UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

The Multi-product Location-Routing Problem with Pickup and Delivery

THÈSE

présentée et soutenue publiquement le 11 December 2015

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Younes Rahmani

Composition du jury

Rapporteurs :	Caroline PRODHON : Maître de conférence (HDR), Université de Technologie de Troyes
	Philippe LACOMME : Maître de conférence (HDR), Université de Clermont-Ferrand
Examinateurs :	Ammar OULAMARA : Professeur, Université de Lorraine (directeur de thèse)
	Wahiba RAMDANE CHERIF : Maître de conférence, Université de Lorraine (co-directrice)
	El-Houssaine AGHEZZAF : Professeur, Faculty of Engineering, Ghent University
	Christelle GUÉRET : Professeur, Université d'Angers

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thesul.

Remerciements

Foremost, I would like to express my sincere gratitude to my advisors Prof. Ammar Oulamara and Dr. Wahiba Ramdane Cherif for their continuous support during my Ph.D. study and related research, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis.

Besides, I would like to thank my thesis committee for their encouragement, insightful comments, and hard questions which incented me to widen my research from various perspectives.

Also, I thank Lorraine Universty for funding my Ph.D. thesis and I'm grateful to all my collaborators in Lorraine Research Laboratory in Computer Science and its Applications and industrial department of Ecole des Mines de Nancy for providing seminars, informations, frequent meetings and discussions.

My sincere thank also goes to Loria laboratory and Mines de Nancy staff for all the administrative help, instrumental guidance and support and also for creating such a wonderful working atmosphere.

I am grateful to Prof.Dominique Mery, director of IAEM doctoral school, for all his care and attention for Ph.D. students.

My appreciation also extends to my laboratory colleagues, Kamal, Ons, Berna for hard days we were working together, and for all the fun we have had during past years.

My very very especial thank goes to Valia, Bertrand, Bamdad, Meysam, Mahta and Maryam who made my days in Nancy delighted, with whom I share unforgettable memories. They took such an especial place in my heart and I can never imagine my stay in Nancy without their presence.

Last but not least, I thank my family for supporting me spiritually throughout my life and giving me the confidence I needed to accomplish challenges. They are biggest inspiration of my life ; I don't know how I could make it through hard days without his continuous encouragements and positive vibes.

Je dédie cette thèse
à mes parents.
Leila & Alireza.

Table des matières

Liste des tableaux

Notations

$\boldsymbol{P}$: set of product types.

c(i,j) : cost of the shortest path in G between vertices i and j, $(i, j \in N)$.

d_{is} : demand of client i for c-product $s \in P$.

p_{ks} : amount of c-product that could be provided by potential processing center $k \in N_0$.

FD_k : a fixed cost associated with each potential processing center $k \in N_0$.

V_1 : a set of identical primary vehicles, available at the main depot.

V_2 : a set of identical secondary vehicles, available at the processing center sites.

$\boldsymbol{V}$: set of all primary and secondary vehicles.

$C_{V1}(C_{V2})$: capacity of primary (secondary) vehicle.

F_{V1} (F_{V2}) : exploitation cost including the acquiring cost of the primary (secondary) vehicles.

dp_{ks} : primary product demand of processing center $k \in N_0$ for product $s \in P$.

H_{is} : is equal to 1 if node i asks for product $s \in P$, and 0 otherwise.

Ω_{ks} : is equal to 1 if processing center k produces c-product $s \in P$, and 0 otherwise.

$\mathbf{P}_j$: set of c-products types that customer $j \in N_c$ asks for them.

Time-Limit : maximum duration of vehicle routes.

$\boldsymbol{M}$: a large positive value.

Abbreviations

TSP : *Traveling Saleman Problem*
VRP : *Vehicle Routing Problem*
CVRP : *Capacitated Vehicle Routing Problem*
VRPTW : *Vehicle Routing Problem with Time Window*
HFVRP : *Heterogeneous Fleet Vehicle Routing Problem*
MDVRP : *Multi Depot Vehicle Routing Problem*
DVRP : *Dynamic Vehicle Routing Problem*
BB : *Brach and Bound*
AP : *Assignment Problem*
LR : *Lagrangian Relaxation*
DP : *Dynamic Programming*
ILP : *Integer Linear Programming*
NN : *Nearest Neighbor*
LS : *Local Search*
TS : *Tabu Search*
GA : *Genetic Algorithm*
2E-VRP : *Two-Echelon Vehicle Routing Problem*
SDVRP : *Split Delivery Vehicle Routing Problem*
VRPSF : *Vehicle Routing Problem with Satellite Facilities*
FLP : *Facility Location Problem*
SCP : *Set Covering Problem*
PMP : *P-median Problem*
2E-FLP : *Two-echelon Facility Location Problem*
LRP : *Location Routing Problem*
2E-LRP : *Two-echelon Location Routing Problem*
LRPSPD : *Location Routing Problem with Simultaneous Pickup and Delivery*
MMLRP : *Many to Many Location Routing Problem*
USBRP : *Urban School Bus Routing Problem*
MPLRP : *Multi-Product Location Routing Problem*
MIP : *Mixed Integer Problem*
MDVRPSD : *Multi Depot Vehicle Routing Problem with Split Delivery*
DARP : *Dial-a-Ride Problems*

Chapitre 1

Introduction

The subject of this thesis, which is in the field of transportation, concerns the resolution of two new extensions of location-routing problem. The location routing problems (LRPs) and their variants are models of the literature that allow combining strategic decisions (related to the selection of potential sites) with tactical and operational decisions (related to the assignment of customers to the selected potential sites and the construction of vehicle routes in order to serve all customer demands). The objective of the LRP is to minimize the total cost including routing costs, vehicle fixed costs, and potential site opening costs. This introductory chapter located the issue, in the context of transport and logistics, and gives a summary of the contributions of this thesis.

1.1 Transportation and logistic

One of the most important problems in a supply chain is the design of logistic network. The transportation costs often represent an important part of the logistic network cost and substantial saving can be reached by improving the transportation system. This transportation system design must also cope with the new challenges of sustainable development. For instance, decisions about the number and the localization of facilities (platforms, factories, depots) are among the important tactical decisions to consider in the design of the transportation system.

Interest in this kind of activity is increasing in the present life, since it can be applied to many areas, as well the transfer of physical products with mail delivery, parcel delivery, milk collection, the arrangement of public transport lines, salting roads, garbage collection, etc.

Furthermore, globalization involves logistic costs increasingly important for companies. This is why the latter has a vested interest to integrate into supply chain, a distribution networks in order to minimize the costs and ensure their competitiveness and sustainability. We should not overlook another aspect taking a dimension of growing interest : the environmental awareness that encourages sustainable development for example a reduction in emissions of greenhouse gases, which transportation is partly responsible.

For these reasons, much efforts are needed to develop flexible and effective transportation systems facing current concerns about the economy and our quality of life. Operations research on such problems is therefore essential. But to tackle these problems, it is important to know what kind of decisions should be considered.

1.2 Summary of the contributions of the thesis

This thesis proposes to study two new variants of location routing problems. These variants allow modeling applications in complex transportation networks that may include factories, warehouses, customers and suppliers. In these complex transportation networks, consolidating freight through one or more processing centers on the same route, allows considerable savings by sharing the used resources, and allows to meet with the environmental requirements. The originality in the two new variants, is the use of non classical Vehicle Routing Problem (VRP) routes in Location Routing Problem (LRP) which is not common in the literature. This is by including (1) multi-product constraint (2) pickup and delivery in the processing center that must be opened and (3) the use of the processing centers as intermediate facilities in the routes. To the best of our knowledge, these three constraints have not been considered simultaneously in LRP literature. The first variant studied in this thesis is related to two-level distribution networks, and is named Two-Echelon Multi-products Location-Routing problem with Pickup and Delivery (2E-MPLRP-PD). The second variant considers one level distribution networks, and can be reviewed as an extension of the pickup and delivery vehicle routing problems with the constraints cited above. It is named Multi-products Location-Routing problem with Pickup and Delivery (MPLRP-PD). Several applications are concerned by our new models like drink distribution, home delivery service and grocery store chains.

The thesis is organized into seven chapters. After this introductory chapter, the second chapter presents an overview of basic concepts related to the VRP and LRP. First we talk about the basic concepts to model the transportation problems. A brief reminder of classical resolution methods (exacts and heuristics) for combinatorial optimization problems, basic logistic problem for LRP and VRP, are defined. Basic mathematical formulations were also presented for VRP and LRP. Then in the chapter, we will focus on VRP and LRP variants that come close to our problem. The chapter will focus on study and analyze the variants of LRP and/or VRP from the literature that include the new constraints, and more particularly the integration of multi-echelon, pickup and delivery, multi-depot and multi-commodity aspects.

Chapter 4 and 5 are dedicated to the first extension named Two-Echelon Multi-products Location-Routing problem with Pickup and Delivery (2E-MPLRP-PD). In chapter 4, a definition and a mathematical formulation are given for the problem.

We propose a mixed integer linear model for the problem and solve the small-scale instances with Cplex. Furthermore, we propose non-trivial extensions of nearest neighbour and insertion approaches. We develop clustering based approaches that have not been extensively investigated with regards to location routing. Four strategies are tested for each method. These methods are tested on instances derived from LRP literature instances with up to 200 customers and 10 processing centers.

Chapter 5 investigates an improvement methods for 2E-MPLRP-PD to enhance the result of the proposed heuristics. We propose two types of local search approach, three «routing improvement Local Search» that related to routes improvement, and two «processing centers location improvement Local Search» that related to processing centers location improvement. The concept of 2-Opt is extended to deal with the pickup and delivery and multi-products constraints. A new Local Search specific to the 2E-MPLRP-PD, named *Merging Product* (MP) is introduced. We also investigated a metaheuristic called *Iterative Local Search (ILS)* for solving the 2E-MPLRP-PD. This metaheuristic is widely developed to solve VRP and rarely used for Location-Routing Problems. The proposed *ILS* is based on local search method integrating different types of neighborhood search procedures, and a specific perturbation procedures.

In chapter 6, we address the second new extension called Multi-products Location-Routing problem with Pickup and Delivery (MPLRP-PD). A definition and a mathematical formulation is given for this problem. A heuristic approach is proposed and compared with the lower bound of Cplex. We propose an «Adaptive Large Neighborhood Search» to improve the heuristic solution. 6 destroy operators and 4 repair operators, and 1 integrated destroy and repair operator are proposed. The performance of the proposed approaches are evaluated on adapted 2E-MPLRP-PD instances. The meta-heuristic $ALNS$ uses the generated solutions from different heuristics, proposed in previous section, as initial solutions.

Finally, a conclusion of the work discussed in this thesis and some perspectives of research are given in Chapter 7.

Chapitre 2

Basic concepts for transportation and location problems

Contents

2.1 Introduction

In general, a combinatorial optimization problem consists in finding an extreme point of a function (maximum or minimum), often called cost-function or objective function, on a discrete set. This solution is then called optimal. Finding an optimal solution in a discrete and finite set is an easy issue in theory. We need just to try all solutions and compare their qualities to see the best. However, in practice, the enumeration of all solutions may take too much time ; Or searching time for the optimal solution is a very significant factor and that's why the combinatorial optimization problems are considered so difficult. The complexity theory provides some tools in order to measure this search time. Moreover, since the set of feasible solutions is defined in an implicit manner, it is sometimes very hard to find even one feasible solution.

Distribution system is a combinatorial optimization problem, that consists in identifying and optimizing the flow of freights from manufacturing center to clients by using this transportation system. The basic problem for this transport system is known as «Routing Problem» in the operational research domain. This problem deals with essential optimization issues which lead to not only reduces the costs (km, duration, etc) but also the environmental impact (fuel, emissions, noise, etc).

In the vehicle routing problem, the customers can be described by various characteristics, such as : their demand (goods, quantity, delivery and/or pickup), their location in the network, time service, their availabilities, represented by the time window, and the type of vehicles needed to satisfy the customer demand. A routing problem is often modeled by graph. The nodes represent a set of clients and sometimes the number of vehicles available at depot is limited. A vehicle is exploited in each route, starting and ending at the same node named depot, by satisfying a set of client demands. We note that each client must be served only by one vehicle. The objective is to create the routes with minimal cost. The Traveling Salesman Problem (TSP) and the Vehicle Routing Problem (VRP) are the two most important types of Routing Problem. If the vehicle capacity allows to visit all customers, only one "big" route is required. This problem is known as the Traveling Salesman Problem (TSP). Otherwise, when one vehicle is not enough to serve the whole set of customers, several routes must be resulted. The resulted problem is known as Vehicle Routing Problem (VRP).

In some applications several depots may be required, the latter can store the goods, or may represent intermediate processing centers or cross-docking center. The location of these depots is also a very important tactical decision problem in logistic field because the flow of transport will be handled from these nodes. This logistic problem is known as facility location problem when the customers are directly served from depots by transport flows. We face Location-Routing Problem when these two decisions : (i) routing level and (ii) location of depots are combined.

This chapter introduces the basic concepts, related to Routing and Location problems. The two latter problems belong to the class of combinatorial optimization problems. We introduce the most common methods, used in the literature, for solving the combinatorial optimization problems in section 1.2. A more formal definition of *Vehicle Routing Problems (VRP)* and *Location-Routing Problem (LRP)* and relevant classical models are reported in section 1.3 and 1.4 respectively. These sections also present the VRP and LRP variants with their resolution methods. Section 2.5 concludes the chapter.

2.2 Basic concepts

2.2.1 Graph definition

The graphs are considered as a powerful tool to model many problems related to transport. This representation is exploited to describe the structure of a complex set which its elements are related. The graph is quite adapted to define any network, such as a road or electricity network. When the relations between elements in a model are not symmetric , we are dealing with a *direct graph*. Otherwise an *undirected graph* appears with the symmetric relations.

The transport logistics problems is defined on a complete graph (without loop) and weighted with a set of nodes (also called vertices) $N = \{0, 1, \ldots, n\}$. The nodes represent the depots or/and clients. Some models consider an undirected graph $G = (N, E, C)$, with a binary variable for each edge, equal to 1 if and only if the edge is used without specifying the direction of crossing. The variables can be written with two indices corresponding to the two nodes of the edge (x_{ij}) or a single index edge (x_w). In the first case, x_{ij} and x_{ji} represent the same variable. Other models consider a directed graph $G = (N, A, C)$. In this case x_{ij} is different from x_{ji} because the direction is taken into account and the links between the nodes are called *arc*. Each arc or edge has a non-negative cost $c(i, j) \in C$, in practice a distance or travel time.

In an undirected (directed) graph, a chain (path) is a series of consecutive edges (arcs) while a cycle (circuit) is a special case which both start-point and end-point are identical. A vehicle route is often described as a cycle. If there is a path from a node i to a node j in a weighted graph, the cost usually is equal to the sum of arc values (or edge values), crossing to get by nodes. If there are several paths between two nodes, the shortest path is considered to calculate the cost.

In order to compute *shortest paths* from a start node s to all other nodes, we can use a known labeling algorithm. For each vertex i is assigned a label value of the shortest paths from s to i. Two major families of these algorithms exist : (i) those that attach labels like Dijkstra's algorithm [79], and those with correction labels like Bellman Aalgorithm (Gondran and Minoux [87]).

Numerous books about graphs exist in the literature. Yellen and Gross [268] present a work regarding the theory of graph. From a computational point of view, we can mention the book of Gibbons [89], or more recently the book by Lacomme et al. [161].

2.2.2 Combinatorial Optimization methods

VRPs and LRPs are part of the combinatorial optimization problems. In most general form, a combinatorial optimization problem (also called discrete optimization) involves finding an optimal discrete set among one of the best feasible subset (or solutions). The concept of best solution is defined by an objective function. The combinatorial optimization problems can be viewed as searching for the best element of some set of discrete items ; therefore, any sort of search algorithm or metaheuristic can be used to solve them. However, generic search algorithms are not guaranteed to find an optimal solution, nor are guaranteed to run quickly (in polynomial time). In the following we will explain the most popular resolution methods.

The first group of solution methods are *exact methods* which usually have difficulties to deal with large instances, and may require long calculation time even on small instances. For this reason, the use of approximate methods is very useful to reach a good quality solution during reasonable execution time. These approximate methods can be divided into two classes : *heuristic* and *metaheuristic*.

2.2.2.1 Exact methods

The exact methods are based on the use of algorithms that allow the global optimum to be found in combinatorial optimization fields. However, they are often computationally expensive. For the NP-hard problems, the complete enumeration of solutions is usually considered as an exact approach but the complexity remains exponential.

Two major methods are addressed in order to enumerate the solutions : 1) Dynamic Programming and 2) Tree Search methods.

2.2.2.1.1 Dynamic Programming

Dynamic Programming is a very strong method to solve the optimization problems by splitting it into a set of sub-problems in a recursive manner and solve these sub-problems to optimality independently. This process starts with the smallest part of problem. Finding the solution of a small sub-problem can help to solve the larger, until all sub-problems are resolved. This concept was introduced by Bellman [33], in order to solve an optimization problem.

Dynamic programming is based on a simple assumption, called the optimality principle of Bellman : Any optimal solution is built on sub-problems, locally solved to optimality. In practice this means that we may deduct the optimal solution for a problem by combining optimal solutions of a series of sub-problems. Problem solution begins with solving of the smallest sub-problems and then progressively deduct the overall solutions of initial problem (see study of Sniedovich [226]).

2.2.2.1.2 Methods based on Tree Search

The methods based on tree search split the solution space according to properties. In the following, the most classical exact approaches of combinatorial optimization problems are presented.

2.2.2.1.2.1 Branch and bound

A branch and bound algorithm is a generic method for solving combinatorial optimization problems. The Branch-and-bound was proposed for the first time by Land and Doig [162].

A naive method to solve this kind of problems is to enumerate all solutions for the problem, calculating the cost for each, and then obtain the minimum. Sometimes it is possible to avoid enumerating solutions which are already known, by analyzing the properties of the problem. A non-promising solution cannot construct an optimal solution.

In other words, mainly it involves the construction of a search tree that represents the solutions space and pruning unnecessary branches of the tree that contain non-promising or infeasible solutions.

Constructing a tree has three components : 1) branching rule, 2) evaluation function to prune the solutions and 3) tree search strategy. The method starts with a root which represents implicitly all solutions. The *branching* breaks down the search space into subparts. For example, during a binary linear program solving, the current node can be separated into two branches : one with the value of a fractional variable, fixed to 0 and another to 1.

The objective is to reduce the search tree. Thus, the evaluation of a node in the tree is needed to determine a lower bound for the minimization problem. If this estimation is greater than the best solution found in previous iterations, there is no need to extend the branches of the tree from this node. The latter is then «pruned». Through this, we avoid branch all the sub-trees problem.

The tree search strategy is to define the selection order of the nodes in the development of the tree. The most common strategies are (i) depth-first search and (ii) the frontier search.

The «Branch-and-bound» can find optimal solutions for VRP and LRP, but in general the resolution time increases when the size of the problem increases. It works well for problems of small sizes, but otherwise, it may generate very large branches. The exploration is done with evaluations of branches and comparisons with a bound or a threshold criterion to optimize. The difficulty of these methods is to obtain a good quality of this value to improve the pruning phase (Bounding). Some methods such as Lagrangian relaxation (Miller, [164]) give better bounds than classical relaxation techniques (capacity constraint, constraint connectivity) (Laporte et al,. [141]). In the 70s, stronger relaxations were also developed, the Benders decomposition [31] and more recently the approach by adding bound (Additive Bounding - AB) which is used afterwards to increase size of solvable instances.

2.2.2.1.2.2 Branch and cut

The branch and cut is a branch and bound with dynamic generation of constraints (often called «cuts» because they serve to break the current fractional point). The term «Branch and Cut» was firstly introduced by (Padberg and Rinaldi, [195]) to solve the traveling salesman problem (TSP).

The method strengthens the bound by adding *valid inequalities* or *Cuts* (the constraints that do not remove any integer point of the polyhedron). During the branching the node is abandoned if the obtained «programming linear» has an optimal solution such that cost is greater than an obtained upper bound. Otherwise, the best found solution and upper bound will be updated if the solution is feasible and improves the best found solution. When no improvement can be achieved, the algorithm branch next node.

To solve programming linear, a solver is used and then attempts to find an optimal integer solution that respects the constraints. If this fails, a decomposition phase (Branch) of the problem into two sub-problems is necessary and the cuts phase is applied on generated sub-problems (Toth and Vigo, [247]).

2.2.2.1.2.3 Branch and Price

Firstly called by Savelsbergh [229], this technique was initially proposed by Johnson [128]. The branch and bound algorithms with column generation algorithms are called branch and price. A branch and price involves a tree exploration in the same way as in the branch and bound, but instead of having a simple resolution PL at each node, it has to solve a PL with column generation.

The column generation is a method to effectively solve large linear programming. It is based on the Dantzig-Wolfe decomposition [90] and breaking down all the constraints into two problems : the master problem and the subproblem. The master problem is the original problem with only a subset of variables being considered. The subproblem is a new problem created to identify new variables. The objective function of the subproblem is to reduce cost of the new variables with respect to the current dual variables.

Branch and Price defines first a *master problem* by applying a relaxation of the integrity constraints that are imposed during the descent of branch and bound tree and one *pricing problem* that evaluates each new column or group of columns, added to the master problem.

The central idea is that the large linear programs have too many variables (or columns), when we want to represent them all explicitly. At the optimum, most of the variables are out of base and in many cases, most of them are equal to zero and only a (small) subset of variables must be taken into account to solve the problem. The column generation technique allows to generate only the variables which have the potential to improve the objective function that is to find variables with negative reduced cost (assuming without loss of generality that the problem is a minimization problem).

The Branch and Cut could be considered as the dual version of the branch and price but it is much simpler to implement because binary classical branching on the variables do not lead to any problem, nor the new dual variables during adding the new constraints.

2.2.2.1.2.4 Branch-Cut-and-Price

Decompositions and reformulations of mixed integer programs are classical approaches to obtaining stronger relaxations and reduce symmetry. These often entail the dynamic addition of variables (columns) and/or constraints (cutting planes) to the model. When the linear relaxation in each node of a branch-and-bound tree is solved by column generation, this method is named branch-and-price. Optionally, as in standard branch-and-bound, cutting planes can be added in order to strengthen the relaxation, and this is called branch-cut-and-price.

2.2.2.2 Heuristics

Heuristics build a single solution progressively by a series of partial and definitive choices without backtracking, that means that they do not contain an improvement phase. In general, the construction of a solution is carried out by consecutive decisions, each item is being added by a greedy manner. In order to extend the partial solution, the item is selected from various possibilities, depending on an attribute that optimizes some criterion. Generally, each constructive heuristic is designed specifically for a given problem.

2.2.2.3 Local Search

The local search is a commonly used technique in combinatorial optimization. The aim of the local search approach (LS) is to improve a feasible solution by exploring a neighborhood $N(S)$ of the current solution S. $N(S)$ is the set of solutions that can be obtained by a simple transformation of the solution S, by using pre-defined moves. The local search approach seeks for the first neighbor from $N(S)$ that can be obtained by an improving move *(first improvement)* or by the best neighbor *(best improvement)*. If a neighbor S' is better than S, LS explores the neighborhood of S'. It stops when no improving neighbor is possible.

Local search heuristic as a improvement method, has proven to be an effective heuristic to generate a good quality solution to instances of the classical VRP as well as variants that involve additional constraints. The reader should consult the recent literature reviews [250, 115] for examples of successful local search implementations.

2.2.2.4 Metaheuristic

The aim of metaheuristics is similar to heuristics : get good solutions within reasonable time. We call metaheuristic methods in order to escape local minimum. The metaheuristics are structures of generic algorithms that can theoretically be applied to any type of optimization problem. In the literature, two major families of metaheuristic exist. The methods based on the exploration of a neighborhood are among the most classical metaheursitics. The idea is improving the current solution that is found into search space through each iteration by starting from an initial solution. We can mention the best-known methods in this category : the GRASP method (Greedy Randomized Adaptive Search Procedure), Simulated Annealing (SA), Tabu Search (TS), Iterative Local Search (ILS), Variable Neighborhood Search (VNS).

The second family of metaheuristic uses a population of solution in each iteration by improving a set of solutions in parallel. Genetic algorithms (GA), Dispersed Search (Scatter Search, SS) or Ant Colony algorithms (Ant Colony, AC) are common examples of this type of methods. In the following, some of these methods are explained briefly.

2.2.2.4.1 Genetic Algorithm

Genetic algorithms, belong to the larger class of evolutionary algorithms, proposed by Holland [122] which generate solutions for optimization problems using techniques inspired by natural evolution. They evolve a population of individuals (solutions) by phenomena of reproduction and mutation.

The basic principle is to start from an initial population of solutions, represented as *chromosomes*. Then, at each iteration, *crossover* of two individuals-parents from the population are made, by using a most promising selection (with good fitness). The crossover of two individuals combines features of these two latter to generate a new individual, called child. *Mutations* may occur in the progeny creation, this allows diversification with avoiding premature convergence. Then, two population management modes may be used. (i) Each iteration creates a number of child equal to the size of the initial population, and the latter is replaced by the child population in the next iteration. (ii) Each iteration combines only two parents, their children are then directly integrated by replacing other individuals in the current population.

2.2.2.4.2 Ant Colony

Ant colony algorithms are inspired by behavior of ants and constitute an optimization metaheuristics family.

Initially proposed by Dorigo et al. [72], for the search of optimal paths in a graph. The first algorithm was based on the behavior of ants seeking a path between their colony and a source of food. The original idea has since diversified to solve a larger class of problems and several algorithms have been developed, inspired by various aspects of the behavior of ants. The initialization of the graph is done by assigning zero rate of pheromones on the arcs. Ants are represented by officers who construct the solution. They move in the graph by making choices subjected to probabilities on the arcs to cross. Indeed, the construction of their path is biased in favor of arcs that strongly marked with their pheromones. Then, depending on the value of the obtained objective function, pheromones rates are updated.

2.2.2.4.3 Tabu Search

The idea of the tabu search (proposed by Glover [88]) consists in starting from a given solution to explore the neighborhood and choose the solution in this neighborhood that minimizes the objective function. It is essential to note this operation may lead to increase the value of the function (in a minimization problem). This is the case when all points of the neighborhood have a higher value. By using this mechanism we get out of a local minimum.

However, the risk occurs during next step, when we fall into the local minimum which we just escaped. Therefore it is necessary to consider a memory for heuristic. The mechanism is to prohibit to return to the last explored solutions. The solutions already explored are stored in a tabu list with a given size, which is an adjustable parameter of the heuristic. The algorithm is stopped after a number of iterations and returns the best found solution.

2.2.2.4.4 Greedy Randomized Adaptive Search Procedure, GRASP

GRASP was introduced in article Feo and Resende [81]. This metaheuristic produces a feasible solution. It is executed a number of times and the best solution found is kept. To produce a solution, two phases are executed one after the other : the first consists in a construction phase in greedy manner, which is followed by a local search phase to improve the current solution. GRASP is randomized to have a different initial solution at each iteration.

2.2.2.4.5 Simulated Annealing

Simulated annealing is a metaheuristic method inspired by a process, used in metallurgy. We alternate in the latter cycles of slow cooling and reheating (annealing) which have the effect of minimizing the energy of the material. Simulated annealing was developed by Kirkpatrick et al. [130]. Compared to the local search, simulated annealing also generates a serie of solutions but the cost may increase from one to the other solution.

Starting from a given solution, by modifying it, we get a second solution. Either it improves the criterion which we want to optimize, then we reduce the energy of the system, or it degrades. If we accept a solution improving the criterion, then we tend to look for the optimum in the neighborhood of the initial solution. The acceptance of a «bad» solution allows us to explore a larger part of the solution space and avoid to fall too quickly into a local optimum.

2.3 Vehicle Routing Problem (VRP)

The Traveling Salesman Problems (TSPs) are an important class of operation research and combinatorial optimization problems which consist in defining the route of a salesman, visiting all cities of a predefined set of cities and returning to the departure city (usually called depot). The main goal is to set an order in which each cities will be visited once (the tour is a Hamiltonian cycle) by minimizing the travel distance. The TSP is defined in an undirected, valued and complete graph. By adding some constraints or hypothesis to the TSP, many numbers of routing problems could be defined. The Vehicle Routing Problem (VRP) is a generalization of the TSP that can be described as the routes construction for several vehicles with a minimal cost, from a depot to a set of geographically distributed points (cities, stores, warehouses, schools, customers, etc.) in order to satisfy the required demands in the network and return to the depot when the capacity of vehicle and/or some other criteria is met (as time or distance limit). Since the TSP (simple version of the routing problem) is a NP-hard problem (Nemhauser and Wolsey [183]). The VRP is itself also an NP-hard problem. Figure (2.1) shows an illustrated example of VRP with one depot and nine clients. This solution proposes three routes.

Each route in one acceptable solution for classical VRP should respect three major constraints.

— Each client must be satisfied by one and only one vehicle.

— One vehicle handles just one route.

— All clients should be satisfied and assigned to one route.

Therefore, the VRP could be a generalized version of TSP in which the solution proposes to use more than one vehicle and/or the presence of the capacity constraint. In each created route, the sum of client demands shouldn't exceed vehicle capacity. If we use a homogenous fleet, all vehicles provide a unique capacity. In the VRP, sometimes we consider one constraint of autonomy. This issue proposes a maximal time of trip between departure of vehicle from the depot and its return to the depot. A subset of arcs of the transportation network that the vehicles can traverse and a subset of customers that they should be delivered by vehicles are considered in the model. Finally, the use of a vehicle requires a cost including a fixed part and a time cost of service. The most common objective for vehicle routing problem is the minimization of total transportation, including the cost of routes according to the distance (time of the routes) and the fixed cost of each used vehicle.

The terms conventionally used as "depot", "client", "goods" and "vehicles" may represent entities of different form and are not always linked to their original physical. Therefore, the proposed models and algorithms for the VRP are efficiently used, not only for the problems concerning the delivery/pick-up of goods (commodity), but also passengers transport, routes in services (repair) and more generally for the various real applications of transportation network.

In the literature, the customer demand is categorized in «soft» and «hard» classes. The

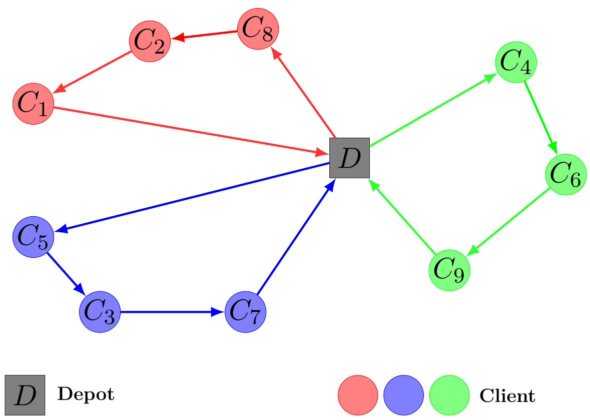

FIGURE 2.1 – A classical VRP example

hard demand must be completely satisfied but "soft" demand allows more flexible constraint, with penalty cost for each non-satisfied customer in function objective. Generally, in this kind of problem, the exploited vehicles can satisfy more than one client.

The vehicle routing problem was firstly proposed in the literature by Dantzig and Ramser [73]. After then, a considerable number of VRP variants has been considered : hard, soft and fuzzy service time windows, maximum route length, pickup and delivery, backhauls, etc. (Cordeau et al,. [51] ; Lopez-Castro and Montoya-Torres [158] ; Thangiah Salhi [223] ; Ozfirat Ozkarahan [223]).

Dantzig and Ramser [73] introduced the "Truck Dispatching Problem", and circumstance modeling of a homogeneous fleet of trucks in which they serve the oil demand of a set of gas stations from a central depot by minimizing travelled path. Few years later this problem was generalized by Clarke and Wright [48]. They propose a linear optimization problem that we face frequently in the logistics field : For example, how to satisfy a set of clients, geographically dispersed around the depot, using a fleet of trucks with limited capacity. This became known as the «Vehicle Routing Problem»(VRP) that is the most extensively studied topics in the domain of Operations Research.

Among the first studies in the literature, "Vehicle Routing" was sometimes replaced by "Vehicle Scheduling" (Clarke and Wright [48]) "Vehicle Dispatching" (e.g. Dantzig and Ramser [73]) or Delivery Problem" (e.g. Balinski and Quandt [26]).

There are two types of vehicle routing problems. When the demands are located on nodes of network, the problem is called node routing problem (Toth and Vigo study [240]). If the demands are located on arc (or edge) of network, the problem is called arc routing problem (Corberan and Laporte [63])

2.3.1 Mathematical formulation

Here, we present the most known formulations for the VRP. The first one is based on vehicle flow. We distinguish the symmetric and the asymmetric case. The second formulation is based on Set-Partitionning.

2.3.1.1 Vehicle flow model

2.3.1.1.1 Asymmetric case

2.3.1.1.1.1 Three-index Vehicle Flow Model

Golden et al. [181] proposed the simplest model to understand. It is an oriented pattern, said three indices with binary variables x_{ij}^v equal to 1 if the arc (i, j) is traversed by the vehicle v and binary variable z_i^v equal to 1 if the client i is visited by the vehicle v. N is a set of nodes , V is set of vehicles. Each vehicle has a capacity CV.

$$\min \sum_{i \in N} \sum_{j \in N} \sum_{v \in V} c(i, j) x_{ij}^v \tag{2.1}$$

Subject to :

$$\sum_{v \in V} z_0^v = \mid V \mid \tag{2.2}$$

$$\sum_{j \in N} x_{ij}^v = \sum_{j \in N} x_{ji}^v = z_i^v \qquad \forall i \in N, v \in V \tag{2.3}$$

$$\sum_{v \in V} z_i^v = 1 \qquad \forall i \in N - \{0\} \tag{2.4}$$

$$\sum_{i \in S} \sum_{j \notin S} x_{ji}^v \leq \mid S \mid -1 \qquad \forall v \in V, 2 \leq \mid S \mid, S \subseteq N - \{0\} \tag{2.5}$$

$$\sum_{i \in N} d_i z_i^v \leq CV \qquad \forall v \in V \tag{2.6}$$

$$z_i^v \in \{0, 1\} \qquad \forall i, j \in N \tag{2.7}$$

$$x_{ij}^v \in \{0, 1\} \qquad \forall i, j \in N, v \in V \tag{2.8}$$

(2.1) is the objective function that minimizes cost. Constraints (2.2) to (2.4) ensure that each customer is visited exactly once, $\mid V \mid$ routes are carried out and that the same vehicle arrives and departs from a client node. Constraints (2.5) are the sub-tour elimination constraints. Constraints (2.6) are the capacity restrictions of vehicles. Finally, constraints (2.7) and (2.8) define the binary variables.

2.3.1.1.1.2 Two-index Vehicle Flow Model

By eliminating the index v from the variables, another formulation can be provided for CVRPs, The fact that how many time the edge (i, j) is exploited by a vehicle will be represented with x_{ij}, $(i < j)$: if $i, j \in N - \{1\}$, then $x_i \in \{0, 1\}$; if $i = 1$ and $j \in N - \{1\}$, then $x_{ij} \in \{0, 1, 2\}$. The case $x_{ij} = 2$ corresponds to the single vertex trip $(1, j, 1)$.

The following formulation was proposed by Laporte et al. [145].

$$\min \sum_{i \in N, i<j} c(i, j)x_{ij} \tag{2.9}$$

Subject to :

$$\sum_{i \in N-\{0\}} x_{0j} = 2M \tag{2.10}$$

$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2 \qquad \forall k \in N - \{0\} \tag{2.11}$$

$$\sum_{i,j \in S, i<j} x_{ij} \geq\mid S \mid -v(S) \qquad \forall S \subseteq N - \{0\}, 2 \leq\mid S \mid\leq n - 2 \tag{2.12}$$

$$x_{0j} \in \{0, 1, 2\} \qquad \forall j \in N - \{0\} \tag{2.13}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in N - \{0\} \tag{2.14}$$

Note that M can be taken as a fixed constant or a variable. It is often practical to impose a lower or an upper bound on M. In sub-tour elimination constraints(2.12), $v(S)$ is a lower bound on the number of vehicles requested to meet S.

Constraint (2.9) defines a first sub-problem, the degree constraints (2.10) and (2.11), ensure the bounds on the variables. The constraints (2.12) eliminates the sub-tours.

2.3.1.1.2 Symmetric case

A compact and elegant formulation, used by many exact methods, was introduced by Laporte and Nobert [149] in 1987. Based on an undirected graph, it uses a variable per edge, with two indices (x_{ij}) or a single (x_w) which present the crossing number per edge, but any way, without vehicle index. The crossing cost by each edge is measured by c_w.

$(\delta(S))$ denotes the set of edges with one endpoint(extremity) which goes out from the subset of client S. If one route does not contain the depot, variables x_w are binary, $x_w \in \{0, 1\}$, otherwise if $w \in \delta(0)$ then $x_w \in \{0, 1, 2\}$. For a subset S of clients, $d(S)$ is the total demand for S and $r(S) = [d(S)/CV]$ the minimum number of vehicles to serve S.

$$\min \sum_{w \in E} c_w x_w \tag{2.15}$$

Subject to :

$$\sum_{i\in\delta(0)} x_w = 2M \tag{2.16}$$

$$\sum_{w\in\delta(i)} x_w = 2 \qquad \forall i \in N - \{0\} \tag{2.17}$$

$$\sum_{w\in\delta(S)} x_w \geq 2r(S) \qquad \forall S \subseteq N - \{0\}, S \neq \emptyset \tag{2.18}$$

$$x_w \in \{0,1,2\} \qquad \forall w \in \delta(0) \tag{2.19}$$

$$x_w \in \{0,1\} \qquad \forall w \notin \delta(0) \tag{2.20}$$

The objective function (2.15) is the total cost to be minimized. Constraints (2.16) specify that M vehicles will use $2M$ edges which is the number of entering and leaving arcs to depot (M entering and M leaving). Constraints (2.17) mean that every customer is visited : two edges are incident to i or the same edge is crossed twice in case of direct delivery. Constraints (2.18) provides in the same time the continuity of routes and assure the respect of each vehicle capacity : Since we need $r(S)$ vehicles to serve a subset S of clients, these vehicles will use at least $2r(S)$ edges, to traverse from S to $N - S$.

2.3.1.2 Set-Partitionning Models

Partitioning models have been proposed by Balinski and Quandt [26] in 1964 for VRPs. We define the set of feasible routes $j \in V = \{1, 2, \ldots, \mid V \mid\}$, each one associated with a cost $c(v)$. In addition, $a_i^v = 1$ if the client i is served by route v. The binary variable x_v takes the value 1 if and only if the route v is chosen in the solution. The problem can then be formulated as follows :

$$\min \sum_{v\in V} c(v)x_v \tag{2.21}$$

Subject to :

$$\sum_{v\in V} a_i^v x_v = 1 \qquad \forall i \in N - \{0\} \tag{2.22}$$

$$\sum_{v\in V} x_v = R \tag{2.23}$$

$$x_v \in \{0,1\} \qquad \forall v \in V \tag{2.24}$$

The objective function presents by (2.21). Constraints (2.22) require that each client i is used only once by a selected tour. For the constraint (2.23), R routes are exactly chosen. While partitioning models have a relatively simple expression, they have an exponential number of tours. Generally, it does not create all feasible routes. It is necessary to use a column generation approach to solve it optimally. In column generation, a reduced problem containing only a restricted subset of all possible columns (variables) is repeatedly solved.

2.3.2 VRP variants

The classical VRP provides delivery routes where each vehicle travels one route, each vehicle owns the same characteristics and there is just one depot. The aim of the VRPs is to minimize the cost of exploited vehicle (vehicle number) such that each client is met exactly once by one route. We consider that each vehicle starts and ends up at the same depot. We note that each vehicle has a capacity and the capacity of the vehicles should be respected in route. In the literature, some authors such as Chen et al. [61] use autonomy constraint for the routes and other authors like Cordeau et al. [49] and Christiansen [50] consider capacity constraint but not constraint of autonomy. A detailed state of the art for VRP variants is given in the book of Toth and Vigo [245] and in the works of Laporte et al. [146] Cordeau et al. [49].

Many variants of VRP have been defined by extending this basic problem, adding different constraints and/or objectives. The literature dedicated to these problems is very vast. To identify the different variants of routing problems, authors generally use initialisms, in which various prefixes and suffixes indicate the presence of different assumptions or constraints. But this identification based on initialisms is inefficient. Ramdane Cherif-Khettaf et al. [209] propose a new notation and classification scheme to identify vehicle routing problems without ambiguity. It describes the problems addressed through their assumptions, constraints and objectives rather than initialisms.

In the general case, a vehicle route begins at a depot and returns to the starting point, but there is a few cases like open VRPs in which the vehicles can be ended up in another point as ending depot (Open VRP , Sariklis and Powell [219]).

The VRPTW is another extension of VRP with Time Windows. This extension ensures that serving a client should be done during a time interval, given by the clients. Soft time windows allow deliveries outside the boundaries against a penalty cost (Tas et al. [238]). Time windows are considered as hard (or strict) otherwise, when we are not allowed to deliver outside of the time interval (Vidal et al. [249]).

Among the variants of VRPs, in Multi-Depot VRP, routes can start from a single depot or potentially several depots (not necessarily with identical characteristics) (Renaud et al. [207] and Mingozzi and Valletta [166]).

Dynamic Vehicle Routing Problems (Dynamic VRP), also known as "online vehicle routing problems", have recently arisen because of the progress of information technology and communication which allows to obtain information and processing these information in real time. In Dynamic Vehicle Routing Problems before starting the work day, the demands can be known in advance, but the day goes on, new demands arrive and the system must integrate them into its daily program, during the day, each driver always will be informed about future visits that he should drive them.

2.3.3 VRP Solution Methods

The number of VRP methods introduced in the literature has increased rapidly over the past years. According to the recent survey provided by Hall [110] thousands of big companies and enterprises, use VRP software. In the state of the art of VRP we can find many works that are related to VRP resolution methods (Gendreau et al. [94, 95], Ficher [84], Toth and Vigo [245]). In the work of Zhu [269], the problem with number of clients (>100) is not resolvable by the exact approaches during a reasonable execution time. Therefore today we have to use heuristics and meta-heuristics to solve this kind of problems.

The **exact methods** generally can effectively solve problems up to 50 customers. In 2003 a Parallel branch and cut method that could solve one problem with 100 customers has been proposed (Ralphs, [205]). A state of the art on the exact approaches for the VRP is given by Laporte and Nobert [149] and more recently in two books, those of Toth and Vigo [245] and Golden et al,. [101].

Laporte et al. [147] in 1986 and Fischetti et al. [82] in 1994 proposed algorithms that use a branch and bound for the assignment problem and bound additions. To do this issue, they use vehicles flow models. These algorithms are based on the work dedicated to the TSP Bellmore and Malone [30] in 1971.

Hadjiconstantinou et al. [112] in 1995, Agarwal et al. [1] in 1989 studied branch and bound algorithms where the lower bound is calculated by solving the dual of the linear relaxation of partitioning model.

Today, the best exact methods for the VRP are dominated by polyhedral approaches (Augerat et al. [14] Lysgaard et al. [155]), and more generally the «branch and cut» methods. These are generally based on vehicle flow models such as those developed by Baldacci et al. [25], or the partitioning models such as Baldacci et al. [22].

Gutierrez-Jarpa et al. [100] proposed an exact «branch and price» algorithm to the multiple Vehicle Routing Problem with Time Windows. We can also mention the branch-and-cut-and-price Fukasawa et al. [86] which combines column generation and branch and cuts. These algorithms were able to solve instances with up to 100 clients.

As another exact approach, Eilon et al. [80] probably first proposed «Dynamic Programming» for VRPs. This method has been used for VRP problems of very small size as in (Rego et al. [208]) to solve problems with 10 to 25 clients.

In comparison of exact methods, the «heuristics» and the «meta-heuristics» would be often more competitive for real applications, because the problems that we are faced today, are considerably larger in scale (e.g., an enterprise may have more than thousands of customers from more than ten depots with many vehicles and subject to a variety of constraints).

The VRP **heuristics** can be divided into two main classes : classical heuristics also called simple heuristics (most developed between 1960 and 1990) and new heuristics or metaheuristics (most developed after 1990).

Among simple approach and most efficient, Clarke and Wright [48] proposed «The Clarke and Wright algorithm» a classical saving algorithm to solve CVRPs. In this case the number of vehicles is unlimited. This method begins with a route that involves the depot and one other node. At each step, two routes among a subset of routes are merged according to the largest saving that can be generated. The idea is to build a trivial solution with a road, dedicated to customer (Marguerite) and merge them by using greedy method in order to obtain new routes. When two routes (0,..., i, 0) and (0, j,..., 0) are aggregated into a single tour (0,..., i, j,..., 0) a gain $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ is obtained. So by carrying out theses positive gains, this heuristic can reduce at the same time the total cost and the number of used routes (see Figure 2.2). Many improvements of this algorithm have been proposed in order to reduce its execution time (Yellow [267] Paessens [189]) or to perform a series of merges in parallel by solving an assignment problem (Desrochers and Verhoog [74]).

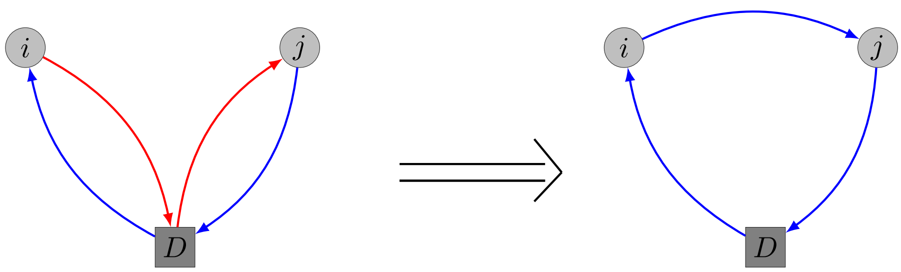

FIGURE 2.2 – A saving Algorithm

The «Insertion Method» is another constructive method that have proven to be popular methods for solving a variety of vehicle routing. The insertion algorithm proceed in two phases ; the first phase selects the nodes to insert and second phase will be applied in order to insert it in a route (Solomon [220]) and (Toth and Vigo [240]).

