kwok, “Multi-objective metaheuristics for a location-routing problem with multiple use of vehicles on real data and simulated data,” *European Journal of Operational Research*, vol. 175, no. 3, pp. 1833–1849, 2006.

[30] Z. Akca, R. T. Berger, and T. K. Ralphs, “Modeling and Solving Location Routing and Scheduling Problems,” *Technical Report*, 2008.

[31] R. Macedo, “Models and algorithms for hard optimization problems,” Ph.D. dissertation, Universidade do Minho, 2011.

[32] R. Macedo, C. Alves, and J. V. de Carvalho, “An exact approach for the location routing problem with multiple routes,” in *International Conference on Engineering UBI2011 (ICEUBI2011) - Innovation & Development*, 2011.

[33] N. Azi, M. Gendreau, and J.-Y. Potvin, “An exact algorithm for a single-vehicle routing problem with time windows and multiple routes,” *European Journal of Operational Research*, vol. 178, no. 3, pp. 755–766, 2007.

[34] N. Azi, M. Gendreau, and J. Y. Potvin, “An adaptive large neighborhood search for a vehicle routing problem with multiple routes,” *Computers & Operations Research*, vol. 41, no. 1, pp. 167–173, 2014.

[35] A. Olivera and O. Viera, “Adaptive memory programming for the vehicle routing problem with multiple trips,” *Computers & Operations Research*, vol. 34, no. 1, pp. 28–47, 2007.

[36] A. Mingozzi, R. Roberti, and P. Toth, “An exact algorithm for the multitrip vehicle routing problem,” *Informs Journal on Computing*, vol. 25, no. 2, pp. 193–207, may 2013.

[37] D. Cattaruzza, N. Absi, D. Feillet, and T. Vidal, “A memetic algorithm for the Multi Trip Vehicle Routing Problem,” *European Journal of Operational Research*, vol. 236, no. 3, pp. 833–848, 2014.

[38] Z.-L. Chen, “Integrated Production and Outbound Distribution Scheduling: Review and Extensions,” *Operations Research*, vol. 58, no. 1, pp. 130–148, 2010.

[39] C. L. Li, G. Vairaktarakis, and C. Y. Lee, “Machine scheduling with deliveries to multiple customer locations,” *European Journal Of Operational Research*, vol. 164, no. 1, pp. 39–51, 2005.

[40] Z.-L. Chen and G. L. Vairaktarakis, “Integrated Scheduling of Production and Distribution Operations,” *Management Science*, vol. 51, no. 4, pp. 614–628, 2005.

[41] L. Lei, S. Liu, A. Ruszczynski, and S. Park, “On the integrated production, inventory, and distribution routing problem,” *IIE Transactions*, vol. 38, no. 11, pp. 955–970, 2006.

[42] M. Boudia, M. A. O. Louly, and C. Prins, “A reactive GRASP and path relinking for a combined production–distribution problem,” *Computers & Operations Research*, vol. 34, no. 11, pp. 3402–3419, 2007.

[43] ——, “Fast heuristics for a combined production planning and vehicle routing problem,” *Production Planning & Control*, vol. 19, no. 2, pp. 85–96, 2008.

[44] M. Boudia and C. Prins, “A memetic algorithm with dynamic population management for an integrated production–distribution problem,” *European Journal of Operational Research*, vol. 195, no. 3, pp. 703–715, 2009.

[45] O. Solyali and H. Süral, “A Relaxation Based Solution Approach for the Inventory Control and Vehicle Routing Problem in Vendor Managed Systems,” in *Modeling, Computation and Optimization*, S. Neogy, A. Das, and R. Bapat, Eds. World Scientific, 2009, ch. 11, pp. 171–189.

[46] J. Bard and N. Nananukul, “A branch-and-price algorithm for an integrated production and inventory routing problem,” *Computers & Operations Research*, vol. 37, no. 12, pp. 2202–2217, 2010.

[47] J. F. Bard and N. Nananukul, “Heuristics for a multiperiod inventory routing problem with production decisions,” *Computers & Industrial Engineering*, vol. 57, no. 3, pp. 713 – 723, 2009.

