N° d'ordre: | 3 | 2 | 4 |

CENTRALE LILLE

THÈSE

présentée en vue d'obtenir le grade de

DOCTEUR

en

Spécialité : Automatique, Génie Informatique, Traitement du Signal et des Images

par

Yihan LIU

Master of Science of Beihang University (BUAA)

Doctorat délivré par Centrale Lille

Titre de la thèse :

Optimisation de Problème de Tournées de Véhicules de Service à Domicile

Soutenue le 27 Juin 2017 devant le jury d'examen :

M. Pierre BORNE	École Centrale de Lille	Président
M. Khaled MELLOULI	IHEC Carthage, Tunisie	Rapporteur
Mme. Shaoping WANG	Beihang University (BUAA)	Rapporteur
M. Abdelkader EL KAMEL	École Centrale de Lille	Directeur de thèse
M. Khaled MESGHOUNI	École Centrale de Lille	Examinateur
M. Dumitru POPESCU	Université Polytechnique de Bucarest	Examinateur
Mme. Zhuoyue SONG	Beijing Institute of Technology (BIT)	Examinateur
Mme. Liming ZHANG	University of Macau	Examinateur

Thèse préparée dans le Centre de Recherche en Informatique, Signal et Automatique de Lille (CRIStAL), UMR CNRS 9189 - École Centrale de Lille
École Doctorale Sciences pour l'Ingénieur - 072

Serial N° : | 3 | 2 | 4 |

CENTRALE LILLE

THESIS

Presented to obtain the degree of

Doctor of Philosophy

in

Topic : Automatic Control, Computer Science, Signal and Image Processing

by

Yihan LIU

Master of Engineering of Beihang University (BUAA)

Ph.D. awarded by Centrale Lille

Title of the thesis :

Optimization of Vehicle Routing Problem for Field Service

Defended on June 27, 2017 in presence of the committee :

Mr. Pierre BORNE	École Centrale de Lille	President
Mr. Khaled MELLOULI	IHEC Carthage, Tunisie	Reviewer
Mrs. Shaoping WANG	Beihang University (BUAA)	Reviewer
Mr. Abdelkader EL KAMEL	École Centrale de Lille	Supervisor
Mr. Khaled MESGHOUNI	École Centrale de Lille	Examiner
Mr. Dumitru POPESCU	Université Polytechnique de Bucarest	Examiner
Mrs. Zhuoyue SONG	Beijing Institute of Technology (BIT)	Examiner
Mrs. Liming ZHANG	University of Macau	Examiner

Thesis prepared within the Centre de Recherche en Informatique, Signal et Automatique de Lille (CRIStAL), UMR CNRS 9189 - École Centrale de Lille
École Doctorale Sciences pour l'Ingénieur - 072

To my parents,
to all my family,
to my professors,
and to my friends.

Acknowledgements

This research work has been realized in École Centrale de Lille, in "Centre de Recherche en Informatique, Signal et Automatique de Lille (CRIStAL)", with the research group "Optimisation : Modèles et Applications (OPTIMA)", from February 2014 to June 2017. This work is financially supported by China Scholarship Council (CSC).

First and foremost, I offer my sincerest gratitude to my supervisor, Prof. Abdelkader El Kamel, for his continuous support of my Ph.D. experience and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this dissertation. I could not have imagined having a better tutor and mentor for my Ph.D. study. He has provided his supervision, valuable guidance, continuous encouragement as well as given me extraordinary experiences through out the work.

Besides my supervisor, I would like to thank Prof. Pierre BORNE for his kind acceptance to be the president of my Ph.D. Committee, as well as Prof. Khaled MELLOULI and Prof. Shaoping WANG, who have kindly accepted the invitation to be reviewers of my Ph.D. thesis, for their encouragement, insightful comments and hard questions. My gratitude to Prof. Khaled MESGHOUNI, Prof. Dumitru POPESCU, Prof Zhuoyue SONG and Prof. Liming ZHANG, for being the examiner of my thesis and for their kind acceptance to take part in the jury of the Ph.D. defense.

I am also very grateful to the staff in École Centrale de Lille. Vanessa Fleury, Christine Yvoz, and Brigitte Foncez have helped me in the administrative work. Many thanks go also to Patrick Gallais, Gilles Marguerite, Jacques Lasue, for their kind help and hospitality. Special thanks go to Christine Vion, Martine Mouvaux for their support in my lodgment life at the dormitory "Léonard de Vinci".

ACKNOWLEDGEMENTS

I would like to take the opportunity to express my gratitude and to thank my fellow workmates in CRIStAL: Bing Liu,, Qi Sun, Jian Zhang, Ben Li for the stimulating discussions for the hard teamwork. Also I wish to thank my friends and colleagues: Daji Tian, Chen Xia, Hongchang Zhang, Yu Du, Lijie Bai, Qi Guo, Lijuan Zhang, etc., for all the fun we have had in the last three years. All of them have given me support and encouragement in my thesis work.

My Ph.D. program was supported by the cooperation between China and Scholarship Council (CSC) and Ecoles-Centrales Intergroup. So I would like to thank all of the people who have built this relationship and contributed to select candidates for interesting research projects. I am extremely grateful to CSC who financed me.

So as to the rest of my family for their love and support. Their encouragement and understanding during the course of my dissertation make me pursue the advanced academic degree.

Villeneuve d'Ascq, France *LIU*
2017

Abstract

The logistics performance of the enterprises and the optimization of transportation have become a great issue in recent years. Field force planning and optimization is a new challenge for the service sector especially for utility companies in the energy, telecommunications and water distribution areas. It generates new variations of combinatorial optimization problems in the fields of manpower scheduling and vehicle routing. The challenges are many: to increase productivity and reduce costs, by increasing the number of visited clients, while reducing the time and cost of transportation and to achieve an efficient internal organization and appropriate human resources planning.

In the literature, most of the work deals with problems involving deliveries of goods. In this thesis, we focus on the service tours, which constitute a less studied problem. We address the problem of the planning and routing of technician visits to customers in the field, for maintenance or service logistics activities undertaken by utilities. The plan must be designed over a multi-period horizon.

This dissertation focuses on the optimization of field service routing problem with meta-heuristics. The addressed problem is abstracted from the realistic problem. This problem can be assimilated to a multi-depot and multi-period routing problem with time window. Various constraints were taken into consider to simulate the real problem.

First, we consider the local search heuristics for solving the problem. Initial feasible solutions are obtained by a constructive heuristic. Several heuristics of local search are adapted to improve the solutions which permit us to obtain a feasible solution in a very short computing time. Second, we consider using genetic algorithm to find the near-optimum solution. The genetic algorithm is applied with new representation of chromosome and new genetic operators to adapt the force constraints of the real-world problem. Third, we consider a genetic algorithm with diversity control to deal with large scale problems. Infeasible solutions are taken account in the population and the diversity contribution is part of the evaluation to avoid the premature of search. Experiments are done to a series of instances which come from the actual production activity. Results showed that these methods' performance meets the demand of real world situation.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

$\mathbb{N}$ - Natural numbers

$\mathbb{R}$ - Real numbers

ACO - Ant Colony Optimization B&B - Branch and Bound

BI - Best Improvement

CVRP - Capacitated Vehicle Routing Problem

DCOP - Deterministic Combinatorial Optimization Problem

EA - Evolutionary Algorithm

FB - First-Best

FI - First Improvement

GA - Genetic Algorithm

GB - Global-Best

IRP - Inventory Routing Problem

ITS - Intelligence Transport System

MACS-VRPTW - Multiple Ant Colony System for Vehicle Routing Problem with Time Window

MDVRP - Multiple Depot Vehicle Routing Problem

MOP - Multi-Objective Problem

OVRP - Open Vehicle Routing Problem

PVRP - Periodic Vehicle Routing Problem

SA - Simulated Annealing

TS - Tabu Search

TSP - Traveling Salesman Problem

VRP - Vehicle Routing Problem

VRPB - Vehicle Routing Problem with Backhauls VRPMT - Vehicle Routing Problem with Multiple Trips

VRPSD - Vehicle Routing Problem with Split Deliveries

ABBREVIATIONS

VRPSTW - Vehicle Routing Problem with Soft Time Window
VRPTW - Vehicle Routing Problem with Time Window

Chapter 1

Introduction

Contents

1.1 Motivation

Logistics has attracted enormous attention from researchers. The definition of logistics is: the process of planning, implementing, and controlling procedures for the efficient and effective transportation and storage of goods including services, and related information from the point of origin to the point of consumption for the purpose of conforming to customer requirements. In the last ten years, logistics finally became recognized as an area that was key to overall business success. It has great importance in the economy, in industry and in environment protection.

Logistics is an importance activity making extensive use of human and material resources that affect a national economy. Figure 1.1 shows that in European Union, logistics represented about 9% of GDP in the last decades. Figure 1.2 tells that this number is significantly higher in developing countries (about 20%

in China) than in developed countries (about 7% in Germany). The routing optimization can give savings of 5% (Hasle *et al.*, 2007) to a company as transportation is usually a significant component of the cost of a product (10%) (Rodrigue *et al.*, 2013). Logistics efficiency has become a main thing that manufacturers need to focus on. Transportation and logistics related costs as a percentage of sales range from 9% to 14% depending on industry sector for companies who do not adopt a logistics efficiency management approach. Transportation costs alone comprise the vast majority of this expense for most companies. By adopting a logistics efficiency management approach, logistics related costs as a percentage of sales drops from 5% to 7% depending on industry sector. For a company with sales of $10,000,000, that's a contribution to corporate profitability of $500,000 to $700,000. Consequently, any savings created by the logistics, even less than 5%, are significant.

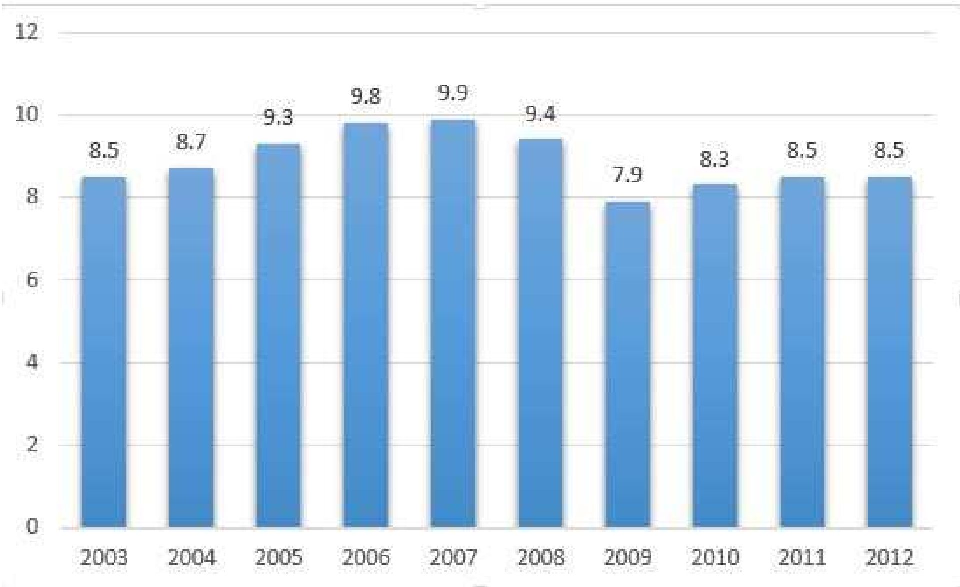

Figure 1.1: Logistics Cost As A Percent of GDP in EU (Source: State of Logistics Report 2014/CSCMP)

When looking at industry and company level, it is essential to be aware that the above costs are average figures taken across a number of companies. The relative make-up of these costs can vary quite significantly between different industries. In general, small companies tend to have proportionately higher logistics costs than large companies because large companies can benefit from economies of scale.

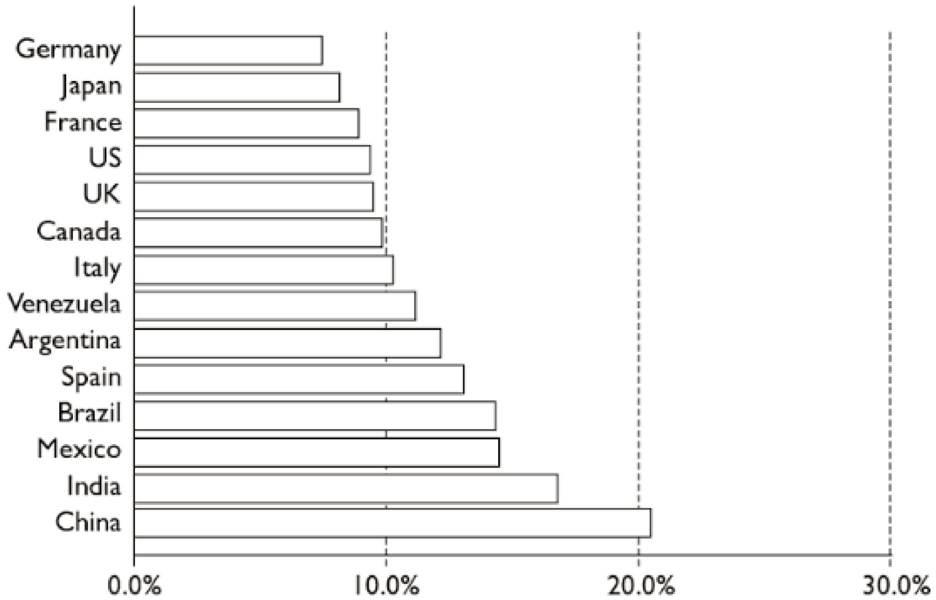

Figure 1.2: Logistics Costs as a Percentage of GDP for Selected Countries (Source: Capgemini Consulting, 2016 State of Third-Party Logistics Study)

Table 1.1: Percentage of Trucks' emissions in total transport related emissions in Ile-de-France (Dablanc, 2013)

	CO_2	SO_2	NO_X	PM_{10}
Trucks	26%	43%	38%	59%

Moreover, the growth of transport-related energy consumption and its negative effects on environment have attracted more and more worldwide concerns. Research on VRP not only reduces the transportation cost but also contributes to the environmental protection. For example, in Paris, a large part of the emissions of harmful gas comes from the truck transportation. As is shown in figure 1.1, truck transportations contribute an important part to the emissions of harmful gas, for CO_2: 26%; for SO_2: 43%; for NO_X: 38%; for PM_{10}: 59%. Trucks are usually used for logistics purposes. We could say that we could reduce environmental pollution by optimizing the routing for logistics.

Planning reasonable routes is essential for companies who offer on-site service or delivery service. The routes are reasonable or not makes great influences to the service cost, working efficiency and customer satisfaction. The research of this vehicle routing problem is beneficial not only to the company, but also to the society.

On one hand, it helps improve the efficiency of delivery or service of the company, make full use of its vehicle resources and increase economic efficiency. More importantly, reasonable routing plan ensures the arrival to customers' at the appointed time which will bring better service to the customers. On the other hand, for the society, rational routing planning can save vehicle resources, alleviate traffic congestion and reduce environmental pollution.

With the development of economic, more and more enterprises need provide low-cost and high-level services. Logistics and other transportation industries are also increasingly inclined to short-distance and short-term transport. Therefore, the problem of vehicle routing problem (VRP) is very worthy of study.

According to the State of Third-Party Logistics Study (Consulting, 2016), the top logistics challenges facing today is shown in figure 1.3. A majority of the study's respondents indicated "cutting transportation cost" as a top challenge, followed by the "business process improvement" and "improved customer service". The enterprises are demanding greater innovation and technology advances while simultaneously remaining cost-conscious.

1.2 Vehicle Routing Problem

VRP is one of the most analyzed problem in the fields of transportation, distribution and logistics. It calls for the determination of the optimal set of routes to be performed by a fleet of vehicles to serve a given set of customers (Toth & Vigo, 2014). It was introduced by Dantzig and Ramser (Dantzig & Ramser, 1959) in 1959, modeling how a fleet of homogeneous trucks could serve the demand for oil of a number of gas stations from a central hub and with a minimum traveled distance. In 1964, Clarke & Wright (1964) generalized this problem to a linear optimization problem which is known as the VRP, one of the most widely studied topics in the field of operation research (Braekers *et al.*, 2015). The VRP has been widely studied during the past decades and it is one of the most important combinatorial optimization problems. It is an integer NP-complete programming problem. So the size of problems that can be solved optimally is limited. The main objective is to minimize the cost of distributing the goods and

Figure 1.3: Top Logistics Challenges (Source: State of Third-Party Logistics Study)

find the shortest path between two points. It generalizes the well-known traveling salesman problem (TSP).

As the development of technology and production, new challenges are brought to researchers. On one hand, new techniques such as Electronic Commerce (EC), Global Positioning System (GPS), Intelligence Transport System (ITS), Geographic Information System (GIS) and Global System of Mobile communication (GMS) provide much more information to solve the problem. On the other hand, demands from customers become various. VRP could be applied to the pick-up of courier mail or packages, the dispatching of buses for the transportation of elderly and handicapped people, distribution of oil to private households and so on. There exists many variants in the family VRP.