More complicated than constructive heuristics (step by step), there are «heuristics in two phases» : grouping customers first, then routing (cluster-first route-second), or first routing, then grouping (route-first cluster-second). The principle of «cluster-first route-second» is to allocate customers in separated sector with a total demand close to the capacity of vehicles and then construct a route in this sector. The best known of this type of method is the «Sweep Algorithm» (Gillett and Miller [98]) which uses angular sectors. This method works well if the depot is relatively central. The origins of the Sweep method can be go back to the work of Wren (1971) [2] for CVRPs with one or several depots, in which vertices are located in the Euclidean plane.

Fisher and Jaikumar [85] propose a segmentation (sectorization) phase based on a generalized assignment problem (Generalized Assignment Problem - GAP). As a clustering method we can mention another heuristic which is based on neighborhood method, called Petals algorithm (Balinski and Quandt [26], Ryan et al. [218]). It generates a large number of routes and makes a selection by solving a partitioning problem.

On the other hand, Beasley [27] proposed a «route-first cluster-second» method for the VRP that works in reverse. It consists this time to generate a huge route like TSP (so it does not take into account the capacity of vehicles) [97]. This route is then splitted into feasible routes (respecting the capacity of vehicles).

For the VRP, many **metaheuristics** have been proposed in the literature and we present the best known. In 1993, a simulated annealing for the VRP was proposed by Osman. [184] It works well, but it was quickly overtaken by the tabu search methods. Cordeau and Laporte [52] in 2002 report ten more efficient tabu search at that time. The three most efficient are the Taillard's algorithm [239], Taburoute by Gendreau et al. [93] and Granular Tabu Search (GTS) by Toth and Vigo [246]. In the VRP literature, we can also mention other efficient methods as «Iterative Local Search» in Li et al. [156], «Adaptive Large Neighborhood Search» proposed by Hemmelmayr et al. [109] and genetic algorithm by Tasan and Gen [244].

2.4 Location Routing Problem (LRP)

When the number and position of the depots is not set in advance, a location problem is added to the routing problem (Wu et al [258], Prins al. [198]). This will select depot to use, each of which can induce an opening cost or cost of infrastructure installation. The depot can be further characterized by the number and types of vehicles (homogeneous or heterogeneous fleet), the storage capacity (Heterogeneous VRP, Prins [194]). If we require each customer to be directly linked to one depot, the LRP becomes a classic facility location problem. On the other hand, if the depot locations are already defined, the LRP reduces to a VRP (Nagy and Salhi [181]).

As mentioned above, in the case where the depots are not given in advance and their number and / or location, and also clients assignment must be defined, another level of logistical decision arises. In this case we have a location and assignment problem which appear under many forms :

- Set Covering problem.
- P-median problem
- Facility Location problem

— Location Routing problem

Using the classification of Magnanti [167], Laporte [147], we mention some basic models that can be interpreted similar with Location Problem.

2.4.1 Set Covering Problem (SCP)

The (SCP) is a common optimization problem which was applied to a huge variety of industrial applications, including scheduling, manufacturing, service planning, and as well as location problems. In this problem, let N_0 be a set of potential depots and N_c denotes a set of customers. We must open the depots such that every customer is covered by at least one open depot.

(i) The cover possibilities are defined by a binary matrix A with $a_{ih} = 1$ if the client i is covered by the depot h. (ii) The decision concerns to open a depot h and is defined by a boolean variable like $y_h = 1$ if the site h is open, with opening cost FD_h. The SCP problem can be formulated as follows :

$$\min \sum_{h \in N_0} FD_h y_h \tag{2.25}$$

Subject to :

$$\sum_{h \in N_0} a_{ih} y_h = 1 \qquad \forall i \in \mathrm{N}_c \tag{2.26}$$

$$y_h \in \{0, 1\} \qquad \forall h \in N_0 \tag{2.27}$$

The objective function (2.25) drives the search toward solutions at minimal cost. Constraints (2.26) ensure that all customers are covered by at least one depot. If constraint (2.26) is not satisfied, the solution is infeasible. If constraint (2.26) is relaxed, the objective function will drive the search toward an empty solution because the empty solution has the lowest cost (0). The variables of integrity constraints are specified by (2.27).

2.4.2 P-median problem

The basic P-median problem (PMP) model has remained almost unchanged during three decade. A binary linear programming formulation of the P-Median Problem can be defined on a weighted bipartite graph. In the PMP, the number of depots to be opened is known in advance and the goal is to choose the p depots by minimizing the average cost of routes. The problem is modeled by several decision variables such as : the cost $c(i, j)$ between nodes i and j and binary variables x_{ih} and y_h. (i) $x_{ih} = 1$ if the customer i is assigned to the depot h.

(ii) $y_h = 1$ if the site h is opened.

The p-median problem can be formulated as proposed by Hakimi [114] :

$$\min \sum_{h \in N_0} \sum_{i \in N_c} c(i,j) x_{ih} \tag{2.28}$$

Subject to :

$$\sum_{h \in N_0} y_h = h \tag{2.29}$$

$$\sum_{h \in N_0} x_{iy} = 1 \qquad \forall i \in \mathrm{N}_c \tag{2.30}$$

$$y_h \geq x_{iy} \qquad \forall i \in \mathrm{N}_c,\ \mathrm{h} \in \mathrm{N}_0 \tag{2.31}$$

$$x_{ih} \in \{0,1\} \qquad \forall h \in N_0, i \in \mathrm{N}_c \tag{2.32}$$

$$y_h \in \{0,1\} \qquad \forall h \in N_0 \tag{2.33}$$

The objective function is represented by (2.28). Constraints (2.29) determine the number of sites to be opened. Constraints (2.30) assign each client to exactly one facility. Constraints (2.31) prohibit the assignment of the clients to a closed facility. The integrity constraints are specified by (2.32) and (2.33).

2.4.3 Facility Location Problem

Before talking about Routing Problem in LRPs, several questions arise : how many satellites should be installed, where, and what customers are assigned to each satellite.

When in a Location Routing Problem, we discard the Routing Problem, a Facility Location Problems will then appear. The FLP concerns in choosing the best location for facilities from a given set of potential sites in order to minimize the total cost while satisfying customer demand. This total cost concerns the sum of the opening facilities costs and the costs of assigning customers to the facilities. A wide range of variants and extensions of this problem is proposed in the literature such as : Aikens [5], Brandeau and Chiu [23].

Firstly, we must define where the satellites can be created : wherever on the plan (continuous location) or on a finite number of potential sites (discrete location). In fact, the installation of a logistic site is very complex and can not be achieved anywhere. We must analyze the availability of land, the connection between these lots and transportation systems, the purchase and construction cost, etc.

The uncapacitated facility location problem (UFLP) involves locating an undetermined number of facilities to minimize the sum of the fixed setup costs and the variable costs of serving the demand from these facilities. In the uncapacitated facility location problem, each facility is assumed to have no limit on its capacity. In this case, each customer receives all its demand from exactly one facility. On other hand, in the capacitated facility location problem, each facility

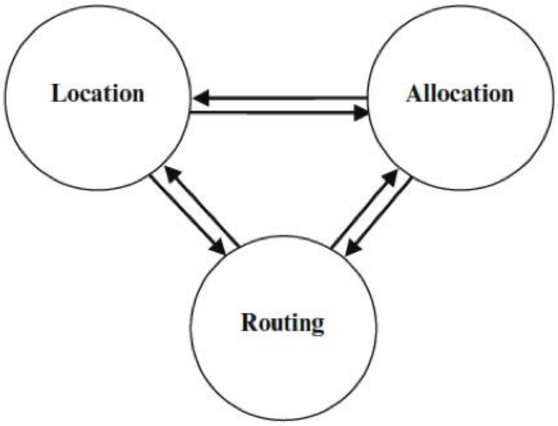

FIGURE 2.3 – LRP component relations

has a limit capacity. A special case of this problem, in which each customer receives supply from exactly one facility, is called the single-source capacitated facility location problem. It has been studied by several authors, including Barcelo and Casanovas [17], Sridharan [123], Klincewicz and Luss [129], Pirkul [46], Beasley [28], Holmberg et al. [123], Klose and Drexl [139].

2.4.4 LRP : definition and formulation

The LRP concerns the location of facilities, the assignment of clients to the selected facilities and the routes of the vehicles for serving all customer demands. Several constraints are considered in the literature such as facility and vehicle capacities. The objective is to minimize total cost including routing costs, vehicle fixed costs, and facility operating costs.

In this problem, the clients are served through routes and not by direct paths. Two decision levels are taken into account : (i) the tactical level to define the number and location of depots and (ii) operational decision related to the construction of routes. Several studies have shown the interest of considering the overall problem instead of separating the two levels (see [276, 198]).

Figure (2.3) is a simple representation of LRP sub-problems and their relationship.

Most LRP studies in the literature don't consider the capacity constraints (see Lin and Lei [143]). Here, we focus on solving the LRP with capacitated depots and routes. Prins et al. [196] gives the following formal mathematical model for the problem.

Let N be a set of nodes divided into a subset N_0 of m potential depot sites and a subset N_c of n customers. Each edge (i, j) has a trip cost $c(i, j)$. Each depot site $h \in N_0$ has a capacity Q_h and an opening cost FD_h. One customer $i \in N_c$ has a demand d_i which must be satisfied by a only one vehicle. A set V of identical vehicles with capacity CV is available. Each route must start and end up at the same depot, and its total charge must not exceed the vehicle capacity. The total client demand, assigned to a depot must not exceed the capacity of the depot. The objective is to minimize the total cost of the opened depots and the constructed routes. A binary variables $y_h = 1$ if and only if depot h is opened, $z_{ih} = 1$ if and only if customer i is assigned to depot h, and $x_{ij}^v = 1$ if and only if edge (i, j) is traversed from i to j in the route carried out by vehicle $v \in V$. We can see the problem formulation in the following binary integer program.

$$\min \sum_{h \in N_0} FD_h y_h + \sum_{i \in N} \sum_{j \in N} \sum_{v \in V} c(i,j) x_{ij}^v \tag{2.34}$$

Subject to :

$$\sum_{v \in V} \sum_{i \in N} x_{ij}^v = 1 \qquad \forall j \in \mathrm{N}_c \tag{2.35}$$

$$\sum_{i \in N} \sum_{j \in N_c} d_j x_{ij}^v \leq CV \qquad \forall v \in \mathrm{V} \tag{2.36}$$

$$\sum_{j \in N_c} d_j z_{jh} \leq \mathrm{Q}_h y_h \qquad \forall h \in \mathrm{N}_0 \tag{2.37}$$

$$\sum_{j \in N} x_{ij}^v - \sum_{j \in N} x_{ji}^v = 0 \qquad \forall i \in \mathrm{N}, v \in \mathrm{V} \tag{2.38}$$

$$\sum_{i \in N_0} \sum_{j \in N_c} x_{ij}^v \leq 1 \qquad \forall v \in \mathrm{V} \tag{2.39}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij}^v \leq \mid S \mid -1 \qquad \forall v \in \mathrm{V}, S \subseteq N_c \tag{2.40}$$

$$\sum_{i \in N_c} x_{hi}^v + \sum_{i \in N_c} x_{ij}^v \leq 1 + z_{jh} \qquad \forall v \in \mathrm{V}, h \in N_0, j \in N_c \tag{2.41}$$

$$x_{hj}^v \in \{0,1\} \qquad \forall h \in N_0, j \in \mathrm{N}_c, \mathrm{j} \in \mathrm{N} \tag{2.42}$$

$$y_h \in \{0,1\} \qquad \forall h \in N_0 \tag{2.43}$$

$$z_{jh} \in \{0,1\} \qquad \forall h \in N_0, j \in N_c \tag{2.44}$$

Constraint (2.34) is the objective function of model involving the opening costs of depots and the routing costs. Constraints (2.35) assure that each customer is satisfied by exactly one vehicle (route), and that each customer has only one predecessor node in the route. Constraints (2.36) ensure respecting the capacity of vehicle and constraints (2.37) handle available capacity for the depots. Constraints (2.38) are known as degree constraint and guarantees that the number of entering and leaving arcs to each node are equal. Constraints (2.39) are continuity constraint for each route. Constraints (2.40) eliminate the sub-tours. Constraints (2.41) ensure that each customer is assigned to only one depot if there is a route connecting these customers and depot. Finally, constraints (2.42), (2.43), and (2.44) define the binary variables.

Further mathematical formulations were presented in the literature and can be found in Prodhon thesis [201].

2.4.5 *LRP* variants

The Location-Routing Problem (LRP) and its variants are the models that allow dealing with location and routing decisions at the same time. Both of these problems can be considered as special cases of the LRP. These two problems belong to the class of the NP-hard problems. The LRP is likewise a NP-hard problem. Webb [259], Christofides and Eilon [60] proposed different models and approaches for the LRP.

We can mention many different types of *LRP* variants in the literature. Here, we try to address the most reviewed LRPs variants according to the classification of Drexl et al. [67]. When *LRP* consists unique planning period, we are dealing with a *static LRP*, against *dynamic* problem which refers to a multiple planning periods (Albareda-Sambola et al. [9]). The *dynamic LRPs* may initially don't have access to some data and they receive the required informations over time. When a multiple planning periods LRP considers complete informations at the beginning, the *periodic LRPs* appears (Prodhon [199]).

The works of Ahn et al. [148] and Harks et al. [8] are the only two papers that reviewed the *Prize-Collecting LRPs (PCLRPs)*. In PCLRPs, some clients may not be assigned to any route, instead a client penalty is token in account. The objective function is to minimize the sum of fixed facility opening, routing cost and penalty value.

When a customer can be served more than once via more than one route to satisfy the demand, we deal with *Split Delivery LRPs*. The researchers are quite active on VRP with split delivery but not on SDLRPs (Harks et al. [8]).

In the LRPs literature, *Inventory LRPs* add the inventory management decisions as an option at the depot. For example, the amount of good storage during how long may be considered as a management decision of ILRPs. Many works can be mentionned here, that introduce this kind of problem (Zhang et al. [272], Ahmadi-Javid and Azad [6]). Many LRP models, reviews, approaches and applications could be found in Prins et al. [198, 197, 196], Belenguer et al. [29] Duhamel et al. [77], those of Wu et al. [258], Min et al. [173], Balakrishnan [18], Nagy and Salhi [179], Derbel et al. [76], Lopesa et al. [144], Ardjmand et al. [13].

The most recent LRPs classification surveys are available in the literature to categorize LRPs by Prodhon and Prins [192] and Drexl et al. [67],

In a multi-echelon system, beside the central depot (or main, large capacity), we have the smaller depots, called "satellites" that serve the points of transfer or intermediate nodes with final customers. These levels own their proper vehicles in order to carry out the required transportation in relevant level. In the next chapter we detailed some other LRP variants that come close to our studied problems : Multi-Commodity and Pick-up and Delivery.

2.4.6 *LRP* solution methods

For the LRPs, quite similar by to the VRPs, the exact methods generally act on the basis of mathematical programming. Because of the complexity of location-routing they can only tackle relatively small instances. General location-routing instances with up to 40 potential depot locations or 80 customers could be solved to optimality (Laporte et al. [150] ; Laporte and Norbert [140]).

A realistic particular version of Location-Routing Problem (LRPs) was proposed by Lin and Kwok [153]. In this work, multiple uses of vehicle are allowed and decisions on location of depots, vehicle routing and assignment each route to vehicle are considered simultaneously. A Tabu Search and Simulated Annealing approaches are developed and tested on random-generated instances and real data.

Duhamel et al. [78] address the capacitated location-routing problem (CLRP). The proposed solution method is a greedy randomized adaptive search procedure (GRASP), calling an evolutionary local search (ELS) and searching within two solution spaces : giant tours without trip delimiters and true CLRP solutions.

Wasner and Zapfel [261] present a real LRP application to postal and freight service provider for a city in Austria. The objective is to define the location and number of depots and facilities. A Local Search method was proposed in order to solve the proposed problem.

Wu et al. [260] merged Simulated Annealing with TS to solve the heterogeneous fleet case of LRP.

A reliable capacitated location-routing problem in which depots are randomly disrupted is proposed by Zhang et al. [271]. Customers whose depots fail must be reinserted into the routes of surviving depots. They presented a based mixed-integer programming model and also a metaheuristic algorithm that is based on a maximum-likelihood sampling method, route-reallocation improvement, two-stage neighborhood search and simulated annealing.

Zarandi et al. [270] studied the fuzzy-based version for Capacitated Location-Routing Problem (CLRP). The authors consider a model with multiple potential depots in which the location of depots should be defined and in the second phase the approach deals with routing problem. Fuzzy travel times between nodes and time window to visit each customer are taken in account. To solve the problem they proposed a Simulated Annealing (SA) procedure.

Rodriguez-Martin et al. [212] consider two-objective for mathematical programming in the multi-depot location-routing problem. They formalized the proposed problem and then solved with both a Scatter Tabu Search approach and the well-known NSGA- II (Non-dominated Sorting Genetic Algorithm II).

In Marinakis [172] a new version of the particle swarm optimization algorithm suitable for discrete optimization problems is presented and applied for the solution of the capacitated location routing problem and for the solution of a new formulation of the location routing problem with stochastic demands. A global and local combinatorial expanding neighborhood topology particle swarm optimization, was developed. The algorithm was compared with a number of different implementations of metaheuristics.

Yu and Li [265] introduce the open location-routing problem that is a variant of the capacitated location-routing problem (CLRP). The objective of OLRP is to minimize the total cost, consisting of facility operation costs, vehicle fixed costs, and traveling costs. They propose a simulated annealing (SA)-based heuristic. They tested the proposed method on a benchmark up to 318 customers and 4 potential depots.

Ko et al. [136] introduce the fleet size and mix location-routing problem with time windows (FSMLRPTW) which extends the location-routing problem by considering a heterogeneous fleet and time windows. They proposed a mixed integer programming formulation, a family of valid inequalities and they develop a powerful hybrid evolutionary search algorithm (HESA) to solve the problem. The hybrid evolutionary search algorithm combines several metaheuristics and offers a number of new advanced efficient procedures to handle heterogeneous fleet dimensioning and location decisions.

A two-phase hybrid heuristic algorithm to solve the capacitated location-routing problem is proposed by Willmer Escobar et al. [254]. In the proposed hybrid heuristic algorithm, after a Construction phase (first phase), a modified granular tabu search, with different diversification strategies, is applied during the Improvement phase (second phase). In addition, a random perturbation procedure is considered to avoid that the algorithm remains in a local optimum for a given number of iterations.

Ting et al. [243] proposed a multiple ant colony optimization algorithm to solve the LRP with capacity constraints.

2.5 Conclusion

This chapter has presented an overview of basic concepts related to the VRP and LRP. We have first recalled the basic concepts to model the transportation problems. A brief reminder of classical resolution methods, exacts and heuristics for combinatorial optimization problems, basic logistic problem for LRP and VRP are defined and basic mathematical formulations were also presented.

In the following chapter, we will focus on VRP and LRP variants that come close to our problem. In our problem, we are interested to two versions of location-routing problem including three constraints that have never been considered simultaneously in the literature. These constraints include 1) multi-product aspect, 2) pickup / delivery, and 3) intermediates multi-depot aspect. The next chapter will focus on each mentioned aspect and describe the related variants of LRP and/or VRP studied in the literature.

Chapitre 3

State of the Art

Contents

3.1 Introduction

In the previous chapter we have described the basic models for *Vehicle Routing Problems (VRP)* and *Location-Routing Problem (LRP)* and some variants of the two problems. The most known methods to solve these kind of problems have been also presented. In the following, we will focus on variants of LRP (and VRP as a special case) that come closest to our problems, and more particularly the integration of multi-echelon, pickup and delivery, multi-depot and multi-commodity aspects.

Section 3.2 provides a bibliography on *Two-Echelon distribution system*. The *VRPs and LRPs with Pick-up and Delivery* studied in the literature, are gathered in section 3.3. A state of the art with the regard to the research on *Multi-Commodity LRPs and VRPs* in section 3.4 and *Intermediate Depots (or facilities)* aspect are reported in section 3.5. More general LRPs problem are gathered in section 3.6 and we conclude the chapter in the last section.

3.2 Location routing and vehicle Routing problems with multi-level distribution system constraints

In a multi-level system, beside the central depot (or main, large capacity), we have the smaller depots, called "satellites" that serve the points of transfer or intermediate nodes with final customers.

Two-level distribution systems are a particular case of N-echelon systems in which the network consists of two-echelons. After leaving its origin, the goods are first delivered to an intermediate facility where some inter-operation such as : storage, merging or transshipment take place. The goods are then transported from the intermediate facility (satellite) to their destination. The flow of freight in one echelon should be coordinated with that in the other echelon. In the following, we describe the most studied variants of VRP and LRP taking into account a multi-level distribution system.

3.2.1 Two-Echelon Vehicle Routing Problem

The 2E-VRP is based on the classical VRP with numerous constraints, such as : a single depot and a fixed number of satellites with limited capacity that are already constructed (opened). The customer demand is fixed and known. A homogeneous fleet serves the demand at every level of routing, and we still do not consider additional options such as time windows, etc.

The first level concerns the routes from the main depot to the satellites, the second level concern routes from satellites to customers. The visited satellites by the first level routes must receive enough goods to serve customers.

When we talk about the Two-Echelon Vehicle Routing Problems (2E-VRPs), we refer to the fact that the problem definition involves only tactical decisions, and the routing is present at both levels. In 2E-VRPs, the set of depots and the set of satellites are given, and it leads to no cost when we use the depot and satellite. As a consequence, routing problems with two-level distribution systems cannot be decomposed into two sub-problems and then solved separately.We define as two-echelon routing problems a class of problems that study how to optimally route in two-echelon transport system.

Soysal et al. [228] propose a comprehensive MILP formulation for a time-dependent two-echelon capacitated vehicle routing problem (2E-CVRP). A case study in a supermarket chain operating in the Netherlands shows the applicability of the model to a real-life problem. They develop several versions of the model, each differing with respect to the objective function. The results suggest that an environmentally friendly solution is obtained from the use of a two-echelon distribution system, whereas a single-echelon distribution system provides the least-cost solution.

Crainic et al. [56] study a variant of the 2E-CVRP including various additional features. They consider that the demand of each customer has an issue such that the demands should be picked up by the vehicle in first level and should be delivered to the customer by using a two-echelon logistic system. An optimal solution can be provided for small instances of 20 clients. The authors in another work introduced a lower bound computed by adding two lower bounds ([62]).

Perboli et al. [191] consider the Two-Echelon Vehicle Routing Problem. They present valid inequalities based on the TSP and CVRP, the network flow formulation, and the connectivity of the transportation system graph.

A hybrid heuristic ant colony algorithm is proposed by Meihua et al. [163] and applied to solving the two-echelon vehicle routing problem.

The 2E-VRP can be also considered as an extension of the vehicle routing problem with multi-depot (VRPMD), where trips must be added to supply the depots from a main platform. In 2007, Crevier et al. [59] proposed another extension of the VRPMD, the vehicle routing problem with satellite facilities (VRPSF) : each vehicle must still start and end its route at the same depot, but it may do some pick-up at other depots.

Another work that we can mention is the ship-road multi-modal distribution system (e.g., see [124]) where the cargo moves from the supplier to a satellite by boat (i.e., the first step) and then charged on a truck and be delivered to its final destination (i.e., the second level).

The truck and trailer routing problem (TTRP) is also a variant of the $2E-VRP$. In the $TTRP$, customers are served by a set of trailers and trucks. There are a limited number of trucks and trailers to satisfy the request of a group of clients. In this case, a trailer is attached to the

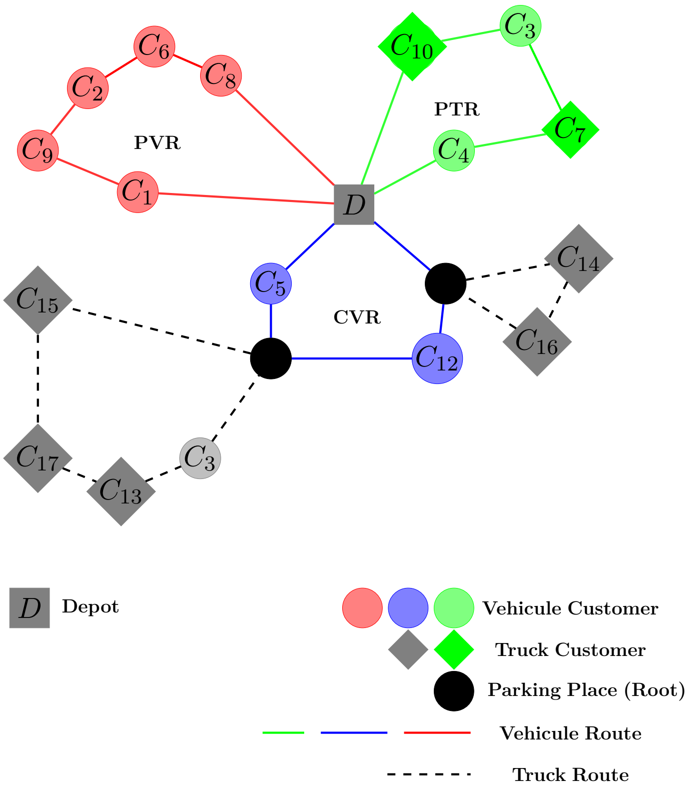

FIGURE 3.1 – A TTRP illustration of different type of vehicle routes.

truck and is pulled by truck. Dependent to TTRP constraints, the truck can leave its trailer wherever it needs to continue the route without trailer.

Some customers in the TTRP could receive their demands from only a single truck.The customers are divided into truck-customers and vehicle-customers. The truck-customers are accessible only by the truck without the trailer. Instead, the vehicle-customers has no accessibility restrictions, and their locations can be used to park the trailer and truck in order to serve the customers (see figure 3.1).

In the literature of the TTRP we find three types of route : (i) the vehicle routes carried by a truck (without trailer) visiting the two types of clients, (ii) the truck routes performed by a complete vehicle (truck with trailer), serving only vehicle-customer and (iii) vehicle routing with sub-routes performed by a truck with trailer and serving then the Truck-customers and vehicle-customers.

Semet and Taillard [230] solved a real problem of 2E-VRP, involving trucks with trailers to deliver groceries in Switzerland. This is an interesting application to deliver goods to areas where there is limited accessibility for example in the mountains. A complete vehicle can visit the villages, accessible by routes. To reach hamlets in the valleys, the driver has to leave the trailer in parking and handle the services of the valleys with the forward side (truck). Villegas et al. [250]

also solved the $TTRP$ by a method called GRASP with evolutionary path-relinking. Tan et al. [237] reviewed a real TTRP which is considered as a variant of the Multi Depot Vehicle Routing Problem (MDVRP) in the literature. The objective is to seek a routes scheduling to serve the demands with minimum traveling distance and number of trucks. The authors consider a Time Windows constraint. The proposed solving method is based on an evolutionary algorithm and Local search method.

A two-phase method to solve $TTRP$ was proposed by Scheuerer [236]. The proposed method in the first phase, an initial solution is obtained by a construction heuristics and then this solution was improved by a Tabu Search method.

In 2008, Gonzalez-Feliu [102] proposed a MIP formulation and a column generation approach for 2E-VRP problem. He succeeded to find the optimum for small instances about 20 clients. For instance with 50 clients, treated by this model, the distance between the lower and the upper bound remains high.

The best heuristic for the 2E-CVRP, to the best of our knowledge is the $ALNS$ (Adaptive Large Neighborhood Search) proposed by Hemmelmayr et al. [109], while the best exact algorithm (branch-and-cut algorithm) was introduced in Baldacci et al. [25].

For further work, we can find a review of two-echelon good transportation optimization problems in Gonzalez-Feliu [106] that tries to identify the main concept. We can also find the survey of Laporte [151] that provides a summary of the largest studies for VRP and Cuda et al. [65] for 2E-VRP.

3.2.2 Two-Echelon Location-Routing Problem

In the context of logistic system with two-echelon, the distribution system consists of first-echelon facilities (depots), second-echelon intermediate facilities (processing centers), and third level including a set of customers. The two-echelon location routing problem (2E-LRP) aims at 1) locating the subsets of the first and the second echelon facilities, 2) constructing the primary (first echelon vehicle routes) and the secondary routes (second echelon vehicle routes). Each primary route starts from an opened depot, visits a subset of processing centers and returns to the starting depot. Each secondary route starts from an opened processing center, visits customers once and returns to the starting processing center. Two sets of homogenous fleet of vehicles are used, one fleet for each level. The fleet of second echelon uses vehicles with capacity smaller than those used in the first level.

As we mentioned before, Two-echelon vehicle routing problem can be reviewed as a special case of 2E-LRP in which all satellites are already opened and induce no cost.

To the best of our knowledge, the first studies on the 2E-LRP are initiated by Jacobsen and Madsen [127], Madsen [170]. These studies consider real applications in a two-level newspaper distribution with 4500 customers with transfer facility, printing center and customers.They consider three types of decision :(i) Decision on location and number of transfer facilities. (ii) Computing the routes from printing centers to transfer facilities and (iii) Calculating the routes from transfer facilities to customers.

The most studied variant of 2E-LRP considers that the distribution system is composed of one of the following variants : (i) more than one depot at the first level and a set of potential processing centers, either without capacity constraints (Lin and Lei [143]), or with capacity constraints (Boccia et al. [38], Contardo et al. [41], Schwengerer et al. [231]), (ii) one fixed depot and a set of potential processing centers with fixed capacities and opening costs (Nguyen et al. [180], Nguyen et al. [178]), (iii) one or more depots and a set of potential processing centers, without capacity constraints and without opening costs, such as the truck and trailer routing problems (Villegas et al. [251]).

Other problems derived from 2E-LRP are considered in the literature. For example, (i) Two-Echelon vehicle routing problem (2E-VRP) when there is no fixed cost for using depots and processing centers, (Crainic et al. [42], and Cuda et al. [65]), (ii) Two-Echelon capacitated facility location problem, in which the clients are directly linked to the facility, (Li et al. [142]).

Several exact heuristics and metaheurisitcs approaches are proposed in the literature to solve the 2E-LRP. In Boccia et al. [39] various mixed integer models for the 2E-LRP are proposed. Contardo et al. [41] and Rodriguez-Martin et al. [216] investigated a branch and cut approach. Metaheuristics such as, genetic algorithm (Lin and Lei [143], Karaoglan and Altiparmak [137]), GRASP with path relinking (Nguyen et al. [180]), multi-start Iterative Local Search (Nguyen et al. [180]), Tabu Search (Boccia et al. [38]), Variable Neighborhood Search - VNS (Schwengerer et al. [231]), Tabu Neighborhood Search (Escobar et al.[262]), multi-start Simulated Annealing (Lim [263]) have been proposed in the literature. A comparison of the performance of these metaheuristics was given in Prodhon and Prins [192].

Govindan et al. [107] introduce a two-echelon location-routing problem with time-windows (2E-LRPTW) in a perishable food supply chain network design and optimizing economical. The proposed method includes a novel multi-objective hybrid approach, a hybrid of two known multi-objective algorithms and adapted multi-objective variable neighborhood search.

A recent survey on LRP and 2E-LRP can be found in Prodhon and Prins [192] and Drexl et al. [67].

3.2.3 Two-Echelon Facility Location Problem

An extension of the facility location problem, with two-echelons of facilities is called «two-echelon uncapacitated facility location problem». In this problem, the deliveries are made from the first-echelon facilities (such as plants or depots) to customers via the second-echelon facilities (such as warehouses). The objective is to determine the number and location of facilities in each echelon, the flow of products between the facilities in different echelons and the assigning of the customers to facilities in the second echelon.

Tragantalerngsak et al. [248] studied the two-echelon Facility Location Problem (2E-FLP). In the proposed distribution system, the goods are delivered from the depots to the satellites and from the satellites to clients. They study a single-source and then multi-source case. In the extension of the single-source, capacitated facility location problem, there exist two-echelons of facilities. The second-echelon facility (satelite) has a limited capacity and can be supplied (delivery/collecte) by only one first-echelon facility (depot), and each client is exactly delivered/collected by a single satellite. In the multi-source case, the request of each satellite/customer can be satisfied by more depots/satellites. The number and location of depots and satellites and also the assignment of the satellites to customers are performed simultaneously. The goal is to minimize both the opening cost of depots and satellites and customers assignment. The authors propose six heuristics, based on a Lagrangian relaxation.

Recently, Gendron and Semet [96] studied a problem of location-distribution with three distribution levels and thus four layers of nodes (hubs, depots, satellites, clients). They have proposed a mathematical model and relaxation methods and search methods with variable neighborhoods.

Kaufman et al. [131] and Ro and Tcha [217] described independently branch and bound algorithms to solve 2E-FLP. Gao and Robinson [91] considered a location problem without capacity constraint at both levels and applied a based dual-based branch and bound method to solve the problem.

Wu et al. [256] propose an approximation algorithm for the two-level facility location problem (2-LFLP) by exploring the approximation oracle concept. They approved that their model can be easily adapted to several variants of the 2-LFLP, including models with stochastic scenario, dynamically arrived demands, and linear facility cost.

3.2.4 Multi-echelon Location-Routing Problem

Multi-Echelon LRPs is another type of LRP that customers are not served directly from a central depot but via level N in a N-level distribution system that contains $N + 1$ levels. Each level $n \in \{1, ..., N\}$ receive the demands from level $n - 1$, and these levels own their proper vehicles in order to carry out the required transportation in relevant level (Gonzalez Feliu et al. [102, 103], Perboli et al.[200]).

In recent years, we are interested in multi-level transportation systems. For a state of the art on the applications of these problems, see the articles Crainic et al. [53] and Taniguchi and Thompson [242], which have proposed a multi-level distribution systems as a tool to reduce urban congestion.

A models for multilevel systems of city distribution was proposed by Crainic et al. [56]. These authors present a multi-level system for the transport of goods in the town. Urban distribution centers play a key role in the majority of these systems.

Most recently, Gonzalez-Feliu [103] presented the concepts and challenges, including a general model for N-Echelon Location Routing (*NE-LRP*).

3.3 Location routing and vehicle routing problem with pickups and deliveries constraints

Pickup and Delivery LRPs (VRPs) appears when two kind of nodes exist in the distribution system : 1) delivery nodes who expect deliveries, and 2) pickup nodes that have available supply to be picked-up. These problems have a wide applications and very large literature concerning many variants of transportation problem with pickups and deliveries. In the following we will focus on VRP with pickups and deliveries and LRP with pickups and deliveries.

3.3.1 Vehicle routing problems with pickups and deliveries

The class we denote VRPPD refers to problems where goods are transported from pickup to delivery points. It can be further divided into two subclasses. The first subclass refers to situations where pickup and delivery locations are unpaired. A homogeneous good are considered. Each unit picked up can be used to fulfill the demand of any delivery customer. In the literature mostly the single vehicle case is tackled, denoted as capacitated traveling salesman problem with pickups and deliveries (Anily and Bramel [3]), One-Commodity Pickup and Delivery Traveling Salesman Problem (1-PDTSP) (Hernandez-Parez and Salazar-Gonzalez [116]), and traveling salesman problem with pickup and delivery (Hernandez-Perez and Salazar-Gonzalez [118]). Since also a multi vehicle application has been reported in the literature (see Dror et al.

The second VRPPD subclass comprises the classical Pickup and Delivery Problem (PDP) and the Dial-A-Ride Problem (DARP). Both types consider transportation requests, each associated with an origin and a destination, resulting in paired pickup and delivery points. The PDP deals with the transportation of goods while the DARP deals with passenger transportation. This difference is usually expressed in terms of additional constraints or objectives that explicitly take the quality of service into account. The single vehicle variant of the PDP has also been referred as pickup delivery traveling salesman problem (Kalantari et al. [132]) and the multi vehicle case as vehicle routing problem with pickup and delivery (Derigs and Dohmer [69]). However, a majority of the work published denotes this problem class as Pickup and Delivery Problem (PDP) (see e. g. Dumas et al. [70]).

The dynamic version is also referred to as demand responsive transport (compare e.g Mageean and Nelson [168]).We denote the single vehicle case of the PDP as SPDP, the single vehicle case of the DARP as SDARP.

The only exact method proposed for the problem at hand was introduced in Hernandez-Perez and Salazar-Gonzalez ([116, 118]). It is a branch and cut algorithm. To speed up the algorithm construction, an improvement heuristics are applied. They are used to obtain feasible solutions from the solution of the LP relaxation at the current node. The construction heuristic is an adaptation of the nearest insertion algorithm. It is improved by 2-opt and 3-opt exchanges (Lin [152]). The test instances solved are adaptations of the ones used in Mosheiov [171] and Gendreau et al. [99], containing up to 75 customers.

Dror et al. [68] propose a heuristic algorithm for the application of the multi vehicle PDVRP to the redistribution of self-service cars. It is related to Dijkstra's algorithm. Also other solution approaches are briefly discussed.

Li et al. [156] propose a metaheuristic method based on iterated local search for the multi-depot vehicle routing problem with simultaneous deliveries and pickups. To strengthen the search, an adaptive neighborhood selection mechanism is embedded into the improvement steps and the perturbation steps of iterated local search. They propose also a new perturbation operators.

Coelho et al. [57] introduce a Single Vehicle Routing Problem with Deliveries and Selective Pickups. In this model, deliveries have to be made to a set of customers and there are also pickup goods to be collected and customers return goods back to the depot such as in postal logistics. In order to solve the problem at hand we propose a hybrid heuristic algorithm, inspired on the metaheuristic General Variable Neighborhood Search combined with an initial solution generation by exact methods. Zhang et al. [273] propose a stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries. They develop a new scatter search approach by incorporating a new chance constrained programming method. A genetic algorithm is also developed and used as a reference for performance comparison.

A variant of the vehicle routing problem in which customers require simultaneous pickup and delivery of goods during specific individual time windows is studied by Wang [257]. A general mixed integer programming model is employed to minimize the routing cost. The authors propose a parallel Simulated Annealing algorithm and an insertion-based heuristic is developed and applied to solve this NP-hard optimization problem.

A survey on different solution methods for VRPPD can be found in Berbeglia et al. [34].

3.3.2 Location-Routing Problems with Pickups and Deliveries

LRP with Pickups and Deliveries occures when a single customer can require at the same time both pickup and delivery demands, and in this case for a customer, pickup and delivery, must be done in the same visit. It is called «Location Routing Problem with Simultaneous Pickup and Delivery (LRPSPD)». This model extends the results of Berbeglia et al. [34] and Parragh et al. [190] on the vehicle routing problem with pickup and delivery.

Yu et al. [263] propose a location routing problem with simultaneous pickup and delivery (LRPSPD). Their study proposes a multi-start simulated annealing algorithm which incorporates multi-start hill climbing strategy into simulated annealing framework. The multi-start simulated annealing algorithm is tested on 360 benchmark instances to verify its performance.

Karaoglan et al. [138] introduce also a LRP with simultaneous pickup and delivery. They propose two polynomial-size mixed integer linear programming formulations for the problem and a family of valid inequalities to strengthen the formulations. While the first formulation is a node-based formulation, the second one is a flow-based formulation. They propose also propose a two-phase heuristic approach based on simulated annealing to solve the large-size LRPSPD and two initialization heuristics to generate an initial solution for the simulated annealing.

In another work Karaoglan et al. [135] develop an effective branch-and-cut algorithm for solving the LRPSPD. The proposed algorithm implements several valid inequalities adapted from the literature for the problem and a local search based on simulated annealing algorithm to obtain upper bounds.

A Capacitated Location-Routing Problem with Mixed Backhauls is considered by Karaoglan [137]. The CLRPMB is defined as finding locations of the depots and designing vehicle routes in such a way that pickup and delivery demands of each customer must be performed with the same vehicle and the overall cost is minimized. They propose a memetic algorithm to solve the problem.

Many-to-many LRP is a particular case of LRP that we will detail in the following.

3.3.2.0.1 Many-to-many LRPs

Many-to-many LRPs (MMLRP) are pick-up and delivery problems where the goal of planning is locating a network of intermediate facilities for the transportation of goods. The pick-up and deliveries are carried out on local multi-stop routes beginning and ending to a hub ; Inter-hub transport is generally straightforward. These problems appear, for example, in postal or package distribution applications.

An MMLRP application was proposed by Cetiner and Sepil Sural [64] in postal logistics and performs the following hypotheses : For each pair of clients, a required flow of goods in both directions is defined. The Hubs are considered as uncapacitated facilities, and there is no loading constraints for the vehicles, instead, a maximum length for the routes is specified. The pickup and delivery at the customer are performed simultaneously. Each client can be assigned to more than a hub, i.e, may send its outgoing products to more hubs and receive its incoming goods from several hubs. The objective is hierarchical : Firstly, minimize the number of vehicles and secondly minimize the total transportation costs. There are no hub or vehicle fixed costs, but a number pre-specified p of hubs must be used. To solve the problem, an iterative embedded matheuristic in two stages is proposed. In the first step, a multiple allocation p-hub median problem is solved, in the second stage, multiple TSPs with length route constraint are addressed for each hub, opened in the first step. For the two problems, exact solution is calculated using MIP models of literature and a standard MIP solver. After each iteration, the distance between the customers and the hubs in the problem of the first step are being updated using the results of the second step. This process is repeated until the solution stops changing between iterations. Computational experiments were conducted with seven modified benchmark instances generated from four different sources belonging to the facility location literature and a real instance, with 81 clients. The obtained results with this procedure are compared to those achieved when only one iteration of the procedure is carried out, and a reported saving is more than 20%.

De Camargo et al. [66] also investigate a MMLRP. The assumptions underlying their problems are similar to those of Cetiner et al. [64], but Camargo et al. [66] ensures that (i) each client is assigned to exactly one hub (single assignment), so each customer is visited exactly once and (ii) the location of each customer is considered as a potential location hub. The objective concerns minimizing the sum of the fixed costs of facility hubs, handling fee incurred to transfer goods at hubs, the fixed costs of allocating vehicles in order to open the hubs and the costs that are based on distance by vehicle routes and inter-hub transports. The authors propose a based MIP model where is founded an arc-variable model, combined the model of Skorin-Kapov et al. [227] for the single assigning hub location problem and the model of Claus [45] for the TSP problem. These models are used due to the power of the linear relaxation.

In the computational experiments, test instances with 10 to 100 customers are generated from a group of data for hub location problems. The authors reviewed and compared their proposed method with a direct solution of the standard MIP solver for instances with up to 30 clients and achieve an acceleration factor of 2 to 100. The biggest example solved with optimality for Benders decomposition algorithm contains 100 clients. As an instance with 100 customers that corresponds to 10000 products and has over four million integer variables.

Rodriguez-Martin et al. [210] consider a MMLRP in which each customer site in the network is a potential hub and exactly p hubs should be constructed. No fixed costs exist for the construction of a site, and the inter-hub transportations are direct. At most q customers can be attributed to a hub, and at each hub, there is just one vehicle to visit clients that are assigned to one multiple stop path. A MIP model based on an undirected graph was presented by the authors with five types of decision variables. An exact branch-and-cut algorithm is introduced to solve the model. To enhance the LP relaxation, a number of valid inequalities and the evidence of their validity are introduced. These additional valid inequalities and sub-tour eliminator constraints are separated. The routines of used separation are described in detail. Computational experiments are performed with modified instances from the literature of the hub location. The algorithm can solve instances with up to 50 sites to optimality within two hours on a PC.

3.4 Location Routing and Vehicle Routing problem with multi-commodity constraints

The majority of works in the LRP and the VRP literature consider a situation with a single product. However in the real application multi-commodity, also called Multi-Product, condition is more realistic. To the best of our knowledge in the literature, the works of Bowerman [19], Burks [21], Gunnarsson et al. [92], Yi and Ozdamar [264] and Sajjadi et al. [232] deal with a multi-commodity problem.

The term «commodity» in the routing problem is also used in other similar telecommunication issues or in the network design. Since each product is moved along a single path from one source to its destination, this product is sometimes recognized as an object. In this case, we keep the assumption that all objects to transfer are different, which justifies the term «multi-commodity» to be called these kind of problems. The Multi-commodity problems have different requirements according to variants, characteristics and optimization features.

3.4.1 Multi Commodity VRP

Zhang and Chen [275] propose an optimization model in frozen food delivery industry that manages the delivery of a variety of products. In this scenario, a set of customers make requests for a variety of frozen foods which are being loaded together. The delivery cost includes the transportation cost, the cost of refrigeration, the penalty cost and cargo damage cost based on the characteristics of different frozen food products. They develop a Genetic Algorithm method to solve the model.

A Multi-Commodity Multi-Trip Vehicle Routing Problem with Time Windows is proposed by Cattaruzza et al. [58] in which the products belonging to incompatible commodities. Two commodities are incompatible if they cannot be transported together into the same vehicle. The objective is to minimize the number of used vehicles. An Iterated Local Search that outperforms the previous algorithm designed for the problem by the authors.

Moin et al. [175] consider a many-to-one distribution network consisting of an assembly plant and many distinct suppliers where each supplies a distinct product. A finite horizon, multi-periods, multi-suppliers and multi-products where a fleet of capacitated homogeneous vehicles, housed at a depot, transport products from the suppliers to meet the demand specified by the assembly plant in each period is considered. A mathematical formulation of the problem is given and CPLEX 9.1 is run for a finite amount of time to obtain lower and upper bounds. A hybrid genetic algorithm, which is based on the allocation first route second strategy and which considers both the inventory and the transportation costs, is developed.