[48] ——, “The integrated production–inventory–distribution–routing problem,” *Journal of Scheduling*, vol. 12, no. 3, p. 257, 2009.

[49] M. Ruokokoski, O. Solyali, J.-F. Cordeau, R. Jans, and H. Sural, “Efficient Formulations and a Branch-and-Cut Algorithm for a Production-Routing Problem,” *GERAD Technical Report G-20 10-66, HEC Montreal Canada*, 2010.

[50] V. Armentano, A. Shiguemoto, and A. Løkketangen, “Tabu search with path relinking for an integrated production–distribution problem,” *Computers & Operations Research*, vol. 38, no. 8, pp. 1199 – 1209, 2011.

[51] C. Archetti, L. Bertazzi, G. Paletta, and M. G. Speranza, “Analysis of the maximum level policy in a production-distribution system,” *Computers & Operations Research*, vol. 38, no. 12, pp. 1731 – 1746, 2011.

[52] N. Nananukul, "Clustering model and algorithm for production inventory and distribution problem," *Applied Mathematical Modelling*, vol. 37, no. 24, pp. 9846 – 9857, 2013.

[53] N. Absi, C. Archetti, S. Dauzère-Pérès, and D. Feillet, "A two-phase iterative heuristic approach for the production routing problem," *Transportation Science*, vol. 49, no. 4, pp. 784–795, 2014.

[54] Y. Adulyasak, J.-F. Cordeau, and R. Jans, "Benders Decomposition for Production Routing Under Demand Uncertainty," *Operations Research*, 2015.

[55] R. Macedo, B. Ramos, C. Alves, J. V. de Carvalho, S. Hanafi, and N. Mladenović, "Integer Programming Based Approaches for Multi-Trip Location Routing." Springer International Publishing, 2016, pp. 79–90.

[56] B. Ramos, C. Alves, and J. Valério de Carvalho, "Column Generation Based Approaches for Combined Routing and Scheduling," *Electronic Notes in Discrete Mathematics*, vol. 64, pp. 155–164, 2018.

[57] D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen, "An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems," *Networks*, vol. 44, no. 3, pp. 216–229, 2004.

[58] F. Clautiaux, C. Alves, and J. Valério de Carvalho, "A survey of dual-feasible and superadditive functions," *Annals of Operations Research*, vol. 179, no. 1, pp. 317–342, 2010.

[59] R. Macedo, C. Alves, S. Hanafi, B. Jarboui, N. Mladenović, B. Ramos, and J. Valério de Carvalho, "Skewed general variable neighborhood search for the location routing scheduling problem," *Computers & Operations Research*, vol. 61, pp. 143–152, 2015.

[60] R. Macedo, S. Hanafi, B. Jarboui, N. Mladenovic, C. Alves, and J. J. M. de Carvalho, "Variable neighborhood search for the Location Routing problem with multiple routes," in *In-*

ternational Conference on Industrial Engineering and Systems Management (IEEE-IESM'2013), 2013, pp. 1–6.

[61] H. Derbel, B. Jarboui, S. Hanafi, and H. Chabchoub, "Genetic algorithm with iterated local search for solving a location-routing problem," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2865–2871, 2012.

[62] N. Mladenović and P. Hansen, "Variable neighborhood search," *Computers & Operations Research*, vol. 24, no. 11, pp. 1097–1100, Nov. 1997.

[63] P. Hansen, N. Mladenović, and J. A. Moreno Pérez, "Variable neighbourhood search: methods and applications," *4OR A Quarterly Journal of Operations Research*, vol. 6, no. 4, pp. 319–360, Nov. 2008.

[64] ——, "Variable neighbourhood search: methods and applications," *Annals of Operations Research*, vol. 175, no. 1, pp. 367–407, Oct. 2009.

[65] P. Hansen and N. Mladenović, "Variable neighborhood search: Principles and applications," *European Journal of Operational Research*, vol. 130, no. 3, pp. 449–467, May 2001.