The standard VRP is the Capacitated Vehicle Routing Problem (CVRP). In the CVRP, there is limitation to the loading capacity of each vehicle. This is the most basic VRP, and all other variants of the VRP are based on the standard problem.

The VRP can be represented as the following graph-theoretic problem. Let

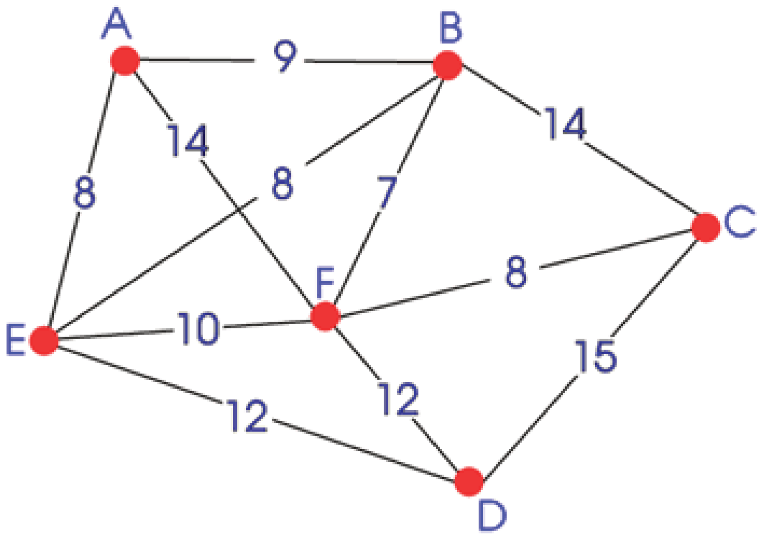

Figure 1.4: A Traveling Salesman Problem

$G = (V, A)$ be a complete graph where $V = \{0, 1, ..., n\}$ is the vertex set and $A = \{(i, j)|i, j \in V, i \neq j\}$ is the arc set. Vertices $j = 1, ..., n$ correspond to the customers, each with a known non-negative demand d_j, whereas vertex 0 corresponds to the depot. For vehicle routing problems, we define the following symbols:

- c_{ij}: Each arc (i, j), $i \neg j$ is associated to a non-negative cost c_{ij}. c_{ij} represents the travel costs between depots and customers. All the costs correspond to each arc constitute the cost matrix of the VRP.

- d_{ij}: This is the space distance between two vertices. If $c_{ij} = c_{ji}$ and $d_{ij} = d_{ji}$, the problem is said to be a symmetric VRP; otherwise, it is called an asymmetric VRP.

- δ_i: Vehicle service time at a customer i.

- m: The number of vehicles involved in the service.

- R: Set of all vehicles. $R = \{1, 2, ..., m\}$.

The objective of CVRP is hierarchical: minimize the number of vehicles and minimize the traveling cost. In the literature, the minimum number of vehicles is the primary objective. On this basis, the travel costs should be minimized. The

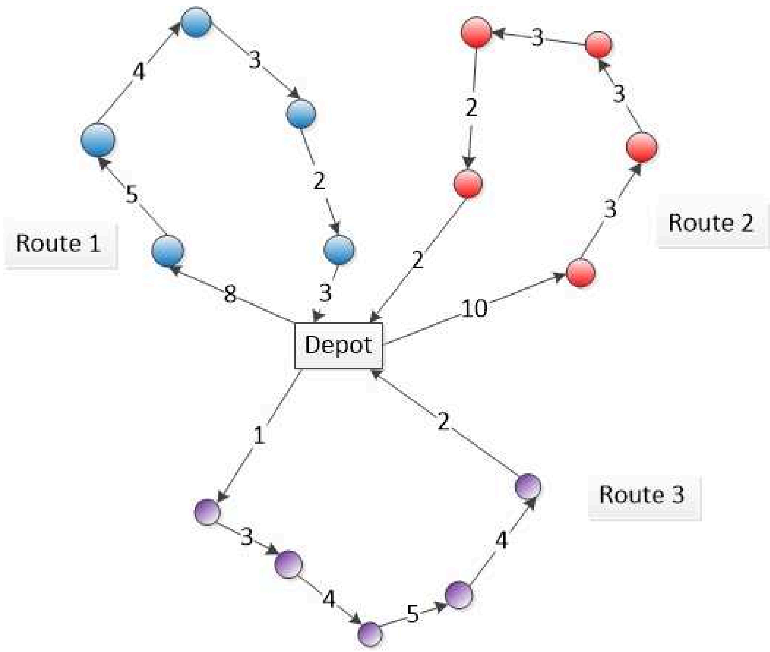

Figure 1.5: A Vehicle Routing Problem

mathematical model of CVRP is given below. To each arc (i, j), we define the variables:

$$x_{ijv} = \begin{cases} 1 & if\ vehicle\ v\ travels\ from\ i\ to\ j \\ 0 & otherwise \end{cases}$$

$$y_{iv} = \begin{cases} 1 & if\ customer\ i\ is\ served\ by\ vehicle\ v \\ 0 & otherwise \end{cases}$$

The mathematical model of CVRP can be represented as below:

$$Minimize\ F(x) = M \sum_{i=1}^{n} \sum_{v=1}^{m} x_{0iv} + \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{v=1}^{m} x_{ijv} c_{ij} \tag{1.1}$$

$$s.t. \quad \sum_{v=1}^{m} \sum_{i=0}^{n} x_{ijv} \geq 1 \qquad \forall j \in V^{'} \tag{1.2}$$

$$\sum_{i=0}^{n} x_{ipv} - \sum_{j=0}^{n} x_{pjv} = 0 \qquad \forall p \in V, \forall v \in R \tag{1.3}$$

$$\sum_{v=1}^{m} y_{iv} = 1 \qquad \forall i \in V^{'} \tag{1.4}$$

$$\sum_{i=1}^{n} d_i y_{iv} \leq Q \qquad \forall v \in R \tag{1.5}$$

$$y_{iv} = \sum_{i=1}^{n} \qquad \forall j \in V', v \in R \tag{1.6}$$

In the formula, $F(x)$ represents the objective function, and M is an infinite integer. During solving the problem, adding M in the target function can ensure the number of vehicles as first goal and the cost as second goal. A solution with less vehicles is better than a solution with more vehicles but smaller travel distance. Equation 1.2 indicates each customer is at least served by a vehicle once. Equation 1.3 is a traffic constraint that requires a vehicle must leave the customer after service. Equation 1.4 means customer i can only be served by one vehicle. Equation 1.5 is a vehicle capacity constraint that represents the sum of demand for the all customers by vehicle v on its service route can not be greater than the vehicle's loading capacity Q. Equation 1.6 indicates that the customer j can only be served by one of the vehicles from customers i.

1.3 Presentation of Problem Studied

As we all know, except for the delivery of goods, another important logistics activity is to offer on-site services to customers. Many companies regard providing good on-site service as an important factor to enhance the influence of enterprises. The problem we study in the thesis is a field service routing problem. This is a practical problem from the real production management. There is a time horizon consisting a number of period. A company, for example, a water conservancy company, should carry out different kind of service tour. Instead of delivering merchandises, it need to perform on-site service door to door, such as repair, equipment maintenance or examination. Different missions ask for various abilities. Technicians who are capable to finish the missions are sent to complete the works.

In this problem, customers are divided into two categories: *obligatory customers* and *optional customers*. The obligatory customers are the demands declared by the their clients. Each obligatory customer is associated with a time window. Technicians should arrive at customers' during certain periods of time fixed by the customer to offer the service. Obligatory customers are not available beyond its compatible periods or exceeding its time window. The other type of customer is optional customer. These customers are the maintenance operations planned by the company, for example, the renewal of water meter and other preventative interventions. These demands have no time window. Technicians are allowed to drop in anytime during the time horizon. It is important to notice that there is a difference in priority of these two kinds of customers. Obligatory customers are considered more importance than the other.

For a consideration of both service quality and economic reason, company would like to carry out all the missions with a minimum cost.

1.4 Contributions of the Dissertation

This dissertation considers the optimization of real world field service routing problem. The aim is to find optimized routing plan for companies to offer quality service with minimum cost.

We developed our research by investigating two categories of algorithms: heuristics and genetic algorithms. The main contributions of this dissertation are summarized as follows:

- The field service routing problem is an important problem in the modern industry. Yet the research done to the real-world problem is very limited. Realistic problems are usually more constrained and complicated. Problem studied in this dissertation is modeled for further investigation. Various constraints were taken into consider to simulate the realistic problem so as to apply to solve real problem.

- Exact methods and meta-heuristic methods are all widely used to general VRP. However, there is not many which can be applicable to the proposed real-world field service routing problem. We adopted genetic algorithm to solve the problem with new designed chromosome representation and operators to adapt to the problem. These components forms a new genetic algorithm which is suitable to the problem addressed. This method solves the problem by getting desired effect.

- Evolutionary algorithms often meet the problem of premature. To deal with this problem, a new genetic algorithm which allows the exploration in the infeasible solution is used. A diversity control procedure helps avoid the premature. Infeasible solutions are repaired by a repair operator. This method can solve large scale field service routing problem with good efficiency.

1.5 Organization of the Dissertation

This dissertation is organized as follows.

- Chapter 1 gives the motivation of research and a general introduction to the problem that we studied in this dissertation.

- Chapter 2 reviews the literature on VRP and the similar problems of our field service routing problem. The review includes the classification of vehicle routing problem and the existed solution methodologies. In particular, literatures on the multi-period VRP, the multi-depot VRP and the VRP with time window are studied for a deeper understand of the problem studied.

- Chapter 3 begins the discussion on the multi-period and multi-depot field service routing problem with time window. Neighbourhood searches are applied to obtain feasible solutions. First, we formalized the problem with its objective and constraints. Then a mathematical model is given for a more comprehensive sight and to be used in the dissertation. Heuristics of construction and heuristics of improvement are proposed to get reasonable solutions. Experiments are carried out to test the methods.

- Chapter 4 proposes a genetic algorithm for the problem addressed. A new chromosome representation is described and new crossover operators are presented which improve the rate of feasible offspring. We test the algorithm on a set of instances. The crossover operators are compared. Then we discuss the diversity controlling genetic algorithm for the addressed problem. Diversity contribution is taken as a part of the fitness to evaluate the individual in a population. Infeasible solutions are allowed into the population and a repair procedure is proposed to make infeasible solutions feasible. Experiments are realized with the new genetic algorithm with diversity control to evaluate its performance.

- To conclude, Chatper 5 summaries the key points of our research and outlines avenues for further research.

Chapter 2

State of the Art

Contents

In the precedent chapter, we have presented the standard vehicle routing problem. In fact, VRP has many different variants. In this chapter, we introduces the general context and constituents to the variants of VRP. A literature review on VRP is conducted. The researches on problems related to our problem are concluded.

2.1 Classification of Vehicle Routing Problem

In this section, the main characteristics and constituent elements of VRP are presented. On this basis, we introduce the different variants of VRP.

2.1.1 Constituent Elements of Vehicle Routing Problem

A typical VRP includes the below elements: road network, customer, depot, vehicle, side constraint and operational objective. An introduction to each constituent is given below.

Road Network

The road network is the basis of the carriage of goods, which is one of the most important elements of VRP. A road network is usually represented by a weighted graph consisting of vertexes and arcs. The vertex represents the depot or the customer, and the arc represents the road connection between the customer and the depot or the customer. According to the different characteristics of the road connecting two points in the transport network, the arc can be divided into directed arc and undirected arc. The directional arc refers to the road where the vehicle can only travel in one direction, and a typical example is the one-way road in the urban transport network. The undirected arc refers to the two-way road where the vehicle can travel in both directions. Each arc can be given a non-negative cost weight. According to the actual needs of the study, it can be given different meanings. For example, it can represent the travel distance or the travel time and so on.

Customer

The customer can represent any type of service object in the actual VRP, and its typical characteristic attributes include the following aspects:

- Customer corresponds to a vertex in the road network diagram.

- The customer point has a service demand, which can be the amount of goods delivered from the depot to the customer, or the amount of goods that need to be collected from the customer to the depot.

- Customer service time. It indicates the unloading time or cargo collection time of the customer.

- Customer service time window or time period. Some certain types of customers may have a service time window, which refers to the time interval during which the vehicle can start service to the customer, including the earliest allowed service time and the latest allowed service time.

Depot

The depot is the starting point or end point of each vehicle route. The vehicles deliver goods from the depot to the customer or collect goods from the customer to the depot. The depot is stationed with a group of vehicles to complete the delivery or collection service to the customers. In the general literature, VRP is assumed to have only one depot.

Vehicle

A group of vehicles complete the distribution or collection of goods for customers in VRP. The typical features include the following aspects:

- Type.

- Loading capacity.

- Cost. The vehicle cost here mainly refers to the fixed cost for use of the vehicle fixed costs and the variable cost. The variable cost refers to the cost for per kilometer or per hour.

- Duration. The vehicle used for goods delivery or collection has a maximum allowable travel distance or time. In the actual problem it indicates the maximum daily working hours of the vehicle driver.

Side Constraint

There are several types of side constraints for a typical standard VRP.

- The sum of the demands for all customers on each vehicle route cannot be greater than the loading capacity of the vehicle.

- The distance or travel time of a vehicle traveling on each vehicle route cannot be greater than its maximum duration.

- The demand for each customer must be met.

- The demand for each customer can only be completed by one vehicle and can only be accessed once.

Operational Objective

According to the characteristics of VRP, the optimal operational objectives can be divided into two categories: multiple-objective and single-objective. The typical single-object optimization functions are:

- Minimize the travel distance.

- Minimize the number of vehicles

- Minimize the total costs, including the fixed costs, variable costs and so on.

- Hierarchical optimization of the objective function. The number of vehicles is the primary optimization goals, and then the corresponding vehicle travel distance will be optimized.

The multi-object optimization mainly refers to the VRP that has more than one object needs to be optimized at the same time. In the actual distribution management, many vehicle routing problems are multi-objective decision-making optimization problems.

2.1.2 VRP Extended Criteria

The previous chapter has given definitions and mathematical models of standard VRP. The standard VRP is the simplest and most popular type of VRP in the field of operations research. Compared with standard VRP, VRP in the actual production operation management has new attributes and characteristics, such as service time window, undetermined quantity of demand, undetermined quantity of travel time and so on. The model and framework of standard VRP obviously cannot be used to model and analyze the new VRP. In order to meet the actual needs of production management, operational research studies the new

VRP extensions by relaxing the assumptions of standard VRP and introducing new constraints to the standard VRP. These new expansion problems gradually broaden the breadth and depth of study for VRP.

We have two ways to relax standard VRP according to the constitutive elements of VRP:

- By relaxing the assumptions of the standard VRP to generate new extended problems;
- By introducing new side constraints and integrating new service elements in the standard VRP based on the actual needs of production management.

The new VRP will gradually adapt to the actual needs of production management. Combining these extended elements with the standard VRP can build different types of VRP. The extended criteria are shown below.

Objective Function

This extension criterion mainly considers whether the VRP studied is a single-objective problem or a multi-objective problem (MOP). A MOP can be stated as follows:

$$(MOP) = \begin{cases} min\ F(x) = (f_1(x), f_(x), ..., f_n(x)) \\ s.t.\ x \in D \end{cases} \tag{2.1}$$

where $n \geq 2$ is the number of objective functions; $x = (x_1, x_2, ..., x_r)$, the decision variable vector; D the feasible solution space and $F(x)$ the objective vector.

The multi-objective routing problems are mainly used in three ways:

- to extend classic academic problems in order to improve their practical application;
- to generalize classic problems;
- to study real-life cased in which the objectives have been clearly identified by the decision-maker and are dedicated to a specific real-life problem.

Service Characteristics of Customer

This extended criterion describes the typical characteristics of the customer, including six sub-criteria.

- The time attribution, which describes whether the customer has time constraints for service. It can be divided into time window constraint and time limit constraint.
- The source of demand, which mainly describes whether the demand for service of the customer is allowed to be split. The customer must be served only by one car in standard VRP. However, in actual distribution management, transportation costs can be saved in situations where customer demand is split.
- The service time period, which mainly describes whether the service time is across more than one day. The service time period is one day in the standard VRP, but in the cycled VRP, it's a multi-day period.
- Demand characteristics. The customer's demand can be determined in advance, but also can be a random variable, and for some situations, the customer also needs to send goods and collect goods in the same time.
- The existence of the customer. The standard VRP assumes that all the customers exist, but the customer demand is not fixed corresponding to the increasingly competitive market economy. It may be a certain probability of random existence.
- The priority constraint, which is used to describe whether the service of the customer in VRP has the order of precedence. For the actual VRP, the customer may be divided into different levels, and the preferred customer can be given priority to the delivery service or collection service.

Vehicle Route Characteristics

The vehicle route characteristic criterion mainly describes whether the vehicle route is a closed Hamiltonian tour or a non-closed Hamiltonian path.

Road Network Characteristics

The road network extended criterion mainly includes the network travel cost and the network symmetry attribute. The standard VRP assumes that travel costs are known as fixed constants in advance. But in actual road networks, travel costs are a random variable subject to traffic conditions.

Depot Characteristics

This extended criterion gives the information on whether the depot is unique.