Psaraftis [193] also propose a dynamic programming solutions for a multi-commodity, capacitated pickup and delivery problem in which solution approaches are developed for the single-vehicle and two-vehicle cases.

An binary MIP model regard to opening decisions for hubs, binary arc variables for the routing is presented by Rieck et al. [211]. They propose several types of valid inequalities as well. A multi-commodity fixed-charge network flow problem is solved to determine the inter-hub routes.

3.4.2 Multi Commodity LRP

Hamidi et al. [119, 121] reviewed a three-echelon LRP with multiple commodities, capacitated facilities on levels 1, 2 and 3. An opening cost for facilities exist on levels 2 and 3. The authors consider a limited-number of homogeneous vehicles with a limited travel time and a limited capacity. The facilities, installed on both levels can serve the customers. The vehicle can satisfy only the customers demands. The proposed heuristic uses Greedy Randomized Adaptive Search Procedure and two probabilistic tabu search strategies of intensification and diversification to tackle the problem.

3.5 Location Routing and Vehicle Routing Problem with multi-depot constraint

3.5.1 Multiple Depots Vehicle Routing Problem

Multiple Depots Vehicle Routing Problem (MDVRP) is a variant of the vehicle routing in which more than one facility-depot is presented in model.

It is often hard adopting a VRP exact algorithm for solving the Multiple Depots Vehicle Routing Problem (MDVRP). A splitting method was proposed by Ray et al. [202] and Li et al. [159] and proposed an MIP for a version with shared depots and time constraints. A fuzzy-based approach and mathematical model was proposed by Adelzadeh et al. [10].

A MDVRP with heterogeneous fleet is proposed by Lin and Kwok [154]. They address two versions of a savings and a nearest neighbor heuristic to solve the problem. Schittekat and

Sorensen [235] propose a MDVRP with capacitated facilities and heterogeneous vehicles. In the last case, vehicle capacity, time windows and maximum travel time for the vehicle are considered as the constraints.

The multi-depot vehicle-routing problem is also reviewed by Baldacci, et al. [36]. They divide a LRP into a limited set of MDVRP and develop bounding approaches. The proposed methods are based on dynamic programming.

Crainic et al. [43] proposed a constructive heuristic to solve a MDVRP. They apply then three Local Search (LS) improvement methods : (i) splitting the route in two, (ii) moving a customer from one route to another one, and (iii) swapping two customers between two different routes.

A metaheuristic based on iterated local search for the multi-depot vehicle routing problem with simultaneous deliveries and pickups is proposed by Li et al. [156].

Kachitvichyanukul et al. [270] studied a generalized multi-depot vehicle routing problem with multiple pickup and delivery requests. Two solution representations for solving problem are proposed. The representations are used in conjunction with GLNPSO, a variant of PSO with multiple social learning terms.

The work of Contardo and Martinelli [55] present an exact algorithm for the multi-depot vehicle routing problem (MDVRP) under capacity and route length constraints. The MDVRP is formulated using a vehicle-flow and a set-partitioning formulation, both of which are exploited at different stages of the algorithm. Several classes of valid inequalities are added to strengthen both formulations, including a new family of valid inequalities used to forbid cycles of an arbitrary length.

Another multi-depot routing with heterogeneous vehicles is proposed by Salhi et al. [221]. A mathematical formulation is given and lower as well as upper bounds are produced using a three hour execution time of CPLEX. An efficient implementation of variable neighborhood search that incorporates new features in addition to the adaptation of several existing neighborhoods and local search operators is proposed. The proposed algorithm is highly competitive as it produces 23 new best results when tested on the 26 data instances published in the literature.

Another version of Multi-Depot Vehicle Routing Problem is introduced by Bernardes de Oliveira et al. [24]. They propose a cooperative coevolutionary algorithm to minimize the total route cost of the MDVRP. Their work presents a problem decomposition approach for the MDVRP in which each subproblem becomes a single depot VRP and evolves independently in its domain space. Customers are distributed among the depots based on their distance from the depots and their distance from their closest neighbor.

3.5.2 VRP with Intermediate Facility

Another problem related to VRP is the Multiple Depots Vehicle Routing Problem with Satellite Facilities (MDVRP-SF). Some authors called this problem «Multi Depot Vehicle Routing Problem with Inter-Depot Routes (MDVRPI). The *MDVRP-SF* has not received much attention from researchers. This is a multi-depot VRP where vehicles are not required to return to original depot. They may end up at another depot and some stops at a satellite to allow them to pick-up (collect). The idea is an interesting principle to deal with the situations in which customer demand is not precisely known or when a «classic» route does not fulfill a work period of the driver and a return to the main depot to complete the schedule is too costly. Thus, the duration of the route is limited by a total time of delivery / collection (pick-up).

Compared to the VRP, the MDVRP-SF deals with a single route for a set of clients which the total demand exceeds the capacity of a single vehicle. The problem has two types of routes : routes with departing from and returning to the same depot without stopping, and routes with inter-depot links.

A simple version of the MDVRP-SF is presented by Jordan and Burns [126] which assume that customer demands are all equal and that the routes are of the back-and-forth between two depots. These authors convert the problem into a problem of matching, which is solved by a greedy algorithm. Angelelli and Speranza [11] developed a heuristic for a periodic vehicle routing problem (PVRP) in which the discharge (delivery) is allowed to satellites (waste collection problem). Their algorithm is based on Tabu Search of Cordeau et al. [52] for the PVRP. A variation with time windows is considered by Servilla and Blas [234], using neural networks and an approach with ant colonies. Several similar applications are encountered in the context where the route of a vehicle can be composed of multiple stops at intermediate depots in order for the vehicle to be replenished. When trucks and trailers are used, the replenishment can be done by a switch of trailers. Angelelli and Speranza [12] presents an application of a similar problem in the context of waste collection.

More recently, Crevier et al. [59] propose a heuristic that combines adaptive memory and the sub-problems solved by a Tabu Search and linear integer programming. It is related to a distribution system in Montreal.

Guastaroba et al. [105] have also surveyed a VRP with intermediate facilities in distribution system.

3.5.3 Multiple Depots Location-Routing Problem

In Multiple Depots Location-Routing Problem, a set of depots must be located. Many researchers use LRP definition but they mean MDLRP.

Contardo et al. [54] reviewed a capacitated LRP with multiple depots. They propose a cut column generation method.

Another multi-objective Multi-Depot LRP in which location of depot and vehicle routing is dealing simultaneously was propose by Tavakkoli-Moghaddam et al. [241]. They present a Scatter Search Algorithm to maximize serving demand and to minimize the total operational cost where involve the opening of depots and delivery costs. Experimental results show that the proposed method work better than an Elite Tabu Search (ETS) approach.

Ozyurt and Aksen [186] solved a multi-depot location-routing problem by using a hybrid approach based on Lagrangian relaxation and TS. Still for the LRP in the literature, Matos and Oliveira [169] proposed a case study reviewed for waste collection system with 202 facilities implemented in Viseu City, Portugal and as well as an Ant Colony Optimization (ACO).

Zarandi et al. [270] studied the fuzzy-based version for Capacitated Location-Routing Problem (CLRP). The authors consider a model with multiple potential depots in which the location of depots should be defined and in the second phase the approach deals with routing problem. Fuzzy travel times between nodes and time window to visit each customer are taken in account. To solve the problem they proposed a Simulated Annealing (SA) procedure.

3.6 More general LRP problems

In the precedent section, we have reviewed the LRP and VRP models that include pickup and delivery constraints, multi-products demands constraints and the use of mult-depots or intermediate facilities in the routes. To the best of our knowledge, the three constraints, listed above have not been considered simultaneously in LRP and LRP-2E literature.

The only study which take more aspect mentioned above is the paper of Rieck et al. [211], that considers many-to-many LRP with inter-hub transport and multi-commodity pickup and delivery, but the vehicle route may be pure pickup, pure delivery or mixed. Some pickup locations have to be visited before serving the first delivery locations. In the study of Rieck et al. [211], the intermediate hubs are not considered in the vehicle routes like in our model, but only direct paths between hubs are considered. In packages (parcel) or mail delivery applications, each origin-destination pair provides a unique product. Rieck et al. [211] generalizes this issue : There are several different products, each of which is provided at one or more locations and requested one or more other locations. At one location, zero or more products are provided, and zero or more products are requested. A limited fleet of homogeneous vehicles is parked at the hubs and authorized to perform three types of routes : (i) multi-stop pickup routes starting empty from a hub, visiting one or more pickup points (location) before zero or more delivery points and return empty or partially charged to the hub, (ii) direct inter-hub vehicles from one hub to another one and go back, and (iii) multi-stop delivery routes getting started, partially loaded from a hub, visiting zero or more pick up points before visiting one or more delivery locations and returning empty at the hub. Each vehicle can perform at most a route of each kind, and should perform the routes in this sequence to guarantee the flow conservation at hubs. An MIP model with using binary variables in order to decide opening a hub, binary arc variables for routing, and 36 types of constraints was presented. They produces an initial feasible solution in two phases. First, a modified version of the savings heuristic is used to determine the type of routes (type-1 and type-3). Second, a multi-product fixed-charge network flow problem is solved to define the inter-hub routes (vehicles). In each iteration of this procedure, a subset of the opened hub and / or the routing variable is fixed, and a lower bound is calculated.

A multi-commodity version of the 2E-LRP in the context of city logistics was proposed Crainic et al. [56]. They consider that the demand of each customer has a issue such that the demands should be picked up by the vehicle in first level and should be delivered to the customer by using a two-echelon logistic system. A first-echelon vehicle will be derived to a second level facility if it exists enough second-echelon vehicles to be satisfied by the first-echelon vehicle. The load cannot be stocked at the second level facilities and the vehicles must not wait.

Recently, some studies were interested in more complex and more general versions of LRP. For example Haijun et al. [113] consider a split delivery constraint. Time windows and uncertainty are studied in M. H. Fazel Zarandi et al. [83]. In M. Mokhtarinejad et al. [165], authors include the location of crossdocking and scheduling problem of vehicles in the location routing problem.

Huang [120] proposes a LRP with pickup and delivery routes considering multi-item and stochastic demands. This paper considers an advanced capacitated location routing problem in a distribution network with multiple pickup and delivery routes, and each customer placing random multi-item demands on it. The pickup and delivery services need two fleets of vehicles and will form two different sets of routes. The problem is solved by tabu search in which can be more efficient and effective.

A theater distribution problem is reviewed by Burks [21]. The proposed 2E-LRP involves a multiple products version, heterogeneous fleet, time windows constraint, and capacitated facilities. The vehicle depots must be selected from a set of potential depot sites. The available vehicles has to be assigned to one depot. In their work, direct transports from first level facilities to customers are allowed. The objective consists of minimizing cost of vehicles use, opening facilities and vehicle routing. The vehicles features vary into costs, capacities, temporal availability and the transported freights. This research utilizes advanced tabu search techniques, including reactive tabu search and group theory applications, to develop a heuristic procedure for solving the proposed model.

Lopes et al. [20] present some heuristic approaches to tackle the location-arc routing problem. The LARP considers scenarios where the demand is on the edges rather than being on the nodes of a network. A constructive and improvement methods are presented and used within different metaheuristic.

Perboli et al. [200] proposed a model that is derived on multi-products distribution network design and uses the flow of goods on each edge as main decision variable. They provided a MILP formulation with valid inequalities and two matheuristics to solve the 2E-CVRP. The proposed heuristics for the 2E-CVRP based on the information that can be obtained by solving the linear relaxation of the model.

3.7 Conclusion

This chapter has developed an overview of the state of the art, related to LRP variants (and VRP as a special case) that come closest to our problems, and more particularly the integration of multi-echelon, pickup and delivery, multi-depot and multi-commodity aspects. We proposed a summary of the addressed issues in the literature. Real applications in supply chain characterized by complex transportation networks that may include factories, warehouses, customers and suppliers motivated us to propose new location routing models in which freight can be consolidated through one or more processing centers on the same route, to allow considerable saving. In the next chapter, we will study the first new location routing.

In the next chapter, we propose to study a new location routing problem including non classical VRP constraints. More precisely, we consider the following constraints : (i) pickup and delivery on the same route, (ii) the use of one or more intermediate processing centers on the same route, and (iii) multi-product demands. This new proposed model is called «Two-Echelon Multi-products Location-Routing problem with Pickup and Delivery». It concerns a generalization of several models presented in this chapter - The traveling salesman location problem with pickup and delivery - The LRP-SPD (LRP with Simultaneous Pickup and Delivery) - Many-to-many LRP - Multi-commodity pickup and delivery vehicle routing problem - Vehicle routing problem with intermediate facilities or with satellite facilities

In particular, the next chapter presents a set of constructive heuristics and hybrid methods to solve Two-Echelon Multi-products Location-Routing problem with Pickup and Delivery.

Chapitre 4

The Two-Echelon Multi-products Location-Routing problem with Pickup and Delivery

Contents

4.1 Introduction

In this chapter, we propose to study a new location routing problem including non classical VRP constraints. More precisely, we consider the following constraints : (i) pickup and delivery on the same route, (ii) the use of one or more intermediate processing centers on the same route, and (iii) multi-product demands. This new proposed model is called 2E-MPLRP-PD for Two-Echelon Multi-products Location-Routing problem with Pickup and Delivery. To the best of our knowledge, the three constraints listed above have not been considered simultaneously in LRP and 2E-LRP literature except in Rahmani et al. [213, 214]. Recently, Rieck et al. [211] considered many-to-many LRP with inter-hub transport and multi-commodity pickup and delivery, but the vehicle route may be pure pickup, pure delivery, or mixed, where some pickup locations have to be visited before serving the first delivery locations. In the study of Rieck et al. [211], the intermediate hubs are not considered in the vehicle routes as in our model, but only direct paths between hubs are considered. The 2E-MPLRP-PD allows modeling problems arising in a number of applications like drink distribution, home delivery service and grocery store chains, e.g. Carrera et al. [44]. These applications are characterized by complex transportation networks, that may include factories, warehouses, customers and suppliers. Consolidating freight through one or more processing centers on the same route, allows considerable savings. The 2E-MPLRP-PD is also a general case for several problems, such as :

1. The traveling salesman location problem with pickup and delivery introduced by Mosheiov [176].
2. The LRP-SPD (LRP with Simultaneous Pickup and Delivery), introduced by Karaoglan et al. [138], in which pickup and delivery are considered simultaneously in the vehicle routes. The addressed vehicle routing problem is known in the literature as vehicle routing problem with simultaneous pickup and delivery, named VRPSPD (Berbeglia et al. [34], Parragh et al. [188]).
3. Many-to-many LRP introduced by Nagy and Salhi [177], in which several customers wish to send goods to others and flows between processing centers are permitted.
4. Multi-commodity pickup and delivery vehicle routing problem (Hernandez-Perez and Salazar-Gonzalez [117], Rodriguez-Martin and Salazar-Gonzalez [215]).
5. Vehicle routing problem with intermediate facilities or with satellite facilities (Moin et al. [175]).

This chapter is organized as follows. Section 3.2 and 3.3 present the considered 2E-MPLRP-PD problem and its mathematical model. Section 3.4 describes the heuristic approaches and four strategies for each heuristic. The Hybrid Clustering Algorithm (HCA) is presented in section

3.5 and five improvement solution (Local Search) are detailed for the 2E-MPLRP-PD in section 3.6. The Experimentation and concluding remarks are discussed in section 3.7 and section 3.8, respectively.

4.2 Problem Description

In the Two-Echelon Multi-products Location-Routing problem with Pickup and Delivery - 2E-MPLRP-PD, two levels are considered. At the first level, routes are constructed from a main depot to a set of processing centers that must be selected, and at the second level, a set of vehicles of smaller capacities visit customers from the selected processing centers. We denote *primary* and *secondary routes*, the routes constructed at the first and the second levels, respectively.

The 2E-MPLRP-PD is modeled as an undirected and weighted graph $G = (N, A, f)$. N refers to the set of nodes, where $N = \{0\} \cup N_0 \cup N_c$, in which N_0 represents the set of potential processing center nodes, N_c represents the set of customers and node 0 is considered as a depot. A is the set of edges and f refers to a function that associates a positive cost (typically travel time) to each arc. At the depot, there is a set V_1 of homogeneous fleet of *primary* vehicles. Each primary vehicle has a limited capacity C_{V1} and a fixed cost F_{V1}. At the processing center sites, another set V_2 of homogeneous fleet of *secondary* vehicles is available. Each secondary vehicle has a limited capacity C_{V2} and a fixed cost F_{V2}. We consider the general case where C_{V1} is different from C_{V2}. Each potential processing center has an opening cost.

Each client asks for one or several types of products, denoted c-products, known in advance and can be satisfied. At each processing center, pickup and delivery operations are performed. Primary products, denoted h-products, are delivered from the main depot to opened processing centers. Each active processing center can receive only one type of h-products. The h-products are transformed into final products, denoted c-products. In this study, we consider specialized processing centers, such that each processing center (factory or logistic platform) should provide exactly one secondary c-product. This can model all applications, in which varied and personalized demands are required by the clients.

We consider two types of vehicles as explained above. The primary vehicles should pick up the h-products from the main depot and deliver them to the processing centers which have been opened. Each processing center is visited only once in each primary route. When satisfying the demand of processing centers, the secondary vehicles can pick up c-products, which are available in the processing centers, and continue their trips in a way that each customer and processing center is visited at most once by each secondary trip. The secondary trips, start from an active processing center, which will represent the departure node, serve several customers, can visit one or several processing centers and must end up at the departure node. We assume that products have the same size, and splitting the demand of customers for a given c-product is not allowed.

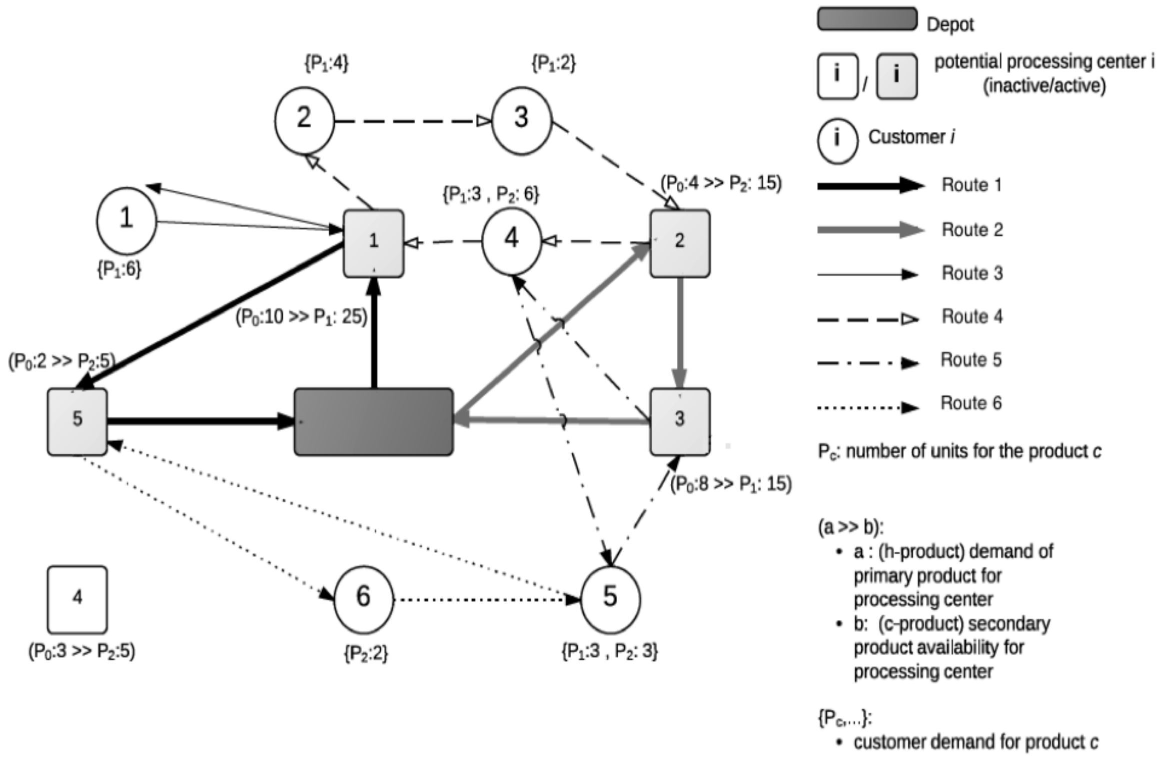

FIGURE 4.1 – Example of 2E-MPLRP-PD with 5 processing centers and 6 clients.

The goal of 2E-MPLRP-PD is to determine the location of processing centers, the assignment of customers to the opened processing centers and the construction of the corresponding primary and secondary routes with a minimum total cost. The total cost includes the opening cost of processing centers, the exploitation cost of vehicles and the sum of edges costs traversed by vehicles. An illustrative example of the two-echelon model is given in Figure (4.1).

4.3 Mathematical model

In this section, we present a mathematical formulation of the two-echelon LRPMPPD. It represents a concise formulation for a new problem. The following parameters are used in our mathematical model.

P : set of product types.

$c(i, j)$: the cost of the shortest path in G between vertices i and j, $(i, j \in N)$.

d_{is} : the demand of client i for c-product $s \in P$.

p_{ks} : amount of c-product that can be provided by potential processing center $k \in N_0$.

FD_k : a fixed cost associated with each potential processing center $k \in N_0$.

V_1 : the set of identical primary vehicles available at the main depot.

V_2 : the set of identical secondary vehicles available at the processing center sites.

V : the set of all primary and secondary vehicles.

C_{V1} (C_{V2}) : the capacity of primary (secondary) vehicle.

F_{V1} (F_{V2}) : the exploitation cost including the acquiring cost of the primary (secondary) vehicles.

dp_{ks} : h-product demand of processing center $k \in N_0$ for product $s \in P$.

H_{is} : is equal to 1 if node i asks for product $s \in P$, and 0 otherwise.

Ω_{ks} : is equal to 1 if processing center k produces c-product $s \in P$, and 0 otherwise.

P_j : set of c-products types that customer $j \in N_c$ asks for them.

Time-Limit : the maximum duration of vehicle routes.

M : is a large positive value.

The following decision variables are used in our mathematical model.

$x_{ij}^v = 1$ if the vehicle v travels directly from node i to node j, and 0 otherwise, $(i, j \in N)$.

$y_{ks} = 1$ if site k is setup to provide $s \in P$, and 0 otherwise, $(k \in N_0)$.

$q_{is}^v = 1$ if vehicle $v \in V$ satisfy demand s of node $i \in N$, and 0 otherwise.

l_{is}^v : total residual load for product s in vehicle v, just after having serviced customer (processing center) i .

U_{is}^v : quantity of product s to pick up from processing center i by vehicle v.

$h_v = 1$ if secondary vehicle $v \in V_2$ is exploited in a route and, 0 otherwise.

μ_{iv} : is the position index of node i in route v.

The problem can be formulated as follow :

$$\min \sum_{i \in N} \sum_{j \in N} \sum_{v \in V} c(i,j) x_{ij}^v + \sum_{k \in N_0} \sum_{s \in P} FD_k y_{ks} + \sum_{k \in N_0} \sum_{v \in V_1} F_{V1} x_{0k}^v + \sum_{v \in V_2} F_{V2} h_v \tag{4.1}$$

The objective function ((4.1)) minimizes the total travel cost, the opening cost of the selected processing center's and the vehicles fixed costs. In the following, we present three sets of constraints. The first set of constraints ((4.2) to (4.8)), concerns the first decision level. The second decision level is presented by the constraints ((4.9) to (4.18)). The last set of constraints ((4.19) to (4.31)) are valid for both levels (general constraints).

First level constraints :

$$\sum_{i \in N_0} x_{0i}^v \leq 1 \qquad \forall v \in V_1 \tag{4.2}$$

$$\sum_{i \in N_0} x_{i0}^v \leq 1 \qquad \forall v \in V_1 \tag{4.3}$$

Constraints (4.2) and (4.3) ensure that each primary vehicle is assigned to at most one route. Its route starts and ends at the depot.

$$y_{ks} \leq \Omega_{ks} \qquad \forall k \in N_0, s \in P \tag{4.4}$$

Constraint (4.4) ensures that each opened processing center produces only one type of c-product.

$$\text{if}\, x_{ij}^v = 1 \forall i \in N_0 \cup \{0\} \Rightarrow l_{is}^v - dp_{js} q_{js}^v = l_{js}^v \qquad \forall j \in N_0, s \in P, v \in V_1 \tag{4.5}$$

Constraint (4.5) assures the compatibility between vehicle load and processing center demand in primary routes.

$$\sum_{s \in p} l_{is}^v \leq C_{V1} \qquad \forall i \in N_0 \cup \{0\}, v \in V_1 \tag{4.6}$$

Constraint (4.6) ensures that the total load on each processing center does not exceed the primary vehicles capacity.

$$l_{0s}^v = \sum_{k \in N_0} q_{ks}^v dp_{ks} \qquad \forall s \in P, v \in V_1 \tag{4.7}$$

Constraint (4.7) guarantees that the total load in a vehicle at depot is equal to the requested quantity of all h-products by the processing centers, which will be satisfied by the given vehicle.

$$\sum_{s \in P} \sum_{v \in V_1} q_{ks}^v = \sum_{s \in P} y_{ks} \qquad \forall k \in N_0,\ H_{ks}= 1 \tag{4.8}$$

Constraint (4.8) assumes that the h-product demand of each opened processing center should be satisfied by a primary vehicle.

Second level constraints :

$$\sum_{i \in N_0} \sum_{j \in N} x_{ij}^v \geq h_v \qquad \forall v \in V_2 \tag{4.9}$$

$$\sum_{i \in N_0} \sum_{j \in N} x_{ij}^v \leq M \times h_v \qquad \forall v \in V_2 \tag{4.10}$$

Constraints (4.9) and (4.10) ensure that a secondary vehicle passes through at least one active(opened) processing center to pick-up the c-product.

$$\sum_{i \in N} \sum_{v \in V_2} x_{ij}^v \leq |P_j| \qquad \forall j \in N_c \tag{4.11}$$

$$\sum_{i \in N} \sum_{v \in V_2} x_{ij}^v \geq 1 \qquad \forall j \in N_c \tag{4.12}$$

Constraints (4.11) and (4.12) guarantee that the number of vehicles that pass through each customer is at most equal to the number of required c-products by the customer and at least equal to one.

$$\text{if}\, x_{ij}^v = 1 \forall i \in N \Rightarrow l_{is}^v - d_{js} q_{js}^v = l_{js}^v \qquad \forall j \in N_c, s \in P, v \in V_2 \tag{4.13}$$

Constraint (4.13) assures the compatibility between vehicle load and client demand in secondary routes.

$$\text{if}\, x_{ij}^v = 1 \forall i \in N \Rightarrow l_{is}^v + U_{js}^v = l_{js}^v \qquad \forall j \in N_0, s \in P, v \in V_2 \tag{4.14}$$

Constraint (4.14) ensures the compatibility between vehicle load and quantity of product s to pick up from processing center.

$$\sum_{s \in p} l_{is}^{v} \leq C_{V2} \qquad \forall i \in N_0, v \in V_2 \qquad (4.15)$$

Constraint (4.15) ensures that the total load on each processing center does not exceed the secondary vehicles capacity.

$$\sum_{v \in V_2} \sum_{s \in P} q_{is}^{v} = \mid P_i \mid \qquad \forall i \in N_c \qquad (4.16)$$

$$\sum_{s \in P} q_{is}^{v} \leq \mid P_i \mid \sum_{j \in N} x_{ij}^{v} \qquad \forall v \in V_2, i \in N_c (4.17)$$

Constraint (4.16) sets the number of variables q_{is}^{v} ($q_{is}^{v} = 1$) to $\mid P_i \mid$, and the constraint (4.17) assures that, when a vehicle visits a client i, it satisfies at least the demand of one product of the client i.

$$\sum_{v \in V_2} U_{is}^{v} \leq p_{is} y_{is} \qquad \forall i \in N_0, \Omega_{is} = 1 \qquad (4.18)$$

Constraint (4.18) guarantees that the sum of loaded c-product from each opened processing center by different secondary vehicles is less than the available c-product at a given processing center.

General constraints :

$$\sum_{i \in N} \sum_{j \in N} c(i,j) x_{ij}^{v} \leq Time - Limit \qquad \forall v \in V \qquad (4.19)$$

Constraint (4.19) enforces that route time will not exceed *Time-Limit*.

$$\sum_{i \in N} x_{ij}^{v} \leq 1 \qquad \forall j \in N, v \in V \qquad (4.20)$$

Constraint (4.20) ensures that a vehicle can visit each node at most once.

$$\sum_{v \in V} \sum_{i \in N} x_{ij}^{v} \geq y_{js} \qquad \forall j \in N_0, \Omega_{js} = 1 \qquad (4.21)$$

Constraint (4.21) ensures that when a processing center is opened, there is at least one visit to this processing center.

$$\sum_{j \in N} x_{ji}^{v} - \sum_{j \in N} x_{ij}^{v} = 0 \qquad \forall i \in N, v \in V \qquad (4.22)$$

Constraint (4.22) is known as degree constraint and guarantees that the same number of arcs enter and leave each node.

$$\sum_{i \in N_c} d_{is} \leq \sum_{k \in N_0} p_{ks} y_{ks} \qquad \forall s \in P \qquad (4.23)$$

Constraint (4.23) ensures that the sum of c-product, available in all opened processing centers is more than the sum of clients demand for each product.

$$\sum_{i \in N} \sum_{s \in P} q^v_{is} \geq \sum_{j \in N} x^v_{ij} \qquad \forall i \in N, v \in V \qquad (4.24)$$

$$\sum_{i \in N} \sum_{s \in P} q^v_{is} \leq M \times \sum_{j \in N} x^v_{ij} \qquad \forall i \in N, v \in V \qquad (4.25)$$

Constraints (4.24) and (4.25) enhance the compatibility between demand satisfaction and the used vehicles.

$$\mu_{iv} - \mu_{jv} + 1 \leq (\mid N \mid -1)(1 - x^v_{ij}) \quad \text{j} \in N, v \in V, i, j \neq 1 \qquad (4.26)$$

$$2 \leq \mu_{iv} \leq \mid N \mid \qquad \forall i \in \{2, ..., \mid N \mid\} \qquad (4.27)$$

$$x^v_{ij} \in \{0,1\} \qquad \forall i, j \in N, v \in V \qquad (4.28)$$

$$h_v \in \{0,1\} \qquad \forall v \in V_1 \qquad (4.29)$$

$$y_{ks} \in \{0,1\} \qquad \forall k \in N_0 \qquad (4.30)$$

$$q^v_{is} \in \{0,1\} \qquad \forall s \in P, k \in N_0, v \in V \qquad (4.31)$$

In order to eliminate the sub-tours, a formulation with polynomial size introduced by Miller, Tucker, and Zemlin [174] is used in constraints (4.26) and (4.27). (4.28)-(4.31) are known as integrity constraints.

Constraint (4.13) are linearized by adding two inequalities which are detailed in the following :

$$l^v_{is} - d_{js} q^v_{js} - l^v_{js} \leq (1 - x^v_{ij}) \times M \forall v \in V_2, i \in N, j \in N_c, s \in P \qquad (4.32)$$

$$l^v_{is} - d_{js} q^v_{js} - l^v_{js} \geq (x^v_{ij} - 1) \times M \forall v \in V_2, i \in N, j \in N_c, s \in P \qquad (4.33)$$

Constraints (4.5) and (4.14) can be linearized in the same way.

4.3.1 Solution Representation

Table 4.1 presents a feasible solution. The solution can be shown as a set of routes assigned to the primary and secondary vehicles in the two-echelon model. In the first level the vehicles start and end up at the central depot and in the second level each the routes start and end up at the same processing center (*starting processing center*). Here, each route is recorded in one array. The primary routes contain only the processing center to deliver their demands ; While the secondary routes in second level contain the customers and one or more *processing center*. In second level, we attribute a cell per product after appearing each client in solution array. For instance (see 4.2), in Route 4, after client 7, two cells in following contain a binary information (0 : Not delivered and -1 : delivered) corresponding to p_1 and p_2 respectively. As you can see

TABLE 4.1 – Example of a solution to 2E-MPLRP-PD

Routes	Ordering	Vehicle Type
1	0-1-5-0	primary route
2	0-2-3-0	primary route
3	1-6-1	secondary route
4	1-7-2-8-1	secondary route
5	3-9-10-3	secondary route
6	5-11-10-5	secondary route

in this example client 7 is delivered for product p_1 and client 8 is delivered for both p_1 and p_2 and finally route 4 ends at *starting processing center* 1.

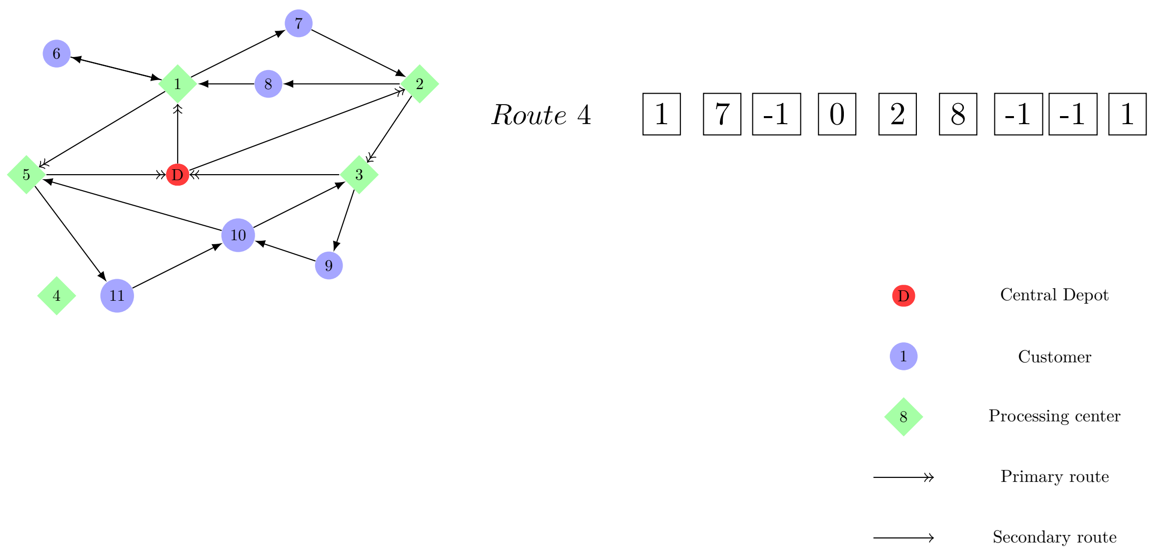

FIGURE 4.2 – Solution Representation

4.4 Heuristic approach

As the *LRP-MPP-2E* is an NP- hard problem and since it results from the combination of complex constraints, large instances can hardly be solved by exact methods. So, the best way to tackle this problem is using heuristic approaches. In this section, we investigated non-trivial extensions of classical vehicle routing heuristics, namely the Nearest Neighbour Heuristic (NNH) and a Sequential Best Insertion Heuristic (SBIH). Another approach, based on clustering analysis namely Hybrid Clustering Algorithm (HCA) is also proposed. In the following we provide details of each heuristic.

4.4.1 The Nearest Neighbour Heuristic (NNH)

The Nearest Neighbour Heuristic (NNH) consists of two steps. The first step aims at locating processing centers and provides the first level routes. The goal of the second step is to use the selected processing centers opened by the first step to construct the second level routes. Both steps are detailed in the following.

Step 1. **First-level routes construction**.

This first level aims at selecting the processing centers and constructing the first-level routes. At first, all processing centers are considered inactive (closed processing center). Each primary route starts at the depot, visits a set of processing centers by satisfying their h-product demands. Therefore, the visited processing centers are considered as active (opened processing center). To construct the routes, the nearest neighbour strategy is used. More precisely, we seek for the nearest closed processing center k from the last inserted node in the current route. This processing center k is chosen in a set L_{pc} of pre-selected processing centers, such that the sum of c-products available at the pre-selected processing centers is at least equal to the total customers demand for each product s. The pre-selection of processing centers in L_{pc} is done either randomly or according to the score value calculated by formula (4.34). As each processing center is specialized in one product, choosing a processing center involves the choice of the product. A nearest neighbour candidate k that provides c-product s is inserted in the route if (i) there is enough amount of h-product to satisfy the demand of k, (ii) the route duration doesn't exceed the *Time-Limit* and (iii) the sum of the c-product s available in the opened processing centers is less than the total customers demand of product s. Otherwise a second neighbour candidate from L_{pc} will be checked until no processing center can be inserted in the current tour. In that case a new tour is created. The construction of the routes is repeated until $L_{pc} = \emptyset$.

$$k^* = \text{argmin}_{k \in N_0} \left\{ c(k,0) + \frac{p_{ks}}{FD_k} + \sum_{i \in Z} c(k,i) \right\} \tag{4.34}$$

Where Z in (4.34) is the set of non-satisfied clients that request c-product of processing center k. Note that each processing center k provides only one c-product s in P.

Step 2. **Second-level routes construction.**

At this level, the secondary vehicles try to fulfil the customers' requests. Figure 4.3 shows the routes construction of this level. In this step, we ignore all closed processing centers that have not been opened in step 1. Let $\overline{P(j)}$ be the set of non satisfied products of costumer j and $\overline{N_c}$ be the set of non satisfied customer j, such as $|\ \overline{P(j)}\ | \neq 0$. Firstly, a processing center k is selected as departure node to start a new route l, either randomly or according to the score calculated by formula (4.35). Then, a nearest neighbour candidate i (active processing center or client) is selected. The procedure *Check-feasibility(i,l)* is applied to verify the following constraints : 1) i is not already visited by route l, 2) vehicle capacity and *Time-Limit* constraints are not violated and 3) if i is a client candidate, then i must be satisfied at least for one product s in $\overline{P(i)}$. In this last case, if i can be completely satisfied in l, i.e., the quantity of demand $\sum_{s \in \overline{P(i)}} d_{is}$ can be satisfied by the inserted processing centers in l then i is inserted in the last position in l to satisfy all product $\overline{P(i)}$ and i is deleted from $\overline{N_c}$. Else, if only a subset of products $S(i)$ in $\overline{P(i)}$ can be satisfied then, i is inserted in the route to satisfy the products $S(i)$ and $\overline{P(i)}$ is updated by deleting the $S(i)$ products. If i cannot be inserted in the current route, the second nearest neighbour candidate from the last inserted node will be checked. Once we cannot insert neither processing center nor client, the current route is closed and we create a new route. This process is repeated until all client requests are satisfied (Figure 4.3).

$$k^* = \text{argmin}_{k \in N_0} \left\{ \sum_{i \in Z} c(k,i) \right\} \tag{4.35}$$

where Z in (4.35) is the set of non-satisfied clients that request c-product of processing center k.

4.4.2 Best Sequential Insertion Heuristic (BSIH)

In this section, we describe the insertion heuristic method developed to solve the 2E-MPLRP-PD. The insertion method consists of two phases : (i) initialization phase and (ii) insertion phase (Figure. 4.4). The initialization phase determines the first processing center in each new route and the second phase tries to insert the maximum number of clients and processing centers into the current route. The two phases are repeated until the total customer demand is satisfied. The routing between the selected processing centers for the first level (first-level routes) is obtained by the nearest neighbour heuristic described in section 4.4.1, step 1.

Step 1. **Initialization Phase.** The goal of this step is to determine, for each c-product s, the processing center that will start a new route. In order to select the best processing center, we use the score value of formula (4.34) for each s. The score function takes into account the distance between the processing center and the depot, the sum of distances between the processing center and non-satisfied clients that request s and the opening cost of the processing center. A processing center with minimal score is selected to start the route. The Initialization function will check for each product s whether the already opened processing centers could satisfy the clients, who asked for s.

Step 2. **Route insertion phase.**

After inserting a processing center via the initialization step, the construction of a route is performed by two processes 2.1 and 2.2 described below. These two processes will be

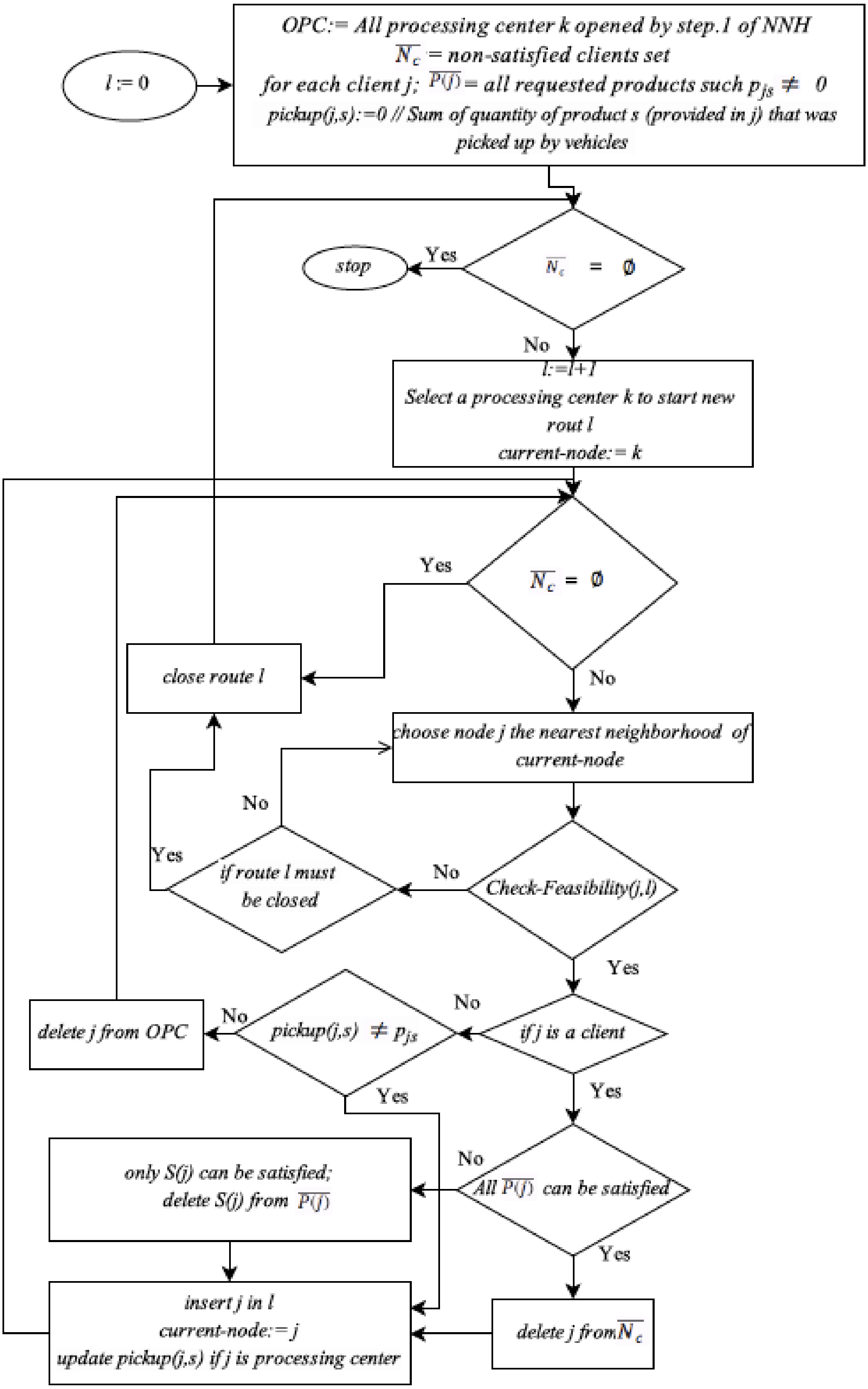

FIGURE 4.3 – Secondary Level Routing of NNH Algorithm

repeated until the *Time-Limit* of the route is reached. In this case, if all clients requests are satisfied the algorithm stops ; otherwise, a new route is initialized by selecting a processing center according to Step 1.

2.1. *Client insertion.* Let (i, S) be a pair of a customer i and its request of n $c-$products $S = \{s_1, \ldots, s_n\} \subseteq P_i$. Let $\overline{N_c}$ be the list of pairs $(i, \{s_1, \ldots, s_n\})$ of non satisfied clients-products sorted according to one of this two strategies : (i) Random Client Insertion (RCI), in which the customers are sorted randomly and (ii) Nearest Neighbour Client Insertion (NNCI), in which customers are sorted according to their proximity to the route under construction.

In the client insertion process, the request of client i can be splitted into (i, S_1) and (i, S_2) such that $S_1, S_2 \subseteq \{s_1, \ldots, s_n\}$ and $S_1 \cup S_2 = \{s_1, \ldots, s_n\}$. The client insertion process can be described as below :

Select a client i from the head of $\overline{N_c}$. The algorithm inserts the client (i, S) at the best position $\delta^*(i)$ that maximizes the sum of satisfied demands of client i over the insertion cost Δ_δ. More precisely, if *SP* is the subset of $c-$products requested by client i that may be inserted at position δ between two already inserted nodes a and b in L, where L is the subset of all feasible positions in current route, then

$$\delta^*(i) = \text{argmax}_{\delta \in L} \left\{ \frac{\sum_{s \in SP} d_{is}}{\Delta_\delta} \right\} \tag{4.36}$$

where $\Delta_\delta = c(a, i) + c(i, b) - c(a, b)$, $i \in \overline{N_c}$.

If the total demand of client i is satisfied in position $\delta^*(i)$, then the pair (i, S) is removed from $\overline{N_c}$, otherwise all products already satisfied are removed from subset S and the list $\overline{N_c}$ is updated. If $\overline{N_c}$ is empty then all demands are satisfied and the algorithm stops, otherwise we scan the list $\overline{N_c}$ to insert the rest of the clients. Once all clients are tested the algorithm continues with the processing center insertion process.

2.2 *Processing center insertion.* All available processing centers are candidates for insertion. A processing center k is chosen by its closeness to the current route. Two insertion strategies are used BIH_1 and BIH_2. In BIH_1, the goal is to shift the insertion of k at the last position in a current route l, when the construction of l is at the beginning and the route time of l is far enough from *Time-Limit*. Let T be the duration time of the current route and HP a subset of clients in the current route that could be satisfied by a given processing center k. If $T \leq Time - Limit \times \alpha$, where $\alpha \in [0, 1]$, then k is inserted at the last feasible position, otherwise all positions are checked to compute the best insertion position $\delta^*(k)$ according to (4.37). In BIH_2 only the best insertion position rule is used where the best position $\delta^*(k)$ is calculated according to (4.37).

$$\delta^*(k) = \text{argmax}_{\delta \in L} \left\{ \frac{\sum_{i \in HP} d_i s}{\Delta_\delta} \right\} \tag{4.37}$$

where $s \in P$, $k \in N_0$, $\Omega_{ks} = 1$.

If a processing center is inserted, then the algorithm continues with the client insertion procedure, otherwise when there is no processing center to be inserted into the current route, then the current route is closed and the algorithm restarts with the initialization phase (Figure. 4.4).

4.5 Hybrid Clustering Algorithm

The clustering approach has not been considered for 2E-LRP. Only few studies have considered clustering approaches for LRP (Ozdamar and Demir [185], Zare Mehrjerdi and Nadizadeh [276], Guerrero and Prodhon, [108]). However, several authors such as Bruns and Klose [40], Barreto et al. [37] have recognized the potential of cluster analysis.

The proposed HCA algorithm is a non-trivial extension of a greedy clustering method proposed in [276] for a classical LRP with a fuzzy demands. The HCA algorithm proceeds in five steps (see Figure. 4.5). In the first step, customers are clustered using an algorithm based on nearest neighbour, such that each cluster involves only clients that request the same product (Figure 4.5.a). In the second step, the center of gravity of each cluster is calculated. This allows to select a set of potential processing centers (Figure 4.5.b). In the third step, clusters are merged as well as possible in order to create Global-Clusters (GC) in which only one vehicle will be exploited, i.e., each Global-Cluster represents one feasible secondary route. This merging step considers the distance between the center of gravity of the clusters as well as the route *Time-Limit* (Figure 4.5.c). In order to ensure the feasibility of the solution in each Global-Cluster, the merged clusters should not have any common client, since the exploited vehicle for each Global-Cluster must visit each customer and each processing center only once. The clusters are allocated to the opened processing centers in the forth step, considering the distances between the processing centers and the center of gravity of the clusters as well as the capacity of the processing centers (Figure. 4.5.d). Finally, in the fifth step, Cplex solver is used to find a feasible secondary route in each Global-Cluster (Figure. 4.5.e). The routing between the selected processing centers in the first level (primary routes) is obtained by a vehicle routing nearest neighbour heuristic (section 3.1, step1). Details of the HCA steps are given bellow.

4.5.1 Clustering the customers

The customers are separated into different groups considering their intra distances, the sum of their demands, the vehicle capacity, the route time limit, and an estimation of the

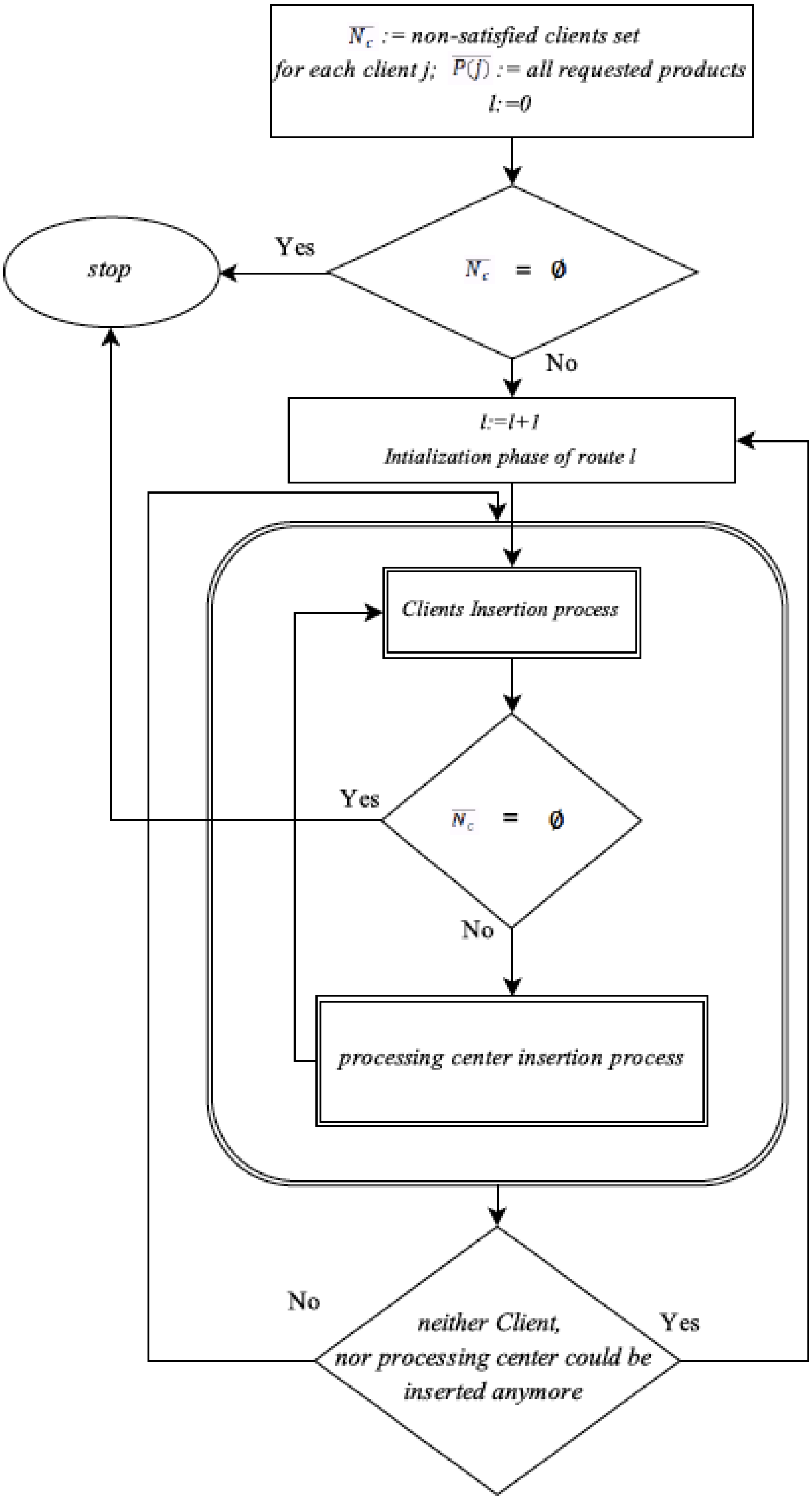

FIGURE 4.4 – General scheme of BSIH.

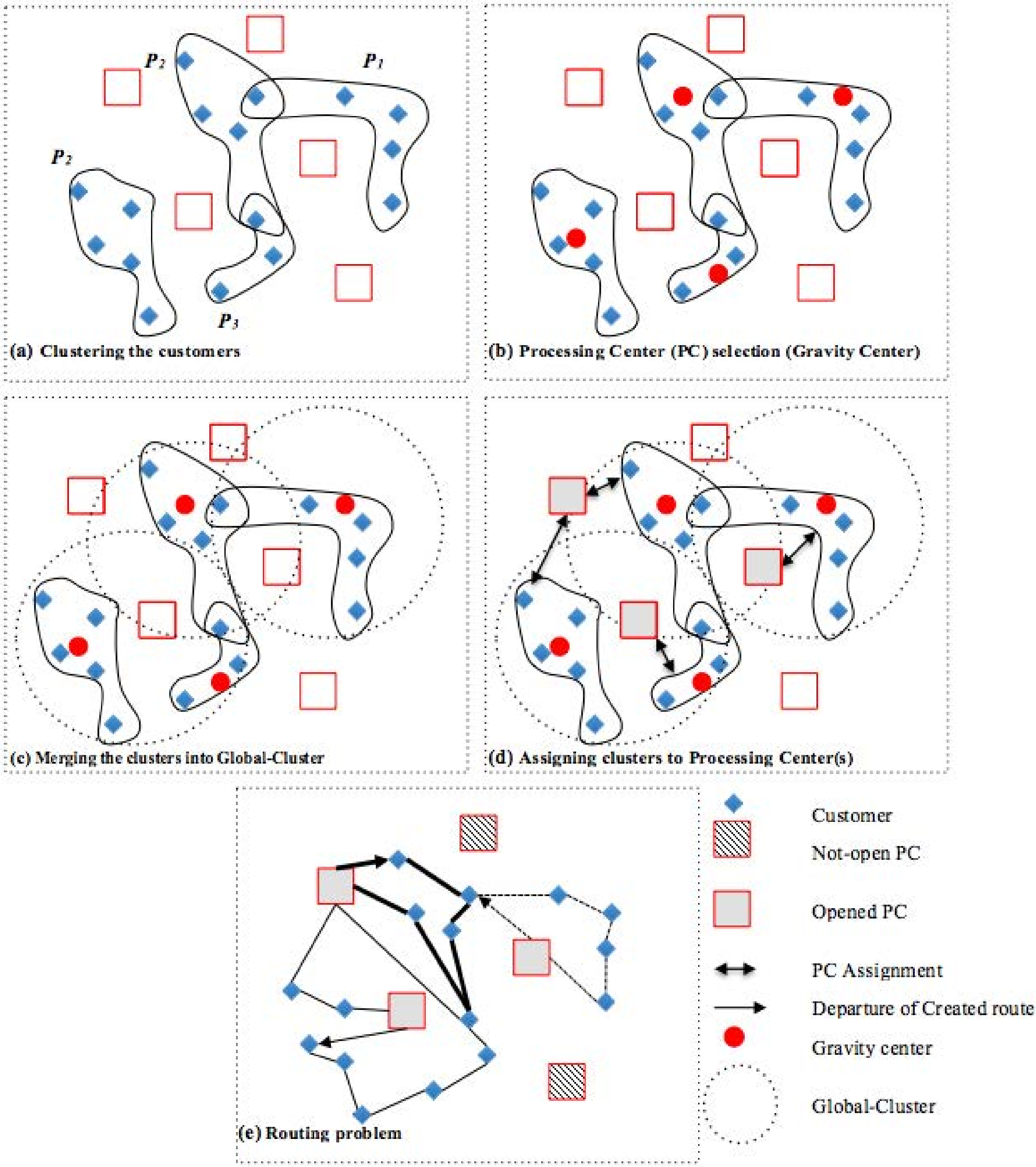

FIGURE 4.5 – HCA for the 2E-MPLRP-PD.

route travel time given in formula (4.38), in which N_{cl}^{c} and N_{cl}^{0} represent the number of clients and processing centers in cluster cl, respectively. D_{C}^{Max} is the maximal distance between each pair of clients in cluster cl. The maximal distance between the processing centers and the clients in cluster cl is denoted by D_{k}^{Max}.

$$T = ((N_{cl}^{c} - N_{cl}^{0}) \times D_{C}^{Max}) + 2 \times (N_{cl}^{0} \times D_{k}^{Max}) \tag{4.38}$$

The value T associated to a cluster cl, is an overestimation of the route that starts from a processing center, visiting all the customers assigned to the cluster cl and ending at the starting processing center. More precisely, for each $c-$product s, a set of non-clustered customers (NCC_s) is initialized by all customers j such as $d_{js} > 0$, where d_{js} is the quantity of product s demanded by the customer j. At first, a customer is selected randomly from a set NCC_s, then the nearest customer to the last selected customer of the current cluster is chosen from NCC_s. Therefore, the clusters are formed for a single $c-$product. The nearest customer is selected as follows : when a new customer j asking for product s is selected from NCC_s set (j is the closest customer with respect to the distance to the last inserted customer in the cluster cl), before its assignment to cl, the procedure *check-feasibility-clustering(j,cl)* is applied to verify two conditions, (i) the sum of the amounts requested by the assigned clients to the cluster cl should not exceed the secondary vehicle capacity C_{V2}, (ii) the estimated travel time in a cluster cl, i.e., value T in (4.38) with $N_{cl}^{0} = 1$, doesn't exceed the route time limit. If these two conditions are fulfilled, the new customer is assigned to the current cluster, otherwise, the algorithm searches in NCC_s for the next closest customer to the last added customer. The algorithm creates a new cluster if there is no customer to be assigned to the current cluster. The algorithm stops when there is no unassigned customer. Figure (4.6) illustrates the cluster's selection algorithm.

4.5.2 Processing Center (PC) selection

In the second step of the HCA, the method of [276] is used to establish the list of opened processing centers. This method is based on a gravity center criterion as illustrated by equation (4.39), in which (X_{cl}, Y_{cl}) is the coordinates of the gravity center of the cluster cl and (x_i, y_i) is the coordinates of customer i, where n_{cl} is the number of customers assigned to cluster cl.

$$(X_{cl}, Y_{cl}) = \left(\frac{\sum_{i \in cl} x_i}{n_{cl}}, \frac{\sum_{i \in cl} y_i}{n_{cl}} \right) \tag{4.39}$$

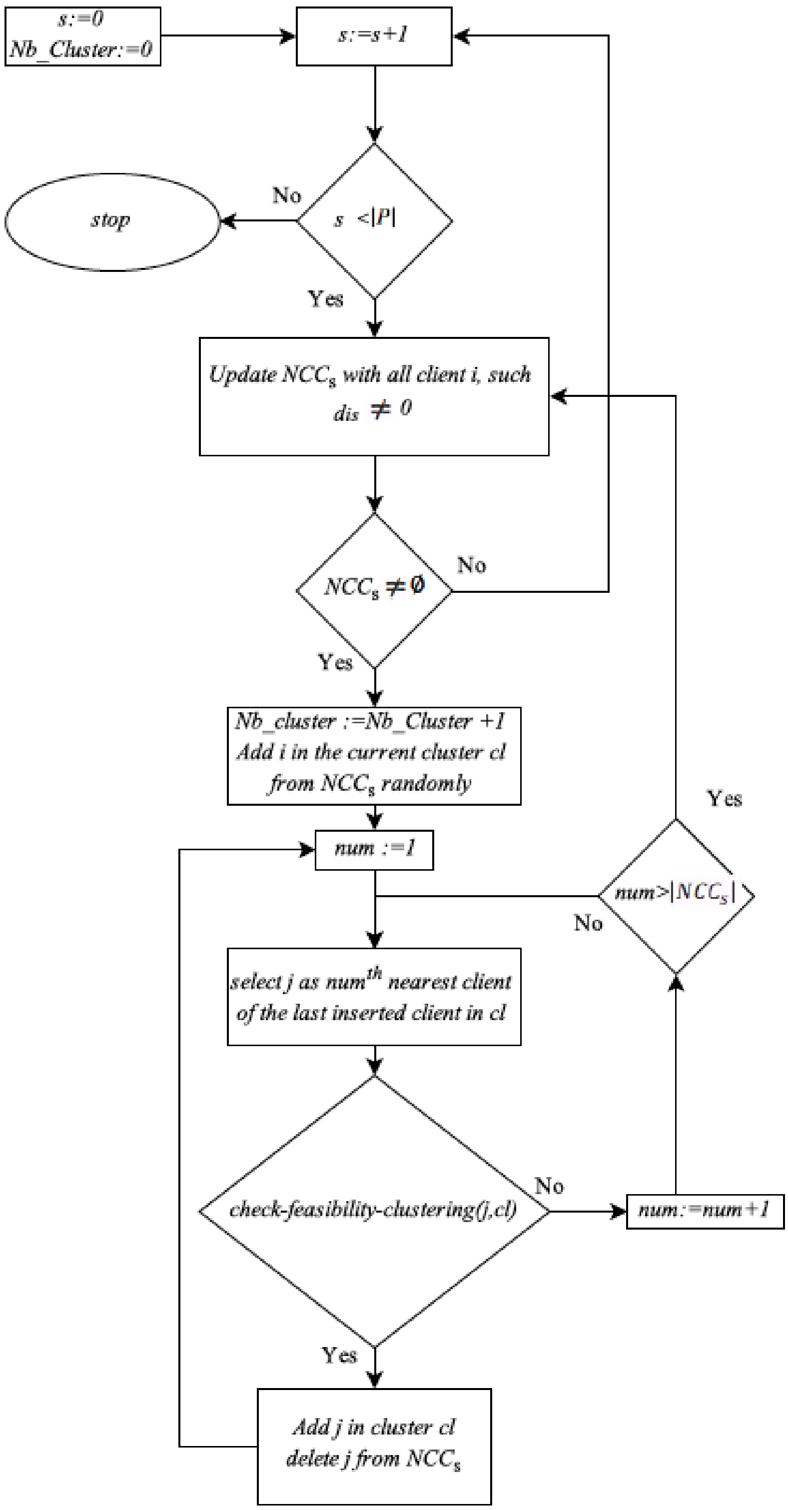

FIGURE 4.6 – Clustering Customers Algorithm

For each processing center, we calculate the sum of distances between this potential site and all centers of gravity. The potential sites are re-indexed in non-decreasing order of their Euclidean distances to the center of gravity of the clusters. If the current opened top-ranked potential site is not able to fulfill all the remaining customers' demands, the next potential site of the sorted list is selected to be opened. This procedure is repeated until all the clusters are covered. Therefore, each selected processing center will be assigned to one or more clusters and each cluster is covered by one or more processing centers.

4.5.3 Merging the clusters into Global-Cluster

In this step, the clusters are merged in order to create a set of Global-Clusters GC, in which GC represents one feasible secondary route. Since the assigned vehicle to each GC must visit customers and processing centers only once, the merged clusters should not have any common client. At first, a cluster cl is selected randomly, and then a sorted list of the unmerged clusters cl' is constructed according to the distance between the centers of gravity of cl and cl'. The first cluster in the list is added into the current Global-Cluster GC if the value of T calculated by (4.38) when N^0_{cl} is equal to the number of merged clusters in GC does not exceed the route time limit (Figure. 4.5.c). This procedure is repeated until no cluster can be added to the current Global-Cluster. In that case, either the process stops because all the clusters are merged or the process is restarted to search for a new Global-Cluster.

4.5.4 Assigning clusters to Processing Centers

In the forth step of the HCA, the clusters are allocated to the processing centers that were ranked and opened in the processing center selection step. Each processing center serves as many clusters as possible according to its capacity (availability of the provided c-products). Note that we can't allocate two clusters cl_1, cl_2 to the same processing center when they were merged in the same GC, because a vehicle cannot visit a processing center twice in a given route. In order to allocate the clusters to the processing center, the Euclidian distance between the center of gravity of each cluster and the opened processing center is calculated. Then the unassigned clusters are ranked in an ascending order based on the distances of their centers of gravity to the processing center. The top-ranked cluster cl_1 will be allocated to the top-ranked processing center p_r if (i) the processing center p_r has enough capacity to cover the total demand of the cluster cl_1, and (ii) the processing center p_r is not already assigned to a cluster cl_2, such that cl_1 and cl_2 belong to the same Global-Cluster. The allocation process for the processing center p_r is completed when there is not enough capacity to allocate a new cluster. In that case, the allocation procedure is repeated for the next top-ranked processing center until all clusters are allocated.

4.5.5 Routing problem

In the fifth and last phase of the HCA, the routing problem is solved for each Global-Cluster GC with the relevant processing centers and assigned clients. Each Global-Cluster is served by exactly one vehicle and the vehicle is not allowed to visit any node two times. The addressed routing problem in each GC is a complex variant of the TSP, named multi-product traveling salesman problem with pickup and delivery. Cplex solver is used to create one secondary route per one Global-Cluster.

4.6 Computational experiments

4.6.1 Implementation and Benchmark instances

The proposed heuristics were computationally compared with the exact approach. Results are evaluated in terms of solution quality and running time. All the experiments were carried out on a PC with Intel (R) Core (TM) Solo CPU 1.40 GHz, 2GB of RAM. The tested algorithms were coded in C++. The routing steps of HCA and the mixed integer linear model use a Cplex solver version 12.4.
Prodhon et al. [14] have proposed a set of *2E-LRP* Euclidean instances. Prodhon's *2E-LRP* benchmark contains 30 instances, derived from Prodhon's *CLRP* benchmark by converting depots into satellites-depots, and adding one main depot at the origin $(0, 0)$. These instances are grouped in four subsets with the following features : number of customers $n \in \{20, 50, 100, 200\}$, uniform distribution demands in the interval $[11, 20]$, number of satellites depots $m \in \{5, 10\}$, with their opening costs, number of clusters $\beta = \{1, 2, 3\}$ of clients, vehicle capacity of second level C_{V2}, vehicle capacity of first level C_{V1}. Since there is some similar instances with just different vehicle capacities in Prodhon's benchmark, we obtained only 18 instances among 30 instances of Prodhon's Benchmark by fixing capacity $C_{V1} = 200$ and $C_{V2} = 70$. Each instance is described by three parameters : $n - m - \beta$.
In order to adapt these instances to our problem, we have considered the following hypotheses :

— Each satellite depot corresponds to a processing center in our problem.
— We consider 3 distinct products : one $h-$product p_0 and two others $c-$products p_1 and p_2.
— Each client asks for products, p_1 or p_2 or both products with equal probability.
— We consider two homogeneous fleets of vehicles, with capacities C_{V1} and C_{V2}. We note that C_{V1} must be greater than the quantity of all $h-$product demands.
— We added the $h-$product demand for each processing center, such that the demand of each $h-$product is equal to $\frac{1}{5}$ of $c-$product availability at this processing center.

These instances are used to generate another set of 18 instances having the same characteristics as the first set of instances except for the processing center opening cost. The imposed changes are such that all processing centers with the same secondary product p_c get a unique opening cost equal to the minimum opening cost of the considered processing centers. Table 1. provides the characteristics of the instances. Vector p_0 represents the $h-$product demands of all processing centers, and vectors p_1 and p_2 give the indices of processing centers in which the products p_1 and p_2 are available. These generalized instances are accessible at [206].

Instance	#C	# hub	$c-$products (p_1), (p_2)	$h-$products (p_0)
coord20-5-1-2e	20	5	p_1(2,4,5), p_2(1,3)	p_0(28,28,28,28,28)
coord20-5-1b-2e	20	5	p_1(2,4,5), p_2(1,3)	p_0(60,60,60,60,60)
coord20-5-2-2e	20	5	p_1(2,4), p_2(1,3,5)	p_0(14,28,14,28,28)
coord20-5-2b-2e	20	5	p_1(3,5), p_2(1,2,4)	p_0(30,30,60,60,30)
coord50-5-1-2e	50	5	p_1(1,4), p_2(2,3,5)	p_0 (84,84,70,70,70)
coord50-5-2-2e	50	5	p_1(1,4), p_2(2,3,5)	p_0(70,70,70,70,70)
coord50-5-2bBIS-2e	50	5	p_1(1,3,4), p_2(2,5)	p_0(60,60,60,60,60)
coord50-5-2BIS-2e	50	5	p_1(1,3,5), p_2(2,4)	p_0(70,70,70,70,70)
coord50-5-3-2e	50	5	p_1(1,4), p_2(2,3,5)	p_0(70,84,70,84,84)
coord100-5-1-2e	100	5	p_1(2,4,5), p_2(1,3)	p_0(154,140,154,154,154)
coord100-5-2-2e	100	5	p_1(2,4,5), p_2(1,3)	p_0(140,154,154,154,168)
coord100-5-3-2e	100	5	p_1(2,4,5), p_2(1,3)	p_0(154,168,154,154,154)
coord100-10-1-2e	100	10	p_1(2,4,5,6,8), p_2(1,3,7,9,10)	p_0(98,84,84,98,112,98,98,98,84,112)
coord100-10-2-2e	100	10	p_1(2,4,5,6,8), p_2(1,3,7,9,10)	p_0(98,112,112,98,98,112,112,98,84,98)
coord100-10-3-2e	100	10	$p1$(2,4,5,6,8), p_2(1,3,7,9,10)	p_0(98,84,98,112,112,112,84,112,98,98)
coord200-10-1-2e	200	10	p_1(4,6,8,9), p_2(1,2,3,5,7,10)	p_0(200,182,182,200,196,200,200,200,200,200)
coord200-10-2-2e	200	10	p_1(4,6,8,9), p_2(1,2,3,5,7,10)	p_0(182,182,196,200,196,196,299,196,196,182)
coord200-10-3-2e	200	10	p_1(1,4,6,8,9), p_2(2,3,5,7,10)	p_0(196,182,200,200,200,196,196,200,200,200)

TABLE 4.2 – Characteristic of instances

4.6.2 Comparative analysis Heuristics

In this section, we analyze the performance of the proposed methods using four strategies for each method. In all, we compare and test 12 heuristics for the 2E-MPLRP-PD. These heuristics are evaluated by using a lower bound LB obtained by Cplex with the relaxation of sub-tours elimination and secondary vehicles capacity constraints. In tables 4.3 and 4.4, the values in parenthesis represent the Gap between the lower bounds LB and the solutions obtained by the heuristics. The Gap is calculated as $\frac{Cost(H)-LB}{Cost(H)}$.
Results of tables 4.3 and 4.4 show that Cplex is unable to find the optimal solution in any of the instances, and it cannot generate any solution after 1 hour of execution time for large instances ($n \geq 200$ for the first set of instances and $n \geq 100$ for the second set of instances).
In tables 4.3 and 4.7, the first column indicates the problem name "$coord.n-m-\beta-2e$" and the last row indicates for each column : (i) the average solution compared to the LB, and (ii) the average running time.

Instances	Nearest Neighbor processing center selection				Score based processing center selection			
	NNPCS-RI		NNPCS-SI		SBPCS-RI		SBPCS-SI	
	Time	Cost(%)	Time	Cost(%)	Time	Cost(%)	Time	Cost(%)
coord20-5-1-2e	0	41085(15,70)	1	**37447(7,51)**	1	37488(7,62)	0	**37447(7,51)**
coord20-5-1b-2e	1	29875(39,22)	1	24113(24,70)	0	24692(26,46)	0	**19877(8,65)**
coord20-5-2-2e	0	36938(3,94)	1	50969(30,38)	2	41526(14,55)	1	**36800(3,58)**
coord20-5-2b-2e	2	48719(58,48)	0	35053(42,29)	1	**27937(27,59)**	3	27941(27,60)
coord50-5-1-2e	0	44147(32,37)	1	53833(44,54)	2	**34800(14,21)**	4	44125(32,34)
coord50-5-2-2e	3	56540(12,81)	1	57900(14,86)	2	**54395(9,38)**	3	55242(10,77)
coord50-5-2bBIS-2e	2	36093(22,65)	1	35784(21,98)	2	34678(19,49)	4	**34174(18,30)**
coord50-5-2BIS-2e	4	**35479(17,72)**	2	35556(17,90)	2	36224(19,41)	2	35556(17,90)
coord50-5-3-2e	2	52546(25,13)	1	53234(26,10)	2	44461(11,52)	3	**43859(10,30)**
coord100-5-1-2e	4	208053(13,65)	2	204336(12,08)	3	204441(12,12)	4	**196936(8,78)**
coord100-5-2-2e	3	**216913(7,88)**	3	223017(10,40)	3	231267(13,60)	5	217352(8,07)
coord100-5-3-2e	5	210661(5,43)	2	211117(5,64)	3	212231(6,13)	4	**209812(5,05)**
coord100-10-1-2e	2	285302(26,24)	2	324373(35,13)	2	336742(37,51)	4	**271890(22,60)**
coord100-10-2-2e	4	213916(11,56)	2	229191(17,46)	4	225103(15,96)	5	**199861(5,34)**
coord100-10-3-2e	4	245942(26,16)	2	265139(31,51)	2	**219894(17,41)**	4	243523(25,43)
coord200-10-1-2e	8	434322	4	459966	5	419625	5	**351502**
coord200-10-2-2e	5	**453836**	3	458940	5	519466	6	534204
coord200-10-3-2e	6	450966	4	388641	4	384127	8	**374447**
coord20-5-1-2e	0	**31713(5,96)**	1	32624(8,59)	1	38072(21,67)	0	32158(7,26)
coord20-5-1b-2e	0	**19820(8,57)**	0	20329(10,86)	0	20029(9,53)	0	20329(10,86)
coord20-5-2-2e	1	46886(25,66)	1	46350(24,80)	1	40485(13,91)	1	**36655(4,91)**
coord20-5-2b-2e	1	27348(27,80)	0	27345(27,79)	1	37814(47,78)	0	**25092(21,31)**
coord50-5-1-2e	0	**36901(43,09)**	2	45215(53,55)	2	44874(53,20)	2	39933(47,41)
coord50-5-2-2e	2	54218(13,82)	1	53010(11,86)	1	**51839(9,87)**	1	53010(11,86)
coord50-5-2bBIS-2e	1	**31715(24,22)**	1	32858(26,86)	2	33055(27,29)	1	32858(26,86)
coord50-5-2BIS-2e	2	32671(18,74)	1	32668(18,73)	10	**32582(18,52)**	1	32668(18,73)
coord50-5-3-2e	1	41773(34,88)	1	41726(34,81)	2	**35535(23,45)**	1	35989(24,42)
coord100-5-1-2e	4	197096	2	**192741**	11	194026	2	**192741**
coord100-5-2-2e	2	217694	2	**217180**	10	217649	2	**217180**
coord100-5-3-2e	2	207631	2	205755	14	**204793**	2	205197
coord100-10-1-2e	3	275802	1	320904	13	**272930**	1	314142
coord100-10-2-2e	3	215783	2	220747	16	215182	1	**195688**
coord100-10-3-2e	2	193425	2	233956	3	190204	2	**189526**
coord200-10-1-2e	5	**387708**	3	459966	7	417922	3	410020
coord200-10-2-2e	4	422734	3	**422704**	11	436954	2	525280
coord200-10-3-2e	6	422670	3	380102	8	480593	4	**368830**
Average	2,61 S	(21,74%)	1,69 S	(23,35%)	4,38 S	(19,92%)	2,52 S	(16,08%)

TABLE 4.3: Computational results for Nearest Neighbor strategy

SBPCS-SI : Score Based Processing Center Selection - Scored Initialization.

SBPCS-RI : Score Based Processing Center Selection - Random Initialization.

NNPCS-SI : Nearest Neighbor Processing Center Selection - Scored Initialization.

NNPCS-RI : Nearest Neighbor Processing Center Selection - Random Initialization

Instances	BIH_1				BIH_2			
	$BIH_1 - NNCI$		$BIH_1 - RCI$		$BIH_2 - NNCI$		$BIH_2 - RCI$	
	Time	Cost(%)	Time	Cost(%)	Time	Cost(%)	Time	Cost(%)
coord20-5-1-2e	2	36705(5,65%)	1	**36551(5,25%)**	2	37199(6,90%)	2	42573(18,65%)
coord20-5-1b-2e	1	**19946(8,96%)**	1	19952(8,99%)	1	20629(11,98%)	1	26702(32,00%)
coord20-5-2-2e	1	**37039(4,20%)**	2	37476(5,32%)	3	37564(5,54%)	0	38008(6,64%)
coord20-5-2b-2e	2	**27436(26,26%)**	0	27912(27,52%)	1	28007(27,77%)	0	27994(27,73%)
coord50-5-1-2e	5	46481(35,77%)	6	**45784(34,79%)**	6	47051(36,55%)	4	47326(36,92%)
coord50-5-2-2e	4	**55224(10,74%)**	4	55656(11,43%)	5	57670(14,52%)	4	57222(13,85%)
coord50-5-2bBIS-2e	4	**34862(19,92%)**	5	35369(21,06%)	6	36465(23,44%)	4	37449(25,45%)
coord50-5-2BIS-2e	5	**34610(15,65%)**	6	35903(18,69%)	4	37122(21,36%)	5	37189(21,50%)
coord50-5-3-2e	5	**44666(11,92%)**	5	44826(12,24%)	6	47045(16,38%)	3	46533(15,46%)
coord100-5-1-2e	10	**198039(9,28%)**	9	199568 (9,98%)	15	202088 (11,10%)	6	202634(11,34%)
coord100-5-2-2e	10	**217799(8,25%)**	9	218459 (8,53%)	16	222830 (10,33%)	5	222432(10,17%)
coord100-5-3-2e	10	**209136(4,74%)**	9	210427 (5,33%)	16	214537 (7,14%)	6	213981(6,90%)
coord100-10-1-2e	11	**271464(22,48%)**	7	272681(22,83%)	19	277508(24,17%)	8	276443(23,88%)
coord100-10-2-2e	11	206995(8,61%)	6	**204574(7,52%)**	13	205600(7,99%)	7	205807(8,08%)
coord100-10-3-2e	9	**245693(26,09%)**	8	245966(26,17%)	19	250185(27,41%)	13	250219(27,42%)
coord200-10-1-2e	16	**349657**	15	350907	30	353933	26	358573
coord200-10-2-2e	19	**536103**	22	538763	40	543690	33	546601
coord200-10-3-2e	20	**370812**	23	371676	29	378365	25	378308
coord20-5-1-2e	1	**31830(6,31%)**	1	36551(18,41%)	3	31838(6,33%)	2	32791(9,05%)
coord20-5-1b-2e	0	**19857(8,74%)**	1	19952(9,18%)	0	20347(10,94%)	1	20927(13,41%)
coord20-5-2-2e	1	**36894(5,53%)**	2	37476(7,00%)	3	37419(6,85%)	3	46205(24,57%)
coord20-5-2b-2e	1	**25243(21,78%)**	0	27912(29,26%)	1	25854(23,63%)	2	28457(30,61%)
coord50-5-1-2e	5	**40984(48,76%)**	6	45784(54,13%)	7	42340(50,40%)	6	49073(57,20%)
coord50-5-2-2e	5	**52234(10,55%)**	4	55656(16,05%)	6	54620(14,45%)	5	54076(13,59%)
coord50-5-2bBIS-2e	5	**33555(28,38%)**	5	35369(32,05%)	6	35947(33,14%)	3	35005(31,34%)
coord50-5-2BIS-2e	5	**31722(16,31%)**	6	35903(26,06%)	6	34234(22,45%)	3	33220(20,08%)
coord50-5-3-2e	7	**37045(26,57%)**	5	44826(39,32%)	7	38089(28,58%)	4	38197(28,78%)
coord100-5-1-2e	14	**193382**	9	199568	10	196190	13	198093
coord100-5-2-2e	10	**218267**	9	218459	14	223042	9	221776
coord100-5-3-2e	13	**205256**	9	210427	13	210019	10	208456
coord100-10-1-2e	15	315304	7	**272681**	16	320729	9	319604
coord100-10-2-2e	15	**196437**	6	204574	11	201251	9	202025
coord100-10-3-2e	14	**190480**	8	245966	9	194256	6	195907
coord200-10-1-2e	35	413665	15	**350907**	34	421009	23	419356
coord200-10-2-2e	39	**527055**	22	538763	29	536500	23	536739
coord200-10-3-2e	33	**367286**	23	371676	25	375808	22	375460
Average	10,08 S	(16,31%)	7,66 S	(19,05%)	11,97 S	(18,72%)	8,47 S	(21,44%)

TABLE 4.4: Computational results for Insertion method

BIH : Best Insertion Heuristic. NNCI : Nearest Neighbor Client Insertion. RCI : Random Client Insertion.

Table 4.3 gives the results of *NNH* Algorithm with four strategies compared to the *LB*. In columns 2, 3, 4 and 5, we consider that the processing center is chosen according to the nearest neighbor procedure and in columns 6, 7, 8 and 9, we consider the best score as the criterion for opening processing centers. For both previous cases, we present the results in the case where the first processing center is selected randomly (columns 3 and 7) and in the case where the first processing center is chosen according to the score criterion (columns 5 and 9). The columns *Time* and *Cost* represent the computation time (seconds) and the value of the total cost, respectively. Results of table 4.3 show that on the average *NNH* finds a solution within 4 seconds. The minimum and maximum average *Gap* between the *NNH* solution and the *LB* are 16.08% and 23.35%, respectively. Furthermore, results of table 4.3 show that the strategy of choosing processing centers according to the score value (SBPCS-SI) is better on 11 instances out of 18 of the first set of instances and 7 instances out of 18 of the second set of instances. Since all processing centers in the second subset of the benchmark instances with the same *c*-product have a unique opening cost, the dominance of the score based processing centers strategy is less evident in the second set of instances. The score based processing center strategy is more efficient in solving the first subset of the benchmark. We can also note that when the score based processing centers strategy is not used (column 3 and 5) and Scored Initialization strategy is not too useful.

In table 4.4, the results of the BIH_1 strategy is reported in columns 2, 3, 4 and 5, and those of BIH_2 strategy are gathered in columns 6, 7, 8 and 9. For each strategy, we test two rules for selecting clients, namely, the Nearest Neighbour Client Insertion (NNCI) rule and Random Client Insertion (RCI). The results of *NNCI* are presented in columns 2, 3, 6 and 7 and those of *RCI* in columns 4, 5, 8 and 9. The columns *Time* and *Cost* present the computation time (seconds) and the total cost, respectively. From table 4.4, the best results are obtained by the strategy $BIH_1 - NNCI$ (31 best solutions among 36 instances). The maximum and minimum gaps between the solutions obtained by this strategy and LB are 48.7% and 4.2%, respectively and the average running time is 10 seconds. BIH_2 does not obtain any best solutions. This confirms that using a best insertion cost, as the unique criterion for processing center insertion is not enough. The strategy BIH_1 avoids unprofitable insertion positions. Several tests were performed to fix the value of α used in $BSIH$ (step 2 of section 4.4.2) and the best results were obtained with $\alpha = 0.9$.

Table 4.5 gathers the results of the four strategies for clustering *C-NNH, HCA-MA, HCA-AM, and HCA-CH*. In *C-NNH*, firstly *NNH* is applied in order to create the routes, then all nodes in each secondary route is considered as a Global-Cluster and Cplex solver is used for each Global-Cluster to create the secondary routes. In *HCA-MA* (HCA with Merge First and Assignment second), the step of merging clusters into Global-Cluster is done before the clusters assignment to processing centers. These two steps are reversed in *HCA-AM* (HCA with Assignment first and Merge second). In *HCA-CH* , the processing centers are included in the construction process of clusters. In this case, the customers clustering step starts with a processing center according to the score value.

Instances	Clustering Method			
	C-NNH	HCA-MA	HCA-AM	HCA-CH
coord20-5-1-2e	36853(6,02%)	35799(3,26%)	35799(3,26%)	**35312(1,92%)**
coord20-5-1b-2e	19826(8,41%)	19824(8,40%)	19834(8,45%)	**19332(6,07%)**
coord20-5-2-2e	36584(3,01%)	36123(1,77%)	36097(1,70%)	**35670(0,52%)**
coord20-5-2b-2e	27315(25,94%)	26737(24,65%)	**26734(24,33%)**	26770(24,43%)
coord50-5-1-2e	45769(34,77%)	43014(30,59%)	43014(30,59%)	**42139(6,33%)**
coord50-5-2-2e	55322(10,89%)	**51740(4,73%)**	52255(5,66%)	52366(5,86%)
coord50-5-2bBIS-2e	34300(18,60%)	**31466(11,27%)**	31938(12,58%)	31629(11,73%)
coord50-5-2BIS-2e	35556(17,90%)	**32192(9,32%)**	32222(9,40%)	32262(9,51%)
coord50-5-3-2e	45035(12,65%)	**41508(5,22%)**	43064(8,65%)	41675(5,60%)
coord100-5-1-2e	201283(10,75%)	**192376(6,61%)**	193815(7,31%)	192535(6,69%)
coord100-5-2-2e	217284(8,04%)	212746(6,08%)	214621(6,90%)	**212353(5,90%)**
coord100-5-3-2e	212641(6,31%)	**204080(2,38%)**	206128(3,35%)	204376(2,52%)
coord100-10-1-2e	-	**263998(20,29%)**	266003(20,89%)	266299(20,98%)
coord100-10-2-2e	-	**197236(4,08%)**	-	-
coord100-10-3-2e	-	**192582(5,70%)**	239958(24,32%)	-
coord200-10-1-2e	-	-	-	-
coord200-10-2-2e	-	-	-	-
coord200-10-3-2e	-	-	-	-
coord20-5-1-2e-2	32029(6,89%)	30982(3,74%)	**30980(3,73%)**	31009(3,82%)
coord20-5-1b-2e-2	20280(10,65%)	19808(8,52%)	19808(8,52%)	**19285(6,04%)**
coord20-5-2-2e-2	36077(3,39%)	**35454(1,69%)**	35947(3,04%)	47109(26,01%)
coord20-5-2b-2e-2	24653(19,91%)	25049(21,17%)	25052(21,18%)	**24580(19,67%)**
coord50-5-1-2e-2	38961(46,10%)	**37314(43,72%)**	38316(45,19%)	37409(43,86%)
coord50-5-2-2e-2	50932(8,26%)	**49148(4,93%)**	50157(6,84%)	49337(5,29%)
coord50-5-2bBIS-2e-2	32858(26,86%)	**30103(20,16%)**	30593(21,44%)	30684(21,68%)
coord50-5-2BIS-2e-2	32685(18,78%)	29322(9,46%)	29271(9,30%)	28917(8,19%)
coord50-5-3-2e-2	37262(27,00%)	**33194(18,05%)**	34170(20,39%)	33783(19,48%)
coord100-5-1-2e-2	192746	**187821**	189768	188085
coord100-5-2-2e-2	217268	212659	-	**212362**
coord100-5-3-2e-2	205192	**199624**	-	199776
coord100-10-1-2e-2	314032	**307634**	-	307958
coord100-10-2-2e-2	195560	**190501**	-	190794
coord100-10-3-2e-2	189263	**185085**	-	-
coord200-10-1-2e-2	-	-	-	-
coord200-10-2-2e-2	-	-	-	-
coord200-10-3-2e-2	-	-	-	-
Average	(15,77%)	(10,49%)	(13,35%)	(11,91%)

TABLE 4.5: Computational results for Clustering method

C_NNH : Clustering Nearest Neighbor Heuristic. HCA_MA : Hybrid Clustering Algorithm(First Merging second Assigning). HCA_AM : Hybrid Clustering Algorithm(First Assigning second Merging). HCA_CH : Hybrid Clustering Algorithm (Client_Hub)

In table 4.5, columns 2, 3, 4 and 5 provide the results of *C-NNH, HCA-MA, HCA-AM*, and *HCA-CH*, respectively. Preliminary experimentations confirmed that the solution returned by CPLEX for the routing step of *HCA* remains stable after 10 minutes. We then set a time limit equal to 10 minutes for all instances. Note that, for some large instances $n \geq 100$, Cplex cannot generate any solution.

The minimum and maximum *Gap* between the *C-NNH* solution and *LB* solution are 3% and 46.1%, respectively. These values are equal to 1.7% and 43.7% for *HCA-MA*, 1.7% and 45.1% for *HCA-AM* and 0.5% and 43.8% for *HCA-CH*. The *HCA-MA* method obtains 19 best results versus 2 for *HCA-AM*, 8 for *HCA-CH*, and 0 time for *C-NNH*. Note that *HCA-AM* gives a weak result in comparison with *HCA-MA* because when we fix the assignment before merging, the algorithm fails to form good global clusters. *C-NNH* gives bad results for all instances compared to the other strategies of HCA. The use of distances between the centers of gravity as a criterion to form mono-product clusters provides better results than the use of a nearest neighbour approach. Note that the gaps for solutions to the second set of instances are similar to those for the first set. This issue shows that increasing similarity in opening cost of the processing centers does not influence on the results.

Table 4.6 summarizes the results of all proposed methods. The dominant strategy for each method has been reported. The results of the best insertion strategy corresponding to $BIH_1 - NNCI$ are presented in columns 2 and 3. The results of the best strategy of *NNH* corresponding to score based processing center selection and scored initialization of table 4.3 are reported in columns 4 and 5. The results of the best strategy of *HCA* (*HCA-MA*) are gathered in columns 6. The value in parenthesis for each strategy represents the gap compared to the best obtained solution.

The results of table 4.6 confirm that the clustering approach is more competitive, it finds all the best results for $n < 200$. For the large instances, i.e., $n \geq 200$, the clustering method fails to provide the best results and it is not able to provide any solutions within 10 mins. In these cases, the insertion method is more competitive.

Table 4.7, is similar to table 4.6 except that the reported results for each method are the best solutions found by all tested strategies. The results of table 4.7 confirm that the clustering approach is the best approach when $n < 200$ and the insertion approach is more efficient for large instances.