Vehicle Characteristics

It mainly includes four sub-criteria.

- Vehicle type, which describes whether the vehicle used for distribution and collection is homogeneous or shaped.

- Vehicle service type, which describes whether the vehicle is allowed to serve multiple routes. The vehicle in standard VRP can serve only in one route. In the actual production management, the number of vehicles may be limited and the service period may be shorter, so at this time allowing the vehicle to serve multiple routes may be the only viable option.

- The number of vehicles. The number of vehicle is unlimited in the standard VRP. In fact, the limited number of vehicles can reflect the actual situation of distribution management much better.

2.1.3 Vehicle Routing Problem Extensions

Combined different extended criteria with the standard VRP, we can get different types of VRP extensions. The main types are discussed below.

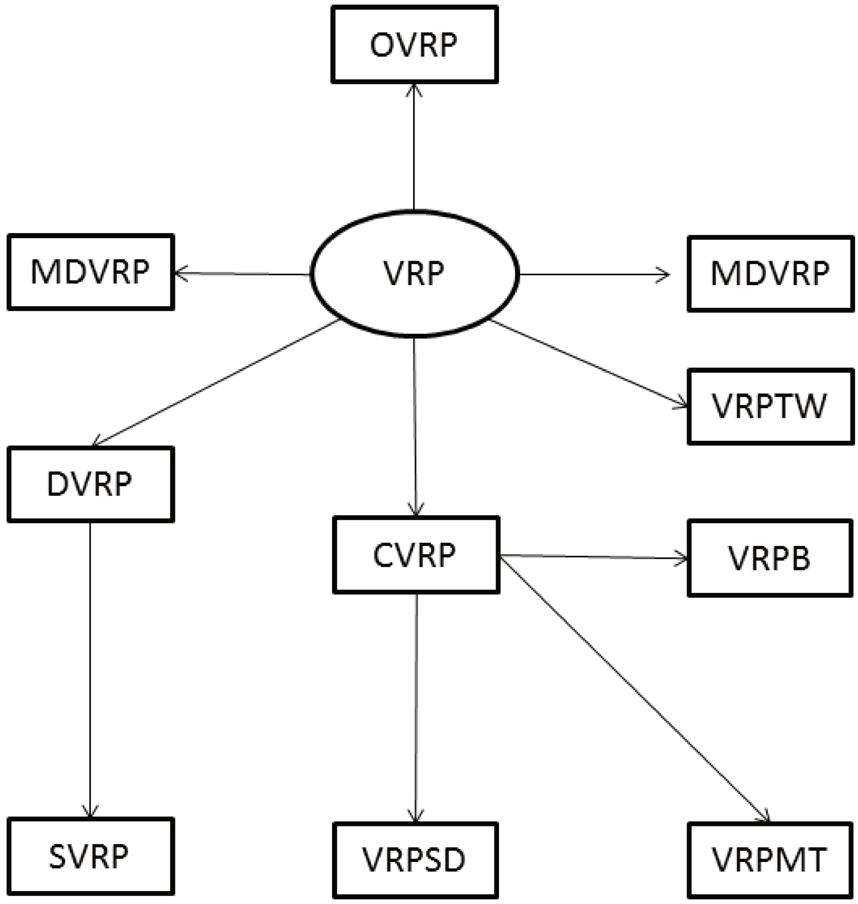

Figure 2.1: Different Variants of VRP

Vehicle Routing Problem with Time Window

The vehicle routing problem with time windows is an important problem in logistics distribution management. VRPTW sets a service time window for the customer based on the standard VRP. The service for customer can only be completed during the time window. According to the nature of the time window, it can be divided into two categories.

- Vehicle Routing Problem with Hard Time Window (VRPHTW). VRPHTW refers to the service for each customer must be completed in the time window. If the service is beyond the time window, the solution is not a feasible solution.

- Vehicle Routing Problem with Soft Time Window (VRPSTW). VRPSTW refers to the service for each customer cannot be completed in the time window. If the service time is beyond the time window, the solution of the objective function will be punished.

Vehicle Routing Problem with Split Deliveries

Vehicle Routing Problem with Split Deliveries (VRPSD) is an important relaxation problem for standard VRP. In the standard VRP, the delivery service of each customer can only be completed by one car. In VRPSD, the customer's demand can be divided by several vehicles at the same time. When considering the split deliveries, the number of vehicles and travel costs can be reduced, as described by an example below.

Assume the demands of each customer are: $d_1 = 3$, $d_2 = 4$, $d_3 = 3$. Assume the distances between customer and depot are: $c_{0i} = 10$, $i = 1, 2, 3$; $c_{12} = c_{23} = 1$, $c_{13} = 2$. The loading capacity of the vehicle is 5. The corresponding optimized solution for the standard VRP is: the number of vehicles is 3, and the sum of distance is 60. When considering the split deliveries, the optimized solution is: the number of vehicles is 2, and the sum of distances is 42. The VRPSD can save vehicle numbers and travel distances.

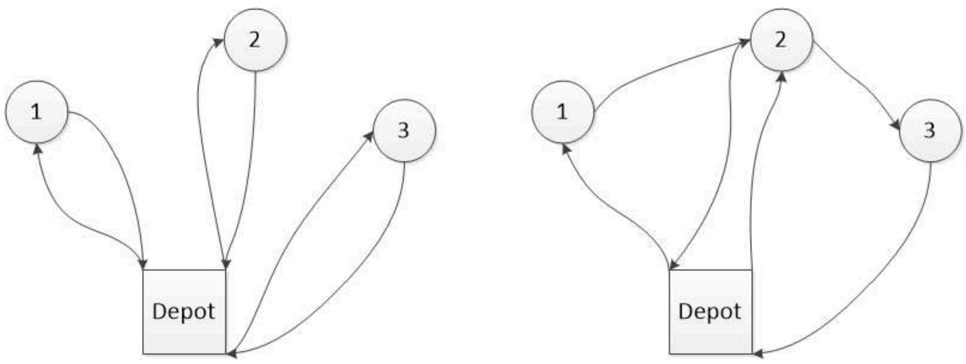

Figure 2.2: Example of VRPSD

Multiple Depot Vehicle Routing Problem

In the standard VRP, there is only one depot. All the vehicles start from the depot to the customers, and ultimately return to the depot. If a company has a number of depots, the demands of customers can be completed by vehicles form any depots. If the customers are clustered into different depot, the corresponding sub-issues are independent standard VRP. If the depots and the customers are mixed together, the problem is translated into a Multiple Depot Vehicle Routing Problem (MDVRP).

Period Vehicle Routing Problem

In the standard VRP, the planned delivery period of the vehicle is one day. Unlike the standard VRP, in Period Vehicle Routing Problem (PVRP), the planning period of the vehicle service is extended to several days, during which each customer is serviced at least once, and the set of dates is not fixed The visiting schedule of each customer is a table. If the period is one day, it is converted to a standard VRP.

Open Vehicle Routing Problem

The biggest difference between Open Vehicle Routing Problem (OVRP) and the standard VRP is that, the vehicle route is a Hamiltonian tour in the standard VRP but a Hamiltonian path in OVRP. In OVRP the vehicles does not need to be back to the departure depot. If they are asked to return to the depot, they must return along the same route. There are many applications in actual situations, especially in the economic activities with the characteristics of outsourcing business, such as newspaper distribution, milk distribution, etc. In such issues, the enterprises do not own vehicles, but outsource their distribution business to other vehicles or fleets. These companies do not need the vehicles to get back to the depot after serving the last customer, and the expenses paid for vehicles are affected by the travel distance. The figure shows an example of an OVRP.

Dynamic Vehicle Routing Problem

Thanks to recent advances in information and communication technologies, vehicle fleets can now be managed in real-time. Real-time data such as current vehicle locations, new customer requests, and periodic estimates of road travel times can be offered by some new devices and systems. In this context, dynamic vehicle routing problems (DVRP) are getting increasingly important. The dynamic vehicle routing problem is defined as below:

1. Not all information relevant to the planning of the routes is known by the planner when the routing process begins.

2. Information can change after the initial routes have been constructed.

In Figure 2.3 a simple example of a dynamic vehicle routing situation is shown. Pillac *et al.* (2013) classifies routing problems from the perspective of information quality and evolution. The authors give a presentation of dynamic routing and present a comprehensive review of applications and solution methods for DVRP. The last decade has seen the development of Intelligent Transport Systems (ITS), which are based on a combination of geolocation technologies, with precise geographic information systems and increasingly efficient hardware and software for data processing and operation planning. It is widely applied to home service (Borenstein *et al.*, 2010), (Tagmouti *et al.*, 2011), transport of goods (Barceló *et al.*, 2007), (Campbell & Savelsbergh, 2005), (Smolic-Rocak *et al.*, 2010), and transport of persons (Beaudry *et al.*, 2010), (Wen *et al.*, 2010).

Xiang *et al.* (2008) considers the dynamic multi-period vehicle routing problem which deals with the distribution of orders from a depot to a set of customers over a multi-period time horizon. Khouadjia *et al.* (2012) adapted PSO and VNS for solving VRP with dynamic requests and compared the results on benchmarks.

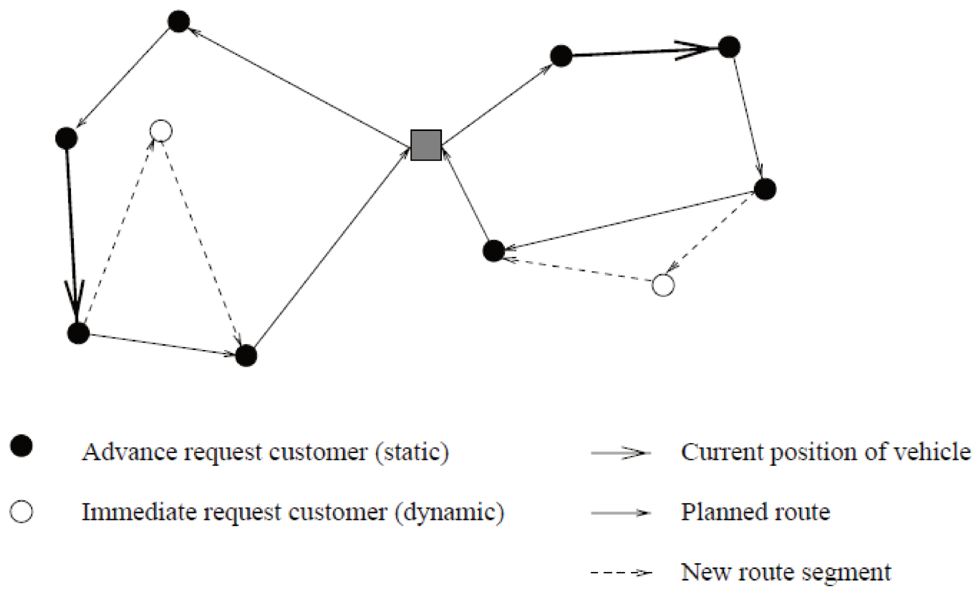

Figure 2.3: A DVRP with 8 advance and 2 immediate request customers

Stochastic Vehicle Routing Problem

Stochastic Vehicle Routing Problem (SVRP) belongs to DVRP. It has the below characteristics.

- Random customers. Each customer i exists at probability p_i, with probability $1 - p_i$ absent.

- Demand of customers. The demand of customer i is a random variable d_i.

- Random time. The random time here mainly refers to the random service hours and travel costs.

Vehicle Routing Problem with Backhauls

Vehicle Routing Problem with Backhauls (VRPB) is an extension of the standard VRP. In this type of problem, the customers are divided into two subsets. The first subset is the set of outset customers, that is, the vehicles need to send a certain amount of goods to the customers from the depot. The second subset is the set of inset customers, that is, the vehicles need to return a certain amount of goods to the depot from the customers. In VRRB, there is an important assumption for order. All the customers in the first subset must be served before the customers in the second subset. The demands of customers in both subsets are known and fixed.

Vehicle Routing Problem with Multiple Trips

Vehicle Routing Problem with Multiple Trips (VRPMT) is a relaxation problem for the the standard VRP. In the standard VRP, the number of vehicles is usually assumed to be infinite and each vehicle can only serve one route. However, the assumption is unreasonable in many practical applications. The actual number of vehicles is limited. When the loading capacity is very small or the planning period is very long, making a car to serve multiple routes may be the unique feasible choice.

2.2 Solution Methodologies

VRP is a very important combinatorial optimization problem. This is a NP-hard problem. Only small scale VRP can find global optimal solution so far. The current VRP algorithm can be divided into two categories: exact algorithms and heuristic algorithms. The heuristic algorithms include classical heuristics and metaheuristics.

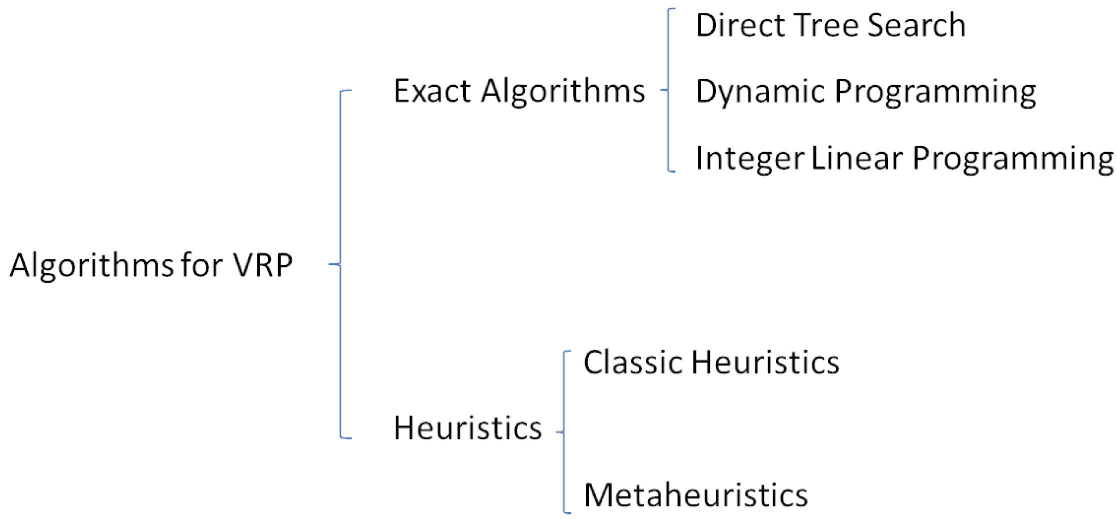

Figure 2.4: Algorithms for Vehicle Routing Problem

In this section, we will present the different algorithms for solving VRP. First, a brief introduction to the exact algorithm is given. Then, we describe the common heuristic algorithms. At the end of this section, we will focus on the metaheuristic algorithms which are the most commonly used method of solving large-scale vrp.

2.2.1 Exact Algorithms

Several families of exact algorithms have been proposed for the VRP with a symmetric cost structure. The exact algorithms can be classified into three classifications: (1) direct tree search method; (2) dynamic programming and (3) integer linear programming. At present, the exact algorithm can only solve the problem of small-scale VRP. Golden *et al.* (2008) indicates that for a vrp with more than 50 customers, exact algorithms do not agree on the optimal solution.

Laporte (1992) provide six representative examples of exact algorithms for VRP including two direct tree search methods based on different relaxations, a dynamic programming formulation and three integer programming algorithms.

The most effective exact algorithm for VRP is *branch-and-cut* (BC) algorithm based on a two-commodity network folw formulation of the problem (Baldacci *et al.*, 2008). The method of Fukasawa *et al.* (2006) proposed a new *branch-and-cut-and-price* (BCP) algorithm based on the two-index and the *set partitioning* (SP) formulations. The lower bound is computed with a column-and-cut generation method that uses *k-cycle-free* q-routes instead of feasible CVRP routes and the valid inequalities.The first exact algorithm for the VRPTW based on the SP formulation was the *branch-and-price* (BP) algorithm of Desrochers & Laporte (1991). In general, any exact algorithm for the VRPTW based on the SP model can be easily adapted to solve the CVRP by simply relaxing the time window constraints in the pricing algorithm.

2.2.2 Heuristics

Classical heuristics for the VRP are naturally divided into constructive heuristics and improvement heuristics. The descent heuristics always proceed from a solution to a better one in its neighbourhood until no further gain is possible. In contrast, methheuristics allow the consideration of non-improving and even infeasible intermediate solutions.

Construction heuristics mainly includes savings algorithm, route-first cluster-second, cluster-first route-second and insertion heuristics. Two types of improvement algorithms can be applied to VRP solutions: intra-route heuristics and inter-route heuristics.

Several heuristics are introduced below.

Savings Heuristic

Savings heuristic was proposed by Clarke & Wright (1964) to solve the problem of which the number of vehicles is not fixed. The saving heuristic is successive approximation algorithm based on saving criteria. The main idea of saving heuristic is to generate n routes consisted of only one depot and one customer point. Then calculate the saving cost for combing each of the two routes $s_{ij} = c_{i0} + c_{oj} - c_{ij}, (i, j = 1, 2, ..., n and i \neq j)$ and sort the values. According to the results and the feasible conditions to combine the routes and repeat this process until no more routes to combine. The procedure of savings heuristic is shown in Figure 2.5.