Instances	Dominant Strategy				
	$\text{BIH}_1 - NNCI$		SBPCS-SI		HCA-MA
	Time	Cost(%)	Time	Cost(%)	Cost(%)
coord20-5-1-2e-2	2	36705(2,47%)	0	37447(4,40%)	**35799(0%)**
coord20-5-1b-2e-2	1	19946 (0,61%)	0	19877 (0,27%)	**19824 (0%)**
coord20-5-2-2e-2	1	37039 (2,47%)	1	36800 (1,84%)	**36123 (0%)**
coord20-5-2b-2e-2	2	27436 (2,55%)	3	27941 (4,31%)	**26737 (0%)**
coord50-5-1-2e-2	5	46481 (7,46%)	4	44125 (2,52%)	**43014 (0%)**
coord50-5-2-2e-2	4	55224 (6,31%)	3	55242 (6,34%)	**51740 (0%)**
coord50-5-2bBIS-2e	4	34862 (9,74%)	4	34174 (7,92%)	**31466 (0%)**
coord50-5-2BIS-2e-2	5	34610 (6,99%)	2	35556 (9,46%)	**32192 (0%)**
coord50-5-3-2e	5	44666 (7,07%)	3	43859 (5,36%)	**41508 (0%)**
coord100-5-1-2e	10	198039 (2,86%)	4	196936 (2,32%)	**192376(0%)**
coord100-5-2-2e	10	217799 (2,32%)	5	217352 (2,12%)	**212746(0%)**
coord100-5-3-2e	10	209136 (2,42%)	4	209812 (2,73%)	**204080(0%)**
coord100-10-1-2e	11	271464 (2,75%)	4	271890 (2,90%)	**263998(0%)**
coord100-10-2-2e	11	206995 (4,71%)	5	199861 (1,31%)	**197236(0%)**
coord100-10-3-2e	9	245693 (21,62%)	4	243523 (20,92%)	**192582(0%)**
coord200-10-1-2e	16	**349657(0%)**	5	351502 (0,52%)	-
coord200-10-2-2e	19	536103 (0,35%)	6	**534204(0%)**	-
coord200-10-3-2e	20	**370812(0%)**	8	374447 (0,97%)	-
coord20-5-1-2e	1	31830 (2,67%)	0	32158 (2,31%)	**30982 (0%)**
coord20-5-1b-2e	0	19857 (2,88%)	0	20329 (2,70%)	**19808 (0%)**
coord20-5-2-2e	1	36894 (3,90%)	1	36655 (3,28%)	**35454 (0%)**
coord20-5-2b-2e	1	25243 (2,63%)	0	25092 (2,04%)	**25049 (0%)**
coord50-5-1-2e	5	40984 (9,96%)	2	**39933 (0%)**	37314 (1,10%)
coord50-5-2-2e	5	52234 (5,91%)	1	53010 (5,19%)	**49148 (0%)**
coord50-5-2bBIS-2e	5	33555 (8,83%)	1	32858 (3,54%)	**30103 (0%)**
coord50-5-2BIS-2e	5	31722 (8,84%)	1	32668 (11,25%)	**29322 (0%)**
coord50-5-3-2e	7	37045 (10,40%)	1	35989 (6,59%)	**33194 (0%)**
coord100-5-1-2e	14	193382 (2,74%)	2	192741 (2,42%)	**187821(0%)**
coord100-5-2-2e	10	218267 (2,71%)	2	217180 (2,22%)	**212659(0%)**
coord100-5-3-2e	13	205256 (2,74%)	2	205197 (2,52%)	**199624(0%)**
coord100-10-1-2e	15	**315304(0%)**	1	314142 (0,09%)	307634 (11,36%)
coord100-10-2-2e	15	196437 (3,02%)	1	195688 (2,65%)	**190501(0%)**
coord100-10-3-2e	14	190480 (2,83%)	2	189526 (2,34%)	**185085(0%)**
coord200-10-1-2e	35	**413665(0%)**	3	410020 (9,49%)	-
coord200-10-2-2e	39	527055 (19,79%)	2	**525280 (0%)**	-
coord200-10-3-2e	33	**367286(0%)**	4	368830 (0,4%)	-
Average	10,08 Seconds	(5,44%)	2,52 Seconds	(4,39%)	(0,72%)

TABLE 4.6: Comparison of the Dominant Strategy for each method

BIH : Best Insertion Heuristic. NNCI : Nearest Neighbor Client Insertion. HCA_MA : Hybrid Clustering Algorithm(First Merging second Assigning). SBPCS-SI : Score Based Processing Center Selection - Scored Initialization

Instances	Best Solution				
	Insertion method		NNH		HCA
	Time	Cost(%)	Time	Cost(%)	Cost(%)
coord20-5-1-2e	2	36551(3,38%)	0	37447 (5,70%)	**35312 (0%)**
coord20-5-1b-2e	1	19946(3,07%)	0	19877 (2,74%)	**19332 (0%)**
coord20-5-2-2e	1	37039(3,69%)	1	36800 (3,07%)	**35670 (0%)**
coord20-5-2b-2e	2	27436(2,55%)	3	27937 (4,31%)	**26734 (0%)**
coord50-5-1-2e	5	45784(23,99%)	4	**34800 (0%)**	42139 (17,41%)
coord50-5-2-2e	4	55224(6,30%)	3	54395 (4,88%)	**51740 (0%)**
coord50-5-2bBIS-2e	4	34862(9,74%)	4	34174 (7,92%)	**31466 (0%)**
coord50-5-2BIS-2e	5	34610(6,98%)	2	35479 (9,26%)	**32192 (0%)**
coord50-5-3-2e	5	44666(7,07%)	3	43859 (5,36%)	**41508 (0%)**
coord100-5-1-2e	10	198039(2,85%)	4	196936 (2,32%)	**192376 (0%)**
coord100-5-2-2e	10	217799(2,50%)	5	216913 (2,10%)	**212353 (0%)**
coord100-5-3-2e	10	209136(2,41%)	4	209812 (2,73%)	**204080 (0%)**
coord100-10-1-2e	11	271464(2,75%)	4	271890 (2,90%)	**263998 (0%)**
coord100-10-2-2e	11	204574(3,59%)	5	199861 (1,31%)	**197236 (0%)**
coord100-10-3-2e	9	245693 (21,62%)	4	219894 (12,42%)	**192582 (0%)**
coord200-10-1-2e	16	**349657 (0%)**	5	351502 (0,52%)	-
coord200-10-2-2e	19	536103 (3,10%)	6	**519466 (0%)**	-
coord200-10-3-2e	20	**370812 (0%)**	8	374447 (0,97%)	-
coord20-5-1-2e	1	31830 (2,67%)	0	31713 (2,31%)	**30980 (0%)**
coord20-5-1b-2e	0	19857 (2,88%)	0	19820 (2,70%)	**19285 (0%)**
coord20-5-2-2e	1	36894 (3,90%)	1	36655 (3,28%)	**35454 (0%)**
coord20-5-2b-2e	1	25243 (2,63%)	0	25092 (2,04%)	**24580 (0%)**
coord50-5-1-2e	5	40984 (9,96%)	2	**36901 (0%)**	37314 (1,10%)
coord50-5-2-2e	5	52234 (5,91%)	1	51839 (5,19%)	**49148 (0%)**
coord50-5-2bBIS-2e	5	33555 (8,83%)	1	31715 (3,54%)	**30593 (0%)**
coord50-5-2BIS-2e	5	31722 (8,84%)	1	32582 (11,25%)	**28917 (0%)**
coord50-5-3-2e	7	37045 (10,40%)	1	35535 (6,59%)	**33194 (0%)**
coord100-5-1-2e	14	193382 (2,74%)	2	192741 (2,42%)	**188085 (0%)**
coord100-5-2-2e	10	218267 (2,71%)	2	217180 (2,22%)	**212362 (0%)**
coord100-5-3-2e	13	205256 (2,74%)	2	204793 (2,52%)	**199624 (0%)**
coord100-10-1-2e	15	**272681 (0%)**	1	272930 (0,09%)	307634 (11,36%)
coord100-10-2-2e	15	196437 (3,02%)	1	195688 (2,65%)	**190501 (0%)**
coord100-10-3-2e	14	190480 (2,83%)	2	189526 (2,34%)	**185085 (0%)**
coord200-10-1-2e	35	**350907 (0%)**	3	387708 (9,49%)	-
coord200-10-2-2e	39	527055 (19,79%)	2	**422704 (0%)**	-
coord200-10-3-2e	33	**367286 (0%)**	4	368830 (0,4%)	-
Average	11,09 Seconds	(6,25%)	3,02 Seconds	(4,41%)	(0,42%)

TABLE 4.7: Comparison of the best obtained results by all strategies

NNH : Nearest Neighbor Heuristic. HCA : Hybrid Clustering Algorithm.

4.7 Conclusion

This chapter addresses a new extension of the two-echelon location routing problem by including new constraints that have not been considered simultaneously in the literature, namely, multi-product, pickup and delivery and the use of the processing center as intermediate facility in the second-level routes. This new variant, named 2E-MPLRP-PD (The Two-Echelon Multi-products Location-Routing problem with Pickup and Delivery), has many realistic applications in supply chain. We have proposed a mixed integer linear model for the problem and used Cplex Solver to solve small-scale instances. Furthermore, we have proposed non-trivial extensions of nearest neighbour and insertion approaches. We have developed clustering based approaches that have not been extensively investigated with regards to location routing. Four strategies were tested for each method. These methods are tested on instances derived from LRP instances with up to 200 customers and 10 processing centers. Extensive computational experiments show that the clustering approach is very competitive and outperforms the other heuristics when $n < 200$. For larges instances, the insertion approach is more efficient. In the next chapter, we will present local search methods to improve the heuristics results and we will investigate a metaheuristic approaches for 2E-MPLRP-PD.

In further researches, we aim to improve HCA with metaheuristic techniques and an iterative process. For instance, we can use a metaheuristic instead of Cplex to solve the routing problem and restart HCA several times with a different initial solution. It will also be interesting to develop more efficient lower bounds. Another perspective is to include more real-life constraints to deal with some other realistic constraints, such as splitting of demand per product and the use of non-specialized processing centers.

Chapitre 5

Improvements methods for the 2E-MPLRP-PD

Contents

5.1 Introduction

In previous chapters we have proposed different heuristics in order to solve our two-echelon multi products location-routing problem with pickup and delivery. In this chapter, we investigate an improvement method based on local search and metaheuristic called *Iterative Local Search (ILS)*. This metaheuristic is widely developed to solve VRP and rarely used for Location-Routing Problems. The review of the literature shows that the ILS method yields to good results on some routing problem variants. The only works in the literature that use ILS application on LRP are those of Derbel et al. [75], and Nguyen et al. [182].

We have proposed an *ILS* method based on local search methods integrating different types of neighborhood search, and a specific perturbation procedures. The computational results are more promising compared to the result obtained by the heuristic methods.

Section 5.2 presents the applied local search methods. Section 5.3 describes the general principle of ILS and outlines of ILS methods from the literature applied to LRPs.

Section 5.4 details the general algorithm of our ILS developed for the *2E-MPLRP-PD*. The following section 5.5 describes the proposed general algorithms. In section 5.6 we discuss the results of an extensive computational experiment carried out with two purposes : (i) to identify the key features of the algorithm and (ii) to determine the set of parameter values that produce the best algorithm performance. We conclude the chapter in section 5.7.

5.2 Local Search methods for the 2E-MPLRP-PD

In this section, we develop local search methods for 2E-MPLRP-PD.

The aim of the local search approach (LS) is to improve a feasible solution by exploring a neighborhood $N(S)$ of the current solution S. $N(S)$ is the set of solutions that can be obtained by a simple transformation of the solution S, by using pre-defined moves. The local search approach seeks for the first neighbor from $N(S)$ that can be obtained by an improving moves *(first improvement)* or by the best neighbor *(best improvement)*. If a neighbor S' is better than S, LS explores the neighborhood of S'. It stops when no improving neighbor is possible. Several moves have been proposed in the literature for vehicle routing problems. The most classical and known moves are those developed for the Traveling Salesman Problem (TSP). These methods consist in moving the position of a node in the cycle, exchange of nodes, k-OPT (Lin and Kernighan, [157]), Or-OPT (Or, [187]). An overview of local search procedures, applied to routing problems can be found in Ibaraki et al. [125].

In this section, we propose two local search methods ; «the routing improvement local search» and «the processing centers location improvement local search». The first one consists of three movements : *Intra-Route Multi Product Two-Opt Local Search (2-OPT-MP1), Inter-Route Multi Product Two-Opt Local Search (2-OPT-MP2)* and *Merging Pro-*

duct Local Search (MP). These moves are related to clients and intermediate processing centers but do not affect the processing centers used as departure node of routes. The second method "processing centers location improvement procedure" contains moves that affect the processing centers of routes. It acts on the set of opened processing centers. These movements are called «*change-state-PC*» and «*Swap-routes-PC*». The «*Change-State-PC*» mutates the state of two processing centers, and the *Swap-routes-PC* swaps the position of two processing centers used as departure node of routes. Note that the presence of pickup and delivery and multi-products distribution requires checking several feasibility constraints.
In the following, before explaining the proposed local search procedures, we will present the «Adjusting-loading procedure» that we use to check the feasibility of each solution. Let S represents the current solution.

5.2.1 Adjusting-Loading procedure

Each sequence between two processing center in route k is considered as a macro-client i with demand Q_i. The position of each processing center and each macro-client is updated according to k. Let l_n be the last position of macro-client i in the route k. The goal of the algorithm is to check if the demand Q_i of a macro-client i (in position l_i) can be satisfied by one or more processing center j (in position l_j, with $l_j \lneq l_i$) while satisfying vehicle capacity and processing center availability. The algorithm tries to allocate the demand of the macro-client i by considering the first available processing center j ($l_j \lneq l_i$). For each processing center j, let $pickup(j,i)$, the assigned quantity for macro-client i. If $pickup(j,i) \geq Q_i$, the algorithm moves to the next macro-client in position $l_i = l_i + 2(l_i \leq l_n)$. If $pickup(j,i) \lneq Q_i$, the macro-client i is partially satisfied. The algorithm allocates $pickup(j,i)$ to macro-client i and tries to allocate the remaining demand of client i ($Q_i - pickup(j,i)$) by considering the next processing center at position $l_j + 2(l_j + 2 \leq l_i)$. The algorithm stops either when 1) it does not succeed to affect a demand of a macro-client i. In this case, the solution is infeasible, or 2) it succeeded to allocate all demands of all macro-clients. In this case, the algorithm updates the pickup's quantity at each processing center in the route k.
Before explaining the proposed local search, we specify some notations used in the following sections. Let k_1 and k_2 be two routes of S, D $\subseteq N_0$ the set of processing centers, used as departure node of routes, $I \subseteq N_0$ the set of processing centers used as intermediate processing centers. Note that we can have $(D \cap I) \neq \emptyset$. Let spc_1 and spc_2 (spc_1 and $spc_2 \in D$) the departure node of k_1 and k_2, respectively and ipc_1 and $ipc_2 \in I$, be intermediate processing centers, in routes k_1 and k_2, respectively. We denote i, j nodes

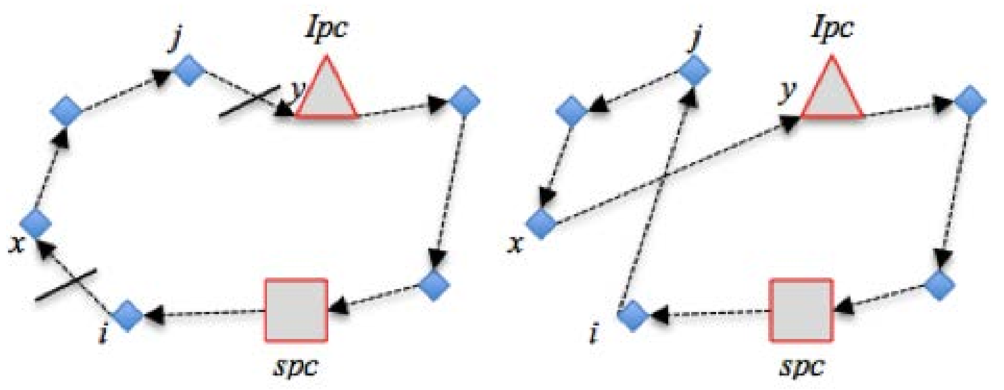

FIGURE 5.1 – Example of *2-Opt-MP1(S,k_1,i,j)* Local Search method. The edges (i, x) and (j, y) in the left route are deleted and a new route (in the right) is created.

(either clients or intermediate processing center) in given route. Let x (resp. y) be the successor of i (resp. j) in the route k_1 (resp. k_2). The moves are made only if they lead to a feasible solution and lowest cost.

5.2.2 Routing Improvement Local Search methods

The routing improvement local search methods doesn't change the location of processing centers in D but can change the position of all processing centers. The set of the opened processing centers can be decreased when a move eliminates an intermediate processing center *ipc*, with $ipc \notin D$.

5.2.2.1 The Intra-Route Multi Product Two-Opt Local Search Method

The Intra-Route Multi Product Two-Opt local search *2-Opt-MP1(S,k_1,i,j)* involves one route. It can change the position of customers, intermediate-processing centers, and the allocation of clients to intermediate processing centers.
The *2-Opt-MP1(S,k_1,i,j)* performs several iterations. In each iteration, it scans all pairs of node (i, j) in route k_1 and deletes the arcs (i, x) and (j, y) from the route. The resulting sequences are connected in order to form a new route (Fig. 5.1). The adjusting-loading procedure is then applied to update the quantity of pickups at intermediate processing centers and to verify the feasibility of the moves. The algorithm stops when iteration reports no improvement.

5.2.2.2 The Inter-Route Multi Product Two-Opt Local Search method

The Inter-Route Multi-Product Two-Opt Local Search *2-Opt-MP2(S,k_1, k_2,i,j)* involves two routes. It extends the 2-OPT concept for multi-depot proposed by Prodhon (2010) to deal with the multi-product and the pickup delivery constraints.
The *2-Opt-MP2(S,k_1, k_2,i,j)* performs several iterations. In each iteration, it scans all pairs of node i in k_1 and j in k_2 ($k_1 \neq k_2$). The arcs (i, x) and (j, y) are deleted from the routes, and the resulting sequences are connected to form two new routes. After the fusion, the new routes may contain a double occurrence of a given client i. These occurrences correspond to the delivery of different products. For example in the case where the number of products is equal to two, let l_1 (resp. l_2) be the occurrence of the client i in route k_1 (resp. in route k_2). The position l_1 (resp. l_2) corresponds to the delivery of product P_1 in k_1 (resp. P_2 in k_2). After the fusion, if a route contains the two occurrences l_1 and l_2 of client i, these two occurrences must be merged in either l_1 or l_2. If α is the number of customers with two occurrences in new obtained route after fusion ($\alpha \geq 0$), the number of all possible ways to merge the identical clients in the new route is at most (2^α). This is why we decided to limit the number of fusions to be explored by the current move.
Let *LRouteDouble* be the set of route k' obtained after fusion with at least one client in double, and L' the list of clients in double in a new route k'. Since the number of possible fusions can be exponential, if $\alpha \gneqq 2$, we have decided to keep a limited number of fusions equal to 2 for each route in *LRouteDouble*. To do this, L' is ordered in decreasing order according to the quantity. This quantity is calculated per client as the quantity's sum of its occurrences. For the client in the top of L', we consider the two possible merging. For other clients in L', one merging position is chosen randomly. This gives us two possible fusions per new route k'.
If $\mid LRouteDouble \mid = 0$ and k_1 and k_2 have the same starting processing center, there is 2 ways to connect the routes. For example, this corresponds to the case where $spc_1 = spc_2$ in Fig. 5.2. The resulting routes will be affected to the same processing center. Otherwise if $spc_1 \neq spc_2$, there is 4 ways to connect the routes (Fig. 5.2). If $\mid LRouteDouble \mid \neq 0$, we will consider two fusions per new route k' as explained above.
The *adjusting-loading procedure* (described in section 5.2.1) is then applied to each fusion. This procedure updates the quantity of pickups and verifies the feasibility of the moves. The algorithm stops when iteration reports no improvement.
In Fig. 5.3 clients c_1 and c_2 received their c-product P_1 and P_2 from two different routes. After applying *2-Opt-MP2*, the edges (i, x) and (j, y) were removed and two new routes are created. In the left one (bold route in Fig.5.3) clients c_1 and c_2 have two occurrences

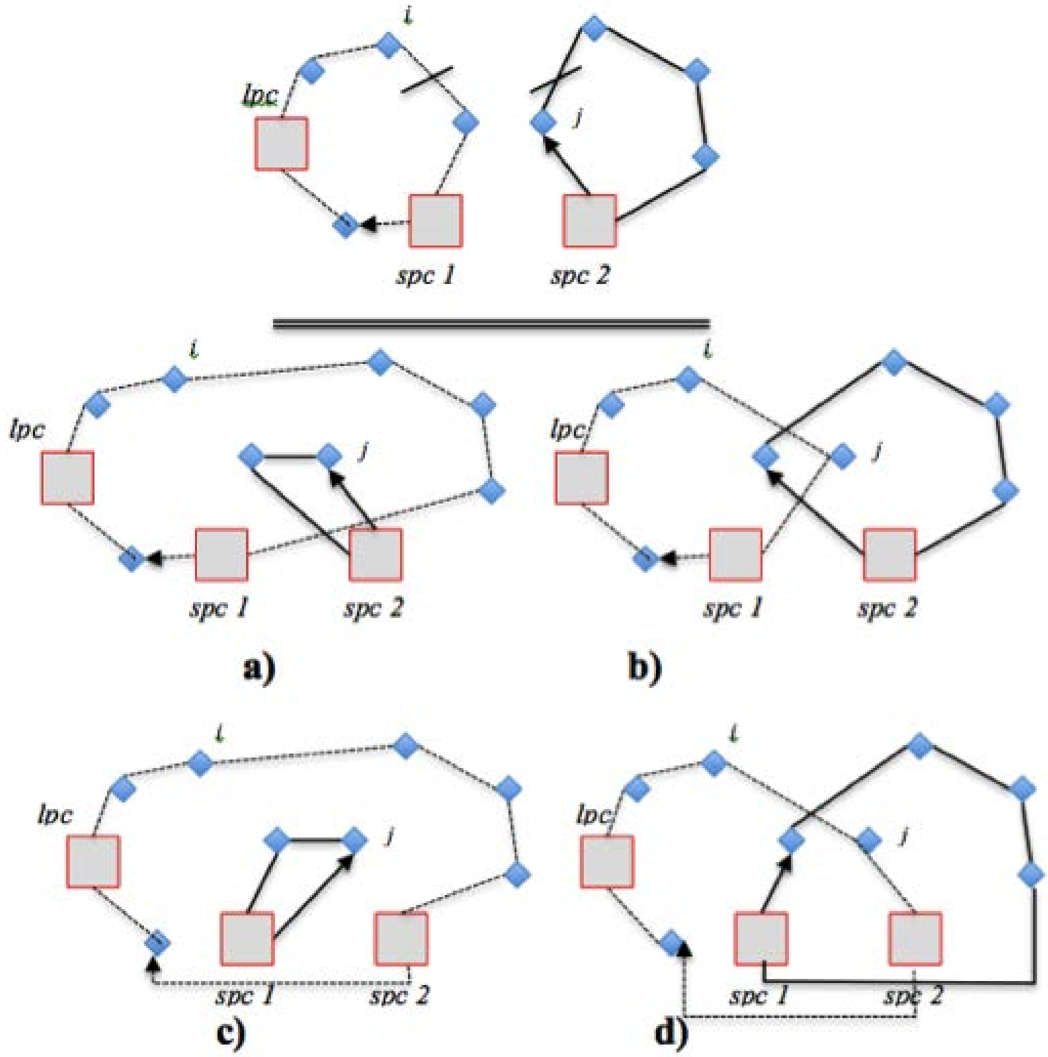

Figure 5.2 – *2-Opt-MP2* Method in the case : $spc_1 \neq spc_2$ and $\mid LRouteDouble \mid = 0$. The edges (i, x) and (j, y) are moved. The routes in c (resp. in d) are created by exchanging the processing center spc_1 and spc_2 in routes a (resp. in b). We achieve four possibilities of new routes.

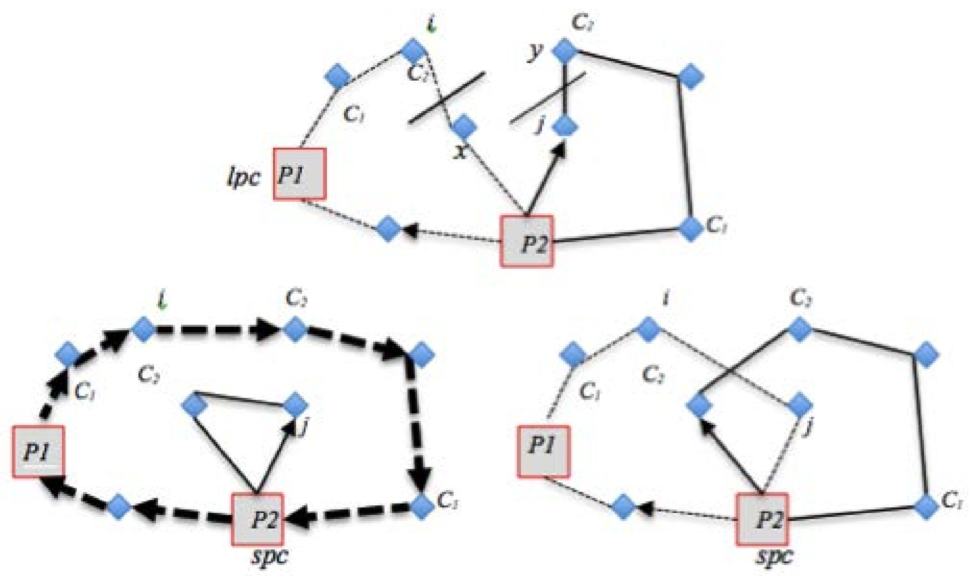

FIGURE 5.3 – *2-Opt-MP2* Method in the case : $spc_1 = spc_2$, and $\mid LRouteDouble \mid = 1$, $L' = \{c_1, c_2\}$. In bold route, $\alpha = 2$ and $\alpha = 0$ for the other routes.

in the same route. Figure 5.4 shows all possible ways to merge c_1 and c_2.

5.2.2.3 The Merging-Product Local Search method

The Merging-Product Local Search $MP(S, i)$ is applied to each client i. It tries to decrease the number of deliveries for each client. For example, if a client i is delivered by routes $(k_1, k_2, k_3, ..., k_m)$ for each demand c-product and the positions of client i in these routes are $(l_1, l_2, l_3, ..., l_m)$. The Local Search MP, at the first step, considers the merging of the m deliveries in all positions $(l_1, l_2, l_3, ..., l_m)$. If this merging leads to a feasible solution, the best fusion is saved. The process continues by considering the merging of $m - 1$ deliveries, chosen randomly in $(k_1, k_2, ..., k_m)$, then the merging of $m - 2$ deliveries, until exploring the merging of two deliveries.
For example, at the top of Fig. 5.5 client c receives its demand P_1 by route R_1 and demand P_2 by route R_2. At the bottom of figure 5.5, the MP method merges the two demands P_1 and P_2 of client c in the route R_2.

5.2.3 Processing centers location improvement local search method

These procedures contain the moves that affect the location of processing centers.

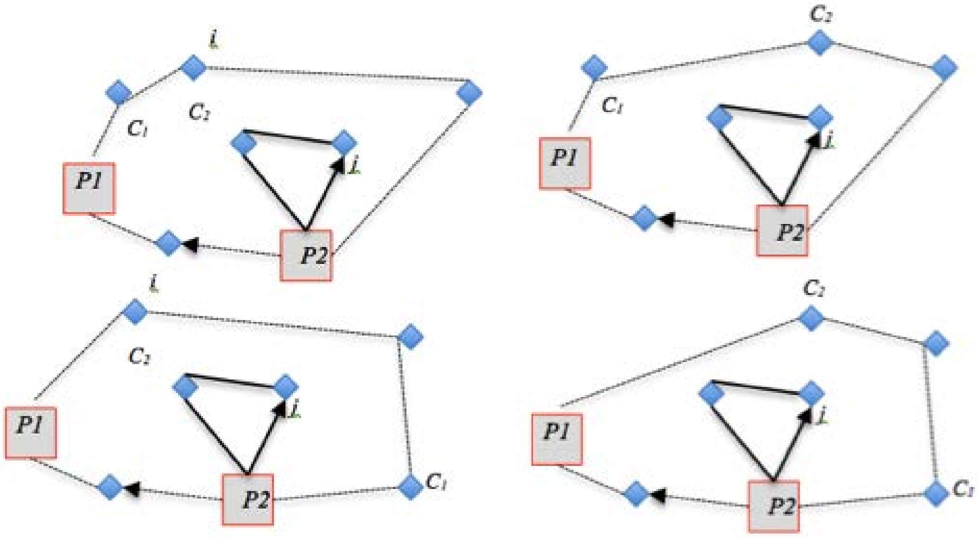

FIGURE 5.4 – All possible fusions for the bold route from Fig.5.3.

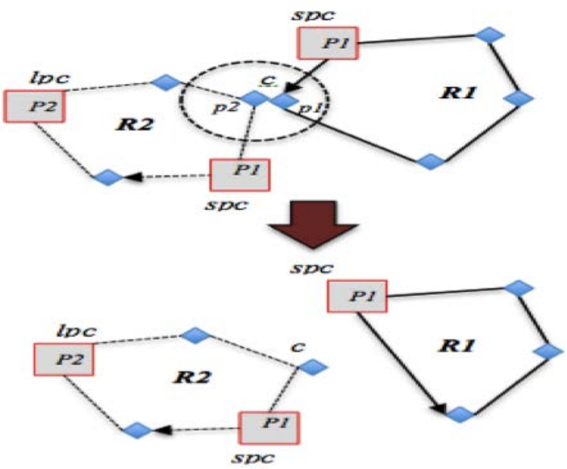

FIGURE 5.5 – Example of Merging Product Local Search.

5.2.3.1 $change - state - PC(S, pc_1, pc_2)$:

This Local Search method closes an opened processing center $pc_1 \in (D \cup I)$ and opens a closed processing center pc_2. The routes (or only sequence of clients) are transferred from pc_1 to pc_2. These routes are then optimized by *2-Opt-MP1* local search. The cost of this move may include the variation of the routing cost of the first echelon routes.

5.2.3.2 $Swap - routes - PC(S, pc_1, pc_2)$:

This Local Search method swaps two opened processing centers pc_1 and pc_2 (pc_1 and $pc_2 \in D$). The routes are then optimized by *2-opt-MP1* local search.

5.3 Iterative Local Search

An ILS Meta-heuristic consists of four phases : initial solution generation, local search procedure, perturbation, and stopping criterion. Let s_0 be the initial solution for the ILS method which is generated and then a local search heuristic is applied to reach a better solution s^*. The solution s^* will be perturbed randomly by using some modifications on the current local optimal solution (perturbation). Instead of generating a new initial solution, the perturbation mechanism generates a promising initial solution and this phase provides an intermediate solution $s^{'}$. The local search is then again applied to solution $s^{'}$. This procedure leads to reaching a new local optimum solution, denoted $s^{''}$. If the resulting solution $s^{''}$ outperforms the incumbent solution s^* according to an acceptance criterion, solution $s^{''}$ replaces s^*. This process is repeated until a given stopping criterion is met (see Algorithm 1).

Despite simplicity of ILS, the algorithm has proven to be a very successful approach to solve combinatorial optimization problems (Stutzle [233] ; Voudouris and Tsang [253] ; Vansteenwegen et al. [252]). A detailed explanation of the ILS metaheuristic can be found in Lourenco et al. [148]. A small number of ILS methods were proposed to solve LRP in the literature. The only methods that we have found those of Derbel et al. [75] for LRP and Nguyen et al. [182] to solve a 2E-LRP.

Derbel et al. [75] consider a LRP with capacitated depots and a single un-capacitated vehicle dispatched at each open depot such that the demand of each customer can be satisfied. The authors developed an iterated local search. They also suggest four neighborhood structures : N1, N2, N3 and N4, where N1 and N2 are performed between two different routes with an attempt to improve the solution by moving customers between depots, whereas N3 and N4 are developed inside the same route. The neighborhood N1 creates solutions with an interchange of two clients between two routes which disturbs the route while keeping the number and position of clients assigned to each depot. To create solutions by N2, they take up a client from one route and insert it into another one. In order to move from one solution to another one in the neighborhood N3, they switch the position of two clients inside the route. As for the neighborhood N4, they insert a client between two other clients in the same route. The perturbation procedure is related to the depot. A depot is selected at random, and all assigned customers to the selected depot are inserted randomly to another depot either opened or closed.

The local search is applied to a given initial solution according to the first neighborhood structure N1. The obtained local optimum will be the starting point for the new local search with respect to the second neighborhood structure N2 and so on. This process is repeated until a local optimum of the four structures of neighborhood is reached.

The same authors [76] propose also an issue to use their ILS method to intensify the search space of genetic algorithm. Numerical experiments show that the hybrid algorithm improves, for all instances, the best known solutions previously obtained by the tabu search heuristic.

Another iterated local search is proposed by Nguyen et al. [182] to solve a two-echelon location-routing problem. The latter work presents an ILS by using a tabu list to avoid the local optimum. Three greedy randomized heuristics are used to get initial solutions. Each ILS run alters between two search spaces in order to perturbate the solution. The authors propose to transform the set of routes into one *giant route* and perform some movements on this giant route. The perturbation procedure considers three kinds of moves. The first move randomly selects certain pairs of clients and swaps them. The second move determines the four longest edges and deletes two at random. The substring between the two selected edges is removed and a cheapest insertion in the remaining sequence is performed. The last move adds the edge linking the first and last clients and then breaks the resulting cycle by randomly deleting one of the four longest edges. Each call for perturbation procedure randomly selects one of these move types. As improvement procedure, six moves for the first and second levels were proposed : Relocate, Exchange, 2-Opt, 3-Opt, Or-Opt, and Reinsert. The method is competitive since it outperforms two GRASP algorithms and one tabu search on the LRP-2E. In a comparison with four published CLRP metaheuristics in the literature, only one Langrangean relaxation and granular tabu search method (LRGTS) by Prins et al. [196] does better.

Algorithm 1 The pseudo code of the basic ILS

(1) Generate Initial Solution sol_0
(2) sol^*=LocalSearch (sol_0)
repeat
    (3) $sol^{'}$=Perturbation (sol^*)
    (4) $sol^{''}$=LocalSearch $(sol^{'})$
    (5) sol^*=AcceptanceCriterion $(sol^*, sol^{''})$
until Termination condition =0

5.4 ILS for 2E-MPLRP-PD

We developed the ILS which uses the *Nearest Neighbor Heuristic* (NNH) and a randomized version of our insertion heuristic called *Best Sequential Insertion Heuristic* (BSIH), addressed in chapter 4 as initial solutions, and three local search procedures. (1) The Intra-Route Multi-Products Two-Opt Local Search Method (*2-OPT-MP1*), (2) The Inter-Route Multi-Products Two-Opt Local Search method (*2-OPT-MP2*), (3) The Merging-Products Local Search method (*MP*) and (4) the change-state-PC, detailed in second section of this chapter. In this section, to complete the required components for ILS, we propose two perturbation procedures, called, *client-perturbation(S,S')* and *sequence-perturbation(S,S')*. In setup parameters section, different number of iterations and also different value of parameters, used in perturbation procedures, are tested. By evaluating the results, we fix these parameters to reach the best performance of ILS algorithm.

5.4.1 Initial solutions

The heuristics that have already been presented in the previous chapter will be considered as the initial solutions. The main of heuristic is briefly reminded below.

5.4.1.1 NNH Algorithm

At the first stage of NNH heuristic, a processing center is selected as departure node to start a new route, according to one calculated score. Then, a nearest neighbour candidate is selected. A checking feasibility procedure is applied to verify vehicle capacity constraint. If the selected candidate cannot be inserted in the current route, the second nearest neighbour candidate from the last inserted node will be checked. Once we cannot insert neither processing center nor client, the current route is closed and we create a new route. This process is repeated until all client requests are satisfied. The same process is applied to the first vehicle routes.

5.4.1.2 Best Sequential Insertion Heuristic (BSIH)

The insertion method that is applied to the second level routes consists of two phases : (i) initialization phase and (ii) insertion phase (Figure. 4.4). The initialization phase determines the first processing center in each new route. The second phase tries to insert the maximum number of clients and processing centers into the current route. The algorithm selects the best position according to the time of route for each insertion. The two phases are repeated until the total customer demand is satisfied. The client and processing center insertion will be repeated until the route time limit is reached. If all clients requests are satisfied the algorithm stops ; otherwise, a new route is initialized by selecting a processing center. The routing between the selected processing centers for the first level (first-level routes) is obtained by the nearest neighbour heuristic described in section 4.4.1.

5.4.2 Perturbation procedure

We proposed two perturbation specific procedures to solve our problem. These procedures try to make some modifications on routes in S and then create a new solution S'.
The first one is named *client-perturbation* and is related to client moves either in the same or in different route. The second perturbation concerns a re-insertion of a set of clients that appear between two processing center into the same route. This perturbation, called *sequence-perturbation*, can eliminate in some cases an intermediate processing center if all customers in the sequence are removed and assigned to another processing centers.
These two procedures are applied to the second level route. For the first level route only the first procedure is applied.

5.4.2.1 *Client-perturbation(S,S')*

This procedure aims at applying α successive moves of clients chosen randomly in S. We remind here that each client (i, S), such that $S = \{s_1, s_2, ..., s_n\}$ is the set of products which are requested by client i can appear in one or in several splitted positions. Let be $(i, k, S^{'}_{np})$ client node i and $S^{'}_{np}$ is the set of np delivered products for client i at position k.
Each client $(i, k, S^{'}_{np})$ is chosen randomly in route r and deleted from position k , then we try to insert the client $(i, S^{'}_{np})$ in another position in arbitrary way either into r or in another route. First, we try the insertion without splitting $S^{'}_{np}$. The *adjusting-loading procedure* is applied after each insertion try. The procedure checks three constraints : 1) vehicle capacity, 2) route time limit, 3) duplicate insertion problem (more one visit in route is not allowed) and 4) availability of product, provided by processing centers.

Among these four constraints, if only the availability constraint is not respected, the procedure can add a processing center to the route and re-check the constraints.
If the procedure is not able to insert the client $(i, S_{np}^{'})$ after β insertion trials, algorithm attempts to split $S_{np}^{'}$ in np different nodes, and tries to insert each node separately. (see algorithm 2)

Algorithm 2 *client-perturbation(Sol,Sol')* pseudo code

Input : Solution $sol_0 := \{r_1, r_2, ..., r_k\}$. i refers to route r_i.
Output : Feasible solution $sol^{'}$
$sol^{'} := sol_0$
α_counter :=0
while α_counter $\leq \alpha$ **do**
    choose randomly a position k delivered by $r_s \in sol^{'}$
    let be $(i, k, S^{'}_{np})$ the node associated to position k
    β_counter :=0
    insert := fault
    repeat
        Let d be index of random route r in $sol^{'}$, $r_s \neq r_d$
        $r^{'}_d := r_d$
        re-insert client $(i, S^{'}_{np})$ in a random position k in route $r^{'}_d$
        if *Adjusting-loading procedure* is feasible for route $r^{'}_d$ **then**
            replace r_d by $r^{'}_d$
            eliminate $(i, S^{'}_{np})$ from position k in route r_s
            insert := true
        end if
        if *availability of product* is not respected **then**
            insert an intermediate processing center
            if *Adjusting-loading procedure* is feasible **then**
                insert := true
            end if
        else
            β_counter++ or insert := true
        end if
    until β_counter := β
    if insert == false **then**
        split $(i, S^{'}_{np})$ on np node and reinsert randomly each node separately
    end if
    α_counter++
end while=0

5.4.2.2 *sequence-perturbation(S,S')*

The principle of the second perturbation procedure, denoted *sequence-perturbation(S,S')* is similar to client-perturbation except the chosen clients to move will be selected from a sequence in S. A sequence refers to the clients which exist between two processing centers in a given route.

The sequence-perturbation tries to move α sequence of clients chosen randomly in route

r. Each client that belongs to this sequence will be inserted randomly in another position in r or in another route $r \neq r'$. If we don't succeed to move the clients after β' trials, it remains in its initial position. Algorithm 3 details the sequence-perturbation.

Algorithm 3 *sequence-perturbation(Sol,Sol')* pseudo code

Input : Solution $sol_0 := \{r_1, r_2, ..., r_k\}$. i refers to route r_i.
Output : Feasible solution $Sol^{'}$
$sol^{'} := sol_0$
α_counter :=0
while α_counter $\leq \alpha$ **do**
    choose randomly a sequence of client Se from route $r_s \in Sol^{'}$
    let be $(i, k, S^{'}_{np})$ the node associated to position k
    let be n the number of client $(i, S^{'}_{np})$ exist in sequence Se
    $i := 1$
    while $i \leq n$ **do**
        take i^{th} client $(i, S^{'}_{np})$ in sequence Se
        β_counter :=0
        repeat
            Let d be index of random route r in $sol^{'}$, $r_s \neq r_d$
            $r^{'}_d := r_d$
            re-insert client $(i, S^{'}_{np})$ in a random position k in route $r^{'}_d$
            if *Adjusting-loading procedure* is feasible for route $r^{'}_d$ **then**
                replace r_d by $r^{'}_d$
                eliminate $(i, S^{'}_{np})$ from position k in route r_s
                insert := true
            end if
            if *availability of product* is not respected **then**
                insert an intermediate processing center
                if *Adjusting-loading procedure* is feasible **then**
                    insert := true
                end if
            else
                β_counter++ or insert := true
            end if
        until β_counter := β
        if insert == false **then**
            split $(i, S^{'}_{np})$ on np node and re-insert randomly each node separately
        end if
        i++
    end while
    α_counter++
end while=0

5.4.3 Improvement procedure

In the improvement part of algorithm ILS we used the two kinds of local search methods that we have proposed in section two of this chapter.
As local search procedures, we selected (1) *Intra-Route Multi Product Two-Opt Local Search (2-OPT-MP1)*, (2) *Inter-Route Multi Product Two-Opt Local Search (2-OPT-MP2)*, (3) Merging-Products Local Search method (*MP*) and (4) the change-state-PC.
In order to consider larger neighborhood including processing center changes we propose to implement a neighborhood descent local search, the obtained version of ILS is named *ND-ILS*(Neighborhood Descent ILS) and will be presented in the next section. In this case, the improvement procedure uses four neighborhood structures in the following order : (2-OPT-MP1 → 2-OPT-MP2 → MP → change-state-PC). This version is inspired from Variable Neighborhood Descent by Hansen and Mladenovic [111]. Algorithm 4 detailed the structure of the *ND-ILS*.

5.5 General algorithms

The major components of the proposed algorithms are addressed in previous sections. By using these local search methods and perturbation procedures, here, the general schemes of three ILS algorithms are presented : (i) *Randomized ILS (R-ILS)*, (ii)*Neighborhood Descent ILS (ND-ILS)* and (iii) *Score based ILS (S-ILS)*.
In each iteration of *Randomized ILS*, one improvement procedure (*Intra-Route Multi Product Two-Opt Local Search (2-OPT-MP1)*, *Inter-Route Multi Product Two-Opt Local Search (2-OPT-MP2)* and *Merging-Product Local Search (MP))* will be used randomly. After perturbing the current solution (by using the procedures *client-perturbation(S,S')* or *sequence-perturbation(S,S'))* the selected local search is applied in order to improve the solution.
Neighborhood Descent ILS (ND-ILS) considers a *pointer*, called, *LS_selection* to make decision about local search which will be applied in each iteration. The pointer *LS_selection* starts from smallest improvement procedure (according to the size of the neighborhood). The most complicated neighborhoods are tested only when the previous improvement procedure leads to no improvement as following : (2-OPT-MP1 → MP → 2-OPT-MP2 → change-state-PC). When the algorithm reaches any improvement, the pointer goes back to the first local search and so on.
The last ILS strategy that is developed, called *Score based ILS (S-ILS)* uses an adaptive technique to choose a local search and perturbation procedure. This adaptive technique is inspired from an Adaptive Large Neighborhood Search (ALNS) proposed by Ropke

and Pisinger [203].

The main idea is to take advantage of the best combination of local search and perturbation. In order to explore better the search space, it is sometimes necessary to change the perturbation and sometimes only the local search, and sometimes both of them. To take in account this idea, we consider the following three local searches 2-OPT-MP1(sol', sol''), 2-OPT-MP2(sol', sol'') and MP(sol', sol''). We use local searches and client-perturbation(S, S'), sequence-perturbation(S, S') as perturbation procedures. We have the choice in each iteration to use one pair (local search, perturbation) among six possible pairs.

Before applying each iteration, the *S-ILS* should choose a couple *local search* and perturbation procedure which will be applied into each iteration, called *LS-Pert*$_i$. Each *LS-Pert*$_i$ holds two values (i) weight denoted, w_i and (ii) score denoted p_i.

The used method to select a *LS-Pert*$_i$ is inspired by Roulette Wheel Selection method [160]. The idea is based on the fact that if a *LS-Pert*$_i$ leads to more improvement, gains more score and the probability of being chosen increases.

After applying this neighborhood search, the scores will be updated. New weights are calculated according to the recorded scores at the end of each segment. Formula 5.1 calculates the weight of heuristic i(a chosen pair of local search and perturbation named *LS-Pert*$_i$) in segment j, denoted $w_{i,j}$. We assigned the weights of all heuristics to 1 in the first segment. At the end of each segment j, the weight for all heuristics i to applied in segment $j+1$ are calculated as follows :

$$w_{i,j+1} = (1-r) * w_{i,j} + (r) * \frac{p_i}{\theta_i} \tag{5.1}$$

where $r = 0.1$.