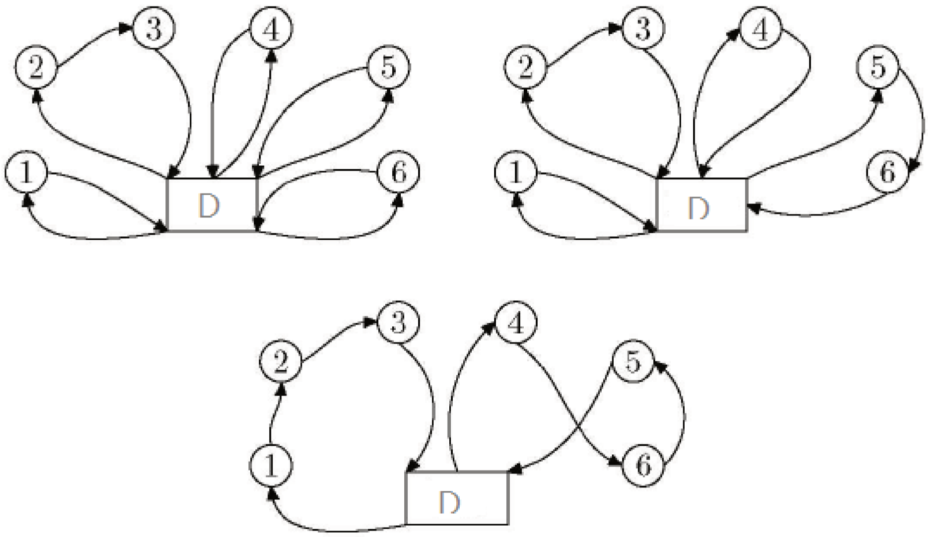

Figure 2.5: The Savings Heuristic

Saving heuristics is easy to realize. This method is suitable to the small-scale optimization. However, it does not work well when the scale of customer increases.

Nearest Neighbor Method

Nearest neighbor method is proposed by Rosenkrantz *et al.* (1977). This method start from the depot and search for the nearest unvisited customer as the next customer. Repeat this procedure in the condition of not exceeding the capacity limit until all customers are visited. The advantage of this algorithm is that the computation speed is fast and we can easily get initial solutions. The disadvantage is that it is easy to fall into local optimum.

Sweep Algorithm

The main idea of this algorithm is "partition first, route later". SAP (2002) presents the sweep algorithm and its application to VRP. The sweep algorithm applies to planar instances of the VRP. The depot is joined with an arbitrarily chosen point. All other nodes are joined to the depot and then aligned by increasing the angles which are formed by the segment that connects the nodes to the depot. It consists of two parts:

- Split: Feasible clusters are initiated formed rotating a ray centered at the depot based on their capacity (figure 2.6);
- TSP: A vehicle routing is then obtained for each cluster by solving a TSP.

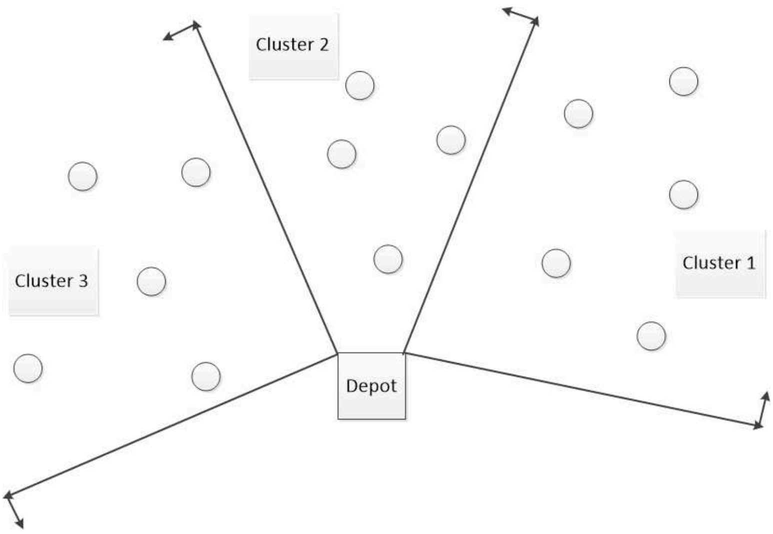

Figure 2.6: Clustering Process of Sweep Algorithm

2-Phase Algorithm

The problem is decomposed into its two natural components: (1) clustering of vertices into feasible routes then use $k - opt$ to optimize the routes respectively and (2) reduce the total travel cost by swapping between routes and optimize the routes with $k - opt$. The other 2-phase algorithm is proposed by Fisher & Jaikumar (1981). The main idea is to solve a Generalized Assignment Problem (GAP) to decide the feasible cluster and solve the TSP in each route.

Insertion Heuristics

Insertion heuristics are popular methods for solving a variety of vehicle routing and scheduling problems. The main principle of insertion heuristics is to start from a single node that is usually called a seed node and that forms the initial route from the depot. Other nodes are inserted one by one evaluating certain functions to select a node and the place in the route for insertion.

2.2.3 Meta-heuristics

The field of metaheuristics for the application to combinatorial optimization problems is a rapidly growing field of research. In Blum & Roli (2003), the authors give a thorough presentation of mateheuristic. To understand metaheuristic, we first give a formal definition of deterministic combinatorial optimization problem (DCOP)

Definition *(Deterministic Combinatorial Optimization Problem)* Given a finite set S of feasible solutions x, and a real valued cost function $G(x)$, find

$$\min_{x \in S} \quad G(x) \tag{2.2}$$

The set S is usually called search space. Its structure may be made complex by the presence of constraints on solutions. The solution x^* with minimal objective function value, that is, $G(x^*) <= G(x) \forall x \in S$, is called a globally optimal solution. For many DCOPs belonging to the class of NP-hard optimization problems, algorithms that guarantee to find the optimal solution within bounded time (exact algorithms) may require exponential computation time. Even for small instances of a problem, exact algorithms may require too much computation time for practical purposes. That is why there is a great interest in designing algorithms that find in a reasonable computation time a solution that is as good as possible, but not necessarily optimal. We call these algorithms approximate algorithms. Heuristics and metaheuristics are typical approximate algorithms.

Heuristics are basic approximate algorithms that search the solution space to find a good solution. There are two types of heuristics: constractive algorithms and local search algorithms. Constructive algorithms build a solution by joining together pieces until a solution is complete. Local search algorithms start from a pre-existent solution and try to improve it by modifying some of its components in an appropriately defined neighborhood of current solution. The neighborhood is defined as follows:

Definition A neighborhood structure is a function $\mathcal{N} : \mathcal{S} \to 2^{\mathcal{S}}$ that assigns to every $s \in \mathcal{S}$ a set of neighbors $\mathcal{N}(s) \subseteq \mathcal{S}$. $\mathcal{N}(s)$ is called the neighborhood of

s.

After knowing the definition of neighborhood, we can define the concept of *locally minimal solutions.*

Definition A *locally minimal solution* (or *local minimum*) with respect to a neighbourhood structure $\mathcal{N}$ is a solution $\hat{s}$ such that $\forall s \in mathcal N\hat{s}: \; f(\hat{s}) \; <= f(s)$. We call $\hat{s}$ a strict locally minimal solution if $f(\hat{s}) \; < \; f(s) \; \forall s \in \mathcal{N}(\hat{s})$

In Bianchi *et al.* (2009), the authors give a definition of metaheuristic: In computer science and mathematical optimization, a metaheuristic is a higher-level procedure or heuristic designed to find, generate or select a heuristic that may provide a sufficiently good solution to an optimization problem, especially with incomplete or imperfect information of limited computation capacity. In recent years, metaheuristics are emerging as successful alternatives to more classical approaches also for solving optimization problems that include in their mathematical formulation uncertain, stochastic, and dynamic information.

Blum & Roli (2003) lists the properties that characterize most metaheuristics:

- Metaheuristics are strategies that guide the search process.
- The goal is to efficiently explore the search space in order to find near-optimal solutions.
- Techniques which constitute metaheuristic algorithms range from simple local search procedures to complex learning processes.
- Metaheuristic algorithms are approximate and usually non-deterministic.
- Metaheuristics are not problem-specific.

There are a wide variety of metaheuristics. Some properties are use to classify them, such as local search or global search, single-solution or population-based, hybridization or memetic algorithms, parallel metaheuristcs and mature-inspired metaheuristcs.

The most common and studied metaheuristics include *Ant Colony Optimization*, *Simulated Annealing* , *Tabu Search* and *Evolutionary Algorithm*. Genetic algorithm, evolutionary programming and memetic algorithm belong to EA. The classification and main metaheuristics in each class is showed in figure 2.7.

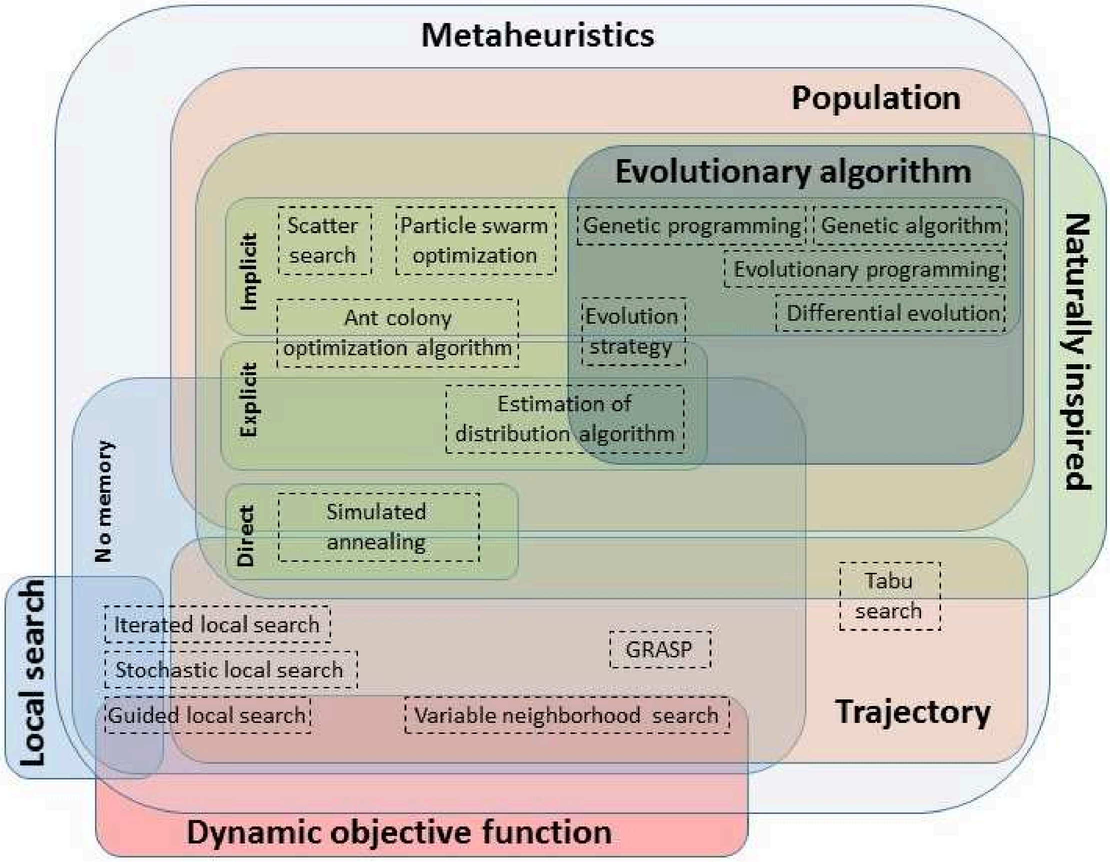

Figure 2.7: Classification of Metaheuristics

Tabu Search

Tabu Search was proposed by Glover (1989). The basic principle of TS is to pursue local search whenever it encounters a local optimum by allowing non-improving moves; cycling back to previously visited solutions is prevented by the use of memories, called tabu lists, that record the recent history of the search, a key idea that can be linked to Artificial Intelligence concepts.

Tabu search enhances the performance of local search by relaxing its basic rule. First, at each step worsening moves can be accepted if no improving move is available (like when the search is stuck at a strict local minimum). In addition, prohibitions (henceforth the term tabu) are introduced to discourage the search from coming back to previously-visited solutions.

The implementation of tabu search uses memory structures that describe the visited solutions or user-provided sets of rules. If a potential solution has been previously visited within a certain short-term period or if it has violated a rule, it is marked as "tabu" (forbidden) so that the algorithm does not consider that possibility repeatedly. We define a set of neighboring solutions in the search space, denoted $N(S)$ (the neighborhood of S): $N(S)$ ={solutions obtained by applying a single local transfromation to S}.

The procedure of tabu search is showed in figure 2.8.

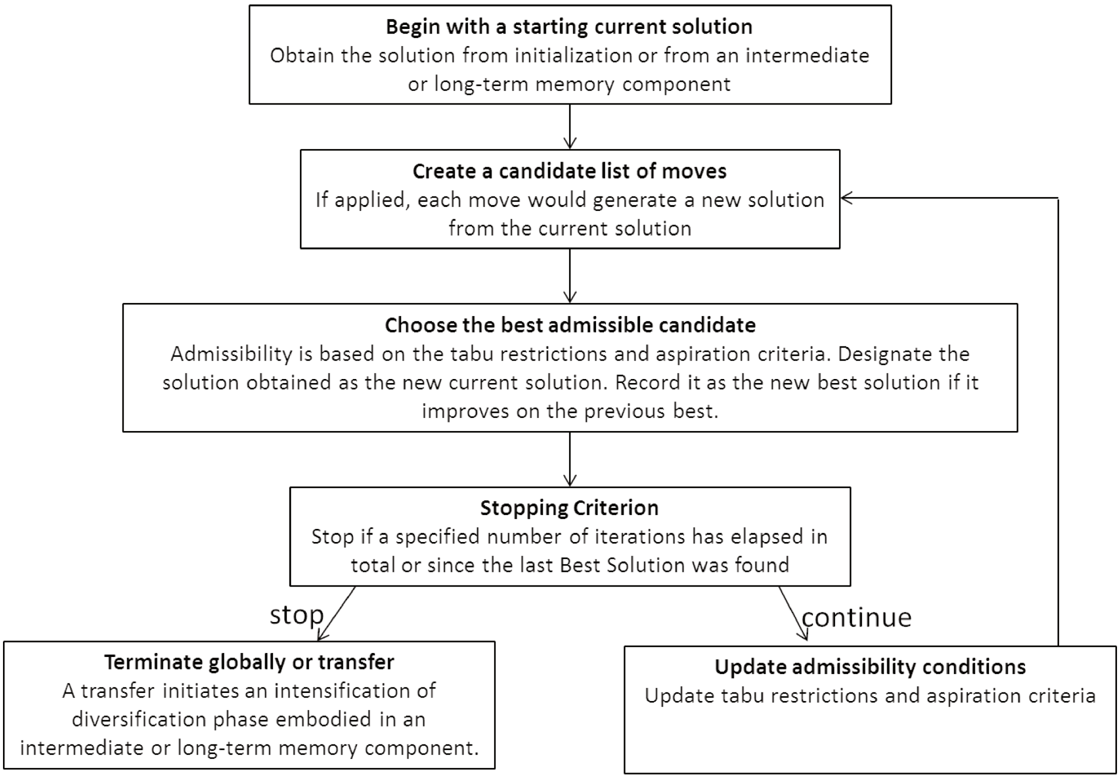

Figure 2.8: Tabu Search Procedure

Tabu search has been widely used to solve VRP. Basu (2012) review the tabu search literature on the TSP and its variations. Montané & Galvao (2006) developed a tabu search algorithm to solve VRP with simultaneous pick-up and delivery. In Cordeau & Maischberger (2012), the author introduced a parallel iterated tabu search heuristic for solving four different routing problems including the classical vehicle routing problem, the periodic VRP, the multi-depot VRP and the site-dependent VRP. In addition, tabu search is often used as hybrid method with other algorithms to solve VRP and its variants. Du & He (2012) presents a new and effective hybrid metaheuristic algorithm for large-scale vehicle routing problem. The algorithm combines the strengths of the well-known Nearest Neighbor Search and Tabu Search into a two-stage procedure. More precisely, Nearest Neighbor Search is used to construct initial routes in the first stage and the Tabu Search is utilized to optimize the intra-route and the inter-route in the second stage. Ngueveu *et al.* (2009) presents a hybridization of a perfect b-matching within a tabu search frame-work for the m-Peripatetic Vehicle Routing Problem. Liu *et al.* (2014) proposed a tabu search method combined with different local search schemes including both feasible and infeasible local searches. Khalifa *et al.* (2011) and Khalifa *et al.* (2010) applied tabu search to indoor navigation problem to solve the itinerary optimization problem inside hypermarkets.

Simulated Annealing

SA is a stochastic relaxation technique, which has its origin in statistical mechanics. It is based on an analogy from the annealing process of solids, where a solid is heated to a high temperature and gradually cooled in order for it to crystallize in a low energy configuration. SA can be seen as one way of trying to allow the basic dynamics of hill-climbing to also be able to escape local optima of poor solution quality. SA guides the original local search method in the following way. The solution S is accepted as the new current solution if $\Delta \leq 0$, where $\Delta = f(x) - f(x_i)$. To allow the search to escape a local optimum, moves that increase the objective function value are accepted with a probability $e^{-\Delta/T}$ if $\Delta > 0$, where T is a parameter called the 'temperature'. The value of T varies from a relatively large value to a small value close to zero. These values are controlled by a cooling schedule, which specifies the initial, and temperature values at each stage of the algorithm.

At iteration t of Simulated Annealing, a solution x is drawn randomly in $N(x_i)$.

$$x_{i+1} = \begin{cases} x & \text{with probability } p_i \\ x_i & \text{with probability } 1 - p_i \end{cases}$$

where p_i is usually a decreasing function of t and of Δ. It is common to define p_i as $e^{-\Delta/T}$.

There are three common stopping criteria:

- The value f^* of the incumbent x^* has not decreased by at least $\pi_1\%$ for at least k_1 consecutive cycle of T iterations;
- The number of accepted moves has been less than $\pi_2\%$ of T for k_2 consecutive cycles of T iterations;
- k_3 of T iterations have been executed

In the literature, SA has been used to solve VRP and its variants since 1990s, such as Osman (1993) and Chiang & Russell (1996). Vincent *et al.* (2010) proposed a SA based heuristic for solving location routing problem. Tavakkoli-Moghaddam *et al.* (2011) proposed a new mathematical model for solving VRPTW with SA. In BañOs *et al.* (2013), the author deals with a multi-objective variant of the VRPTW and proposed a multi-objective procedure based on SA called Multiple Temperature Pareto Simulated Annealing.

Ant Colony Algorithm

The first ant system for VRP has been designed very recently by Chen & Ting (2006), who considered the most elementary version of the problem: CVRP.

For more complex versions of VRP, Gajpal & Abad (2009) have developed a multiple ant colony system for VRPTW (MACS-VRPTW) which is organized with a hierarchy of artificial ant colonies designed to successively optimize a multiple objective function: the first colony minimizes the number of vehicles while the second colony minimizes the traveled distances. Cooperation between colonies is performed by exchanging information through pheromone updating.

There are two basic ant system phases: construction of vehicle routes and trail update. The AS algorithm is explained here.

- *Ant System Algorithm*

After initializing the AS, the two basic steps construction of vehicle routes and trail update, are repeated for a number of iterations. Concerning the initial placement of the artificial ants it was found that the number of ants should be equal at each customer at the beginning of an iteration. The 2-opt-heuristic (it is an exhaustive exploration of all the permutations obtainable by exchanging 2 cities) is used to shorten the vehicle routes generated by the artificial ants, considerably improves the solution quality. In addition to this straight forward local search we also introduce candidate lists for the selection of customers which are determined in the initialization phase of the algorithm. For each location d_{ij} we sort $V - \{v_i\}$ according to increasing distances d_{ij} to obtain the candidate list. The proposed AS for the CVRP can be described by the following schematic algorithm:

- Initialize.
- For I^{max} iterations do:
  - For all ants generate a new solution using Formula 2.3 and the candidate lists
  - Improve all vehicle routes using the 2-opt-heuristic
  - Update the pheromone trails using Formula 2.4
- Construction of Vehicle Routes.

To solve the VRP, the artificial ants construct solutions by successively choosing cities to visit, until each city has been visited. Whenever the choice of another city would lead to an unfeasible solution for reasons of vehicle capacity or total route length, the depot is chosen and a new tour is started. For the selection of a (not yet visited) city, two aspects are taken into account: how good was the choice of that city, an information that is stored in the pheromone trails τ_{ij} is associated with each arc (v_i, v_j), and how promising is the choice of that city. This latter measure of desirability, called visibility and denoted by η_{ij}, is the local heuristic function mentioned above.

With $\Omega = \{v_j \in V : v_j \text{ is feasible to be visited}\} \bigcup \{v_0\}$, city v_j is selected to be visited as follows:

$$p_{ij} = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{k \in \Omega} [\tau_{ik}]^\alpha [\eta_{ik}]^\beta} & \text{if } v_j \in \Omega \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

This probability distribution is biased by the parameters α and β that determine the relative influence of the trails and the visibility, respectively. The visibility is defined as the reciprocal of the distance, and the selection probability is then further extended by problem specific information. There, the inclusion of savings and capacity utilization both lead to better results. On the other hand, the latter is relative costly in terms of computation time (as it has to be calculated in each step of an iteration) and will therefore not be used in this paper. Thus, we introduce the parameters f and g, and use the following parametrical saving function for the visibility: $\eta_{ij} = d_{i0} + d_{0j} g d_{ij} + f|d_{i0} d_{0j}|$.

- *Trail Update*

After an artificial ant has constructed a feasible solution, the pheromone trails are laid depending on the objective value of the solution. This update rule is as follows:

$$\tau_{ij}^{new} = p\tau_{ij}^{old} + \sum_{\mu=1}^{\sigma 1} \Delta\tau_{ij}^{\mu} + \sigma\Delta\tau_{ij}^{*} \tag{2.4}$$

where p is the trail persistence (with $0 \leq \rho \leq 1$), thus the trail evaporation is given by $(1-\rho)$. Only if arc (v_i, v_j) was used by the μ-th best ant, the pheromone trail is increased by a quantity $\Delta\tau_{ij}^{\mu}$ which is then equal to $(\sigma\mu)/L_{\mu}$, and zero otherwise (cf. second term in 2.4). In addition to that, all arcs belonging to the so far best solution (objective value L^*) are emphasized as if σ elitist ants had used them. Thus, each elitist ant increases the trail intensity by an amount $\Delta\tau_{ij}^{*}$ that is equal to $1/L^*$ if arc (v_i, v_j) belongs to the so far best solution, and zero otherwise (cf. third term in 2.4).

There are a large number of literature using ant colony algorithm to solve VRP. Yu & Yang (2011) proposed an improved ant colony optimization to solve period vehicle routing problem with time window and get better results than best-know solutions. Gajpal & Abad (2009) use a multi-ant colony system to solve vehicle routing problem with backhauls with a new construction rule and two multi-route local search schemes. Yu *et al.* (2009) use an ant colony optimization possessing a new strategy to update the increased pheromone. Rizzoli *et al.* (2007) discusses the applications of ACO to the different real-world problem such as supermarket chain, distribution company and freight distribution.

Evolutionary Algorithms

Among the set of search and optimization techniques, the development of Evolutionary Algorithms has been very important in the last decades. EA is a set of modern generic population-based metaheuristics used successfully in many applications with great complexity. EAs use mechanisms inspired by biological evolution, such as reproduction, mutation, recombination and selection. The family of EA includes genetic algorithm, genetic programming, evolutionary programming, gene expression programming, evolution strategy, differential evolution, neuro-evolution, learning classifier system, etc. In the evolutionary systems, two fundamental forces are essential: variation operators (recombination and mutation) create the necessary diversity and thereby facilitate novelty; selection acts as a force pushing quality.

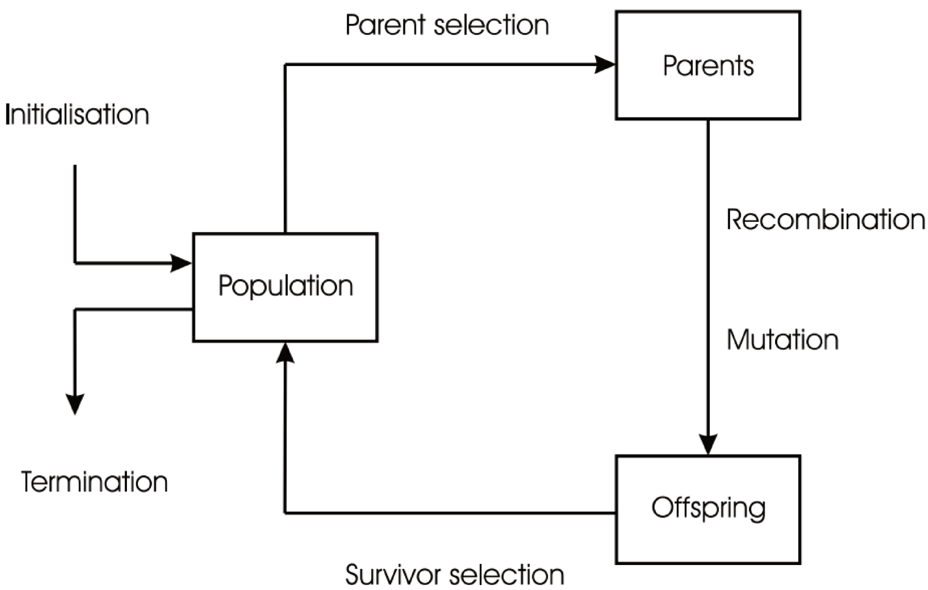

Figure 2.9: Flow-chart of Evolutionary Algorithm

EA has been widely used to solve VRP and its variants in the research of recent years. Garcia-Najera & Bullinaria (2011) proposed a novel multi-objective evolutionary algorithm incorporating methods for measuring the similarity of solutions to solve VRPTW. In Mester & Bräysy (2007), an active-guided evolution strategies metaheuristic is presented which combines the strengths of the well-known guided local search and evolution strategies into an iterative two-stage procedure. Memetic algorithm is used in Labadi *et al.* (2008), Nguevue *et al.* (2010) and Cattaruzza *et al.* (2014) for solving various variants of VRP and the results show the advantages of EA in large-scale Combinatorial optimization problem.

GA is the most important and most applied algorithm in the family of EA. We will give a detailed description of GA in chapter 4.

2.3 Similar Variants to Our Problem

In this section, we focused on the variants of VRP that share characteristics with our problem, including the priority of customers, limitation of fleet, multi-depot and multi-period VRP. We will present the problem which are similar to ours in one or more respects.

2.3.1 Multi-Period Vehicle Routing Problem

In classical VRPs, typically the planning period is a single day. VRP of multi-period considers a planning horizon consisted of many period. In this way, the vehicles should execute many tours in the plan. There are two main types of MPVEP: the Periodic Vehicle Routing Problem (PVRP) and the Inventory Routing Problem (IRP). In the PVRP, each customer may be served more than once. The objective is to minimize the cost while serve all customers for a certain number of times. IRP has become a spot of research during last decades. It is a more global approach than the PVRP. It integrates inventory management, vehicle routing and delivery scheduling decisions (Coelho *et al.*, 2013).

Prodhon (2008) combines the Location-Routing Problem (LRP) and PVRP into the Periodic LRP and proposed a metaheuristic based on Randomized Extended Clarke and Wright Algorithm and tried to take into consideration several decision levels when making a choice during the construction of a solution. Alonso *et al.* (2008) and Prodhon (2011) both choose tabu search for PVRP. Exact algorithms are often used for multi-period VRP in the literature. Mourgaya & Vanderbeck (2007) use a truncated column generation procedure followed by a rounding heuristic to find approximate solutions. This method can only deal with problems with 50-80 customers over five working days. Dayarian *et al.* (2015) proposed a mathematical model based on a two-stage a priori optimization paradigm. The first stage is solved by a branch-and-price approach and the subproblem is solved by a dynamic programming based label-correcting algorithm. The PVRP is reviewed in Francis *et al.* (2008). Exact methods Baldacci *et al.* (2011) are able to solve some instances with up to 100 customers and 6 time periods. Several efficient neighborhood-centered searches have been designed Cordeau *et al.* (2001) and Cordeau & Maischberger (2012). The population-based approach of Alegre *et al.* (2007) dedicated to large temporal horizons, focuses on assignment optimization, while using constructive methods to create routes.

2.3.2 Multi-Depot Vehicle Routing Problem

A company may have several depots from which the vehicle can serve its customers. If the customers are clustered around depots, then the distribution problem should be modeled as a set of independent VRPs. However, if the customers and the depots are intermingled then a Multi-Depot Vehicle Routing Problem should be solved. The MDVRP deals with a number of depots. Each vehicle is assigned to a single depot, which is generally both the origin and the destination of the vehicle's route.

A review on MDVRP is given in Montoya-Torres *et al.* (2015). Exact algorithms and heuristics are used to solve the MDVRP. For example, branch-and-cut algorithms were proposed by Benavent & Martínez (2013) and Braekers *et al.* (2014).

Salhi & Sari (1997) proposed a multi-level composite heuristic which sharply decrease the computing time. This method was tested on different problems in Salhi & Nagy (1999) and Nagy & Salhi (2005). Jin *et al.* (2004) modeled the MDVRP as a binary programming problem and proposed a two-stage approach that decomposes and solves the problem into two independent subproblems: assignment and routing. A variable neighborhood search has been applied into MDVRP for the first time in Polacek *et al.* (2004). Most of the existed approach for multi-depot can be divided into two stage called 'cluster first, route second': assignment and routing, then solve the two subproblems separately. Lim & Wang (2005) proposed a one-stage methodology and compared to the traditional two-stage method and the results show that the new one-stage algorithm outperforms the two-stage methods.

Various meta-heuristics have been studied for MDVRP. Among them, we can highlight the SA algorithms in Wu *et al.* (2002) and Lim & Zhu (2006), the Variable Neighborhood Search in Polacek *et al.* (2004), Tabu Search algorithm of Lim & Wang (2005) and Aras *et al.* (2011). GA has been proposed in Hwang (2000) and Villegas *et al.* (2010). Yu *et al.* (2011) developed a parallel improved ant colony optimization.

2.3.3 Vehicle Routing Problem with Time Window

VRP with time windows (VRPTW) is the most extensively studied VRP variant. Each depot or customer is associated to an available time duration. Due to the restriction on problem scale that the exact methods can solve, a lot of researches have been done to heuristics for VRPTW. Garcia-Najera & Bullinaria (2011) proposed and analyzed a novel multi-objective evolutionary algorithm, which incorporates methods for measuring the similarity of solutions, to solve the VRPTW. Macedo *et al.* (2011) considers the VRP with time windows and multiple routes. The authors proposed a new exact algorithm for the problem relies on a pseudo-polynomial network flow model. A Tabu Search method is proposed in Taş *et al.* (2013) to solve a VRP with soft time windows and stochastic travel times with a post-optimization method.

As for limited fleet number, there is little research in this type of VRP. In a typical VRP, the objective is to minimize the vehicle used to fulfill all the customer visits. In a variant with limited fleet, the number of vehicles is limited. The primary objective is to maximize the number of visited customers. There are not many researches which combine the above characteristics together. To our information, Tricoire (2007) solve the multi-period and multi-depot routing problem for service technicians with memetic algorithm; Vidal *et al.* (2013) introduces a hybrid genetic search with advanced diversity control which can be applied to solve MDVRPTW and MPVRPTW.

A lot of researches have been done to solve VRP variants focused on the practical usage of the real world problems. Results from these researches show that compared to other metaheuristic algorithms, GA has advantages in both aspect of performance and final result on time constraints and limited compute ability. There exist some other metaheuristics able to find better solution than GA. However, GA make a balance between the solution quality and computing time. Vaira & Kurasova (2014) proposed a genetic algorithm based on insertion heuristics for the vehicle routing problem with constraints. The author uses random insertion heuristic to get initial solutions and to reconstruct the existing ones. Tasan &

Gen (2012) proposes a GA based approach to solve the VRP with simultaneous pick-up and deliveries. A genetic algorithm for the multi-compartment vehicle routing problem with continously flexible compartment sizes is proposed in Koch *et al.* (2016). The paper Lau *et al.* (2010) deals with the optimization of VRP of multi-depot, multi-customers and multi-products. They propose a stochastic search technique to dynamically adjust the crossover rate and mutation rate after a certain generations. Genetic algorithm is frequently used to solve VRPTW thanks to its ability to always find feasible solution. Genetic algorithms for multi-objective VRPTW are discussed in Ombuki *et al.* (2006) and Ghoseiri & Ghannadpour (2010). The former takes advantage of the Pareto ranking technique for evaluating the solution in a balanced way. The latter presents a new model and combined goal programming with genetic algorithm.

Concerning the multi-period multi-depot vehicle routing problem with time window, not many works appear in the literature. Chiu *et al.* (2006) presented a two-phase heuristic method. Bostel *et al.* (2008) solve the problem of the planning and routing of technician visits to customers in the field, for maintenance or service logitics acivities undertaken by utilities by a memetic algorithm and a column generation/branch and bound heuristic.

2.4 Conclusion

This chapter has presented the variants of vehicle routing problem. The main variants of VRP are presented and the existed algorithms are introduced for solving the problem. The similar problems to our problem studied are discussed to get a clear vision for the addressed problem. There are many studies on different variants of VRPs in the literature. However, few of these studies can be used directly to solve the multi-depot and multi-period VRP with time window. Therefore, it is necessary to carry out a more in-depth study of the question raised in this dissertation.

Chapter 3

Heuristics of Construction and Improvement

Contents

In this chapter, we first give a thorough presentation of the addressed problem, including the description of real-world problem and the mathematical model. Then an investigation of local search is done to solve the problem as the basic knowledge for the following research. We will present a series of heuristics to construct and improve the multi-depot and multi-period field service routing problem with time window and fixed fleet. These heuristics could produce the feasible solutions which could serve as the start point for the metaheuristics search for the following chapter. In addition, we need a quick heuristic for get a reasonable and feasible solution in the actual production life.