The obtained score of heuristic i during the last segment and the number of times that we have tried to use heuristic i during the last segment are denoted by p_i and θ_i respectively. Regulating that how quickly the weight adjustment algorithm behaves to changes in the performance of the heuristics is performed by reaction factor r.

$$p_i = DV_i \tag{5.2}$$

$$\text{where } DV_i = \begin{cases} 6: & \text{if the chosen } \textit{LS-Pert}_i \text{ return a feasible solution, but no improvement.} \\ 10: & \text{if current iteration improves the solution of previous iteration.} \\ 30: & \text{if current iteration improves the best found solution.} \end{cases}$$

The probability of being selected for a *LS-Pert*$_i$ in the next segment $(j+1)$ is calculated by formula (5.3). The algorithm stops after applying n consecutive iterations in main body of algorithm.

$$prob_i = \frac{w_{i,j+1}}{\sum_{k=1\ to\ 6} w_{k,j+1}} \tag{5.3}$$

Algorithm 4 *S-ILS* pseudo code

Input : Initial Solution sol_0. **Output :** Feasible solution sol^*

```
sol* := sol_0
NI_ILS_counter :=0
ILSWeight[i] := 1 ; ILSScore[i] := 0 ; i ∈ {1, ..., 6}
while NI_ILS_counter ≤ nb_segment do
    addtoScore := 0
    for s_n from 1 to Max_{s_n} do
        switch LS_Pertr do
            case 1
                call 2-OPT-MP1(sol*, sol') and then
                call client-perturbation(sol', sol'')
            end
            case 2
                call 2-OPT-MP2(sol*, sol') and then
                call client-perturbation(sol', sol'')
            end
            ...
            case 6
                call 2-OPT-M(sol*, sol') and then
                call sequence-perturbation(sol', sol'')
            end
        end
        if Cost(sol'') ≤ Cost(sol*) then
            sol* := sol''
            addtoScore := 30
        if Cost(sol*) ≤ Cost(sol'') and Cost(sol'') ≤ Cost(sol_prec) then
            addtoScore := 10
        if sol'' is just feasible solution then
            addtoScore := 6
        sol_prec := sol''
        ILSScore[LS_Pert] := addtoScore
        s_n++
    end
    update ILSWeight[LS_Pert]
    update prob[LS_Pert]
    choose LS_Pert according to roulette wheel selection
    NI_ILS_counter++
end
```

5.6 Experimental results

5.6.1 Tested instances and implementation

The developed algorithms are implemented in MS Visual studio C++ 2010 and tested on adopted instances for 2E-MPLRP-PD , using a Dell PC with Intel (R) Core (TM) Solo CPU 1.40 GHz, 2GB of RAM and Windows XP Pro. The tested benchmark are accessible on website [206] and are detailed in chapter 4.6.

5.6.2 Local Search Result Analysis

This section shows the results of the proposed Local Search methods, applied on two heuristics, proposed section 4.4. These heuristics present non-trivial extensions of Nearest Neighbor Approach (NNH) and Best Insertion Approach (BIH). The Details of these approaches can be found in section 4.4.

Instances	*NNH*				*BIH*			
	T	FI	T	BI	T	FI	T	BI
I(20,5)	8	16,93%	19	15,03%	13	12,54%	9	14,54%
I(50,5)	56	17,43%	92	20,12%	28	14,81%	30	16,92%
I(100,5)	178	14,49%	444	17,83%	80	17,45%	198	13,17%
I(100,10)	177	21,13%	318	18,84%	138	18,74%	249	14,54%
I(200,10)	1453	14,58%	1218	17,54%	308	13,87%	903	17,54%
Average	374,4	16,91%	418,2	17,87%	113,4	15,48%	277,8	15,34%

Table 5.1: Evaluation of 2-OPT-MP2

Tables 5.1 to 5.4 gather the results of the Local Search methods (2-Opt-MP1, 2-Opt-MP2, MP, change-state-PC) for NNH and BIH respectively. The tested instances are grouped in the first column according to the number of customers and the number of processing centers. The set $I(20-5)$ $\{$resp.$I(50-5), I(100-5), I(100-10), I(20-10)\}$ involves $8\{$resp.$10, 6, 6, 6\}$instances. For each method, the *first improvement strategy (FI)* and *Best improvement strategy (BI)* results are presented. In these table all strategies own 2 columns : (i) time execution in second (column T) and (ii) the gap between the heuristic solution before and after applying a local search procedure. The last row indicates for each column : (i) The average gaps, compared to the heuristic solution, and (ii) the average of running time.

The results show that all proposed Local Search methods succeed to improve the heuristics solutions. The inter-route Multi-product Two-Opt Local Search method (2-OPT-MP2) obtains better gaps (at worst 15,34% on average) compared to the other local Search methods but the computation time is more important (at worst $418,2s$). This result can be explained by the fact that the explored neighbourhood by *2-OPT-MP2* is larger than *2-OPT-MP1* and MP.

Instances	*NNH*				*BIH*			
	T	FI	T	BI	T	FI	T	BI
I(20,5)	0	0,65%	0	1,07%	1	0,81%	0	0,90%
I(50,5)	1	0,34%	2	0,52%	1	0,41%	3	0,40%
I(100,5)	3	1,78%	2	1,90%	3	3,09%	4	3,45%
I(100,10)	3	0,35%	4	0,23%	4	1,53%	5	1,81%
I(200,10)	5	0,36%	6	0,09%	5	0,18%	7	0,35%
Average	2,4	0,70%	2,8	0,76%	2,8	1,20%	3,8	1,38%

Table 5.2: Evaluation of 2-OPT-MP1

The results of *2-OPT-MP2* method are similar to those obtained by MP methods but we can note a better performance for *2-OPT-MP1* compared to MP. This can be explained by the fact that the number of products used in the instances is small.
For all proposed local search method, the *best improvement strategy (BI)* gives almost the same results as *first improvement strategy (FI)*but the computing time of (BI) is more important. This shows that (BI) explores so many infeasible moves, which makes it less competitive. We think that penalizing infeasible moves can provide better results for (BI).
Table 5.4 shows that change-state-PC gives better results than MP. We didn't mention the results of Swap-routes-PC procedure because it is not able to improve any heuristic solution.

Instances	*NNH*				*BIH*			
	T	FI	T	BI	T	FI	T	BI
I(20,5)	1	1,01%	2	1,57%	2	1,74%	3	4,32%
I(50,5)	8	0,22%	10	0,47%	5	0,03%	10	0,03%
I(100,5)	25	0,15%	26	0,20%	14	0,02%	22	0,03%
I(100,10)	26	0,20%	75	0,41%	21	0,61%	22	0,89%
I(200,10)	69	0,16%	79	0,30%	40	0,03%	35	0,03%
Average	25,8	0,35%	38,4	0,59%	16,4	0,49%	18,4	1,06%

TABLE 5.3: Evaluation of MP

Instances	*NNH*				*BIH*			
	T	FI	T	BI	T	FI	T	BI
I(20,5)	1	1,83%	2	1,75%	1	2,70%	2	2,49%
I(50,5)	3	0,58%	4	0,58%	3	3,69%	3	3,54%
I(100,5)	5	0,53%	7	0,53%	7	4,01%	7	4,40%
I(100,10)	6	0,32%	14	0,85%	6	2,95%	8	2,60%
I(200,10)	16	2,39%	36	2,39%	25	3,52%	36	2,81%
Average	6,2	1,13%	12,6	1,22%	8,4	3,37%	11,2	3,16%

TABLE 5.4: Evaluation of change-state-PC

5.6.3 Setup Parameters for ILS

The two proposed perturbation procedures exploit two parameters (α, β for «client-perturbation(S,S')» and $\alpha^{'}$, $\beta^{'}$ for «sequence-perturbation(S,S')»). In order to fix the best parameters for ILS different combination of parameters are considered, while keeping the same perturbation procedure. Several preliminary test were performed. We have decided to present here the most significant tests that are related to the use of 2-OPT-MP1 and 2-OPT-MP2. Tables 5.5 and 5.6 presents all these tests. Table 5.5 is related to «client-perturbation(S,S')» and table 5.6 gathers the results test for «sequence-perturbation(S,S')». Each test involves two parameters. Before analyzing the setup parameters results, we focus on parameters definition. Table 5.5 and table 5.6 show all tested combination.

Column 1 shows the number of iterations for each execution (iter). Column 2 in table 5.5 (in table 5.6) present the parameter α ($\alpha^{'}$ respectively), which refers to the number of *client-product unit* to move for «client-perturbation» (the number of sequences that is supposed to be moved for the «sequence-perturbation» respectively). In column 3 from table 5.5 (respectively table 5.6) β (respectively $\beta^{'}$) concerns the number of trials in order to move one client. The applied local search for each test are shown in forth column of tables 5.5 and 5.6. Fifth column holds the average time in second and the sixth column presents average gap on 18 tested instances. This gap refers to difference between the NNH heuristic solution before and after applying ILS.

Iter	α	β	Applied LS	average Time	average Gap

5	5	15	2-OPT-MP1	10	10,14%
5	5	20	2-OPT-MP1	12	9,38%
5	5	30	2-OPT-MP1	15	10,16%
5	10	10	2-OPT-MP1	8	9,99%
5	10	20	2-OPT-MP1	12	10,54%
10	5	15	2-OPT-MP1	15	10,22%
10	5	20	2-OPT-MP1	17	10,36%
10	10	5	2-OPT-MP1	14	9,97%
10	10	10	2-OPT-MP1	17	10,26%
15	1	20	2-OPT-MP1	15	11,31%
15	2	20	2-OPT-MP1	20	9,67%
15	3	20	2-OPT-MP1	29	8,82%
15	5	20	2-OPT-MP1	35	10,29%
30	5	5	2-OPT-MP1	29	10,95%
40	5	20	2-OPT-MP1	51	10,29 %
15	1	10	2-OPT-MP2	127	31,00 %
15	1	20	2-OPT-MP2	158	31,94 %
15	5	10	2-OPT-MP2	250	30,57 %

TABLE 5.5: client-perturbation setup parameters

Iter	α'	β'	Applied LS	average Time	average Gap
10	1	10	2-OPT-MP1	15	10,46%
15	1	10	2-OPT-MP1	19	11,58%
20	1	10	2-OPT-MP1	25	10,23%
15	1	20	2-OPT-MP1	20	10,40%
10	2	10	2-OPT-MP1	20	10,29%
15	2	10	2-OPT-MP1	27	10,88%
15	2	20	2-OPT-MP1	31	10,33%
10	1	5	2-OPT-MP2	95	29,72%
10	1	10	2-OPT-MP2	124	30,17%
10	2	10	2-OPT-MP2	134	30,44%
20	1	10	2-OPT-MP2	159	29,01%

TABLE 5.6: sequence-perturbation setup parameters

As we can see the better result is obtained by using 2-OPT-MP2 as LS. More precisely, the use of 2-OPT-MP2 gives better gaps compared with the use of 2-OPT-MP1. The average gap presents the improvement of initial solution after applying relevant method. Regarding the number of client moves, the better parameter set obtained for 2-OPT-MP1 and 2-OPT-MP2 is with one client. Perturbation of one client is already sufficient to achieve diversification on solution because the client can be inserted either in one position, satisfying all demand or splitted in multiple nodes position, and then inserted or be inserted with a processing center.

We also noticed that the results of the perturbation procedure using one sequence is in average better that the other procedure. Using one sequence allows a good diversification of the solution. The results of the client perturbation of one customer with the use of 2-Opt-MP2 are similar to the results of the sequence perturbation using one sequence and the 2-Opt-MP2 local search. These last results gives the best parameter for α and $\alpha^{'}$ and was confirmed by another intensive tests.

5.6.4 Numerical results

To analyze the performance of our algorithms, we compared three versions of ILS (*R-ILS*, *ND-ILS* and *S-ILS*) with two other version of ILS in which only one local search is used. It concerns ILS (2-Opt-MP1) and ILS (2-Opt-MP2).

Table 5.7 (respectively table 5.8) presents the results of five ILS strategies with sequence perturbation procedures (respectively one client perturbation procedure). For each ILS method, time execution named T (in second) and improvement gap are provided (named G). The ILS improvement percentage compared to initial solution (gap) is calculated for each tested instances. This gap values (G) is the improved cost obtained by ILS denoted $ImpCost$ and the initial solutions obtained by the heuristics named Cost(H). The *Gap* is calculated as $\frac{Cost(H)-ImpCost}{Cost(H)}$.

The last row of each table mentions the average of the execution time and gap for all 36 instances.

Best results are obtained by *S-ILS*. The results are almost similar in the case of the use of the sequence perturbation procedure (average gap of $32,82\%$ and average time of 173 seconds, with a minimum and maximum gap between $5,21$ and $55,58\%$) and the use of the client perturbation (average gap of $32,26\%$ and average time of 147 seconds, with a minimum and maximum gap between $6,54$ and $53,35\%$).

The *R-ILS* and *S-ILS* give similar performance regarding to the gap because the use of the local search already gives better performance. This proves that when local search provides a good result then ILS works well. *ND-ILS* minimizes the execution time compared to

S-ILS because the structure of the neighborhood leads to the less costly computational time. This method does't give the best performance here, because the perturbation can be less important than the used local search which is composed by 4 neighborhood structure and the solution is less pertubated and may remain in the same local optimum.
ILS with 2-OPT-MP2 provides better results than the version with 2-OPT-MP1, but is not as good as *R-ILS*, *S-ILS*, *ND-ILS*; it gives the best gap only for one instance (table 3) and for three instances (table 4).
Regarding the overall performance, the best strategy is obtained by the *S-ILS* version with $32,82\%$ gap on average.
In table 5.9 (respectively table 5.10) we give the result compared with the best heuristics methods. The negative percentage means that the ILS strategy could not improve the best heuristic solution. We note that the various versions of ILS have succeeded to improve the best solution almost in all cases except in very few cases : 1) In the case with the sequence perturbation procedure, 2 worst cases (1 worst-case respectively) for ILS(2-Opt-MP1) ((ILS(2-Opt-MP2) respectively); 2) in the case with client perturbation, 2 worst cases for ILS(2-Opt-MP1), one worst case for ILS(2-Opt-MP2)) and one for *ND-ILS*.
In the last table (5.11), we study the impact of the initial solution in the *S-ILS* algorithm with the use of sequence perturbation. The last column shows the result of a random selection strategy in which at each iteration BIH or NNH are chosen randomly to start the algorithm. As we can see the *S-ILS* version with BIH initial solution outperforms the other results with a gap average of $32,82\%$.

Instances	*ILS(2-OPT-MP1)*		*ILS(2-OPT-MP2)*		*R-ILS*		*ND-ILS*		*S-ILS*	
	T	G	T	G	T	G	T	G	T	G
coord20-5-1-2e	4	9,68%	24	30,63%	21	**31,31%**	6	**31,31%**	26	30,09%
coord20-5-1b-2e	6	**11,26%**	20	1,32%	31	4,64%	10	4,30%	25	5,21%
coord20-5-2-2e	8	23,99%	24	41,70%	26	**48,43%**	3	40,81%	27	42,89%
coord20-5-2b-2e	7	1,36%	49	9,73%	33	14,71%	11	**28,73%**	55	20,78%
coord50-5-1-2e	10	9,88%	41	32,86%	68	30,19%	19	33,66%	59	**35,61%**
coord50-5-2-2e	6	12,48%	46	27,66%	123	34,39%	28	30,11%	50	**38,66%**
coord50-5-2bBIS-2e	7	7,46%	49	25,03%	88	**37,79%**	18	30,20%	68	35,25%
coord50-5-2BIS-2e	14	8,00%	51	28,45%	61	28,94%	26	29,43%	60	**30,11%**
coord50-5-3-2e	13	15,17%	65	23,51%	69	24,05%	17	28,49%	49	**29,32%**
coord100-5-1-2e	25	5,33%	102	27,28%	281	27,28%	40	**30,67%**	169	29,26%
coord100-5-2-2e	17	12,46%	143	53,66%	200	54,41%	50	54,41%	182	**55,58%**
coord100-5-3-2e	16	13,64%	119	44,69%	258	46,39%	75	50,58%	143	**51,96%**
coord100-10-1-2e	25	23,73%	179	47,64%	233	**50,38%**	61	41,56%	273	39,92%
coord100-10-2-2e	30	8,15%	176	**29,40%**	275	26,98%	40	27,21%	301	28,71%
coord100-10-3-2e	25	8,64%	163	20,87%	284	**25,79%**	40	16,87%	160	20,01%
coord200-10-1-2e	35	11,51%	390	29,72%	674	25,01%	430	**33,43%**	421	32,54%

coord200-10-2-2e	55	14,07%	543	41,70%	651	**42,47%**	227	42,28%	559	36,03%
coord200-10-3-2e	40	13,78%	232	31,38%	898	**31,90%**	356	30,62%	482	28,82%
average	19	11,70%	134	30,40%	237	32,50%	81	32,48%	173	**32,82%**

TABLE 5.7: Comparative results of ILS with sequence-perturbation

Instances	*ILS(2-OPT-MP1)*		*ILS(2-OPT-MP2)*		*R-ILS*		*ND-ILS*		*S-ILS*	
	T	G	T	G	T	G	T	G	T	G
coord20-5-1-2e	5	17,79%	20	**37,16%**	43	30,63%	11	11,49%	11	32,67%
coord20-5-1b-2e	3	1,32%	22	1,32%	37	5,30%	15	1,32%	11	**6,54%**
coord20-5-2-2e	4	25,34%	20	44,39%	34	41,03%	21	43,72%	13	**53,35%**
coord20-5-2b-2e	4	0,00%	23	14,03%	28	**22,40%**	30	15,61%	20	19,65%
coord50-5-1-2e	7	14,51%	51	**32,95%**	54	32,06%	41	29,47%	41	30,06%
coord50-5-2-2e	6	12,24%	42	27,66%	75	18,73%	27	24,11%	32	**29,87%**
coord50-5-2bBIS-2e	8	10,35%	46	30,93%	41	30,93%	75	18,41%	40	**31,02%**
coord50-5-2BIS-2e	7	6,03%	40	28,45%	35	**28,94%**	32	**28,94%**	39	25,23%
coord50-5-3-2e	11	15,17%	50	23,73%	84	**30,88%**	28	28,49%	51	30,04%
coord100-5-1-2e	16	5,18%	139	27,28%	112	27,28%	50	**30,67%**	129	25,85%
coord100-5-2-2e	12	10,73%	127	53,66%	75	**54,41%**	86	53,03%	119	46,57%
coord100-5-3-2e	15	13,64%	121	44,69%	59	46,76%	91	**50,58%**	105	42,30%
coord100-10-1-2e	21	22,12%	125	47,64%	57	43,96%	104	**48,68%**	89	47,62%
coord100-10-2-2e	24	8,38%	136	**35,18%**	91	29,17%	285	24,73%	162	32,82%
coord100-10-3-2e	28	8,36%	134	20,87%	76	**22,14%**	513	8,36%	112	17,54%
coord200-10-1-2e	25	10,55%	576	27,60%	245	25,65%	297	**40,32%**	452	38,05%
coord200-10-2-2e	37	15,22%	486	41,70%	186	44,78%	202	37,72%	345	**45,71%**
coord200-10-3-2e	42	5,83%	683	31,38%	395	30,63%	317	**31,78%**	874	25,83%
average	15	10,09%	158	31,70	96	31,43%	124	29,30%	147	**32,26%**

TABLE 5.8: Comparative results of ILS with client-perturbation

Instances	*ILS(2-OPT-MP1)*	*ILS(2-OPT-MP2)*	*R-ILS*	*ND-ILS*	*S-ILS*
	G	G	G	G	G
coord20-5-1-2e	7,60%	29,03%	**29,73%**	29,73%	28,48%
coord20-5-1b-2e	**8,85%**	-1,37%	2,05%	1,70%	2,63%
coord20-5-2-2e	21,34%	39,67%	**46,64%**	38,75%	40,90%
coord20-5-2b-2e	-0,69%	7,85%	12,94%	**27,25%**	19,13%
coord50-5-1-2e	9,88%	32,86%	30,19%	33,66%	**35,61%**
coord50-5-2-2e	7,74%	23,74%	30,83%	26,32%	**35,34%**
coord50-5-2bBIS-2e	4,11%	22,32%	**35,54%**	27,68%	32,91%
coord50-5-2BIS-2e	-3,61%	19,42%	19,97%	20,52%	**21,29%**

coord50-5-3-2e	9,17%	18,10%	18,68%	23,43%	**24,32%**
coord100-5-1-2e	2,99%	25,48%	25,48%	**28,96%**	27,51%
coord100-5-2-2e	10,50%	52,62%	53,39%	53,39%	**54,58%**
coord100-5-3-2e	11,43%	43,27%	45,02%	49,32%	**50,73%**
coord100-10-1-2e	23,66%	47,59%	**50,33%**	41,50%	39,86%
coord100-10-2-2e	5,64%	**27,47%**	24,99%	25,22%	26,76%
coord100-10-3-2e	6,47%	18,99%	**24,03%**	14,90%	18,11%
coord200-10-1-2e	2,23%	22,35%	17,15%	26,45%	**25,47%**
coord200-10-2-2e	14,07%	41,70%	**42,47%**	42,28%	36,03%
coord200-10-3-2e	13,43%	31,10%	**31,63%**	30,34%	28,53%
average	8,60%	27,90%	30,06%	30,08%	**30,46%**

TABLE 5.9: Comparative results of ILS compared to the best methods (with sequence perturbation procedure)

Instances	*ILS(2-OPT-MP1)*	*ILS(2-OPT-MP2)*	*R-ILS*	*ND-ILS*	*S-ILS*
	G	G	G	G	G
coord20-5-1-2e	15,90%	**35,71%**	29,03%	9,45%	31,12%
coord20-5-1b-2e	-1,37%	-1,37%	2,72%	-1,37%	**4,00%**
coord20-5-2-2e	22,74%	42,45%	38,98%	41,76%	**51,73%**
coord20-5-2b-2e	-2,08%	12,24%	**20,79%**	13,86%	17,98%
coord50-5-1-2e	14,51%	**32,95%**	32,06%	29,47%	30,06%
coord50-5-2-2e	7,48%	23,74%	14,33%	20,00%	**26,07%**
coord50-5-2bBIS-2e	7,11%	28,43%	28,43%	15,46%	**28,53%**
coord50-5-2BIS-2e	-5,83%	19,42%	**19,97%**	**19,97%**	15,79%
coord50-5-3-2e	9,17%	18,33%	**25,99%**	23,43%	25,09%
coord100-5-1-2e	2,84%	25,48%	25,48%	**28,96%**	24,02%
coord100-5-2-2e	8,73%	52,62%	**53,39%**	51,98%	45,37%
coord100-5-3-2e	11,43%	43,27%	45,40%	**49,32%**	40,82%
coord100-10-1-2e	22,05%	47,59%	43,91%	**48,63%**	47,57%
coord100-10-2-2e	5,88%	**33,41%**	27,24%	22,68%	30,99%
coord100-10-3-2e	6,18%	18,99%	**20,29%**	6,18%	15,58%
coord200-10-1-2e	1,17%	20,01%	17,86%	**34,06%**	31,56%
coord200-10-2-2e	15,22%	41,70%	44,78%	37,72%	**45,71%**
coord200-10-3-2e	5,45%	31,10%	30,35%	**31,51%**	25,53%
average	8,14%	29,23%	28,94%	26,84%	**29,86%**

Table 5.10: Comparative results of ILS compared to the best methods (with client perturbation procedure)

Instances	*NNH*		*BIH*		*Random*	
	T	G	T	G	T	G
coord20-5-1-2e	45	19,75%	26	**30,09%**	41	21,46%
coord20-5-1b-2e	41	2,91%	25	**5,21%**	39	3,24%
coord20-5-2-2e	37	31,49%	27	**42,89%**	45	39,45%
coord20-5-2b-2e	72	**25,36%**	55	20,78%	64	22,65%
coord50-5-1-2e	65	31,19%	59	35,61%	55	**37,56%**
coord50-5-2-2e	44	27,78%	50	**38,66%**	44	30,07%
coord50-5-2bBIS-2e	81	30,45%	68	35,25%	59	**36,01%**
coord50-5-2BIS-2e	59	**30,43%**	60	30,11%	73	29,98%
coord50-5-3-2e	55	21,75%	49	**29,32%**	65	19,37%
coord100-5-1-2e	175	24,65%	169	29,26%	201	**31,80%**
coord100-5-2-2e	169	31,28%	182	**55,58%**	189	38,15%
coord100-5-3-2e	165	29,76%	143	**51,96%**	159	49,45%
coord100-10-1-2e	305	38,82%	273	**39,92%**	312	38,48%
coord100-10-2-2e	298	**29,54%**	301	28,71%	287	28,09%
coord100-10-3-2e	286	22,26%	160	20,01%	202	**24,54%**
coord200-10-1-2e	503	24,07%	421	**32,54%**	498	23,73%
coord200-10-2-2e	549	28,59%	559	**36,03%**	487	35,98%
coord200-10-3-2e	528	20,93%	482	**28,82%**	554	19,12%
average	193	26,17	173	**32,82%**	187	29,68%

Table 5.11: The impact of the initial solution for *S-ILS* (sequence perturbation procedure)

5.7 Conclusion

We have proposed improvement methods to enhance the result of heuristic methods proposed in this chapter for the 2E-MPLRP-PD. We have also proposed four local searches. These local searches improve the best heuristic solution with a gap equal to $17,87\%$ on average, but the execution time of 2-OPT-MP2 is quit high. The ILS outperforms these results. The average improvement is equal to $32,82\%$ with an average time equal to 173. We can notice the performance of ILS even for the simplest strategy using only one local search procedure as improvement technique. The best performance of ILS is given with *S-ILS* and the use of BIH heuristic as initial solution. The two proposed perturbation procedure (client-perturbation and sequence perturbation) give almost the same performance.

Chapitre 6

Multi-product Location-Routing Problem with Pickup and Delivery

Contents

6.1 Introduction

The first extension of the LRP addressed in the previous two chapters concerns a two level distribution system, including processing centers that must be delivered in the first level. In this chapter, we are interested in one level distribution system where small customers and processing centers or warehouses are delivered in the same level and in the same route. This distribution configuration becomes increasingly common because it allows the mutualization of resources (warehouses and vehicles) to reduce the transportation and logistic costs. The new proposed problem is named MPLRP-PD for Multi Product Location Routing Problem with Pickup and Delivery. It includes the same constraints as those taken into account in the 2E-MPLRP-PD except that one level distribution system is considered.

The rest of this section is organized as follows. Section 2 presents the problem and its formulation. In Section 3, we detail the proposed heuristic approach. Section 4 presents a metaheuristic approach based on ALNS. Computational results are reported in section 5. Conclusion and future directions are discussed in the last section.

6.2 Problem Description

We consider a many-to-many distribution network. This is a part-supply network of inbound logistic where the network consists of a depot, a set of clients and a set of potential active processing centers (locations), where each active processing center has a demand of primary products, denoted *h-product*, that will be transformed to final products, such as packing h-products in new specific packaging, in order to satisfy the demand of clients, let denote *c-product* the final products. The processing centers are considered as pickup and delivery nodes and they have opening costs. Each active processing center can provide only one type of c-product. A fleet of unlimited number of homogeneous vehicles is available to transport h-products from depot to active processing centers and c-products from active processing centers to clients. At the beginning, vehicles are assumed to be located at the depot, and at the end of each trip, vehicle return to the depot, furthermore, we are not allowed to split the demand h-product of each processing center, i.e. each demand of active processing center is satisfied, when it is possible, by one vehicle. The first vehicle that reaches the processing center must be a delivery vehicle. The transportation cost consists of a fixed charge incurred on each trip and a variable cost proportional to the traveled distance or time. Furthermore, we assume that products are ready to be

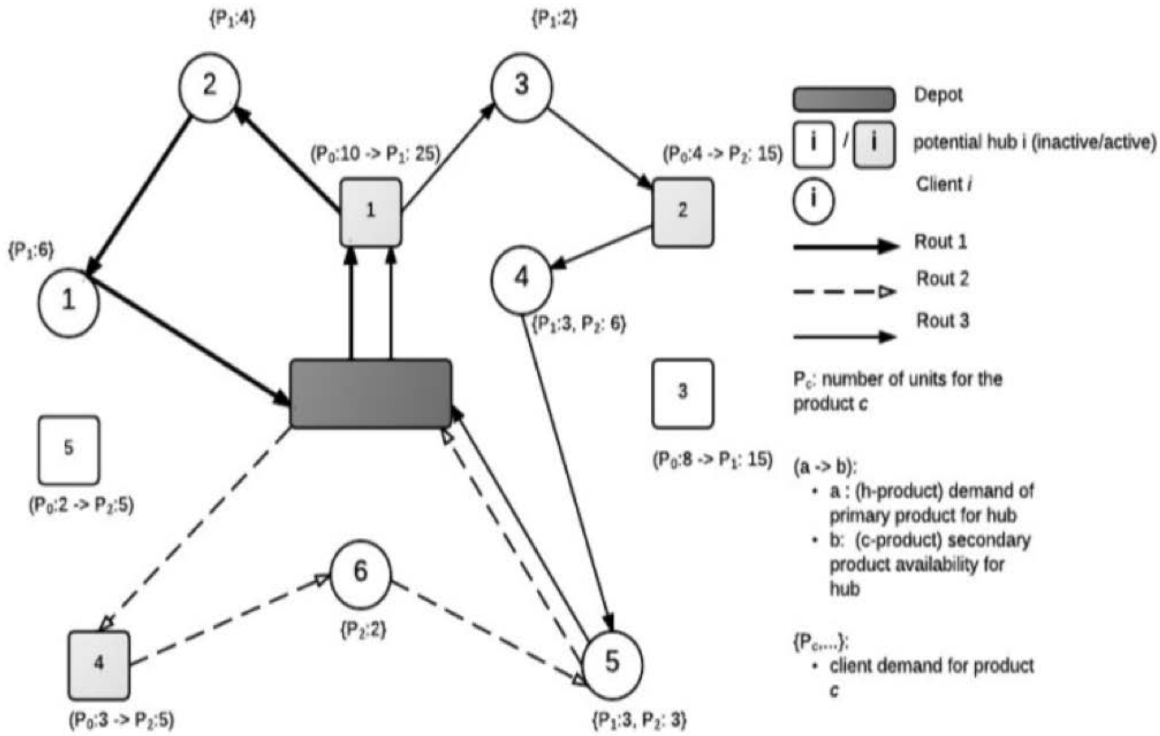

FIGURE 6.1 – Example of MPLRP-PD with 5 processing centers and 6 clients.

collected at the depot and at active processing center when the vehicle arrives. A vehicle passes through a set of processing centers when is necessary to collect sufficient quantity to satisfy the demand of a customer. Therefore, the objective is to minimize the total transportation and processing center opening cost.

Others constraints and assumptions are given below :

— All sites are ready to setup any processing centers.

— Each processing center asks for one primary *h-product* and should provide exactly one secondary product *c-product*.

— Each customer asks for one or several *c-products*, and can be served by only one vehicle for a given product.

— The products have the same unit sizes.

— Each vehicle starts its trip from depot and ends at the depot.

— A vehicle visits a processing center only once.

— Active processing centers receive their request from depot.

— Splitting the demand of client for a given product is not allowed.

— Splitting the demand of processing center is not allowed but several vehicles can visit the processing center to pickup the *c-products*.

— There may be several processing centers with the same *c-product*.

The MPLRP-PD can be viewed as an extension of vehicle routing problem with pickup and delivery, when the processing centers are the pickup and delivery nodes that must be located, and the clients are the delivery nodes.

6.2.1 Illustrative example for MPLRP-PD

Figure 6.1. shows a case with 5 potential active processing centers and 6 clients. In this example we consider the vehicle capacity equal to 12. Besides the third and fifth processing centers are not even opened. Remark that vehicles should pickup the h-products from depot and deliver them to the active processing centers, which have being opened through a trip. After satisfying the demand of processing centers, a vehicle will pick-up the c-product, which are in the unfold processing center, and continue its trip. It should be mentioned that in figure 6.1, client 5 asks for both c-products p_1 and p_2, each of these products is satisfied with a separate vehicle, p_1 by route 3 and p_2 by route 2.
The first and second processing centers can't be satisfied in Route 3, because the sum of h-products quantities requested by processing center 1 and 2 exceed vehicle's capacity, so processing center 1 is satisfied by route 1. Furthermore each processing center will just demand one primary h-product and should provide exactly one secondary c-product. In route 3, the vehicle leaves depot with 4 units of h-product p_0, it picks up 11 units of c-product p_1 in processing center 1, at the next node, the client 3 receives its demand, then, in processing center 2, the vehicle delivers 4 units of h-product p_0 and picks up 6 units of c-product p_2, finally it delivers the p_1 and p_2 demands of client 4 and the p_1 demand of client 5.

6.2.2 Problem Formulation

The given notations and decision variables in section 4.3 are used for *MPLRP-PD* model. We consider the assumption of an unlimited number of homogeneous vehicles with known capacities in order to dimension the fleet of vehicles V_1 from a strategic point of view. If a vehicle has satisfied h-product, demand of processing center k, it could provide its c-product s given in p_{ks} amount. Each customer $i \in N_c$ has delivery demands d_{is} for product $s \in P$. The *MPLRP-PD* consists of two main problems, namely, the facility location problem and the vehicle routing problem with pickup and delivery. The problem is to determine the location of active processing centers, the assignment of clients to the opened processing centers and the corresponding vehicle routes with a minimum total cost. A proposed mathematical formulation is given below :

$$\min \sum_{i \in N} \sum_{j \in N} \sum_{v \in V} c(i,j) x_{ij}^v + \sum_{k \in N_0} \sum_{s \in P} FD_k y_{ks} + \sum_{k \in N_0} \sum_{v \in V_1} F_{V1} x_{0k}^v \tag{6.1}$$

$$\sum_{i \in N} \sum_{v \in V_1} x_{ij}^v \leq |P_j| \qquad \forall j \in N_c \tag{6.2}$$

$$\sum_{i \in N} \sum_{v \in V_1} x_{ij}^v \geq 1 \qquad \forall j \in N_c \tag{6.3}$$

$$\sum_{i \in N} x_{ij}^v \leq 1 \qquad \forall j \in N, v \in V_1 \tag{6.4}$$

$$\sum_{v \in V_1} \sum_{i \in N} x_{ij}^v \geq y_{js} \qquad \forall j \in N_0, s \in P \tag{6.5}$$

$$\sum_{j \in N} x_{ji}^v - \sum_{j \in N} x_{ij}^v = 0 \qquad \forall i \in N, v \in V_1 \tag{6.6}$$

$$\sum_{i \in Y} \sum_{j \in Y} x_{ij}^v \leq |Y| - 1 \qquad \forall v, \in V_1, Y \subset N - \{0\}, |Y| \leq 2 \tag{6.7}$$

$$\sum_{j \in N_0} x_{0j}^v \leq 1 \qquad \forall v \in V_1 \tag{6.8}$$

$$\sum_{i \in N_c} x_{i0}^v \leq 1 \qquad \forall v \in V_1 \tag{6.9}$$

$$\text{if } x_{ij}^v = 1 \forall i \in N \Rightarrow l_{is}^v + U_{js}^v - dp_{js} q_{js}^v = l_{js}^v \qquad \forall j \in N_0, s \in P, v \in V_1 \tag{6.10}$$

$$\text{if } x_{ij}^v = 1 \forall i \in N_0 \cup \{0\} \Rightarrow l_{is}^v - d_{js} q_{js}^v = l_{js}^v \qquad \forall j \in N_c, s \in P, v \in V_1 \tag{6.11}$$

$$\sum_{v \in V_1} U_{is}^v \leq p_{is} y_{is} \qquad \forall i \in N_0, s \in P \tag{6.12}$$

$$\sum_{s\in P} y_{ks} \leq 1 \qquad \forall k \in N_0 \tag{6.13}$$

$$\sum_{s\in p} l_{is}^{v} \leq C_{V1} \qquad \forall i \in N_0 \cup \{0\}, v \in V_1 \tag{6.14}$$

$$l_{0s}^{v} = \sum_{k\in N_0} q_{ks}^{v} dp_{ks} \qquad \forall s \in P, v \in V_1 \tag{6.15}$$

$$\sum_{i\in N}\sum_{j\in N} c(i,j)x_{ij}^{v} \leq Time - Limit \qquad \forall v \in V_1 \tag{6.16}$$

$$\sum_{s\in P}\sum_{v\in V_1} q_{ks}^{v} = \sum_{s\in P} y_{ks} \qquad \forall k \in N_0,\ H_{ks}{=}\,1 \tag{6.17}$$

$$\sum_{v\in V}\sum_{s\in P} q_{js}^{v} = \mid P_j \mid \qquad \forall j \in N_c \tag{6.18}$$

$$\sum_{s\in P} q_{is}^{v} \leq \mid P_i \mid \sum_{j\in N} x_{ij}^{v} \qquad \forall v \in V_1, i \in N_c \tag{6.19}$$

$$\sum_{i\in N}\sum_{s\in P} q_{is}^{v} \geq \sum_{j\in N} x_{ij}^{v} \qquad \forall i \in N, v \in V_1 \tag{6.20}$$

$$\sum_{i\in N}\sum_{s\in P} q_{is}^{v} \leq M \times \sum_{j\in N} x_{ij}^{v} \qquad \forall i \in N, v \in V_1 \tag{6.21}$$

$$x_{ij}^{v} \in \{0,1\} \qquad \forall i,j \in N, v \in V \tag{6.22}$$

$$y_{ks} \in \{0,1\} \qquad \forall s \in P, k \in N_0 \tag{6.23}$$

$$q_{is}^{v} \in \{0,1\} \qquad \forall s \in P, k \in N_0, v \in V \tag{6.24}$$

The objective function (6.1) minimizes the total travel costs, opening potential processing center's costs and vehicle fixed costs. Constraints (6.2) and (6.3) guarantee that the number of vehicles that pass through each client is at most equal to the number of c-products required by the client and at least equal to one. Constraint (6.4) ensures that a vehicle could visit each node maximum once. Constraint (6.5) ensures that when a processing center is opened, there is at least one visit to this processing center. Constraint (6.6) is known as degree constraint and guarantees that the number of entering and leaving arcs to each node are equal. (6.7) is a constraint of sub tour-elimination. Constraints (6.8) and (6.9) handle departure and destination point to depot for each route. Constraints (6.10) and (6.11) assure the compatibility between routes and vehicle capacity. Constraint (6.12) implies that the total amount of pickups from each processing center does not exceed the corresponding processing center's availability (c-product). Constraint (6.13) ensures that each opened processing center produces only one type of c-product. Constraint (6.14) assures that the total load on each processing center does not exceed the vehicle capacity. Constraint (6.15) ensures that the total load in vehicle at depot is equal to the quantity of all requested h-product by the processing centers, which will be satisfied by the given vehicle. Constraint (6.16) enforces that route time will not exceed *Time-Limit*. Constraint (6.17) assumes that the h-product demand of each opened processing center should be satisfied by one vehicle. Constraint (6.18) sets the number of variables q^v_{js} ($q^v_{js} = 1$) to $\mid P_j \mid$, and the constraint (6.19) assures that, when a vehicle visits a client i, it satisfies at least the demand of one product of the client i. Constraints (6.20) and (6.21) enhance the compatibility between demand satisfaction and the used vehicles. (6.22)-(6.24) are known as integrality constraints.

6.3 Heuristic approaches for the MPLRP-PD

In this section we describe the main of our heuristic method developed to solve the multi-products location-routing problem with pickup and delivery. Our heuristic method consists of two phases : (i) initialization phase and (ii) assignment phase. The initialization phase determines the first processing center in each new trip, and the second phase, tries to insert the maximum number of clients and processing centers into the current route. The two phases are repeated until total customers demand is satisfied. Before describing the two phases heuristic, several functions are used in both phases that will be described in the following section.

6.3.1 Description of main functions

— *OpenBestClosedHub*, for requested c-product, this function returns the best processing centers for that product. It assigns a score to each closed processing centers k. This score is computed as the sum of the opening processing centers cost and the total distances between the processing center k and the non-satisfied clients that be can satisfied by k and the distance between the processing center k and the depot.

— *Checking-CapacityOpenedHub* for requested c-product, this function checks whether the processing centers, which have been opened till now, could satisfy their client demands or not. We should note that these clients' requests of c-product are not yet satisfied and these requests concern the same product of the relevant processing centers. In order to find a fictive assignment, *Checking-CapacityOpenedHub* function sorts clients in non-increasing order of their amount of request. Afterward it starts to pack clients into the processing centers. It will continue till that no client could be packed, and in this case it goes to the next processing center, which is selected randomly. If all clients are packed into the opened processing center, then no new processing centers is needed and function return true, otherwise a function return false. In the following we will present the two phases of our heuristic : (i) route initialization phase and (ii) assignment phase.

6.3.2 Route initialization phase

The goal of this phase is to determine, for each c-product, the best processing centers, that will start a new route. In order to select the best processing centers for each c-product, a score value is associated to each processing center. The score function takes into account the distance between the processing center and the depot, the sum of distances between a processing center and non-satisfied clients that request a c-product provided by the given processing center, and the opening cost of the processing centers. A processing center with minimal score is selected to start the route. The Initialization function will check for each product, whether the already opened processing centers of the relevant product could satisfy the clients, who asked the mentioned given products. If not, the *OpenBestClosedHub* function, that uses a score value, will be called to open the best-closed processing centers for this product.

Algorithm 5 Description of Initialization Function

Input : partial solution of MPLRP-PD

Output : the best processing center node to start a new route

Let $Distance_h$ be the sum distance between a processing center h and the non-satisfied clients that request the c-product provided by the given processing center k

Let $ScorInitial_h$ be the processing center h score

Let *BestScorIintial* be the best score

Let *IndexHubBestScorIntial* be the Index of the best processing center h to start route

for *all c-product (i)* **do**
  if *CheckingCapacityHubOpened(i)==false* **then**
    OpenBestClosedHub(i)
  end
end

for *all Opened processing center (h)* **do**
  ScorInitial= 2*DistancefromDepot + $Distance_h$
end

$BestScorInitial_h$ = Min (ScorInitial$_h$)

$IndexHubBestScorInitial$ is the index of the processing center that gives $BestScorInitial_h$

Return IndexHubBestScorInitial

6.3.3 Assignment phase

As an assignment phase we have implemented two constructive methods, named *Nearest Neighbor method* (NNH) and *Insertion method*. In the following we explain briefly these two approaches.

6.3.3.1 Nearest Neighbor method (NNH)

This strategy is based on the nearest neighbor principle. A neighbor candidate (processing center or client) is inserted in route if all constraints are satisfied otherwise the second neighbor candidate will be checked, until neither processing centers nor client can be inserted in route. In that case, a new route is created, and initialization phase is used to select the first processing center. This process is repeated until all client requests are satisfied. For each selected neighbor candidate, the kind of constraints that should be satisfied is related to candidate role. In case that the nearest candidate is a client, assignment phase inserts the candidate in the route when constraints of the maximum duration of trip, vehicle's capacity and product availability, are satisfied. In the case that the nearest candidate is a processing center, we distinguish two possibilities : (i) the processing centers is already opened by a previous route, or (ii) a closed processing centers. If a processing center is already opened and its demand of h-product is already satisfied by previous trips, it will be inserted in the current route. But when the processing centers is closed, the *CheckCapacityOpenedHub* function will be called. In this case, if the function returns false, the best processing centers for that product should be identified by the *OpenBestClosedHub* function. If the identified processing centers corresponds to the processing centers that is being tested then this processing centers will be opened, and be inserted in the route, otherwise, the processing centers that is being tested is not opened.

6.3.3.2 Insertion method

This method tries to insert the maximum number of clients and processing centers into the current route. At each iteration, the algorithm chooses a random client and insert it into the best position on the route. When we can not insert no more client into the current route, algorithm tries to insert a processing center and calls clients insertion phase. The client and processing center insertion will be repeated until the Time-Limit of the route is reached. If we cannot insert any client nor processing center into current route, a new route is initialized by a initialization phase to select a new processing center to start a new route. If all clients requests are satisfied the algorithm stops.

6.4 Adaptive Large Neighborhood Search Heuristic for the MPLRP-PD

In this section, we investigated a metaheuristic approach based on Adaptive Large Neighborhood Search (ALNS) to enhance the results of the heuristic approaches presented in the previous section. The ALNS consists of a number of sub-heuristics that are exploited with a frequency according to their historic efficiency.
The meta-heuristic $ALNS$ uses the generated solutions from the heuristic, proposed in the previous section, as initial solution.

6.4.1 Principle of Adaptive Large Neighborhood Search Heuristic

The *Adaptive Large Neighborhood Search* (ALNS) is described in this section. The ALNS can be used to solve a vast group of optimization problems. The ALNS is a local search (LS) approach, in which a number of fast algorithms attempt to modify and then improve the current solution. In each iteration, one of the proposed algorithm will be selected to destroy the current solution and then, an algorithm is selected to repair the solution. Repairing the solution by algorithm means that the new solution is a feasible solution and it can be accepted if it satisfies the criteria defined by the local search approach.

Ropke and Pisinger [203] presented a general heuristic, which is capable of solving five different variants of the vehicle routing problem : the vehicle routing problem with time windows, the capacitated vehicle routing problem, the multi-depot vehicle routing problem, the site dependent vehicle routing problem and the open vehicle routing problem. All these five problems were transformed to a pickup and delivery problem and they solved them using the Adaptive Large Neighborhood Search framework. The idea is inspired from an extension of *Large Neighborhood Search* framework from Shaw [222]. The ALNS adaptively selects among a number of insertion and removal operators, to intensify and diversify the neighborhood search. The proposed heuristic has a number of conveniences : ALNS obtains solutions of high quality, the algorithm is robust and provides some self-calibration strategies. The authors give a general description of the framework and they discussed how the various components can be designed in a particular setting.

Schrimpf et al. [224] also introduced the *Ruin* and *Recreate* paradigm that can considered as the base of ALNS or the *Ripup* and *Reroute* paradigm proposed in [255]. In each iteration, the current solution is partially destroyed and then repaired by applying relevant heuristics.

ALNS also has similarities with *Very Large Neighborhoods Search* (VLNS) presented by Ahuja et al. [4]. In VLNS the algorithm operates on very large neighborhoods selected in a way that they can be searched efficiently.

We can consider the *ALNS* heuristic as a sequence of fixed and optimized operators instead of a sequence of destroy and repair operations. A number of variables that are fixed at their current value are chosen by a fixed operator. The optimized operator explore the neighborhood to find a close-optimal solution that validate the fixed variables. We have to notice that only non-fixed variables can be modified. The fixed operator is too similar to the destroy operator and the optimized operator is similar to the repair operator. The fixed and optimized operators could be useful when using the heuristic where the destroy and repair operators are not perceptive.

The general outline is given in algorithm 6. The main body of algorithm is repeated in lines 6-15. In each iteration, a destroy and one repair neighborhood will be selected in line 6. A flexible layer stochastically adjusts which neighborhoods should be chosen according to their past efficiency that will be recorded as score. As much as one neighborhood N_i has committed to create the solutions, it takes the larger score p_i. The neighborhood that achieved larger score has a larger probability of being selected in the next iterations. The algorithm uses *roulette wheel selection* to select a destroy and a repair operator which is detailed in section 6.4.2.

In line 6, the current solution *sol* is destroyed using a heuristic searching the neighborhood ξ^- (from destroy heuristic subset) and then repair the solution using the heuristic

corresponding to neighborhood ξ^+ (from repair heuristic subset).
Different number of criteria can be used to evaluate how much a neighborhood commit to the solution process : new best solutions are obviously given a large score, but also not previously visited solutions are given a score.
Since each step of the ALNS heuristic involves two neighborhoods (a destroy and a repair neighborhood), the score obtained in a given iteration is divided equally between them. Every s iterations of the ALNS algorithm, the scores p_i are reset, and the probabilities for choosing the neighborhoods are recalculated. Each neighborhood is assigned a minimum probability for being chosen to ensure that statistical information about its performance can be collected. The probabilities for choosing a neighborhood can also be a weighted sum of the score during the last s iterations, and the overall score since the beginning of the algorithm.
ALNS has widely been applied to vehicle routing problem (Azi et al. [7], Demir et al. [71] and Hemmelmayer et al. [109]). Few contributions concern the LRP : Contardo et al. [41] propose an ALNS meta-heuristic with the objective of finding good-quality solutions quickly to solve the 2E-CLRP. They introduce a two-stage algorithm that deals with two types of destroy operators. There are «large operators» that change the current configuration of opened satellites and «small operators», that remove only a certain number of customers, but do not explicitly open or close satellites. The customers that were removed by the destroy operators are then inserted by means of an insertion operator to minimize the objective function value. After the destroy and repair operators are executed, the first level is improved by recursively calling again the ALNS to solve the first-level CLRP. The destroy and repair operators are selected by a roulette wheel mechanism.
Yang and Sun [266] present also ALNS for an electric vehicles battery swap stations location routing problem (BSS-EV-LRP), which aims to determine the location strategy of battery swap stations (BSSs) and the routing plan of a fleet of electric vehicles simultaneously under battery driving range limitation. They propose a four-phase heuristic called SIGALNS to solve the problem. In the proposed SIGALNS, location stage and the vehicle routing stage are alternated iteratively, which considers the information from the routing plan while improving the location strategy.

Algorithm 6 Adaptive Large Neighborhood Search pseudo code

Result: Improved routes

Input : Initial Solution Sol.

Output : Feasible solution Sol^*

Generate a feasible solution sol

$Sol^* \leftarrow Sol$

repeat

  Select one destroy neighborhood ξ^- and one repair neighborhood ξ^+ by applying roulette wheel selection, according p_i

  Create a new solution $sol^{'}$ from sol by applying the chosen neighborhood in previous line

  if $Sol^{'}$ *is feasible* **then**

    $Sol \leftarrow Sol^{'}$

  end

  Update previously scores p_i for each neighborhood i

  if $Cost(Sol) \leq Cost(Sol^*)$ **then**

    $Sol^* \leftarrow Sol$

  end

  return Sol^*

until *stop condition is met*

6.4.2 ALNS for Multi-products Location-Routing problem with Pickup and Delivery

This section describes how the ALNS heuristic has been applied to the MPLRP-PD.

6.4.2.1 Initial solutions

The heuristics that have already been presented in the previous section will be used to provide the initial solution. The two strategies (NNH and Insertion method) will be used to obtain different initial solutions that will be used to start the ALNS.

6.4.2.2 The proposed destroy operators

We proposed various destroy operators related either to clients or to processing centers. It consists of three operators that acting on clients removal (Random client removal, Worst removal, Sequence removal) and two operators that acting on the processing centers (Processing center removal, Closing processing center).

6.4.2.2.1 Random client removal

Random client removal chooses α positions and eliminates the associated request (we notice by request a client and the set of product demands, satisfied in the considered position) randomly and removes them from the solution. α is also considered as parameter and allows fixing the number of requests to remove.

6.4.2.2.2 Worst removal

The main principle of the worst removal heuristic is to choose which positions in the current solution lead to expansive cost. The expensive cost concerns the longest edge into a route. Removing one expensive request associated to an expansive position may be a good strategy to destroy the structure of the current solution. In the MPLRP-PD it's obviously acceptable to try removing high cost requests and re-insert them at another position to obtain a improved solution.

6.4.2.2.3 Sequence removal

Sequence removal heuristic tries to move α' sequences of clients that exist in the solutions. The sequence is selected randomly and each client request, associated to a position that belong to this sequence will be removed. One sequence refers to all clients that exist between two processing centers in a relevant route.

6.4.2.2.4 Processing center removal

Since in MPLRP-PD more than one route may pass by open processing center k denoted r_k, then several routes could involve the same processing center. The *processing center removal* operator eliminates processing center k from r_k where k and r_k are chosen randomly.

6.4.2.2.5 Closing processing center

A MPLRP-PD solution may have several open processing centers which receive their *primary product* and serve the *final product* to clients. The *closing processing center* gets the number of opened processing center k as the input. The heuristic closes one processing center k randomly and eliminates the processing center k from all created routes in the solution. Notice that eliminating processing center k by *Processing center removal* operates only in one route whereas *closing processing center* deals with eliminating processing center k from all routes that pass through this processing center.

6.4.2.3 The proposed repair operators

We proposed also several destroy operators which consist in two operators that act on clients insertion (Best Insertion of client request, First Insertion of client request) and two operators that act on the processing centers (Opening processing center, Insert processing center).

6.4.2.3.1 Best Insertion of client request

We proposed a basic greedy heuristic which gets as input i number of requests and T number of insertion trials for each request. The heuristic inserts one request in each iteration. Let $\Delta f_{i,t}$ denote the change in objective value performed by inserting request i by insertion trial t at the position that induce a minimum increases the objective value. If it is not possible to insert request i in insertion trial t, then we set $\Delta f_{i,t} = \infty$. We then define c_i by formula 6.25. c_i is the *cost* of inserting request i at its best position among T trials which is called the minimum cost position. Finally we choose the request i that minimizes $ft^* = min_{i \in U} c_i$ and insert it at its minimum cost position. U is the set of unplanned requests. This process continues until all requests have been inserted or no more requests can be inserted among T insertion tries.

$$c_i = \text{argmin}_{t \in T} \{\Delta f_{i,t}\} \tag{6.25}$$

Note that we try to insert request without splitting it according to different product. We have to notice that at each iteration we change just one single route (by inserting a request into), and it is unnecessary to re-calculate insertion costs in all the other routes. This idea is helpful to speed up the insertion heuristics implementation.

6.4.2.3.2 First Insertion of client request

The main procedure of *first Insertion client request* heuristic follows the same routine as *best Insertion of client request* heuristic. The only difference is that in each iteration after first successful insertion, algorithm accept the insertion and goes to the next iteration. Observe that here, the goal is to operate more effect of diversifying the neighborhood search and that we are not intended to find the minimum cost position.

6.4.2.3.3 Opening processing center

After applying one destroy operator for processing center on MPLRP-PD solution, the available products through all open processing centers may be not enough to satisfy the request of clients. In this case, we have to open a closed processing center in order to provide more available products. The *opening processing center* heuristic is implemented to fulfill this purpose. The processing center to be opened is selected randomly among all processing centers which provide the same requested product.

6.4.2.3.4 Insert processing center

When in MPLRP-PD solution, the provided products by all opened processing centers are sufficient but there are some routes that don't own enough products for their present clients. The *Insert processing center* is applied to repair these kind of routes. It consists on inserting one opened processing center in the unsatisfied routes. The strategy choose one open processing center randomly.

6.4.2.4 The proposed integrated destroy/repair operator

We proposed also a new heuristic which operates the destroying and repairing phase in the same time and we can't separate them. It gets two open processing center k_1 and k_2 as inputs and tries to exchange their position in l_{k1} and l_{k2}. In ALNS heuristic, we consider the *Exchange open processing center* as an integrated destroy/repair operator. The two selected processing centers provides the same products.

6.4.2.5 Selecting a destroy/repair operator by roulette wheel selection

In last section, the five heuristics : *random client removal*, *worst removal*, *sequence removal*, *processing center removal* and *closing processing center* were defined as destroy operators of ALNS. The *first insertion of client request*, *best insertion of client request*, *opening processing center* and *insert processing center* operate as repairing of solution. We have also proposed a integrated destroy/repair heuristic (*exchange open processing centers*). The classical ALNS selects one destroy operator and one repair operator and uses these all over the search.

The fact of switching between different destroy/repair operators leads to exploit the more suitable heuristic in a given iteration of ALNS and also applying an appropriate heuristic on different instances. In our proposed ALNS, we decide to work by pairs of operators (destroy and repair) and we select only the most suitable pairs to avoid increasing the computational time. More precisely, for example if when we destroy the solution with *processing center removal* operator, using the *opening processing center* seems unnecessary and it will be more suitable to apply the *insert processing center* operator. Therefore, in proposed ALNS we call *processing center removal* and *insert processing center* as a destroy/repair couple and *closing processing center*, *opening processing center* as another destroy/repair operator. The *exchange open processing centers* operator is also used alone as a destroy/repair heuristic. Here, we mention all selected destroy/repair pairs : random client removal and first Insertion of client request, random client removal and best insertion of client request, sequence removal and first Insertion of client request, sequence removal and best insertion of client request, Processing center removal and Insert proces-

sing center, closing processing center and opening processing center, worst removal and best Insertion of client request, exchange open processing center.
The algorithm uses roulette wheel selection to choose a destroy and a repair pair i in $K = \{1, 2, ..., 10\}$. To do this we need to calculate assigned weights for each destroy/repair operators i, denoted w_i. The *roulette wheel selections* choose neighborhood i with given probability in formula 6.26.

$$prob_i = \frac{w_i}{\sum_{j \in K} w_j} \tag{6.26}$$

6.4.2.6 Adaptive weight adjustment

This section describes how w_j can be automatically balanced by considering ALNS performance from earlier iterations. The reasonable way is to record and to keep track of a score for each considered pairs of destroy/repair operators. This score evaluates how well the pair of destroy/repair operators has behaved recently. A large score is representative for a successful pair. The whole search is divided into a number of segments. A segment involves a number of iterations of the ALNS heuristic. The score of all pairs is set to zero at the start of each segment. The score of pair i denoted p_i is incremented with the τ_1, τ_2 or τ_3 values depending on the new solution $x^{'}$. Notice that the current solution refers to the solution of earlier iteration.

— τ_1 : The destroy/repair operator provided a new global best solution $sol(x^{'})$.
— τ_2 : The last destroy/repair operator provided a feasible solution but the cost of the new solution $sol(x^{'})$ is worse than the cost of current solution.
— τ_3 : The last destroy/repair operator provided a feasible solution that the cost of the new solution $sol(x^{'})$ is better than the cost of current solution.

We fixed $\tau_1 = 30$, $\tau_2 = 12$ and $\tau_3 = 8$. The first case occurs when a pair find a new overall best solution. For both other cases, they occur if a heuristic is able to find a solution that is feasible and is accepted by the imposed criteria in the ALNS search.
Improving a solution is always desired but we are also looking for the diversification of the search and these are taken into account by τ_2.
While each iteration of ALNS, a pair of heuristic is applied : a destroy heuristic and an repair heuristic. In the case of improving a solution we are not able to say which destroy or repair was the reason of solution improvement. That's why after applying this neighborhood search, the scores for both heuristics will be updated by the same value. New weights are calculating according to the recorded scores at the end of each segment. Formula 6.27 calculates the weight of heuristic i in segment j, denoted $w_{i,j}$. We assigned all weights of heuristics to 1 in the first segment. At the end of each segment j, the weight for all heuristics i to be applied in segment $j + 1$ are calculated as follows :

$$w_{i,j+1} = (1 - r) * w_{i,j} + (r) * \frac{p_i}{\theta_i} \tag{6.27}$$

where $r = 0.1$.

We select a pair of destroy/repair operator with given probability in formula 6.26. The past score of a neighborhood i is denoted p_i. Destroy and repair operator (neighborhood) could be applied n times in main body of algorithm.

The obtained score of heuristic i during the last segment and the number of times that we have tried to use pair i during the last segment are denoted by p_i and θ_i respectively. Regulating that how quickly the weight adjustment algorithm behaves to changes in the performance of the heuristics is performed by reaction factor r.

6.5 Computational Experiments

In this section we examinate the performance of the proposed method. Since our problem is not considered in the literature, we have tried to adapt known LRP instances to our problem.

The proposed heuristics are applied on adopted 2E-MPLRP-PD instances, described in section 4.6. In original instances, the vehicle capacity of second level Q_2 (i.e., fleet of vehicles that transport products from satellites-depots to clients), vehicle capacity of first level Q_1 (i.e., fleet of vehicles that transport products from depot to satellites-depots) are considered but in our adopted instances a homogeneous fleet of vehicles, with capacity equal to average quantity of Q_1 and Q_2 is considered.

The proposed heuristic was coded in C++ and we evaluated its performance on a PC with Intel (R) Core (TM) Solo CPU 1.40 GHz, 2GB of RAM.

6.5.1 Numerical results of the heuristic

To measure the performance of our method, we have used Cplex solver to solve the mixed integer-programming model; described in section 6.2.2. The computing time of Cplex is limited to one hour. Then, we compared our approach results with the Cplex solution. Table 6.1 summarizes the computational results for NNH and insertion strategy. For each method, we consider a gap that is calculated between the lower bound (LB) and heuristic solution (UB). It is calculated as $[(UB - LB)/UB]$, where LB is obtained by Cplex. Columns Time indicate the computing time of the heuristic in second and the last row shows the average execution time and average Gap. For the first nine instances, the average gaps are of 4,24% for NNH and 13,34% for insertion method. This percentage of gap would rise up to $24,55\%$ for insertion method and to $22,03\%$ for NNH. The Cplex is unable to generate any lower bound during one hour for the instances with more than 50 clients. The heuristic can reach the solution during 2 second even for the large instances with 200 clients for NNH and 30 second for insertion method. We can conclude that our heuristic method is more efficient when it uses the NNH method in assignment step. We notice that the results by using NNH technique are always better than insertion technique.

Instances	*NNH*		*Insertion method*	
	Time	Gap	Time	Gap
coord20-5-1-2e	0	4,24%	0	5,87%
coord20-5-1b-2e	0	3.59%	0	7,65%
coord20-5-2-2e	0	4.19%	1	4,45%
coord20-5-2b-2e	0	3.78%	1	19,70%
coord50-5-1-2e	0	9.86%	4	20,15%
coord50-5-2-2e	1	9.05%	5	9,35%
coord50-5-2bBIS-2e	0	5.09%	3	15,62%
coord50-5-2BIS-2e	0	22.03%	4	24,55%
coord50-5-3-2e	0	10.33%	6	12,59%
coord100-5-1-2e	0	-	10	-
coord100-5-2-2e	1	-	9	-
coord100-5-3-2e	0	-	13	-
coord100-10-1-2e	0	-	12	-
coord100-10-2-2e	0	-	13	-
coord100-10-3-2e	1	-	14	-
coord200-10-1-2e	1	-	24	-
coord200-10-2-2e	2	-	30	-
coord200-10-3-2e	2	-	25	-
average	0,5	4,24%	10	13,34%

Table 6.1: Comparative results of NNH and Insertion strategies compared to CPLEX

6.5.2 Numerical results of ALNS

In this section, we compared the performance of ALNS with two methods. The first is a local search (named RL) which is an adaptation of the Intra-Route Multi Product Two-Opt Local Search *(2-Opt-MP1(S,k_1,i,j))* proposed for 2E-MPLRP-PD (see section 2 of

chapter 5)). The second method is a LNS technique, which is a particular case of ALNS where we choose to apply one destroy and one repair operators. The destroy operator is *random client removal* and the repair operator is *first Insertion of client request*. We define a segment as 15 iterations integrated to ALNS. Table 6.2 (respectively table 6.3) gather the results of our experimentations for LNS, RL and $ALNS$ applied on two initial heuristic solution : (i) table 6.2 with the use of Insertion strategy as assignment phase on heuristic solution and (ii) 6.3 with NNH method. The RL presents the obtained improvement by 2-Opt-MP1 local search.
The proposed approach is able to improve the best heuristic solutions are applied on $MPLRP$-PD. The computational results show that it is worthwhile to apply several sub-heuristics rather than just one. We assumed that the $ALNS$ is too robust and could be adapted to the different problem including various features.

Instances	*LNS*		*RL*		*ALNS*	
	Time	Gap	Time	Gap	Time	Gap
coord20-5-1-2e	47	8,48%	2	25,44%	36	**46,82%**
coord20-5-1b-2e	52	14,57%	2	**45,34%**	35	29,76%
coord20-5-2-2e	51	17,20%	1	**41,97%**	57	38,56%
coord20-5-2b-2e	48	0,00%	2	0,00%	24	**42,98%**
coord50-5-1-2e	61	8,31%	4	10,09%	52	**26,89%**
coord50-5-2-2e	67	16,08%	3	**39,32%**	69	29,93%
coord50-5-2bBIS-2e	70	7,49%	3	12,98%	58	**29,95%**
coord50-5-2BIS-2e	68	15,33%	2	24,15%	94	**27,48%**
coord50-5-3-2e	66	11,48%	5	20,53%	69	**24,03%**
coord100-5-1-2e	103	3,94%	8	28,11%	167	**32,51%**
coord100-5-2-2e	105	13,72%	8	23,17%	81	**30,76%**
coord100-5-3-2e	103	8,76%	7	**31,82%**	180	16,87%
coord100-10-1-2e	93	7,10%	9	15,79%	139	**18,56%**
coord100-10-2-2e	98	2,42%	9	14,04%	63	**19,97%**
coord100-10-3-2e	101	10,37%	9	24,29%	133	**24,66%**
coord200-10-1-2e	268	7,74%	17	**24,53%**	250	21,27%
coord200-10-2-2e	165	10,07%	21	19,91%	44	**22,38%**
coord200-10-3-2e	155	1,53%	17	19,95%	35	**21,89%**
average	96	9,14%	7	23,41%	88	28,07%

TABLE 6.2: Comparative results on Insertion strategy

The last column in tables 6.2 and 6.3 show ALNS results.
For each strategy we consider two columns : (i) time execution in second and (ii) gap improvement compared to initial heuristic solution.
We have done several test to fix our parameters as following : *segment=10*, *Iteration number=15*, $\alpha = 5$, $\beta = 10$, $\alpha^{'} = 1$, $\beta^{'} = 5$ and *Iteration number of ALNS=50*. As we can see the $ALNS$ provides the best solution with 28,07% average improvement when we use insertion method as initial solution and 23,19% for NNH initialization. The execution time of ALNS remains reasonable.
The results showed that LNS provides the worst results because the chosen operators may not be the most efficient. The ALNS method improves significantly the results of the two initial heuristics. In the case of the initial heuristic with insertion strategy, the method gets the best gap in 13 cases, while RL reaches the best gap in only 5 cases and LNS does not get any better gap. In the case of initial solution, using NNH type, while RL provides an average gap of $3,97\%$, ALNS obtain the average gap of $23,19\%$. The ALNS gets the best gap in 16 cases against one single case for RL and only one case for LNS. We can also conclude that the results of ALNS do not depend on the quality of the initial solution.
RL improve better the insertion method solution than NNH method because it uses an improvement within a route and do not act between several routes. The NNH provides the solutions that the nodes are too close to each other and it's difficult to improve it by the RL.

Instances	*LNS*		*RL*		*ALNS*	
	Time	Gap	Time	Gap	Time	Gap
coord20-5-1-2e	51	3,91%	0	1,42%	70	**21,71%**
coord20-5-1b-2e	69	8,70%	0	0,83%	63	**12,42%**
coord20-5-2-2e	52	3,79%	0	0,00%	39	**25,95%**
coord20-5-2b-2e	49	6,40%	1	0,00%	45	**25,29%**
coord50-5-1-2e	80	5,38%	0	4,70%	40	**24,77%**
coord50-5-2-2e	68	3,22%	1	6,44%	24	**23,17%**
coord50-5-2bBIS-2e	72	8,58%	1	4,33%	29	**33,16%**
coord50-5-2BIS-2e	76	6,55%	1	2,59%	91	**26,98%**
coord50-5-3-2e	73	9,76%	1	15,38%	57	**25,97%**
coord100-5-1-2e	113	2,58%	2	13,94%	150	**12,34%**
coord100-5-2-2e	98	4,14%	1	5,48%	104	**24,30%**
coord100-5-3-2e	107	6,98%	1	3,02%	158	**28,52%**
coord100-10-1-2e	110	2,96%	1	1,64%	43	**13,21%**

coord100-10-2-2e	129	7,38%	2	2,80%	70	**31,25%**
coord100-10-3-2e	126	5,31%	1	5,74%	196	**39,67%**
coord200-10-1-2e	183	2,15%	5	1,71%	248	**21,00%**
coord200-10-2-2e	223	**6,29%**	3	1,15%	84	4,85%
coord200-10-3-2e	155	1,79%	3	0,32%	382	**22,84%**
average	102	5,33	1	3,97%	105	**23,19%**

TABLE 6.3: Comparative results on NNH method

Figure 6.4 presents all ALNS pairs used in our methods. Preliminary tests allow us to define the number of client to remove for the random client removal, ($\alpha = 1$ or $\alpha = 2$) ; and the number of sequence to remove in sequence removal $\alpha' = 1$.

Neighborhood	*destroy operator*	*repair operator*
OP1	Random client removal (1)	First Insertion of client request
OP2	Random client removal (1)	Best Insertion of client request
OP3	Random client removal (2)	First Insertion of client request
OP4	Random client removal (2)	Best Insertion of client request
OP5	Sequence removal (1)	First Insertion of client request
OP6	Sequence removal (1)	Best Insertion of client request
OP7	Exchange open processing center	-
OP8	processing center removal	Insert processing center
OP9	Closing processing center	Opening processing center
OP10	Worst removal	Best Insertion of client request

TABLE 6.4: ALNS pairs of destroy/repair operators tested

Here in table 6.5, we present the number of times that one operator reaches : (i) a new global best solution named $N\tau_1$, (ii) a feasible solution with worse cost compared to the current solution, named $N\tau_2$, and (iii) a feasible solution with a better cost than the current solution, named $N\tau_3$. The column 2, 3, 4 and 5 gather the results when the initial solution uses insertion strategy and column 6 to 9 gather the result when the initial solution uses NNH strategy. Column 5 (and 6 respectively) computes the sum of occurrency for each used pair with NNH initialization (insertion method respectively).

Instances	*NNH*	*Insertion method*

	$N_{\mathcal{T}_1}$	$N_{\mathcal{T}_2}$	$N_{\mathcal{T}_3}$	sum	$N_{\mathcal{T}_1}$	$N_{\mathcal{T}_2}$	$N_{\mathcal{T}_3}$	sum
OP1	13	71	25	109	7	130	55	192
OP2	13	85	20	118	15	78	50	143
OP3	6	21	21	48	7	113	25	145
OP4	21	66	38	125	22	68	40	130
OP5	16	70	36	122	9	76	35	120
OP6	24	37	33	94	23	76	59	158
OP7	75	40	1	116	5	102	61	168
OP8	90	25	40	155	77	37	79	193
OP9	39	0	0	39	18	0	0	18
OP10	10	614	11	635	3	954	18	975

TABLE 6.5: comparative results of the use of the destroy/repair operators

Instances	*ALNS/NNH*	*ALNS/Insertion method*
coord20-5-1-2e	0,00%	26,91%
coord20-5-1b-2e	36,17%	22,19%
coord20-5-2-2e	0,00%	21,85%
coord20-5-2b-2e	0,00%	4,10%
coord50-5-1-2e	2,16%	0,00%
coord50-5-2-2e	0,00%	27,06%
coord50-5-2bBIS-2e	0,00%	10,21%
coord50-5-2BIS-2e	0,00%	7,23%
coord50-5-3-2e	0,00%	19,52%
coord100-5-1-2e	28,53%	0,00%
coord100-5-2-2e	0,00%	12,43%
coord100-5-3-2e	0,00%	29,84%
coord100-10-1-2e	0,00%	33,87%
coord100-10-2-2e	0,00%	23,02%
coord100-10-3-2e	0,00%	12,37%
coord200-10-1-2e	0,00%	35,08%
coord200-10-2-2e	1,52%	6,57%
coord200-10-3-2e	0,00%	14,97%
average	3,80%	17,07%

TABLE 6.6: ALNS results compared to best solution

Table 6.5 shows the efficiency of selected operators. All operators were used. We can notice that the most used operators is operator OP10, It seems providing improving of solution cost diversity rather than (Nτ_2 = 614 for NNH and Nτ_2 = 954 for insertion method).
The best combination for the client removal operator seems to be the best insertion client (OP4) with Nτ_1 = 21 for NNH and Nτ_1 = 22 for Insertion method. The processing center operators allow achieving an interesting gains by improving the best solution : Nτ_1 for OP8 is equal to 90 and Nτ_1 for OP7 is equal to 75 in the case of NNH. It would be interesting to define diversity measure for each operator and analyze the efficiency regarding not only to the quality of solution cost but also the diversification.
Table 6.6 gathered the gaps between ALNS solution and the best solution obtained by the proposed heuristics, LNS, RL or ALNS. For example in the first instance, ALNS method with using NNH returns best solution $(0, 0\%)$ and in the case of insertion method this gap is equal to $(26, 91\%)$. The ALNS solution with NNH strategy seems to be the best method. It obtains the best solution in 14 cases whereas there are only 2 best solutions, obtained by ALNS with Insertion strategy. Only 3 best solutions are provided by other methods (LNS or RL).

6.6 Conclusion

In this chapter, we have studied a variant of Muti Product Location-Routing Problem and Pickup and Delivery *(MPLRP-PD)*, in which only one distribution level is considered. This variant can be viewed as an extension of vehicle routing problem with pickup and delivery, where the pickup and delivery nodes must be located. The second approach is a metaheuristic based on ALNS. We have proposed 5 specific destroy operators, 4 specific repair operators and one integrated destroy/repair operator. These operators are related either to client node or processing center nodes. We proposed a mixed integer linear programming formulation for this problem. Since the problem is NP-hard, we have proposed heuristic approaches. The first one is a constructive approach based on two client assignment strategies (neighborhood strategy and insertion strategy).
Since there is no instance compatible with *MPLRP-PD* in the literature, we have adapted known 2E-MPLRP-PD instances to evaluate the performance of our approach. Computational results show that our constructive heuristics can reach good solution during reasonable time, even for the larger instance with the size of 3 products, 200 customers

and 10 processing centers. A comparison with $CPLEX$ shows that the proposed heuristic is efficient to solve small and large size *MPLRP-PD* instances.

$ALNS$ is proposed to improve the heuristic solution. The ALNS is able to improve the heuristic solution by $28,77\%$ on average. It is also most competitive than a LNS version in which only one repair and destroy operator is used, and a local search approach based on an adaptation of the Intra-Route Multi Product Two-Opt Local Search *(2-Opt-MP1(S,k_1,i,j))* developed for the 2E-MPLRP-PD. The best solutions for MPLRP-PD are obtained by ALNS for all instances except for 3 instances.

In further research, developing more efficient lower bounds and metaheuristic approaches can be considered as a perspective.

Chapitre 7

Conclusions and perspectives

In the context of this thesis, we focused on the problems of vehicle routing with constraints of location and processing centers assignment (LRP). The objective is the minimization of opening costs of processing centers, operating costs of routes and exploited vehicle costs. We aim to integrate several aspects that have not been treated simultaneously in the literature of location-routing problem : the multi-product aspect, the opportunity to make the pick-up and delivery in the same route, and the use of processing centers as intermediate facilities in the route. We proposed two extensions of the LRP tacking into account the constraints cited above, that allow modeling real applications in supply chain, characterized by complex transportation networks. These complex transportation networks may include factories, warehouses, customers and suppliers. This is particularly interesting because the freight can be consolidated through one or more processing centers, and so allows considerable saving. The new proposed problems have not been studied in the literature, although they can model several real applications like drink distribution, home delivery service and grocery store chains.

The study was first landed on an overview of the state of the art, related to LRP variants (and VRP as a special case) that come closest to our problems, and more particularly the integration of multi-echelon, pickup and delivery, multi-depot and multi-commodity aspects. We proposed a summary of the addressed issues in the literature.

Following a state of the art on the LRP work, we proposed two new generalizations of LRP. The first generalization, called 2E-MPLRP-PD (Two-echelon Multi-Product Location Routing Problem with Pickup and Delivery) integrates the three constraints cited above and considers a two level distribution system, in which two types of vehicles are considered : primary vehicles to serve the processing centers from the depot and secondary vehicles to serve customers from the processing centers.

The second generalization, named MPLRP-PD (Multi-Product Location Routing Pro-

blem with Pickup and Delivery), is an integrated version of the first model, in which only one distribution network level is considered. The problem can be also viewed as an extension of a vehicle routing problem with pickup and delivery, where the pickup and delivery nodes must be located. For both problems mentioned above, mixed integer linear models have been proposed and implemented with Cplex V14.
Since the studied problems are NP-hard, we have chosen as solving approaches, constructive heuristics, hybrid methods and a metaheuristic approaches. We have generated 36 instances derived from LRP literature instances with up to 200 customers and 10 processing centers.
For the 2E-MPLRP-PD, we have proposed non-trivial extensions of nearest neighbour and insertion approaches. We have developed clustering based approaches that have not been extensively investigated with regards to location routing.
Extensive computational experiments show that the clustering approach is very competitive and outperforms the other heuristics when $n < 100$. For larges instances, the insertion approach is more efficient.
Following the promising results obtained by the insertion method and clustering method, we have developed and tested a number of variants of these two approaches which helped to intensify the test and to get a comparative study of performance for all these heuristics. For the insertion version, four new strategies to integrate several criteria for choosing a client and for choosing a processing center to insert into a current route was developed. The insertion position of a client is selected using a score by maximizing the number of products to be satisfied and minimizing traversed distance of the route. The insertion position of a processing center depends on the length of the current route and is selected to maximize the efficiency of the processing center and to minimize the distance of the route. We have also developed three variants of the hybrid clustering method : 1) iterative approach, 2) approach using clusters from the nearest neighbor heuristic, 3) approach using processing centers as a starting point to form the clusters. In all mentioned variants Cplex is used to construct the route of each cluster. An analysis of the different strategies of clustering approaches, implemented in the proposed heuristics was performed.
We have improved the results obtained by the heuristic methods by local search approaches. We have proposed two local search methods, the «routing improvement local search» and «the processing centers location improvement Local Search». The first one consists of three movements : Intra-Route Multi Product Two-Opt Local Search (2-OPT-MP1), Inter-Route Multi Product Two-Opt Local Search (2-OPT-MP2), and Merging Product Local Search (MP). These moves are related to clients and intermediate processing centers but do not act on the processing centers used as departure node of routes. The second method «processing centers location improvement procedures» contains moves

that act on the processing centers. It acts on the set of opened processing centers. These movements are called «change-state-PC» and «swap- routes-PC». The change-State-PC procedure mutates the state of two processing centers, and the swap-routes-PC swaps the position of two processing centers used as departure node of routes. The 2-OPT-MP2 gives a better improvement, when the other local search procedure performs a limited improvement (between 1% to 4%). These results encouraged us to develop a metaheuristic approach to enhance the results obtained by the local search procedures.

The proposed metaheuristic for 2E-MPLRP-PD called ILS (Iterative Local Search), is widely developed to solve VRP and rarely used for LRP. By using the proposed local search methods and two specific perturbation procedures, we have implemented three versions of ILS : (i) Randomized ILS (*R-ILS*), (ii) Neighborhood Descent ILS (*ND-ILS*) and (iii) Score based ILS (*S-ILS*). The Randomized ILS, selects randomly one of the routing improvement local search procedures at each iteration. The Neighborhood Descent ILS, uses all the routing improvement local search procedures and the «change-state-pc» local search procedure. In this variant, the most complicated neighborhoods are tested only when the previous improvement procedure leads to no improvement as following : (2-OPT-MP1 $\rightarrow$ MP $\rightarrow$ 2-OPT-MP2 $\rightarrow$ change-state-PC). The Score based ILS (*S-ILS*) uses an adaptive technique to choose a local search and perturbation procedure at each iteration.

Regarding the overall performance, the three ILS variants give better gaps than the local search. The best strategy is obtained by *S-ILS*.

For the MPLRP-PD, we have proposed an heuristic method based on two phases : (i) initialization phase and (ii) assignment phase. As an assignment phase we have implemented two constructive methods, named Nearest Neighbor method (NNH) and Insertion method. The heuristic with NNH strategy gives better results than the heuristic with insertion strategy.

A metaheuristic called «Adaptive Large Neighborhood Search» is proposed for the MPLRP-PD to improve the heuristic solutions. We proposed various destroy operators related either to clients or to processing centers. It consists on three operators that acting on clients removal (random client removal, worst removal, sequence removal), and two operators that acting on the processing centers (processing center removal, closing processing center). We proposed also several repair operators which consists in two operators that acting on clients insertion (best insertion of client request, first insertion of client request), and two operators that acting on the processing centers (opening processing center, insert processing center). One integrated destroy/repair operator named «exchange open processing center» operator is also proposed. The proposed ALNS is able to improve considerably the best heuristics which are applied on $MPLRP\text{-}PD$. The computational results show that it is worthwhile to apply several sub-heuristics rather than just one. The meta-heuristic $ALNS$ uses the generated solutions from the heuristic with NNH and insertion strategy as initial solutions. The results show that even if the heuristic with insertion strategy gives a bad results, the ALNS achieves better results when the initialization heuristic is based on insertion strategy.

Our research has been the subject of 1 paper appeared in international journal 4 international conferences with acts ISBN, and 1 national communication. The list of publications is given at the end of this conclusion.

We want to continue this research in several directions :

— considering other realistic constraints as : (1) the split of client demand for a given product, (2) the possibility for a processing center to provide multiple products, (3) integration of the time aspect in the problems as time windows associated with processing centers and/or customers, (4) integration of the selective aspect to relax the constraint on meeting the demands of all customers.

— developing lower bounds and improving the proposed mathematical formulation.

— developing a hybrid approach including the clustering method and the developed metaheuristics. It consists in using the proposed metaheuristic within each cluster, coupled with an iterative process to explore several clustering solution.

— given the interest of the results of the ILS version with score, it would be interesting to test this approach on other versions of VRP and/or LRP to measure its effectiveness.

— computing a diversity measure for each diversity of destroy/repair operators in the ALNS approach, to perform a better analyze of the effectiveness of the proposed operators and to improve the adaptability of the method. This technique can also be adapted to the scored based ILS version.

Summary list of publications :

— Y. Rahmani, W. Ramdane Cherif, A. Oulamara (2015). The Two-Echelon Multi-products Location-Routing problem with Pickup and Delivery : Formulation and heuristic approaches. International Journal of Production Research, 2015.

— Y.Rahmani, W.Ramdane Cherif, A. oulamara (2015). A local search approach for the

Two-Echelon Multi-products Location-Routing problem with Pickup and Delivery, The 2015 IFAC Symposium on Information Control in Manufacturing (INCOM 2015), Ottawa, Canada, May 11-13, 2015. IFAC-PapersOnLine, Volume 48, Issue 3, 2015, Pages 193-199.

— Y. Rahmani, A. Oulamara, W. Ramdane-Cherif (2014), Using Clustering Method to Solve Two Echelon Multi-Products Location-Routing Problem with Pickup and Delivery. ICORES 2014 : 425-433.

— Y. Rahmani, A. Oulamara, W. Ramdane Cherif (2013). Multi-products Location-Routing problem with Pickup and Delivery, ICALT IEEE, 29-31 May, Sousse, Tunisie, pages 115,122 978-1-4799-0313-9/13/$31.00 2013 IEEE, 2013 http ://ieeexplore.ieee.org.

— Y. Rahmani, A. Oulamara, W. Ramdane Cherif (2013). Multi-products Location-Routing problem with Pickup and Delivery : Two-Echelon model. 11th IFAC Workshop on Intelligent Manufacturing Systems (IMS 2013), pages 22-24 May, Sao Paulo, Brazil, pages 124-129, 2013, ISBN : 978-3-902823-33-5, http ://www.ifac-papersonline.net/.

— Y. Rahmani, A. Oulamara, W.Ramdane Cherif (2013), probleme de Localisation-Routage Multi-produits avec collecte et livraison, ROADEF, Troyes, France 13-15 fevrier, 2013.

Bibliographie

[1] Y. Agarwal, K. Mathur, and H.M. Salkin. A set−partitioning-based exact algorithm for the vehicle routing problem. Networks, 19 :731−749, 1989.

[2] A. Wren, Computers in Transport Planning and Operation. Ian Allen, London, 103-106, 1971.

[3] S. Anily, J. Bramel. Approximation algorithms for the capacitated traveling salesman problem with pickups and deliveries. Naval Res Logist 46 :654-670, 1999 .

[4] R.K. Ahuja, O. Ergun, J.B. Orlin, A.P. Punnen.A survey of Very Large Scale Neighborhood Search Techniques, Discrete Applied Mathematics 123, 75-102, 2002.

[5] C.H. Aikens, Facility location models for distribution planning, European Journal of Operational Research 22, 263−279, 1985.

[6] A. Ahmadi-Javid, N. Azad. Incorporating location, routing and inventory decisions in supply chain network design. Transportation Research Part E, 46, 582−597, 2010.

[7] N. Azi, M.Gendreau, J.Y. Potvin. An adaptive large neighborhood search for a vehicle routing problem with multiple routes. Computers & Operations Research, Volume 41, Pages 167-173, 2014.

[8] J.Ahn, de O. Weck, Y.Geng, D.Klabjan. Column generation based heuristics for a generalized location routing problem with profits arising in space exploration. European Journal of Operational Research, 223, 47−59, 2012.

[9] M.Albareda-Sambola, E.Fernndez, S. Nickel. Multiperiod location-routing with decoupled time scales. European Journal of Operational Research, 217, 248−258, 2012.

[10] M.Adelzadeh, V. M. Asi, M. Koosha, (in press). A mathematical model and solving procedure for the multi-depot vehicle routing problem with fuzzy time

window and heterogeneous vehicle. International Journal of Advanced Manufacturing Technology. doi : http ://dx.doi.org/10.1007/s00170-014-6141-8.

[11] E. Angelelli and M.G. Speranza. The periodic vehicle routing problem with intermediate facilities. European Journal of Opearational Research, 137 :233−247, 2002.

.

[12] E. Angelelli, M.G. Speranza, The application of a vehicle routing model to a waste-collection problem : Two case studies, Journal of the Operational Research Society 53 944−952, 2002.

[13] E.Ardjmand, G.Weckman, N. Park, P. Taherkhani, M.Singh, Applying genetic algorithm to a new location and routing model of hazardous materials. International Journal of Production Research , 53(3), pp. 916-928, 2015.

[14] P. Augerat, J. M. Belenguer, E. Benavent, A. Corberan, and D. Naddef. Separating capacity constraints in the CVRP using tabu search. European Journal of Operational Research, 106 :546−557, 1998.

[15] N. Ascheuer, M. Junger, G. Reinelt, A branch-and-cut algorithm for the asymmetric traveling salesman problem with precedence constraints, Computational Optimization and Applications 17, 61−84, 2000.

[16] L. Bianco, A. Mingozzi, S. Riccardelli, M. Spadoni, Exact and heuristic procedures for the traveling salesman problem with precedence constraints, based on dynamic programming, INFOR 32, 19−32, 1994.

[17] J. Barcelo and J. Casanovas. A heuristic lagrangean algorithm for the capacitated plant location problem. European Journal of Operational Research, 15(2) :212226, 1984.

[18] A. Balakrishnan, Integrated facility location and vehicle routing models : recent work and future peospects, American Journal of Mathematical and Management Sciences, Vol.7, No.1&2, pp.35-61, 1987.

[19] R. Bowerman, B. Hall , P. Calamai, A multi objective optimization approach to urban school bus routing : formulation and solution method, Transportation Research, Part A, Vol.22, No.2pp.107-123, 1995.

[20] R. Borges Lopes, F. Plastria, C. Ferreira, B. Sousa Santos. Location-arc routing problem : Heuristic approaches and test instances. Computers & Operations Research, Volume 43, Pages 309-317, March 2014.

[21] R. E. BurksJr. An adaptive taboo search heuristic for the location routing

pickup and delivery problem with time windows with a theater distribution application, Ph.D. Dissertation, Air force institute of technology, 2006

[22] R. Baldacci, N. Christofides, and A. Mingozzi. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. Mathematical Programming Series A and B, 115(2) :351−385, 2008

[23] M.L. Brandeau, S.S. Chiu. An overview of representative problems in location research. Management Science,35 :645−74, 1989.

[24] F. Bernardes de Oliveira, R. Enayatifar, H. Javedani Sadaei, F. G. Guimaraes, J.Y. Potvin. A cooperative coevolutionary algorithm for the Multi-Depot Vehicle Routing Problem. Expert Systems with Applications, Available online 2 September 2015.

[25] R. Baldacci, E. Hadjiconstantinou, and A. Mingozzi. An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. Operations Research, 52 :723−738, 2004.

[26] M.L. Balinski and R.E. Quandt. On an integer program for a delivery program. Operations Research, 12 :300−304, 1964.

[27] J.E. Beasley. Route-first cluster-second methods for vehicle routing. Omega, 11(4) :403−408, 1983

[28] J.E. Beasley. An algorithm for set covering problem. European Journal of Operational Research, 31(1) :85−93, 1987.

[29] J.M. Belenguer, E. Benavent, C. Prins, C. Prodhon, and R. Wolfer-Calvo. A cutting plane method for the capacitated location-routing problem. Computer & Operations Research, 38 :931−941, 2011.

[30] M. Bellmore and J.C. Malone. Pathology of travelling salesman subtour elimination algorithms. Operations Research, 19 :278−307, 1971.

[31] J.F. Benders. Partitioning procedures for solving mixed-variables programming problems. Numerische Mathmatik, 4(3) :238−252, 1962.

[32] E. Balas, M. Fischetti, W.R. Pulleyblank, The precedence-constrained asymmetric traveling salesman polytope, Mathematical Programming 68 − 241−265, 1995.

[33] R. Bellman. Dynamic programming and the numerical solution of variational problems, Operations Research, 5, 277−288, 1957.

[34] G. Berbeglia, J-F.Cordeau, I. Gribkovskaia, and G. Laporte, Static Pickup and Delivery Problems : A Classification Scheme and survey. TOP, 15 :1-31, 2007.

[35] J. Berger and M. Barkaoui. A new hybrid genetic algorithm for the capacitated vehicle routing problem. Journal of the Operational Research Society, 54 :1254−1262, 2003.

[36] R.Baldacci, A. Mingozzi, R. Wolfler-Calvo. An exact method for the capacitated location-routing problem. Operations Research, 59(5), 1284−1296, 2011.

[37] S.Barreto, C. Ferreira, J.Paixao, and Sousa Santos, B. Using clustering analysis in a capacitated location-routing problem. European Journal of Operational Research 179, pp. 968-977, 2007.

[38] M.Boccia, T.Crainic, A.Sforza, and C. Sterle, A metaheuristic for a two echelon location-routing problem. In P. Festa (Ed.), Symposium on Experimental Algorithms - SEA 2010. Lecture Notes in Computer Science, Vol. 6049, pp. 288-301, 2010.

[39] M.Boccia, T.Crainic, A.Sforza, and C. Sterle, Location-routing models for designing a two-echelon freight distribution system. Tech. Rep., CIRRELT-2011-06, Université de Montreal, 2011.

[40] A.Bruns, and A.Klose, An iterative heuristic for location- routing problems based on clustering. In : Proceedings of the Second International Workshop on Distribution Logistics, The Netherlands, pp. 1-6, 1995.

[41] C. Contardo, V. Hemmelmayr, and T.G.Crainic, Lower and upper bounds for the two-echelon capacitated location-routing problem. Computers and Operations Research, 39(12), pp. 3185-3199, 2012.

[42] T.G.Crainic, S.Mancini, G. Perboli, and G. Tadei, R. Heuristics for the two-echelon capacitated vehicle routing problem. WP. CIRRELT-2008-46, CIRRELT, Université de Montréal, Canada, 2008.

[43] TG. Crainic, S. Mancini, G. Perboli, R.Tadei, Clustering-based heuristics for the two-echelon vehicle routing problem. Technical report CIRRELT-2008-4, 2008.

[44] M.C. Carrera, W. Portmann, Ramdane Cherif, Scheduling problems for logistic platforms with fixed staircase component arrivals and various deliveries hypotheses. Proceedings of the 2^{th} International Conference on Applied Operational Research, Finlande, pages 517528, 2010.

[45] A. Claus, A new formulation for the travelling salesman problem. SIAM Journal on Algebraic Discrete Methods, 5, 21−25. 1984.