3.1 Problem Formalization

Real-life VRPs arise daily in a variety of different contexts and applications, and they usually introduce certain complications to the basic VRP that has been addressed extensively in the literature. Most of the complications are related to the following aspects:

- Planning horizon: in basic VRP, a single time period is addressed while in real-life problems, routes are planned for a given planning horizon that may consist of multiple periods. A customer may be served only over a subset of these periods or over more than one period.
- Customer: in real-life problems, different types of services are requested from the customers.
- Depot: there can be multiple depots in a large distribution network. A customer may be served by all the depots or only by a subset.
- Driver: in most real-life problems, distributors need to consider the drivers' working regulations for example the working shift and breaks. In addition, specific qualifications may be required to drive particular vehicles.

As the problem we study is a practical problem, it has differences in the above points with the basic problem. In the following part of this section, details will be given to understand the problem we studied.

3.1.1 Problem Description

We address the problem of the planning and routing of technician visits to customers in the field, for maintenance or service logistics activities undertaken by utilities. Field service routing planning and optimization is a new challenge in logistics for the service sector and especially for utility companies in the energy (gas, electricity), telecommunications and water distribution areas. It generates new variations of combinatorial optimization problems in the fields of manpower scheduling and vehicle routing. This activity consists in planning the work allocations and schedules of commercial or technical personnel in the field, over a set of time periods (usually workdays) to visit industrial facilities or customers for different types of activity: contracting, equipment maintenance or replacement, customer surveys. The challenges are many: to increase productivity and reduce costs, by increasing the number of visited clients, while reducing the time and cost of transportation to reach them; to increase customer service by setting appointments for home visits and to achieve an efficient internal organization and appropriate human resources planning. The demand for services may result from various processes and be generated by the company itself (for example the maintenance of equipment), or by the clients through a call center (for example repair of emergency reasons). The overall objectives of the company are:

- to provide and improve a good customer service, by adequately answering the customer requests for visits and meeting customer appointments;
- to satisfy the internal needs for customers visits for maintenance or commercial activities;
- to achieve a better productivity by reducing transportation time and costs and increasing time spent by technicians at customer sites;
- to implement an efficient company organization , involving better technician schedules.

The formal description of the problem is given as follows:

We consider a multiperiod planning horizon composed of a given number of subsequent days. For each day, a given number of technicians is available for service, each with a known starting and ending location. Demands for visits to customers are known at the beginning of the horizon. They may have been generated by the company or by customer requests. Technicians undertake routes between their starting and ending points to visit a number of clients with their vehicle. The goal concerns the minimization of transportation distances while satisfying all the demands for visits over the planning horizon, and meeting other constraints such as time windows and compatible period. The duration of technician visits at the customer sites is considered as deterministic and known.

This problem can be assimilated as a VRP. It is related but not identical to several variations such as multiple traveling salesman problem or the periodic vehicle routing problem. The characteristics will be presented below.

- In our problem, what vehicles delivery is service instead of merchandise. Therefore, the constraint of capacity is out of consideration. However, the total duration of each tour is limited by the length of a working day. and this is in practice a strong constraint. In addition, some work on capacity problems also considers the available time. Nevertheless, the capacity is generally the strongest constraint, and measuring the time consumed serves mainly to verify that the windows of time are respected.

- We deal with the case of multi-period with a planning horizon of several days. However, these trips are service rounds, and each request must be satisfied only once. It is different to the traditional multi-period problems such as the IRP. A validity period is here associated with each request, and represents a set of days of the horizon during which the demand can be satisfied.

- The size of the fleet is limited, and this limit is a strong constraint. When a tour is performed by a technician, the number of technicians available on day t is the upper bound of the number of tours associated with day t. The availabilities of technicians are known a priori, the total number of possible tours over the whole horizon is known. We introduce here the notion of "resource" (Tricoire, 2006), associated with one day and one technician (vehicle). It represents the availability of one technician in one work period. Each resource can be used at most once. It means that each technician carries out no more than one tour every work day. The total number of available resources is the upper bound of the total number of tours in a solution.

- The depot is not unique. Each vehicle has its start point and end point and potentially different to others'.

- The customers are of different categories: obligatory customers and optional customers as explained in section 1.3.

We differentiate the constraints according to two categories: constraints on demands, and constraints on resources, which must be respected by a tour so that it can be realized.

3.1.2 Constraints on Demands

We are going to give an exhaustive list of the constraints that are applied to demands:

- Each obligatory demands should be served once and only once within the planning horizon.

- For sufficient problems, the optional request should served once and only once during the whole time horizon. For insufficient problems, each optional request should be met at most once.

- Each request is associated to a period of validation. The customer cannot be met unless we serve him within the given periods. This period is determined respectively to each request. It can be vary from one working day to the whole horizon.

- Some requests have a time window during their available period.

3.1.3 Constraints on Vehicles and Periods

The constraints on resources ensure that the tours are realizable.

- Each resource vehicle - working day can be used at most once. It means that each technician carries out no more than one tour every day.

- The depart point and the arrival point are the same. That is to say, a technician should return to the depot where he start after the tour to customers'.

- The duration of one period is limited by the work time of technicians. It could be seen as the time window of the arrival point.

3.2 Mathematical Model

We have presented the problem we focus in this thesis in previous sections. For having a more clear view and perceive, we will give the mathematical description to both the two situations (sufficient fleet and insufficient fleet) stated above. The mathematical model is associated to the real problem and the notations in it will be used in the following chapters.

Let $G = (V, A)$ be a directed graph. A set of vehicles K offer service to customers. There are two types of customers: obligatory customers with time windows and optional customers without time windows. There are some points we should pay attention to:

- There is no arc between two depart points or two arrival points.

- There is no arc whose destination is a depart point or start from an arrival point.

- Every arrival point is associated to a time window with is the working time limitation of the resource.

- For those optional customers who have no time windows, it can be seen that they have a time window of the total working time of the period.

We now give the notations for the mathematical model of the problem. They will be used all through this thesis.

- K set of vehicles (technicians) to serve customers, $|K| = m$.

- L set of periods consisting in the time horizon, $|L| = w$.

- N set of all customer requests.

- A set of obligatory customers.
- Q set of optional customers.
- O set of departure points and arrival points; the departure point and arrival point of vehicle k is o^k.
- V set of all nodes in the graph of the problem. $V = A \cup Q \cup O$.
- $[e_i, l_i]$ time window of customer i.
- σ_i service time of customer i.
- τ_{ij} travel time from customer i to j.
- c_{ij} transportation cost from customer i to j.
- q_i^l binary constant, the value is 1 if customer i is compatible with period l, 0 if not.
- M a large number, $M \in R^+$.

We define variables:

- x_{ij}^{kl} binary decision variables, take value 1 if and only if vehicle k in period l visits v_j immediately after v_i.
- y_i^{kl} binary decision variables, take value 1 if and only if vehicle k visits v_i in period l.
- t_i^{kl} arrival time at customer i by technician k in time period l.
- b_i^{kl} service starting time at customer i by technician k in time period l.

For the situation of sufficient fleet, the objective is to minimize the total cost of the tours.

$$Minimize \sum_{v_i \in V} \sum_{v_j \in V} \sum_{k=1}^{m} \sum_{l=1}^{w} c_{ij} x_{ij}^{kl} \tag{3.1}$$

Subject to:

$$\sum_{i \in V} x_{o^k i}^{kl} - \sum_{i \in V} x_{i o^k}^{kl} = 0 \qquad \forall k \in K, \forall l \in L \tag{3.2}$$

$$\sum_{k\in K,l\in L} y_i^{kl} = 1 \qquad \forall i \in N \tag{3.3}$$

$$y_j^{kl} - \sum_{i\in V} x_{ij}^{kl} = 0 \qquad \forall k \in K, \forall l \in L, \forall j \in N \tag{3.4}$$

$$y_j^{kl} - \sum_{i\in V} x_{ji}^{kl} = 0 \qquad \forall k \in K, \forall l \in L, \forall j \in N \tag{3.5}$$

$$y_i^{kl} \leq q_i^l \qquad \forall i \in A, k \in K \tag{3.6}$$

$$b_i^{kl} + \sigma_i + \tau_{ij} - M(1 - x_{ij}^{kl}) \leq b_j^{kl} \qquad \forall i, j \in V \tag{3.7}$$

$$b_i^{kl} + M(1 - y_i^{kl}) \geq e_i \qquad \forall i \in V \tag{3.8}$$

$$b_i^{kl} - M(1 - y_i^{kl}) \leq l_i \qquad \forall i \in V \tag{3.9}$$

$$\sum_{i\in V/S}\sum_{j\in S} x_{ij}^{kl} \geq y_v^{kl} \qquad \forall v \in S, \forall S \subseteq V, \forall k \in K, \forall l \in L \tag{3.10}$$

$$x_{ij}^{kl} \in \{0, 1\} \qquad \forall (i, j) \in A, \forall k \in K, \forall l \in L \tag{3.11}$$

$$y_{ij}^{kl} \in \{0, 1\} \qquad \forall (i, j) \in A, \forall k \in K, \forall l \in L \tag{3.12}$$

$$b_i^{kl}, t_i^{kl} \geq 0 \qquad \forall i \in V, \forall k \in K, \forall l \in L \tag{3.13}$$

Equation 3.2 ensures that if an arc depart from a depart point of a certain resource, there must be an arc enter the same point as arrival point of a tour. Equation 3.3 impose the satisfaction of every customer request. Equation 3.4 and equation 3.5 are the in-degree and out-degree constraints. The compatibility between customer request and the period of resource is guaranteed by equation 3.6. Equation 3.7 impose restriction on the start time at each customer's. Equation 3.8 and equation 3.9 assure the respect of time window of each customer request. Equation 3.10 ensure the elimination of subtour.

For the situation with insufficient number of vehicles, the objective is to maximize the number of intervention during the tour:

$$Maximize \sum_{i \in Q} \sum_{k \in K}^{m} \sum_{l \in L}^{w} y_i^{kl} \tag{3.14}$$

We modified the constraint 3.3 into two constraints that ensure the obligatory customer is served exactly one time and the optional customer is serve no more that one time.

$$\sum y_i^{kl} = 1, \qquad \forall i \in A \tag{3.15}$$

$$\sum y_i^{kl} \leq 1, \qquad \forall i \in Q \tag{3.16}$$

The other constraints will not change in the insufficient situation.

3.3 Heuristic of Construction

Heuristic approaches have been well studied over the last decade in the operation research and artificial intelligence fields. With the increase in computing power, the heuristics become more complex and more advanced. In this section, we will present a family of heuristics for construction and improvement of solutions for the addressed multi-period field service vehicle routing problems with time window and limited fleet. These heuristics permit to obtain the solutions which are the base for the meta-heuristics in the following chapters. The meta-heuristics require a very fast heuristic for the intensive use when generating new solutions. Furthermore, in an industrial context, the enterprises appreciate having a very fast method (a few seconds at most) of solving this problem. These are the three reasons that the study of heuristics is necessary. The simple heuristics proposed between 1960 and 1990 are used to the standard construction and improvement procedures today. These methods perform a relatively limited exploration of search space and generally produce good quality solutions within modest computing time. Most of these heuristics can be extended to adapt to the diverse constraints encountered in real-life contexts.

We will start with the heuristic we use to construct the solution. Then we will give the details for a variety of improve heuristics for the generated solutions. All of the methods have been tested and the result of experiments are discussed in the last section.

The main techniques for constructing VRP solutions are *savings criterion* and *insertion cost*. The former merges existing routes and the latter assign vertices to each vehicle routes (Laporte *et al.*, 2000).

Most of the existed methods for a multi-period VRP divide the problem into two problems. The first problem is to assign all the customers to each day. The second problem is to solve the sub-problems in each period of the horizon. In this section, we propose a global method of construction the solutions for multi-period vehicle routing problem based on best insertion methods.

Insertion heuristics are popular methods for solving a variety of vehicle routing and scheduling problem. Insertion heuristics were first introduced for a travelling salesman problem and belong to a group of route construction algorithms (Campbell & Savelsbergh, 2004). Insertion heuristics construct a feasible solution, for example, a set of feasible routes, by repeatedly and greedily inserting unarranged customer into a partially constructed feasible solution. This approach is popular because it is fast, it produces decent solutions and can be easily extended to handle complication constraints.

The basic insertion heuristic for the standard vehicle routing problem has a time complexity of $\mathcal{O}(n^3)$ (Laporte *et al.*, 2000). The main principle of an insertion heuristic is to start from a single node that is usually called a seed node and that forms the initial route from depot. Other nodes are inserted one by one evaluating certain functions to a select node and the place in the route for insertion. The insertion heuristic approaches are categorized by the methods used for the node selection to be inserted: random insertion, nearest insertion, farthest insertion and cheapest insertion.

We proposed a method for a global construction of the solutions for our problems, based on best insertion (Solomon, 1987). The procedure is easy and straightforward. The method tries to insert the customer between all the edges in the current route. It selects the edge that has the lowest additional insertion cost. To adapt our addressed multi-depot and multi-period problem with time window, the feasibility check will be executed before each insertion for all constraints. Only feasible insertions will be accepted. We will present the adaptation that we made to the best insertion approach.

3.3.1 Adapt to Compatibility Between Requests and Period

What our problem studied in this dissertation mainly differs from the traditional VRPTW is that our problem is multi-period and multi-depot. In best insertion algorithm, we try to insert each customer into every tour so that we can find the insertion with minimum cost, which is the 'best insertion'. In our situation, it is essential to consider the constraint on the period for each request. There exist available periods for requests. In addition, a vehicle has not to execute a service tour in every period during the horizon. This incompatibility between certain requests and certain tours is a very strong constraint.

We call the pair of {vehicle-period} a resource. It represents the availability of one technician in one work period. Each resource can be used at most once. It means that each technician carries out no more than one tour every work day. We note the resource executed by the vehicle k in the period l R_l^k. It can be also called a tour. We define the compatibility between a tour and a customer request: the tour R_l^k is compatible with the customer request v if the period l is included in the available periods of the customer v. For a customer request v of which the available periods during the planning horizon is E, the list of compatible tour is the set $\{R_l^k | l \in E\}$. The method then extends very simply. In classic version of best insertion, we calculate the cost of insertion of a request into each tour. Here we consider only the insertions in its compatible tour. At each step, the least costly feasible insertion is carried out, and compatibility constitutes one of the feasibility criteria. In addition to this, there are two other criterion of feasibility:

- Respect of the time window.
- Non-exceeding of the maximum duration of a tour (working hours of a work day), which can be assimilated to a time window on the arrival point.

These two conditions should be confirmed before a feasible insertion.

3.3.2 Adapt to Priority of Customers

In our problem, there are two categories of customers: obligatory customers who are more important and optional customers who are less important. That is to say, missing an obligatory customer needs to pay more price than missing an optional customer. With the traditional best insertion method, if we insert first the optional customers, it is possible that the obligatory customers can not be inserted in its right period and time interval. However, the obligatory customers must be inserted. For not missing the critic requests, we propose a heuristic with two steps. Firstly, we insert the obligatory customers into the tours with the constructive heuristic based on best insertion. Secondly, we execute a second time insertion heuristic for the optional customers which are less constrained and easier to insert to complete the tours.

Algorithm 1 describe the pseudo-code of the method we use. An insertion is determined by for variables: customer, vehicle, period, location in a tour, $I = I(c, d, p, l)$. Before execute an insertion, its feasibility should be determined. We consider here the compatibility of period, the respect of time window and the duration when return to the depot.

Figure 3.1 shows the insertion steps of a problem with two depots (two vehicles) and the planning horizon consists two periods. In this figure, square represents the depot; circle in solid line represents the customer location that already inserted; dotted circle is the customer need to be inserted here and now. Step a shows the current solution which is not complete. Step b shows all the possible insertions among all arcs in current solution. After found the insertion with minimum cost, the unplanned customer is inserted to the planning tours in step c.