[46] S. Ceria, P. Nobili, and A. Sassano. A lagrangian-based heuristic for large-scale set covering problems. Mathematical Programming, 81 :215−228, 1998.

[47] N. Christodes and J.E. Beasley. A tree search algorithm for the p-median problem. European Journal of Operational Research, 10 :196−204, 1982.

[48] G. Clarke and J.W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research, 12 :568−581, 1964.

[49] J.F. Cordeau, M. Gendreau, A. Hertz, G. Laporte, and J.S. Sormany. New heuristics for the vehicle routing problem. Technical Report G-2004−33, GERAD, 2004.

[50] C. H. Christiansen. Elements of Vehicle Routing under uncertainty. PhD Thesis, Department of Business Studies, Aarhus School of Business, University of Aarhus. April, 2007.

[51] J.F. Cordeau, M. Gendreau, A. Hertz, G. Laporte, and J.S. Sormany. New heuristics for the vehicle routing problem. In A. Langevin and D. Riopel, editors, Logistics Systems - Design and Optimization, pages 279−298. Springer, 2005.

[52] J.F. Cordeau and G. Laporte. Tabu search heuristics for the vehicle routing problem. Technical Report G-2002−15, GERAD, March 2002.

[53] T.G. Crainic, N. Ricciardi, and G. Storchi. Advanced freight transportation systems for congested urban areas. Transportation Research Part C, 12 :119−137, 2004.

[54] C.Contardo, J.-F. Cordeau, B. Gendron. An exact algorithm based on cut- and-column generation for the capacitated location-routing problem. INFORMS Journal on Computing, 26(1), 88-102, 2014.

[55] C. Contardo, R. Martinelli. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. Discrete Optimization, 12, 129-146, 2014.

[56] T.G. Crainic, N. Ricciardi, and G. Storchi. Models for evaluating and planning city logistic transportation systems. Technical Report 65, CIRRELT, 2007.

[57] I.M. Coelho, P.L.A. Munhoz, M.N. Haddad, M.J.F. Souza, L.S. Ochi. A hybrid heuristic based on General Variable Neighborhood Search for the Single Vehicle Routing Problem with Deliveries and Selective Pickups, Electronic Notes in Discrete Mathematics, Volume 39, Pages 99-106, 2012. .

[58] D. Cattaruzza, N. Absi, D. Feillet, D. Vigo. An iterated local search for the multi-commodity multi-trip vehicle routing problem with time windows,Computers & Operations Research, Volume 51, Pages 257-267, 2014.

[59] B. Crevier, J.F. Cordeau, and G. Laporte. The multi-depot vehicle routing

problem with inter-depot routes. European Journal of Operational Research, 176 :756−773, 2007.

[60] N. Christofides, N. Eilon, An algorithm for the vehicle-dispatching problem. Operational Research Quarterly 20 (3), 309−318, 1969.

[61] A.L. Chen, G.K. Yang, and Z.M. Wu. Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. Journal of Zhejiang University science A, 7(4), pages 607-614, 2006.

[62] T.G. Crainic, S.Mancini, G.Perboli, R.Tadei. Lower bounds for the twoechelon capacitated vehicle routing problem. In : Annual Conference of the EURO Working Group on Metaheuristics, Troyes, France, July 23−24, 2008.

[63] A. Corberan, G. Laporte (Eds). Arc routing : problems, methods and applications. SIAM, Philadelphia, 2014.

[64] S. Cetiner, C.Sepil, H. Sural. Hubbing and routing in postal delivery systems. Annals of Operations Research, 181, 109−124, 2010.

[65] R.Cuda, G. Guastaroba, M.G. Speranza, A survey on two-echelon routing problems. Computers & Operations Research, 55, pp. 185-199, 2015

[66] R.de Camargo, G. de Miranda, A. Lokketangen. A new formulation and an exact approach for the many-to-many hub location-routing problem. Applied Mathematical Modelling, 37, 7465−7480, 2013.

[67] M. Drexl, M. Schneider, 2015. A survey of variants and extensions of the location-routing problem. European Journal of Operational Research, 241(2), pp. 283-308, 2015.

[68] M. Dror, D. Fortin, C. Roucairol. Redistribution of self-service electric cars : A case of pickup and delivery. Tech. Rep. W.P. 3543, INRIA-Rocquencourt, Rocquencourt, France, 1998.

[69] U. Derigs, T. Dohmer Indirect search for the vehicle routing problem with pickup and delivery and time windows. OR Spectrum 30 :149−165, 2008.

[70] Y. Dumas, J. Desrosiers, F. Soumis. The pickup and delivery problem with time windows. European Journal of Operational Research. 54 :7−22, 1991.

[71] E. Demir, T. Bekta, G. Laporte. An adaptive large neighborhood search heuristic for the Pollution-Routing Problem. European Journal of Operational Research, Volume 223, Issue 2, 1, Pages 346-359, 2012.

[72] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system : Optimization by a colony of cooperating agents. IEEE Tansactions on Systems, Man and Cybernetics- Part B, 26 :29−41, 1996.

[73] G.B. Dantzig and K.H. Ramser. The truck dispatching problem. Operations Research, 6 :80−91, 1959.

[74] M. Desrochers and T.W. Verhoog. A matching based savings algorithm for the vehicle routing problem. Les Cahiers du GERAD G-89−04 - Ecole des Hautes Etudes Commerciales de Montreal, 1989.

[75] H.Derbel, B. Jarboui, S. Hanafi, H. Chabchoub. An iterated local search for solving a location-routing problem. Electronic Notes in Discrete Mathematics 36, 875-882, 2010.

[76] 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, Issue 3, pp 2865-2871, 2012.

[77] C.Duhamel, P. Lacomme, C.Prins, and C. Prodhon, A GRASP ELS approach for the capacitated location routing problem. Computers & Operations Research. 37(11), pp. 1912-1923, 2010.

[78] C. Duhamel, P. Lacomme, C. Prins, and C. Prodhon. A GRASP ELS approach for the capacitated location-routing problem. Computers & Operations Research, 37 :1912−1923, 2010.

[79] E. Dijkstra. A note on two problems in connexion with graphs. Numerische Mathematik, 1, 269−271,1959.

[80] S.Eilon, C.D.T. Watson-Gandy,and N. Christofides. Distribution Management : Mathematical Modelling and Practical Analysis. Griffin, 1971.

[81] T.A. Feo and M.G.C. Resende. A probabilistic heuristic for a computationally difficult set covering problem. Operations Research Letters, 8 :67−71, 1989. .

[82] M. Fischetti, P. Toth, and D. Vigo. A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs. Operations Research, 42 :846−859, 1994.

[83] M. H. Fazel Zarandi, A. Hemmati, S. Davari, B. Turksen. Capacitated location-routing problem with time windows under uncertainty, Knowledge-Based Systems, Volume 37, Pages 480-489, 2013.

[84] M.L. Fisher. Optimal solution of vehicle routing problems using minimum fc-trees. Operations Research, 42 :626−642, 1994.

[85] M.L. Fisher and R. Jaikumar. A generalized assignement heuristic for vehicle routing. Networks, 11 :109−124, 1981.

[86] R. Fukasawa, J. Lysgaard, M. Poggi de Aragao, M. Reis, E. Uchoa, and R.F. Werneck. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. Mathematical Programming, 106 :491−511, 2006.

[87] M. Gondran, M. Minoux. Graphes et Algorithmes. Eyrolles, 1995.

[88] F. Glover. Tabu search and adaptive memory programing - advances, applications and challenges. In R.S. Barr, R.V. Helgason, and J.L. Kennington, editors, Interfaces in Computer Science and Operations Research, pages 1−75. Kluwer, 1996.

[89] A. Gibbons. Algorithmic Graph Theory. Cambridge, 1985.

[90] B. George, B. Dantzig ; Philip Wolfe. Decomposition Principle for Linear Programs. Operations Research 8 : 101−111. doi :10.1287/opre.8.1.101, 1960.

[91] L. Gao, and E.J. Powell Robinson. A dual-based optimization procedure for the two-echelon uncapacitated facility location problem. Naval Research Logistics, 39(2) :191−212, 1992.

[92] H. Gunnarsson, M. Ronnqvist, D. Carlsson, A combined location and ship routing problem, Journal of the Operational Research Society, Vo.57, 928-938, 2006

[93] M. Gendreau, A. Hertz and G. Laporte. A tabu search heuristic for the vehicle routing problem. Management Science, 40 :1276−1290, 1994.

[94] M. Gendreau, G. Laporte, and J.Y. Potvin. The vehicle routing problem, chapter Metaheuristics for the vehicle routing problem, pages 129−154. Society for Industrial and Applied Mathematics, 2001.

[95] M. Gendreau, G. Laporte, and R. Seguin. Stochastic vehicle routing. European Journal of Operational Research, 88 :3−12, 1996.

[96] B. Gendron and F. Semet. Formulations and relaxations of a multi-echelon capacitated location-distribution problem. Computers & Operations Research, 36 :1335−1355, 2009.

[97] B.L. Golden, A. Assad, L. Levy and F. Gheysens. The fleet size and mix routing problem. Management Science and Statistics. Working paper, University of Maryland at college park, N 82 020, 1982.

[98] B.E. Gillett and L.R. Miller. A heuristic algorithm for the vehicle dispatch problem. Operations Research, 22 :340−349, 1974.

[99] M. Gendreau, G. Laporte, D. Vigo. Heuristics for the traveling salesman problem with pickup and delivery. Computers & Operations Research. 26 :699−714, 1999.

[100] G. Gutierrez-Jarpa, G. Desaulniers, G. Laporte and V. Marianov, A Branch-and-Price Algorith Routing Problem with Deliveries, Selective Pickups and Time Windows, European Journal of Operational Research, Vol. 206, No. 12, pp. 341-349. doi :10.1016/j.ejor.2010.02.037, 2010.

[101] B.L. Golden, S. Raghavan, and E.A. Wasil, editors. The Vehicle Routing Problem : Latest Advances and New Challenges (Operations Research/Computer Science Interfaces Series). Springer, New York, USA, 2008.

[102] J. Gonzalez-Feliu. Models and methods for the city logistics. The two-echelon capacitated vehicle routing problem. PhD thesis, Politecnico di Torino, Italy, 2008.

[103] J. Gonzalez-Feliu. The multi-echelon location-routing problem : Concepts and methods for tactical and operational planning. Technical report, Laboratoire d'Economie des Transports, Lyon, France, 2010.

[104] L. Gouveia, P. Pesneau, On extended formulations for the precedence constrained asymmetric traveling salesman problem, Networks 48, 77−89, 2006.

[105] G. Guastaroba, M.G. Speranza, D. Vigo, Designing service networks with intermediate facilities : an overview, under review ; 2014.

[106] J. Gonzalez Feliu. Two-echelon freight transport optimization : unifying concepts via a systematic review. Working papers on operations management, 2 :18−30, 2011.

[107] K. Govindan, A. Jafarian, R. Khodaverdi, K. Devika. Two-echelon multiple-vehicle location−routing problem with time windows for optimization of sustainable supply chain network of perishable food, International Journal of Production Economics, Volume 152, Pages 9-28, 2014.

[108] W.J.Guerrero, and C. Prodhon, Capacitated hierarchical clustering heuristic for multi depot location-routing problems. International Journal of Logistics Research and Applications : A Leading Journal of Supply Chain Management, 16(5), pp. 433-444, 2013.

[109] V.C. Hemmelmayr, J.F Cordeau, T. G. Crainic. An adaptive large neighborhood search heuristic for Two−Echelon Vehicle Routing Problems arising in city logistics, Computers & Operations Research, Volume 39, Issue 12, December , Pages 3215-3228, 2012.

[110] R. W.Hall. Vehicle Routing Software Survey. Retrieved August 16, 2013, from http ://www.orms-today.org/surveys/Vehicle-Routing/vrss.html. 2012.

[111] P. Hansen and N. Mladenovic. Variable neighborhood search : Principles and applications. European Journal of Operational Research, 130 :449−467, 2001.

[112] E. Hadjiconstantinou, N. Christofides, and A. Mingozzi. A new exact algorithm for the vehicle routing problem based on q-paths and k-shortest paths relaxations. Annals of Operations Research, 61 :21-43, 1995.

[113] W. Haijun, D. Lijing, M. Shihua. Multi-objective open location-routing model with split delivery for optimized relief distribution in post-earthquake, Transportation Research Part E : Logistics and Transportation Review, Volume 69, Pages 160-179, 2014.

[114] S. Hakimi. Optimum location of switching centers and the absolute centers and medians of a graph. Operations Research, 12 :450−459, 1964.

[115] P. Healy, R. Moll, A new extension of local search applied to the dial-a-ride problem, European Journal of Operational Research 83, 83−104, 1995.

[116] H. Hernandez-Perez, JJ. Salazar-Gonzalez. The one-commodity pickup-and-delivery travelling salesman problem. In : M. Junger, G. Reinelt, G. Rinaldi (eds.) Combinatorial Optimization Eureka,You Shrink Springer, LNCS, vol. 2570, 89−104, 2003.

[117] H.Hernandez-Perez, and J.-J. Salazar-Gonzalez,. The multi-commodity one-to- one pickup-and-delivery traveling salesman problem. European Journal of Operational Research, 196, pp. 987-995, 2009.

[118] H. Hernandez-Perez, JJ. Salazar-Gonzalez. A branch-and cut algorithm for the traveling salesman problem with pickup and delivery. Discrete Appl Math 145 :126−139, 2004.

[119] M. Hamidi, K. Farahmand, S. Reza Sajjadi, K. Nygard. A heuristic algorithm for a multi-product four-layer capacitated location-routing problem. International Journal of Industrial Engineering Computations, 5(1), 87−100, 2014.

[120] S.H. Huang. Solving the multi-compartment capacitated location routing problem with pickup and delivery routes and stochastic demands. Computers & Industrial Engineering, Volume 87, Pages 104-113, 2015.

[121] M. Hamidi, K. Farahmand, S. Reza Sajjadi. Modeling a four-layer location-routing problem. International Journal of Industrial Engineering Computations, 3, 43−52, 2012.

[122] J. Holland. Adaptation in natural and artificial systems. Tech. rep., The University of Michigan Press, Ann Arbor, MI, 1975.

[123] K. Holmberg, M. Ronnqvist, et D.Yuan. An exact algorithm for the capacitated facility location problems with single sourcing. European Journal of Operational Research, 113 , 544559, 1999.

[124] D, Infante, G. Paletta, F.Vocaturo, A ship-truck intermodal transportation problem. Marit Econ Logist,11 :247−59, 2009.

[125] T.Ibaraki, S. Imahori, K. Nonobe, K.Sobue, T.Uno, M.Yagiura. An iterated local search algorithm for the vehicle routing problem with convex time penalty functions. Discrete Applied Mathematics Journal. Pp.2050−2069, 2008.

[126] W.C. Jordan and L.D. Burns. Truck backhauling on two terminal networks. Transportation Research, 18B :487−503, 1984.

[127] S.Jacobsen, O. Madsen, A comparative study of heuristics for a two-level routing-location problem. European Journal of Operational Research, 6, 378−387, 1980.

[128] E.L. Johnson. Modeling and strong linear programs for mixed integer programming. S.W.Wallace (ed). Algorithms and Model Formulations in Mathematical Programming, Springer-Verlag, 1−43, 1989.

[129] J.G. Klincewicz and H. Luss. A lagrangian relaxation heuristic for capacitated facility location with single-source constraints. Journal of the Operational Research Society, 37(5) :495500, 1986.

[130] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. Science, 220(4598) :671−680, 1983.

[131] L. Kaufman, M.V. Ecde, and P. Hansen. A plant and warehouse location problem. Operational Research Quarterly, 28 :547−554, 1977.

[132] B. Kalantari, AV. Hill, SR. Arora. An algorithm for the traveling salesman 0problem with pickup and delivery customers. European Journal of Operational Research. 22 :377−386, 1985.

[133] V. Kachitvichyanukul, P. Sombuntham, S. Kunnapapdeelert. Two solution representations for solving multi-depot vehicle routing problem with multiple pickup and delivery requests via PSO. Computers & Industrial Engineering, In Press, Corrected Proof, 2015.

[134] J.G. Klincewicz and H. Luss. A lagrangian relaxation heuristic for capacitated facility location with single source constraints. Journal of the Operational Research Society, 37 :495−500, 1986.

[135] I.Karaoglan, F. Altiparmak, I. Kara, B. Dengiz. A branch and cut algorithm for

the location-routing problem with simultaneous pickup and delivery. European Journal of Operational Research, 211, 318−332, 2011.

[136] C. Koc, T. Bekta, O. Jabali, The fleet size and mix location-routing problem with time windows : Formulations and a heuristic algorithm. European Journal of Operational Research, Volume 248, Issue 1, Pages 33-51, 2016.

[137] I.Karaoglan, F. Altiparmak, A memetic algorithm for the capacitated location-routing problem with mixed backhauls. Computers & Operations Research, 55, pp. 200216, 2015.

[138] I.Karaoglan, F. Altiparmak, I. Kara, B. Dengiz, The location-routing problem with simultaneous pickup and delivery : Formulations and a heuristic approach, OMEGA - The International Journal of Management Science, 2012.

[139] A. Klose and A. Drexl. Facility location models for distribution system design. European Journal Of Operational Research, 162 :4−29, 2005.

[140] G. Laporte, Y. Nobert, An exact algorithm for minimizing routing and operating costs in depot location. European Journal of Operational Research 6, 224-226, 1981.

[141] G. Laporte, Y. Nobert, D. Arpin, An exact algorithm for solving a capacitated location-routing problem. Annals of Operations Research 6, 293-310, 1986.

[142] J.Li, C. Prins, and F. Chu, A scatter search for a multi-type transshipment point location problem with multi-commodity flow. Journal of Intelligent Manufacturing. 23(4), pp. 1103-1117, 2012.

[143] J.-R.Lin, and H.-C. Lei,. Distribution systems design with two-level routing considerations. Annals of Operations Research, 172, pp. 329-347, 2009.

[144] R. B.Lopesa, , C. Ferreira, B.S. Beatriz Sousa Santo, and S. Barreto,A taxonomical analysis, current methods and objectives on location routing problems. International Transactions in Operational Research 20(6), pp. 795-822, 2013.

[145] G. Laporte, M. Desrochers, and Y. Norbert. Two exact algorithms for the distance-constrained vehicle routing problem. Networks, 14 :161−172, 1984.

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

[147] G. Laporte, F. Louveaux, and H. Mercure. Models and exact solutions for a class of stochastic location-routing problems. European Journal of Operational Research, 39 :71−78, 1989.

[148] H. R. Lourenco, O. C. Martin, and T. Stutzle. Iterated local search. In Fred Glover and Gary A. Kochenberger, editors, Handbook of metaheuristics, pages 320353. Springer, 2003.

[149] G. Laporte and Y. Norbert. Exact algorithms for the vehicle routing problem. Annals of Discrete Mathematics, 31 :147−184, 1987.

[150] G. Laporte, Location-routing problems. In : B.L.Golden, A.A. Assad, (Eds.), Vehicle Routing : Methods and Studies. North-Holland, Amsterdam, pp. 163−198, 1988.

[151] G.Laporte, Fifty years of vehicle routing. Transportation Science, 43 :408−16, 2009.

[152] S. Lin. Computer solutions of the traveling salesman problem. AT&T Tech J 44 :2245−2269, 1965.

[153] C. K. Y. Lin, R. C. W. 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, 175(3), 1833−1849, 2005.

[154] C.Lin, 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, 175, 1833−1849, 2006.

[155] J. Lysgaard. A CVRPSEP package. http ://www.hha.dk/ lys/CVRPSEP.htm.

[156] J. Li, P.M. Pardalos, H. Sun, J. Pei, Y. Zhang. Iterated local search embedded adaptive neighborhood selection approach for the multi-depot vehicle routing problem with simultaneous deliveries and pickups. Expert Systems with Applications, Volume 42, Issue 7, Pages 3551-3561, 2015.

[157] S. Lin and B.W. Kernighan. An effective heuristic algorithm for the TSP. Operational Research, 21 :498−516, 1973.

[158] L. F. Lopez-Castro, J. R. Montoya-Torres. Vehicle routing problem with fuzzy time windows. In Proceedings of the 2011 IEEE workshop on computational intelligence in production and logistics systems (pp. 39−46). IEEE Publishing, 2011.

[159] J.Li, Y. Li, P. M. Pardalos, (in press). Multi-depot vehicle routing problem with time windows under shared depot resources. Journal of Combinatorial Optimization. doi : http ://dx.doi.org/10.1007/s10878-014-9767-4.

[160] A. Lipowski, Roulette-wheel selection via stochastic acceptance (arXiv :1109.3627).

[161] P. Lacomme, C. Prins, M. Sevaux. Algorithmes de graphes. Paris : Eyrolles, 2003.

[162] A. H. Land, and A. G. Doig. An Automatic Method of Solving Discrete Programming. Problems. Econometrica, 28, pages 497- 520, 1960.

[163] W. Meihua, T. Xuhong, C. shan, W. Shumi. Hybrid Ant Colony Optimization Algorithm for Two Echelon Vehicle Routing Problem. Procedia Engineering, Volume 15, Pages 3361-3365, 2011.

[164] D.L. Miller, A matching based exact algorithm for capacitated vehicle routing problems. ORSA Journal on Comupting, 7, pages1−9, 1995.

[165] M. Mokhtarinejad, A. Ahmadi, B. Karimi, S. H. A. Rahmati, A novel learning based approach for a new integrated location-routing and scheduling problem within cross-docking considering direct shipment,Applied Soft Computing, Volume 34, Pages 274-285, 2015.

[166] A. Mingozzi and A. Valletta. An exact algorithm for period and multi-depot vehicle routing problems. In Odysseus 2003 - Second International Workshop on Freight Transportation and Logistics, Palermo, Italy, Mai 2003.

[167] , T. Magnanti, R.Wong. Accelerating Benders decomposition : Algorithmic enhancement and model selection criteria. Operations Research, 29, 464−484, 1981.

[168] J. Mageean, JD. Nelson, The evaluation of demand responsive transport services in Europe. J Transport Geogr 11 :255−270, 2003.

[169] A. C.Matos, R.C. Oliveira. An experimental study of the ant colony system for the period vehicle routing problem. Ant colony optimization and swarm intelligence. In M. Dorigo, M. Birattari, C. Blum, L.M. Gambardella, F. Mondada, T. Stutzle (Eds.), Lecture Notes in Computer Science, Vol. 3172, pp. 1−29, 2004.

[170] O.Madsen, Methods for solving combined two level location-routing problems of realistic dimensions. European Journal of Operational Research, 12, pp. 295−301, 1983.

[171] G. Mosheiov. The traveling salesman problem with pickup and delivery. European Journal of Operational Research. 79 :299−310, 1994.

[172] Y. Marinakis. An improved particle swarm optimization algorithm for the capacitated location routing problem and for the location routing problem with stochastic demands.Applied Soft Computing, Volume 37, Pages 680-701, 2015.

[173] H.Min, V. Jayaraman, and R. Srivastava, Combined location-routing problems : A synthesis and future research directions. European Journal of Operational Research 108(1), pp. 1-15, 1998.

[174] C.E.Miller, A.W. Tucker, and R.A. Zemlin, Integer programming formulation of traveling salesman problems, Journal of ACM, Vol. 7, pp.3269, 1960.

[175] N.H. Moin, S. Salhi, and N.A.B. Aziz, An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem. International Journal of Production Economics, 133(1), pp. 334-343, 2011.

[176] G. Mosheiov, The pickup delivery location problem on networks. Networks 26, pp. 243-251, 1995.

[177] G.Nagy, S. Salhi, The many-to-many location routing problem. TOP. 6, pp. 261-275, 1998.

[178] V.-P.Nguyen, C.Prins, and C. Prodhon, A multi-start iterated local search with tabu list and path relinking for the two-echelon location-routing problem. Engineering Applications of Artificial Intelligence, 25(1), pp. 56-71, 2012.

[179] G.Nagy, S.Salhi, Location-routing : Issues, models and methods. European Journal of Operational Research 177 (2), pp. 649-672, 2007.

[180] V.-P.Nguyen, C. Prins, and C. Prodhon, Solving the Two-Echelon Location Routing Problem by a hybrid GRASPxPath Relinking complemented by a learning process. European Journal of Operational Research, 216, pp. 113-126, 2012.

[181] G. Nagy and S. Salhi. Location-routing : Issues, models and methods. European Journal of Operational Research, 177 :649−672, 2007.

[182] V.-P. Nguyen, C. Prins, C. Prodhon. A multi-start Iterative Local Search with tabu list and Path Relinking for the Two-Echelon Location Routing Problem. Engineering Applications of Artificial Intelligence, 25(1), pp. 56-71, 2012.

[183] G.L. Nemhauser and L.A. Wolsey. Integer and Combinatorial Optimization. Wiley, New York, 1999.

[184] I.H. Osman. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Annals of Operations Research, 41 :421−451, 1993.

[185] L.Ozdamar, and O. Demir, A hierarchical clustering and routing procedure for large scale disaster relief logistics planning. Transportation Research Part E : Logistics and Transportation Review, 48(3), pp 591-602, 2012.

[186] Z. Ozyurt, D. Aksen. Solving the multi-depot location routing problem with Lagrangian relaxation. Operations Research/Computer Science Interfaces Series, 37, 125−144, 2007.

[187] I. Or. Traveling salesman-type combinatorial optimization problems and their relation to the logistics of regional blood banking. Ph.D. dissertation, Northwestern University, Evanston, IL, 1976.

[188] S.N.Parragh, K.F. Doerner, and R.F. Hartl, A survey on pickup and delivery problems Part II : Transportation between pickup and delivery locations. Journal fur Betriebswirtschaft 58(2), pp. 81-117, 2008.

[189] H. Paessens. The savings algorithm for the vehicle routing problem. European Journal of Operational Research, 34 :336−344, 1988.

[190] S.N. Parragh, K.F. Doerner, R.F. Hartl, A survey on pickup and delivery problems. Part I : Transportation between customers and depot. Journal fur Betriebswirtschaft 58(1), 21−51, 2008.

[191] G. Perboli, R. Tadei. New Families of Valid Inequalities for the Two-Echelon Vehicle Routing Problem, Electronic Notes in Discrete Mathematics, Volume 36, Pages 639-646, 2010.

[192] C. Prodhon, and C. Prins, A survey of recent research on location-routing problem. European journal of Operational Research, 238(1), pp. 1-17, 2014.

[193] H.N. Psaraftis. A multi-commodity, capacitated pickup and delivery problem : The single and two-vehicle cases. European Journal of Operational Research, Volume 215, Issue 3, Pages 572-580, 16 December 2011.

[194] C. Prins. Efficient heuristics for the heterogeneous fleet multitrip vrp with application to a large-scale real case. Journal of Mathematical Modelling and Algorithms, 1 :135−150, 2002.

[195] M. Padberg, and G. Rinaldi. Optimization of a 532 city symmetric traveling salesman problemby branch-and-cut. Operations Research Letters, pages 1−7, 1987.

[196] C. Prins, C. Prodhon, A. Ruiz, P. Soriano, and R. Wolfer-Calvo. Solving the capacitated location-routing problem by a cooperative lagrangean relaxation granular tabu search heuristic. Transportation Science, 41 :470−483, 2007.

[197] C. Prins, C. Prodhon, and R. Wolfer-Calvo. A memetic algorithm with population management (MA | PM) for the capacitated location-routing problem. In J.Gottlieb and G. R. Raidl, editors, Lecture Notes in Computer Science, volume 3906, pages 183−194. Proceedings of EvoCOP2006 (Evolutionary Com-

putation in Combinatorial Optimization : 6th European Conference, Budapest, Hungary, April 10-12, 2006), Springer, 2006.

[198] C. Prins, C. Prodhon, R. Wolfler Calvo, Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. 4OR 4,221-238, 2006

[199] C. Prodhon, 2008. A metaheuristic for the periodic location-routing problem. In J. Kalcsics, S. Nickel (Eds.), Operations Research Proceedings, pp. 159−164, Springer, 2007.

[200] G .Perboli, R,Tadei, D. Vigo The two-echelon capacitated vehicle routing problem : models and math-based heuristics. Transportation Science, 45 :364−80, 2011.

[201] C. Prodhon. Le probleme de localisation-routage. PhD thesis, Troyes University of Technology, France, 2006.

[202] S.Ray, A.Soeanu, J. Berger, M. Debbabi, (in press). The multi-depot split-delivery vehicle routing problem : Model and solution algorithm. Knowledge-based Systems. doi : http ://dx.doi.org/10.1016/j.knosys.2014.08.006.

[203] S. Ropke, D. Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Submitted to Transportation Science, 2004.

[204] J. Reese, Solution methods for the p-median problem : an annotated bibliography, Networks 48(3) :125-142, 2006.

[205] T. K. Ralphs . Parallel branch and cut for capacitated vehicle routing. Parallel Computing archive, 29 (5), pages 607−629, 2003.

[206] Y. Rahmani, A. Oulamara, and W.Ramdane Cherif, Multi-Products Location Routing problem with Pickup and Delivery : Two-Echelon model. The test instances are accessible on website : http ://www.loria.fr/∼yrahmani.

[207] J. Renaud, G. Laporte, and F.F. Boctor. A tabu search heuristic for the multi-depot vehicle routing problem. Computers & Operations Research, 23(3) :229−235, 1996.

[208] C. Rego, and C. Roucairol. Le Probléme de Tournées de Véhicules : Etude et Résolution Approchée. Technical Report, inria, Février, 1994.

[209] W. Ramdane Cherif-Khettaf, M. Haj Rachid, C. Bloch and P. Chatonnay, New notation and classification scheme for vehicle routing problem, RAIRO-Oper. Res. 49 161−194, 2015.

[210] I.Rodriguez-Martin, J.-J Salazar-Gonzalez, H. Yaman, A branch-and-cut algorithm for the hub location and routing problem. Computers & Operations Research, 50, 161−174, 2014.

[211] J. Rieck, C. Ehrenberg, J. Zimmermann. Many-to-many location-routing with inter-hub transport and multi-commodity pickup-and-delivery. European Journal of Operational Research, 236, 863−878, 2014.

[212] I. Rodriguez-Martin, J.-J. Salazar-Gonzoalez, and H. Yaman. A branch-and-cut algorithm for the hub location and routing problem. Computers & Operations Research, 50 :161−174, 2014.

[213] Y.Rahmani, A. Oulamara, and W. Ramdane Cherif, MultiProducts Location-Routing problem with Pickup and Delivery : Two-Echelon model.11th IFAC Workshop on Intelligent Manufacturing - IMS Sao Paulo, Brazil, pp. 124-129, 2013. ISBN : 978-3-902823-33-5, 2013.

[214] Y.Rahmani, A. Oulamara, and W. Ramdane Cherif, Multi-products Location-Routing problem with Pickup and Delivery, ICALT IEEE, Sousse, Tunisie, pp. 115-1, 2012.

[215] I.Rodriguez-Martin, and J.-J .Salazar-Gonzalez, The multi-commodity one-to-one pickup-and-delivery traveling salesman problem : A matheuristic. 5th International Conference INOC 2011. Lecture Notes in Computer Science, 6701, pp. 401-405, 2011.

[216] I.Rodriguez-Martin, J.Salazar-Gonzlez, and H. Yaman, A branch-and-cut algorithm for the hub location and routing problem. Computers & Operations Research, 50, pp. 161-174, 2014.

[217] H. Ro and D. Tcha. A branch and bound algorithm for the two-echelon uncapacited facility location problem with some side constraints. European Journal of Opearational Research, 18 :349−358, 1984.

[218] D. M. Ryan, C. Hjorring, and F. Glover. Extensions of the petal method for vehicle routeing. Journal of the Operational Research Society, 44(3) :289−296, 1993.

[219] D. Sariklis and S. Powell. A heuristic method for the open vehicle routing p0roblem. Journal of Operational Research Society, 51 :564−573, 2000.

[220] M.M. Solomon. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. Operations Research, 35, pages 254-265, 1987.

[221] S. Salhi, A. Imran, N.A. Wassan The multi-depot vehicle routing problem with heterogeneous vehicle fleet : Formulation and a variable neighborhood

search implementation. Computers & Operations Research, Volume 52, Part B, , Pages 315-325, 2014.

[222] P. Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In CP-98 (Fourth International Conference on Principles and Practice of Constraint Programming), volume 1520 of Lecture Notes in Computer Science, pages 417431, 1998.

[223] S.Salhi, S. R.Thangiah, F. Rahman. A genetic clustering method of the multi-depot vehicle routing problem. In G. D. Smith, N. C. Steel (Eds.), Artificial neural network and genetic algorithms (pp. 234−237). NY : Springer, 1998.

[224] G. Schrimpf, J. Schneider, H. Stamm-Wilbrandt, and G. Dueck. Record breaking optimization results using the ruin and recreate principle. Journal of Computational Physics,159(2) :139171, 2000.

[225] R. Sridharan. A lagrangian heuristic for the capacitated plant location problem with single source constraints. European Journal of Operational Research, 66 , 305312, 1993.

[226] M. Sniedovich. Dynamic Programming : Foundations and Principles, Taylor & Francis, ISBN 978−0−8247−4099−3, 2010.

[227] D.Skorin-Kapov, J. Skorin-Kapov, M. O−Kelly. Tight linear programming relaxations of uncapacitated p-hub median problems. European Journal of Operational Research, 94, 582−593, 1996.

[228] M. Soysal, J. M. Bloemhof-Ruwaard, T. Bekta. The time-dependent two-echelon capacitated vehicle routing problem with environmental considerations. International Journal of Production Economics, Volume 164, Pages 366-378, 2015.

[229] M.W.P. Savelsbergh. A branch-and-price algorithm for the generalized assignment problem.Operations Research 45, 831−841, 1997.

[230] F. Semet and E. Taillard. Solving real-life vehicle routing problems efficiently using tabu search. Annals of Operations Research, 41 :469−488, 1993.

[231] M.Schwengerer, S. Pirkwieser, and G. R. Raidl. A variable neighborhood search approach for the two-echelon location-routing problem. In J.-K. Hao and M. Middendorf, editors, Evolutionary Computation in Combinatorial Optimisation - EvoCOP 2012. Lecture Notes in computer Science, Vol. 7245, pp 13-24, 2012.

[232] S. Sajjadi, S. Cheraghi, Multi-products location routing problem integrated

with inventory under stochastic demand. International Journal of Industrial and Systems Engineering, 2011.

[233] T. Stutzle. Iterated local search for the quadratic assignment problem. European Journal of Operational Research, 174(3) :15191539, 2006.

[234] F. J. C. Sevilla and C. S. Blas. Vehicle routing problem with time windows and intermediate facilities. In 27 Congreso Nacional de Estadistica e Investigacion Operativa. Edicions de la Universitat de Lleida, 2003.

[235] P.Schittekat, K. Sorensen. OR practice Supporting 3PL decisions in the automotive industry by generating diverse solutions to a large-scale location-routing problem. Operations Research, 57, 1058−1067, 2009.

[236] S. Scheuerer, A tabu search heuristic for the truck and trailer routing problem. Computers and Operations Research ;33 :894–909, 2006.

[237] K. C.Tan, Y. H. Chew, L. H. Lee. A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems. European Journal of Operational Research, 172(3), 855−885, 2006.

[238] D. Tas, N. Dellaert, T. van Woensel, T. de Kok. Vehicle routing problem with stochastic travel times including soft time windows and service costs. Computers and Operations Research, 40(1), 214−224, 2013.

[239] E.D. Taillard. Parallel iterative search methods for vehicle routing problem. Networks, 23 :661−673, 1993.

[240] P. Toth, and D. Vigo. The vehicle routing problem. SIAM Monographs on Discrete Mathematics and Applications, pages 129−154, 2002.

[241] R.Tavakkoli-Moghaddam, A. Makui, Z. Mazloomi. A new integrated mathematical model for a bi-objective multi-depot location-routing problem solved by a multi-objective scatter search algorithm. Journal of Manufacturing Systems, 29(3−4), 111−119, 2010.

[242] E. Taniguchi and R.G. Thompson. Modelling city logistics. Transportation Research Record, 1790 :45−51, 2002.

[243] C.J.Ting, C.H. Chen. A multiple ant colony optimization algorithm for the capacitated location routing problem. International Journal of Production Economics, Volume 141, Issue 1, Pages 34-44, 2013.

[244] A.S. Tasan, M. Gen. A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries. Computers & Industrial Engineering, Volume 62, Issue 3, Pages 755-761, 2012.

[245] P. Toth and D. Vigo, editors. The vehicle routing problem. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001.

[246] P. Toth and D. Vigo. The granular tabu search (and its application to the vehicle routing problems. INFORMS Journal on Computing, 15(4) :333−346, 2003.

[247] P. Toth, and D. Vigo. Models, relaxations and exact approaches for the capacitated vehicle routing problem. Discrete Applied Mathematics, 123, 487-512, 2002.

[248] S. Tragantalerngsak, J. Holt, and M. Ronnqvist. Lagrangian heuristics for the two-echelon, single-source, capacitated facility location problem. European Journal of Operational Research, 102 :611−625, 2007.

[249] T. Vidal, T. Crainic, M. Gendreau, C. Prins. A unified solution framework for multi-attribute vehicle routing problems. European Journal of Operational Research. http ://dx.doi.org/10.1016/j.ejor.2013.09.045, 2013.

[250] J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia, and N. Velasco. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. Engineering Applications of Artifical Intelligence, 23(5) :780−794, 2010.

[251] J.G.Villegas, C. Prins, C. Prodhon, A.L.Medaglia, and N. Velasco, N. GRASPVND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots, Engineering Applications of Artificial Intelligence. 23, pp. 780-794, 2011.

[252] C. Voudouris and E. Tsang. Guided local search and its application to the traveling salesman problem. European Journal of Operational Research, 113(2) :469−499, 1999.

[253] P. Vansteenwegen, W. Souriau, G. Vanden Berghe, and D. Van Oudheusden. Iterated local search for the team orienteering problem with time windows. Computers and Operations Research, 36(12) :3281-3290, 2009.

[254] J. Willmer Escobar, R. Linfati, P. Toth. A two-phase hybrid heuristic algorithm for the capacitated location-routing problem. Computers and Operations Research, Volume 40, Issue 1, Pages 70-79, 2013.

[255] Jr. William A. Dees and Patrick G. Karger.Automated rip-up and reroute techniques. In Proceedings of the 19th conference on Design automation, pages 432-439, 1982.

[256] C. Wu, D. Du, D. Xu. Primal-dual approximation algorithm for the two-level facility location problem via a dual quasi-greedy approach. Theoretical Computer Science, Volume 562, Pages 213-226, 2015.

[257] C. Wang, D. Mu, F. Zhao, J.W. Sutherland. A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup and delivery and time windows. Computers and Industrial Engineering, Volume 83, Pages 111-122, May 2015.

[258] T.H. Wu, C. Low, and J.W. Bai. Heuristic solutions to multi-depot location routing problems. Computers and Operations Research, 38 :1393−1415, 2002.

[259] M. H.J.Webb, Cost functions in the location of depots for multiple delivery journeys. Operational Research Quarterly 19 (3), 311−320, 1968.

[260] T. H.Wu, C. Low, J. W. Bai. Heuristic solutions to multi-depot location routing problems. Computers and Operations Research, 29(10), 1393−1415, 2002.

[261] M.Wasner, G. Zapfel. An integrated multi-depot hub-location vehicle routing model for network planning of parcel service. International Journal of Production Economics, 90(3), 403−419, 2004.

[262] J.Willmer Escobar, R. Linfati, G. M. Baldoquin, and P. Toth, 2014. A Granular Variable Tabu Neighborhood Search for the capacitated location-routing problem. Transportation Research Part B : Methodological, 67, pp. 344-356, 2014.

[263] V.F.Yu, and Lin, S-W. Multi-start Simulated Annealing Heuristic for the Location Routing Problem with Simultaneous Pickup and Delivery. Applied Soft Computing, 24, pp. 284-290, 2014.

[264] W. Yi , L. Ozdamar, A daynamic logistic coordination model for evacuation and support in disaster response activities, European Journal of Operational Research, Vol.179, pp.1177-1193, 2007.

[265] V. F. Yu, S.W Lin. A simulated annealing heuristic for the open location-routing problem. Computers and Operations Research, Volume 62, Pages 184-196, 2015.

[266] J. Yang, H. Sun. Battery swap station location-routing problem with capacitated electric vehicles. Computers and Operations Research, Volume 55, Pages 217-232, 2015.

[267] P. Yellow. A computational modification to the savings method of vehicle scheduling. Operational Research Quarterly, 21 :281−283, 1970.

[268] J. Yellen, J. Gross. Graph theory and its applications. CRC Press. 1998. Physics,159(2) :139171, 2000.

[269] K.Q. Zhu. Heuristic Methods For Vehicle Routing Problem with Time Windows. PhD thesis, National University of Singapore, Departement of Electrical Engineering, May, 1999.

[270] M. H. F.Zarandi, A. Hemmati, S. Davari. The multi-depot capacitated location-routing problem with fuzzy travel times. Expert Systems with Applications, 38(8), 10075−10084, 2011.

[271] Y. Zhang, M. Qi, W.H. Lin, L. Miao. A metaheuristic approach to the reliable location routing problem under disruptions . Transportation Research Part E : Logistics and Transportation Review, Volume 83, Pages 90-110, 2015.

[272] B. Zhang, Z.Ma, S.Jiang. Location-routing-inventory problem with stochastic demand in logistics distribution systems. In 4th International conference on wireless communications, networking and mobile computing (WiCOM-08), pp. 1−4, 2008.

[273] T. Zhang, W.A. Chaovalitwongse, Y. Zhang. Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries. Computers & Operations Research, Volume 39, Issue 10, Pages 2277-2290, 2012.

[274] E.E. Zachariadis, C.T. Kiranoudis. A local search metaheuristic algorithm for the vehicle routing problem with simultaneous pick-ups and deliveries, Expert Systems with Applications, Volume 38, Issue 3, Pages 2717-2726, 2011.

[275] Y. Zhang, X.D. Chen. An Optimization Model for the Vehicle Routing Problem in Multi-product Frozen Food Delivery. Journal of Applied Research and Technology, Volume 12, Issue 2, Pages 239-250, 2014.

[276] Y.Zare Mehrjerdi, and A. Nadizadeh, Using greedy clustering method to solve capacitated location-routing problem with fuzzy demands. European Journal of Operational Research. 229(1), pp. 7584, 2013.

.

.

.

.

Résumé

Dans les problèmes de localisation-routage classiques (LRP), il s'agit de combiner des décisions stratégiques liées aux choix des sites à ouvrir (centres de traitement) avec des décisions tactiques et opérationnelles liées à l'affectation des clients aux sites sélectionnés et à la confection des tournées associées. Cette thèse propose de nouveaux modèles de localisation-routage permettant de résoudre des problématiques issues de réseaux logistiques, devenus aujourd'hui de plus en plus complexes vu la nécessité de mutualisation de ressources pour intégrer des contraintes de développement durable et des prix de carburants qui semblent augmenter de manière irrémédiable. Plus précisément, trois aspects ont été intégrés pour généraliser les modèles LRP classiques de la littérature : 1) l'aspect pickup and delivery, 2) l'aspect multi-produits, et 3) la possibilité de visiter un ou plusieurs centres de traitement dans une tournée donnée. Nous avons étudié deux schémas logistiques, qui ont donné lieu à deux nouveaux modèles de localisation et de routage, le MPLRP-PD (LRP with multi-product and pickup and delivery), qui peut être vu comme une extension des problèmes de tournées de véhicules avec collecte et livraison, intégrant une décision tactique liée à la localisation des centres de traitement (noeud avec collecte et livraison) dans un réseau de distribution à un seul échelon, et le 2E-MPLRP-PD (Two-echelon LRP with multi-product and pickup and delivery) qui est une généralisation du LRP à deux échelons avec les contraintes citées plus-haut. Ces deux modèles ont été formalisés par des programmes linéaires en variables mixtes (MIP). Des techniques de résolution, basées sur des méthodes de type heuristique, clustering, métaheuristique, ont été proposées pour résoudre le MPLRP-PD et le 2E-MPLRP-PD. Les jeux d'essais de la littérature ont été généralisés pour tester et valider les algorithmes proposés.

Mots-clés: transport, localisation routage, tournées de véhicules, optimisation combinatoire, heuristiques, technique de clustering, métaheuristiques.

Abstract

In the framework of Location-Routing Problem (LRP), the main idea is to combine strategic decisions related to the choice of processing centers with tactical and operational decisions related to the allocation of customers to selected processing centers and computing the associated routes. This thesis proposes a new location-routing model to solve problems which are coming from logistics networks, that became nowadays increas-

ingly complex due to the need of resources sharing, in order to integrate the constraints of sustainable development and fuels price, which is increasing irreversibly.
More precisely, three aspects have been integrated to generalize the classical LRP models already existed in the literature : 1) pickup and delivery aspect, 2) multi-product aspect, and 3) the possibility to use the processing centers as intermediate facilities in routes. We studied two logistics schemes gives us two new location-routing models : (i) MPLRP-PD (Multi-product LRP with pickup and delivery), which can be viewed as an extension of the vehicle routing problem with pick-up and delivery, including a tactical decision related to the location of processing centers (node with pick-up and delivery), and (ii) 2E-MPLRP-PD (Two-echelon multi-product LRP with pickup and delivery), which is a generalization of the two-echelon LRP. Both models were formalized by mixed integer linear programming (MIP). Solving techniques, based on heuristic methods, clustering approach and meta-heuristic techniques have been proposed to solve the MPLRP-PD and the 2E-MPLRP-PD. The benchmarks from the literature were generalized to test and to validate the proposed algorithms.

Keywords: Transport, location routing, vehicle routing, combinatorial optimization, heurisitc, clustering technique, metaheuristic.