3.3.3 Algorithm for Construction

The algorithm of constructive heuristic is shown in Algorithm 1. $R(d_R, p_R)$ represents the resource of depot d_R and period p_R. Insertion(d, R) is the insertion to position d in tour R. To determine whether a solution is feasible, we need to verify: (1) all customers are served on their compatible period $q_i^l = 1$; (2) violating time of time window $tw(r) = 0$; (3) duration of each vehicle for each period $dt(r) < duration$ the maximum working time for a period. Insertions

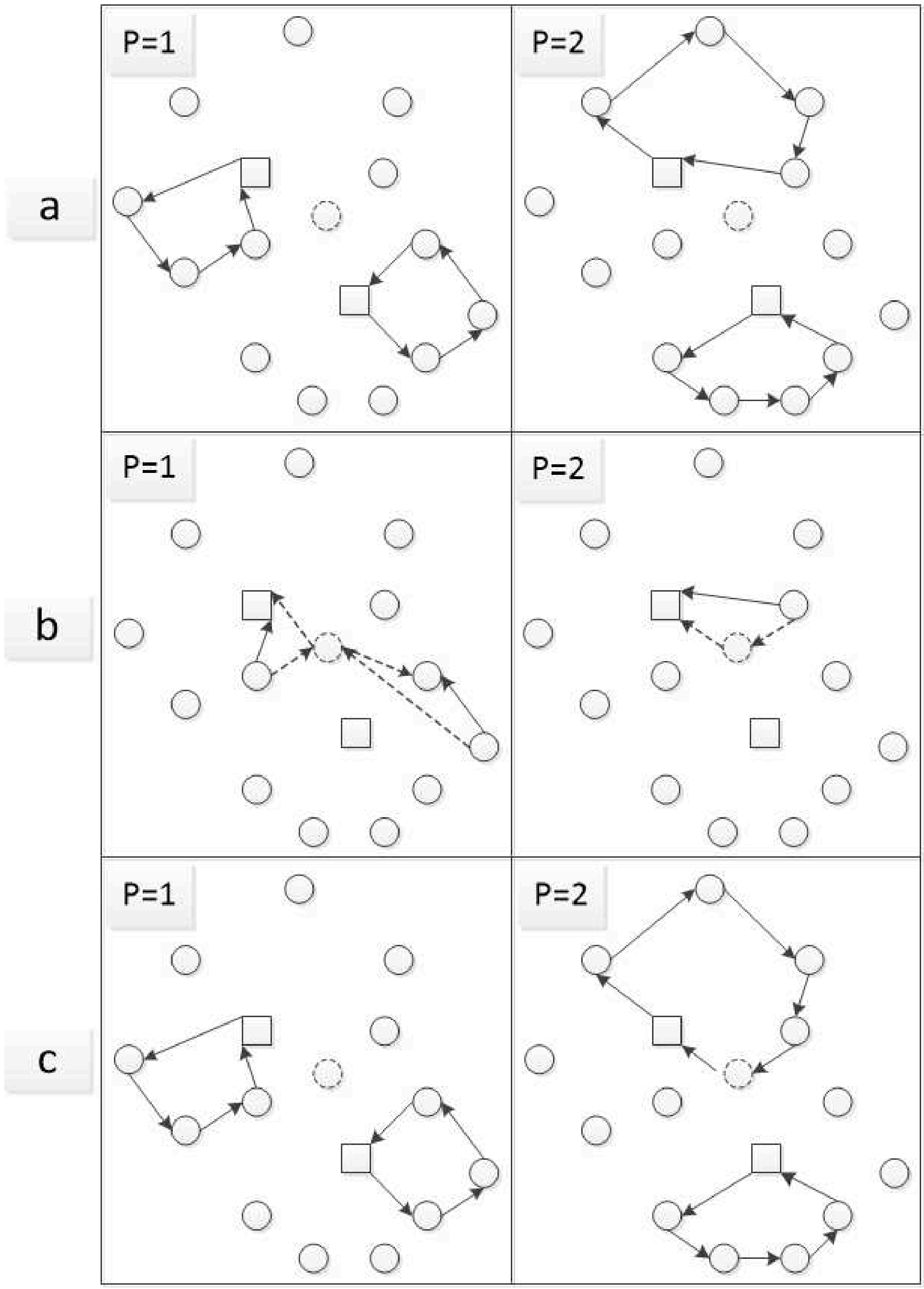

Figure 3.1: Best Insertion Process

which meet these conditions are feasible insertions. This procedure is executed two times: firstly, we insert the obligatory customers then we execute a second time insertion heuristic for the optional customer.

Algorithm 1 Heuristic of Construction Based on Best Insertion

```
A_u = A set of unassigned obligatory customers
Q_u = Q set of unassigned optional customers
while A_u ≠ ∅ do
    a = select a random customer from A_u
    BestInsertion(a) ← FirstFeasibleInsertion(a)
    BestCost ← Cost(BestInsertion)
    for a ∈ A_u do
        for R ∈ CompatibleResource(a) do
            for d location in R do
                if Cost(FeasibleInsertion(d, R))<BestCost then
                    BestInsertion(a) ← FeasibleInsertion(d, T)
                    BestCost ← Cost(BestInsertion)
                end if
            end for
        end for
    end for
    Execute BestInsertion of a
    A_u ← A_u − a
end while
Repeat the heuristic for Q_u
```

3.4 Heuristic for Improvement

Most constructive procedures are followed by an improvement phase. In this section, we define several kinds of neighborhood search operators, then we will use these neighborhood to compose the improvement heuristics.

Improvement heuristics for VRP operate on each vehicle route taken separately, or on several routes at a time. The improvement heuristics start from any feasible solution and improve it by successive small changes. We can see these improvements as a neighbourhood search process, where each route has an associated neighbourhood of adjacent routes.

3.4.1 Local Search Heuristic

An improvement heuristics is a compound heuristic, in which the improvement algorithm is the second phase of the overall compound heuristic. The neighbourhood of a given solution is the set of feasible solutions that are alike the given solution. For example, in TSP, it is possible to define as a neighbourhood of a given tour all the tours that can be generated from that one by applying a 2-opt iteration. For a multi-depot and multi-period VRP, there are more than one potential routes, so we will classify the neighborhood into two types: inter-route improvement and intra-route improvement.

3.4.1.1 Intra-route improvement

2-opt Search

Most improvement procedures can be described in terms of $\lambda - opt$ mechanism. λ edges are removed from the tour and the λ remaining segments are reconnected in all possible ways. This local search changes at most λ components of the solution. If any profitable reconnection (the first or the best) is identified, it is implemented. The procedure stops at a local minimum when no further improvements can be obtained. We choose the $2 - opt$ for the intra-route improvement. Checking the $\lambda - optimality$ of a solution can be achieved in $\mathcal{O}(n^{\lambda})$ time (Laporte, 1992). The neighbors considered by $2 - opt$ are not too large to compute. The computation time is reasonable. The $3 - opt$ is not suitable to our problem because the results of a $3 - opt$ change a lot of the direction and the sequence of the arcs in the routes. It is very easy to cause the violation to the time windows. In addition ,the running time will be substantially grow.

We use $2 - optSwap(route, i, k)$ to represent a swap of $2 - opt$. In this function, $route$ is the route whose neighbors we are searching, i and k are two arcs to exchange. Figure 3.2 gives a example of a 2-opt swap in which $i = 3, k = 5$. That means the exchange to change the arc C→D and arc E→F to arc C→E and arc D→F. On account of the direction of arcs, arc D→E is reversed to arc E→D. Each (i, k) determines a neighbour.

The algorithm of the $2 - opt$ neighbor search is presented in Algorithm 2. It is worth noted that feasibility check should be verified before every move is accepted. Otherwise, there will be violations to the constraints during the search process.

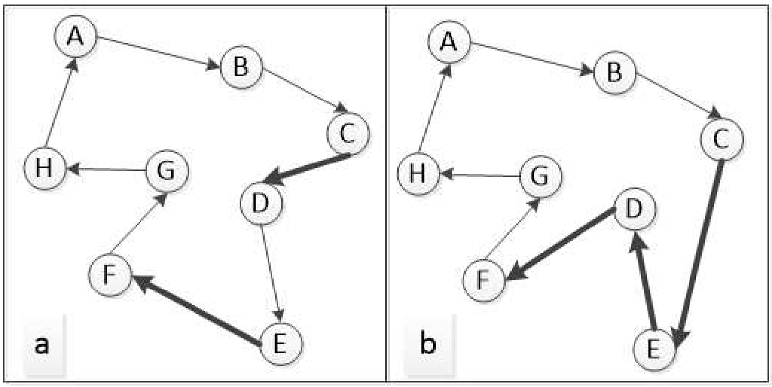

Figure 3.2: A 2-opt swap with $i = 3, k = 5$

Algorithm 2 2-opt Local Search Algorithm

repeat
    BestExchange ←FirstPossibleExchange
    BestCost ←Cost(BestExchange)
    for $route \in T$ **do**
        for $i \in arcs(T)$ **do**
            for $k \in arcs(T)$ **do**
                if 2-optSwap($route, i, k$) is a feasible move and Cost(2-optSwap($route, i, k$))<BestCost **then**
                    BestExchange ←2-optSwap($route, i, k$)
                end if
            end for
        end for
    end for
    if BestCost<CurrentCost **then**
        Execute BestExchange
    end if
until No more improvement is made

λ-interchange

λ-interchange mechanism was introduced for the capacitated clustering problem. It is based on customer interchange between sets of vehicle routes and has been successfully implemented. The local search procedure is conducted by interchanging customer nodes between routes. For a chosen pair of routes, the searching order for the customers to be interchanged needs to be defined, either systematically or randomly. We consider the case of $\lambda = 2$. For example, the operator (1,2) on routes (r_1, r_2) indicates a shift of two customers from r_1 to r_2 and one customer from r_2 to r_1. The other operators are defined similarly. For a given operator, the customers are considered sequentially along the routes. In both the shift and interchange process, only improved solutions are accepted if the moves results in the reduction of the total cost. There are two strategies to select between candidate solutions:

- The first-best (FB) strategy will select the first solution in $N_\lambda(S)$, the neighbourhood of the current solution, that results in a decrease in cost.

- The gloabl-best (GB) strategy will search all solutions in $N_\lambda(S)$, where $N_\lambda(S)$ means the neighbourhood of current solution under λ-interchange operation. GB will select the one, which will result in the maximum decrease in cost.

The procedure of the λ-interchange local search descent is shown in algorithm 3. The result is dependent on the initial solution and GB usually achieves better results than FB because it keeps track of all the improving moves but incurs more expensive computation time.

Algorithm 3 Local Search Descent Method

```
Obtain a feasible solution S.
repeat
    Select a solution S' ∈ N_λ(S).
    if Cost(S') < Cost(S) then
        Accept S' as S.
    end if
until Neighbourhood of S N_λ(S) has been completely searched
```

3.4.1.2 Inter-routes improvement

Van Breedam (1994) classifies the improvement operations as *string cross*, *string exchange*, *string relocation* and *string mix*. String cross exchanges two arcs from two different routes. Two strings of vertices are exchanged by crossing two edges of two different routes. This operation is not suitable to our problem because each vehicle has its own depart point and arrival point. String exchange exchanges the sequences of k nodes between two routes. In our problem, the constraint on the period require the nodes can only be exchanged to their compatible routes. String relocation transfer a sequence of k nodes from a route to another route and string mix selects the best possible movement between string exchange and string relocation. To evaluate these moves, two local improvement strategies are used: first improvement (FI) which consists of implementing the first move that improves the objective function and best improvement (BI) which evaluates all the possible moves and implements the best one.

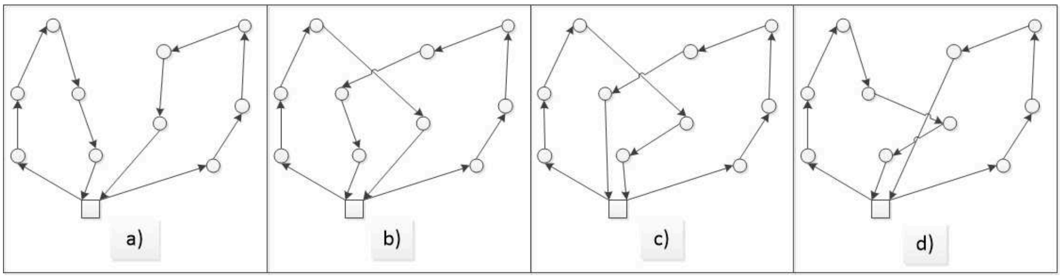

Figure 3.3: Operation Classification of Van Breedam

In (Kindervater & Savelsbergh, 1997), tours are not considered in isolation, so paths and customers are exchanged between different tours. The heuristics include *customer relocation*, *crossover* and *customer exchange*.

- A customer located at on route is changed to another one.
- Two routes are mixed at one point.
- Two customers of two different routes are interchanged between the two routes.

Figure 3.4, figure 3.5 and figure 3.6 show the heuristics of customer relocation, customer crossover and customer exchange.

3.4.2 Local Search Heuristics for MDMPVRPTW

We proposed a series of local search heuristics for the problem studied in this dissertation based on the local search heuristics discussed in the previous section. The heuristic is very essential for a fast progression toward high-quality solutions.

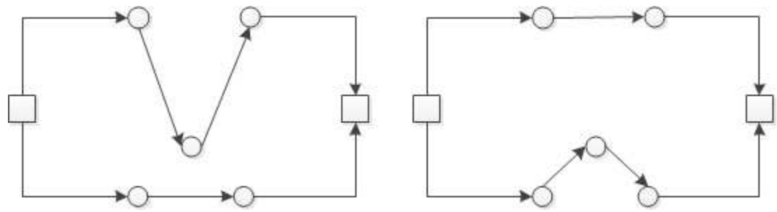

Figure 3.4: Customer Relocation

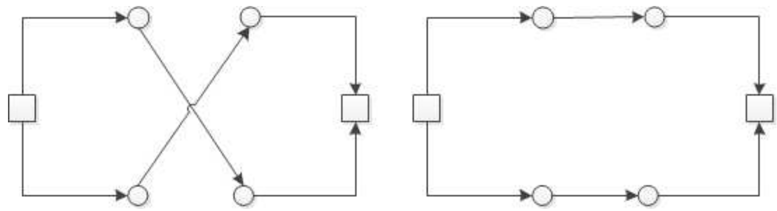

Figure 3.5: Customer Crossover

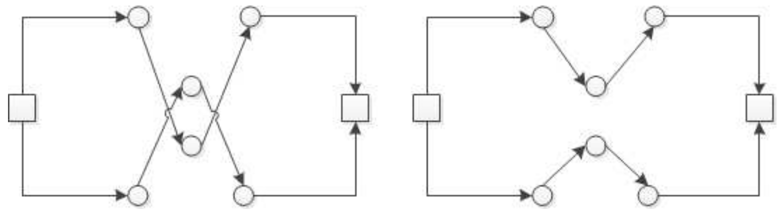

Figure 3.6: Customer Exchange

These procedures hold up the majority of the overall computational effort, such that high computational efficiency is required. We need a suitable choice of neighbourhood, restricted to relevant moves while being large enough to allow some structural solution changes.

Taking into account the specificity of our problem, in addition to the commonly used methods, we propose a procedure focus on the improvement of routes and periods. A routing plan of the problem is a set of routes for each vehicle-period combination. Let v be a customer request and $T(v)$ is the route containing v in the current solution. For customer v_1, let v_2 be a neighbor customer of v_1. The neighborhood of v_1 is defined as the gn closest customers of v_1. g is a granularity threshold which $g \in [0, 1]$ restricting the search to nearby customers. Let v_3 and v_4 represent respectively the successors of v_1 in $T(v_1)$ and v_2 in $T(v_2)$ if exist. We propose nine moves to apply on the solutions. Figure 3.13 to figure 3.15 shows the neighborhood searches in the routes.

- H1: If v_1 is a customer visit, remove v_1 and place it after v_2.
- H2: If v_1 and v_3 are customer visits, remove them, then place v_1 and v_3 after v_2.
- H3: If u and x are customer visits, remove them, then place v_3 and v_1 after v_2.
- H4: If v_1 and v_2 are customer visits, swap v_1 and v_2.
- H5: If v_1, v_2 and v_3 are customer visits, swap v_1 and v_3 with v_2.
- H6: If v_1, v_2, v_3 and v_4 are customer visits, swap v_1 and v_3 with v_2 and v_4.
- H7: If $T(v_1) = T(v_2)$, replace (v_1, v_3) and (v_2, v_4) by (v_1, v_2) and (v_3, v_4).
- H8: If $T(v_1) \neq T(v_2)$, replace (v_1, v_3) and (v_2, v_4) by (v_1, v_2) and (v_3, v_4).
- H9: If $T(v_1) \neq T(v_2)$, replace (v_1, v_3) and (v_2, v_4) by (v_1, v_4) and (v_3, v_2).

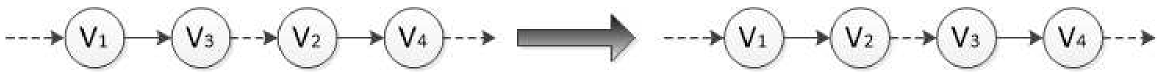

Figure 3.7: H1

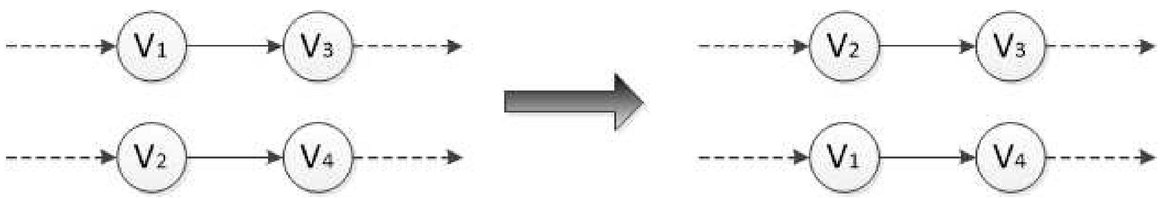

Figure 3.8: H2

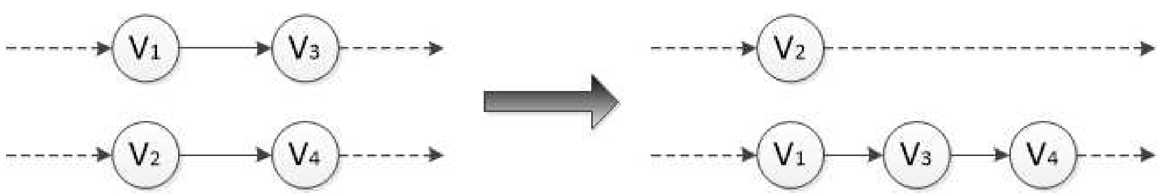

Figure 3.9: H3

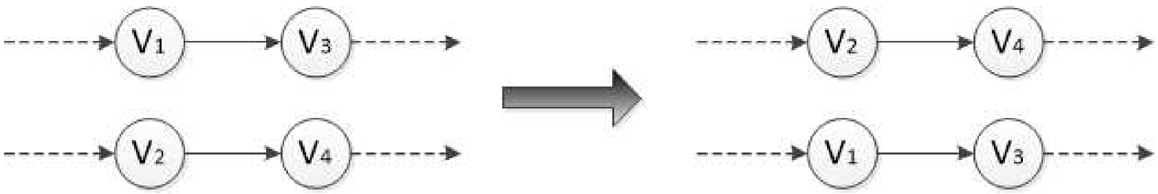

Figure 3.10: H4

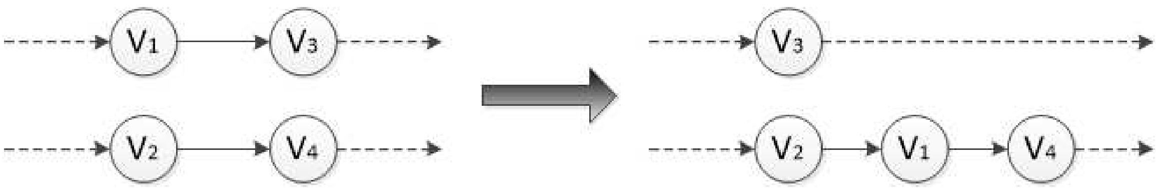

Figure 3.11: H5

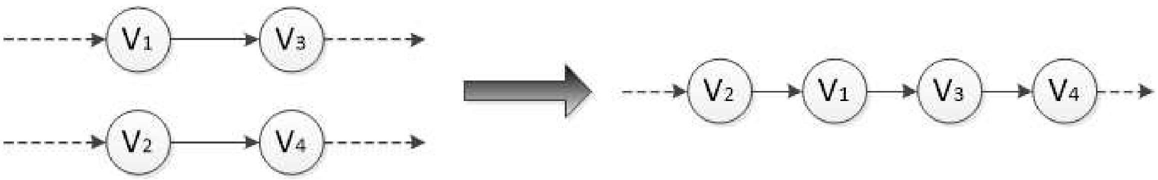

Figure 3.12: H6

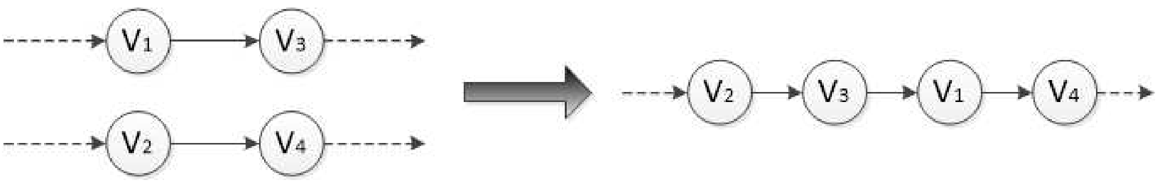

Figure 3.13: H7

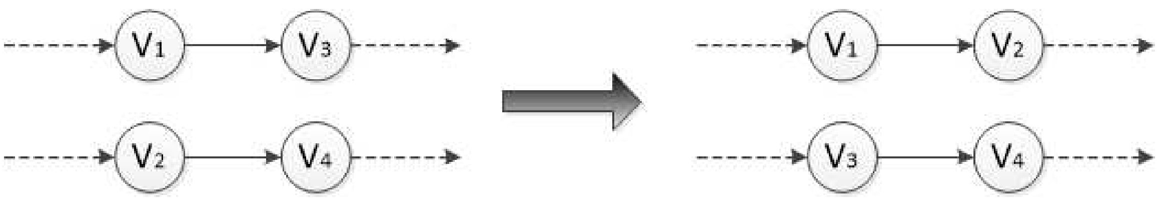

Figure 3.14: H8

The first three moves are insertions. H4 to H6 are swaps. H7 is the $2-opt$ intra-route search while H8 and H9 are $2-opt$ inter-route searches. The aim of these heuristics is to determine an efficient heuristic with reasonable calculation times for intensive use in metaheuristics. We will compare the different neigh-

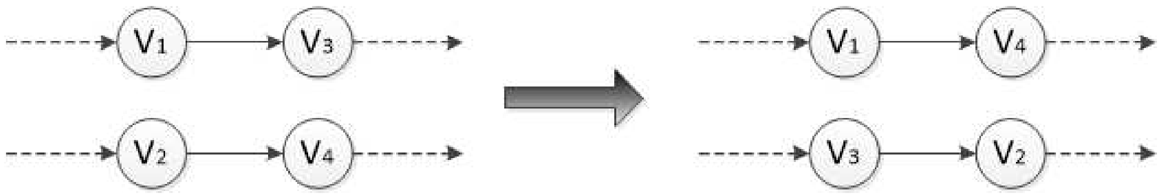

Figure 3.15: H9

borhood search methods and to determine if there exists one or several heuristics that perform better than others. The most efficient heuristics can be use very quickly (mode of emergency use in the company). Finally, obtaining more different solutions allows us to give several starting points to a metaheuristic, and thus to generate a population of different solutions of small size.

The aim of these various heuristics is to obtain good solutions in a rational computing time.

3.5 Experiment Results

We conducted the experiments to evaluate the performance of the methods in this chapter. The algorithms are implemented in JAVA and the experiments are executed on a processor Intel Core i5 1.8 GHz.

Tests will be done to the methodology of construction and the improvement heuristics we propose in this chapter.

3.5.1 Experimental Data

In this dissertation, we use the instances proposed in Tricoire (2006) since there are little data correspond to the multi-period, multi-depot vehicle routing problem with time window without capacity limits. In these instances, there are three vehicles and the planning horizon is consisted of 5 days. There are ten instances in total. The number of customers and the number of obligatory customers of each instance are shown in Table 3.1. The time horizon is consisted of 5 periods of 8 hours to simulate the 5 workdays during a week. The number of vehicles is 3. The number of potential routes is $numberofvehicles * numberofperiods$. In instance C1, there are 5 instances with 100 customers. About 50 of them are

Table 3.1: Number of different type of customers for each instance

Instance	All customer	Obligatory customer
C1_1	100	56
C1_2	100	44
C1_3	100	49
C1_4	100	48
C1_5	100	48
C2_1	180	90
C2_2	180	87
C2_3	180	80
C2_4	180	109
C2_5	180	88

Table 3.2: Duration of different types of service

Type of service	Percentage	Min duration	Max duration
1	20	10	20
2	20	20	20
3	20	15	60
4	15	5	15
5	10	10	20
6	10	15	45
7	5	30	30

obligatory customers with time window. In instance C2, there are 5 instances with 180 customers of which about half are obligatory customers.

The customers are located randomly on a square of 1000 kilometers. The distance between two nodes is the euclidean distance. The transport time is calculated by a factor 0.07 to the distances. This is because we set the speed of vehicles is 35 kilometers per hour. The service time is designed by the real situation in a water company. The type of service (numbered from 1 to 7), its percentage of total service and the duration are presented in Table 3.2.

The instances offer data of location, service time, periods pattern and time window of each nodes.

Table 3.3: Results of Cplex

	$N = 20$		$N = 30$		$N = 40$	
Instance	Cost	Time	Cost	Time	Cost	Time
C1_1	5438.69	47s	9623.31	539s	–	$> 24h$
C1_2	5776.17	126s	7878.29	2468s	–	$> 24h$
C1_3	5048.70	98s	7746.35	1335s	–	$> 24h$
C1_4	4850.00	60s	8665.4	803s	9144.34	$19h$
C1_5	5093.02	187s	8689.29	2386s	–	$> 24h$
C2_1	4894.90	91s	8678.75	3048s	–	$> 24h$
C2_2	5592.41	80s	9615.45	1229s	–	$> 24h$
C2_3	4465.35	86s	8896.28	1086s	–	$> 24h$
C2_4	4405.09	46s	7751.06	338s	9807.23	$22h$
C2_5	4844.56	55s	7112.50	241s	8543.31	$15h$

3.5.2 Results of Cplex

The model is implemented in the Java programming language use CPLEX with Concert Technology in the Eclipse. We do the experiments on new instances extracted from the ten instances above. The number of extracted customers N is 20, 30 and 40 for each original instances. Results are shown in Table 3.3.

We can tell from the results that for our addressed problem, it is not a good choice to use the mixed integer programming to solve it. For a instance of 20 customers, the optimal solution can be computed in a time around 100 seconds. For a instance of 30 customers, the computing time is up to 1 hour to get the optimal solution. We can also observe that on a regular computer, it is not capable to solve a problem with more than 40 customers in a acceptable running time. Taking into account the situation of the actual operation, the problem must be solved within one day to decide the schedule of the next week. In general, the number of customers for a enterprise is between 100 and 200 as in our instance C1 and C2. It is not possible to solve the problem with exact algorithm. As a result, we should ask for help in heuristics.

Table 3.4: Results of Randomly Generated Solutions

Instance	Random Construction			
	Cost	Time(ms)	Violation TW	Feasibility
C1_1	58717.3	25	10.2	0
C1_2	55949.4	25	9.38	0
C1_3	57657.3	27	11.55	0
C1_4	58328.1	22	12.37	0
C1_5	54297.5	27	12.13	0
C2_1	102150.8	46	77.17	0
C2_2	97626.2	50	84.04	0
C2_3	99204.9	46	77.05	0
C2_4	99561.7	56	80.8	0
C2_5	96973.9	50	80.36	0

3.5.3 Results of Construction Heuristics

We compare the results of randomly generation of solution and the algorithm with our proposed constructive approach. We run each algorithm 100 times and we compare the 100 solutions obtained from the two different methods. The results are shown in table 3.4 and table 3.5.

Table 3.4 are results of solutions randomly generated. We can tell that none of the 100 solutions are feasible. For instances in C1, the average number of time window violation is 11.13; for instances in C2, the number is up to 79.88, which means almost all customers with time window don not respect their time window. The more the customers are, the harder to find feasible solutions with a random method.

Table 3.5 shows the results of solutions generated by our proposed method based on best insertion. For instances in C1, 100% of solutions are feasible. For instances in C2, the average number of feasible solutions out of 100 runs is 54. For an instance with 100 customers, out method can guarantee to find a feasible solution. For an instance with 180 customers, the number of feasible results varies a lot. But for sure, we can get adequate feasible solutions by multiple runs of the algorithm. Moreover, costs in table 3.5 are far less than those in table 3.4. The running time of method based on best insertion is greater than that of random

Table 3.5: Results of Solutions with Adapted Best Insertion

Instance	Best Insertion			
	Cost	Time(ms)	Missed Customer	Feasibility
C1_1	23035.4	255	0	100
C1_2	20652.4	271	0	100
C1_3	20662.6	270	0	100
C1_4	22305.2	271	0	100
C1_5	19475.6	267	0	100
C2_1	41101.0	794	1.32	48
C2_2	38361.1	769	2.69	29
C2_3	38581.9	786	1.11	60
C2_4	41166.6	746	1.63	38
C2_5	37120.1	757	0.08	96

method by about 10 times. Considering that the total computing time is very fast, these gaps are not big problems.

3.5.4 Results of Improvement Heuristics

Tests will be done to the 9 different local search heuristics proposed in this chapter. The results are shown in table 3.6 and 3.7. We compare the computing time and the best results of the different heuristics. All the heuristics can improve the result in a short running time. We generate 100 feasible solutions with the proposed constructive heuristic proposed in this chapter and execute for each solution the nine heuristics we proposed one by one. We compare the average costs of each heuristic. Table 3.6 shows the average costs of each heuristics. H1 performs better and obtained better results in 5 of the ten instances. Other heuristics perform average. In table 3.7, we can tell that H1 is the most efficient heuristic among the nine methods. However, there is no dramatic differences between all the heuristics, hence they all may be apply to the metaheuristics in the following chapter.

Table 3.6: Costs of Local Search

Instance	Travel Cost								
	H1	H2	H3	H4	H5	H6	H7	H8	H9
C1_1	20407.69	20412.39	20442.95	20443.11	20412.39	20412.39	20443.11	20627.99	20627.99
C1_2	18067.53	18008.43	18034.49	18134.73	18063.93	18008.43	18134.73	18204.51	18089.69
C1_3	18096.1	18129.17	18213.94	18108.28	18091.54	18108.28	18236.4	18108.28	18108.28
C1_4	19571.15	19587.33	19514.38	19552.95	19568.86	19540.26	19552.95	19823.07	19759.37
C1_5	16984.52	17039.09	17029.57	17029.57	17081.86	17081.05	17029.57	17081.05	17097.68
C2_1	35223.49	35064.6	35255.85	35230.74	35197.33	35230.74	35262.27	35230.74	35230.74
C2_2	33032.86	33130.62	33032.86	33032.86	32997.44	33036.23	33032.86	33105.07	32942.01
C2_3	32816.34	33225.34	33060.05	33040.12	33036.28	33148.64	33040.12	33229.5	33210.08
C2_4	36238.38	36245.59	36006.33	36009.83	36155.86	36245.63	36046.84	36009.83	36030.73
C2_5	32046.37	32306.64	32319.99	32122.79	32181.6	32122.79	32340.27	32122.79	32122.79

Table 3.7: Running Time of Local Search

Instance	Running Time(ms)								
	H1	H2	H3	H4	H5	H6	H7	H8	H9
C1_1	10626	10948	10970	10617	12084	11697	12694	11610	12011
C1_2	9902	10365	10426	9742	11247	10840	12608	10940	11324
C1_3	10430	10767	10862	10283	11743	11261	12736	11282	11675
C1_4	9748	10285	10284	9911	11492	11056	12629	11074	11396
C1_5	10303	10704	10734	10212	11750	11306	12510	11261	11724
C2_1	30035	33093	32737	33842	38298	39503	42561	38652	40548
C2_2	28388	31336	31075	31574	35725	35702	36456	34558	39076
C2_3	34581	37794	37450	39377	44131	41994	50096	46039	46572
C2_4	28861	31816	31427	31980	35695	35922	36460	34467	36747
C2_5	50447	54988	53632	57480	64989	59663	79689	66400	64324

3.6 Conclusion

In this chapter, we formulated the problem of field service routing. We presented a family of heuristics for construction and improvement of the multi-period and multi-depot field service routing problem with time window. Experiments are done to test the algorithms. Construction heuristic is capable to find feasible solution with good cost for the two instances. The heuristics of improvement are compared and all of them are capable to find better solutions. These heuristics will be used in the following chapters.

Chapter 4

Genetic Algorithm for Solving Multi-period and Multi-depot Vehicle Routing Problem with Time Window

Contents

4.1 Introduction

GA is a metaheuristic inspired by the process of natural selection. It belongs to the larger class evolutionary algorithms. We will give a general introduction of genetic algorithm.

Genetic algorithm is a metaheuristic inspired by the process of natural selection. It is one of the best ways to solve a problem for which little is known. They are a very general algorithm and so will work well in any search space. Stadler (2013) gives a detailed presentation of all heuristics and metaheuristcs methods for optimization in engineering science including the GA.

In GAs, we have a pool or a population of possible solutions to the given problem. These solutions then undergo recombination and mutation (like in natural genetics), producing new children, and the process is repeated over various generations. Each individual (or candidate solution) is assigned a fitness value (based on its objective function value) and the fitter individuals are given a higher chance to mate and yield more "fitter" individuals. This is in line with the Darwinian Theory of "Survival of the Fittest".

In this way we keep "evolving" better individuals or solutions over generations, till we reach a stopping criterion.

Genetic Algorithms are sufficiently randomized in nature, but they perform much better than random local search (in which we just try various random solutions, keeping track of the best so far), as they exploit historical information as well.GA is an algorithm that have the ability to deliver a "good-enough" solution "fast-enough". This makes GA attractive in solving optimization problems.

Now it is essential to know some basic component for a GA.

- **Population** - It is a subset of all the possible solutions to the given problem. It is composed of candidate solutions.
- **Chromosomes** - A chromosome reprensents one solution to the given problem.
- **Gene** - A gene is one element position of a chromosome.
- **Fitness Function** - A fitness function is a function which takes the solution as input and produces the suitability of the solution as the output. In some cases, the fitness function and the objective function may be the same, while in others it might be different based on the problem.
- **Genetic operators** - The genetic operators alter the genetic composition of the offspring, including crossover, mutation, selection, etc.
  - **Selection** which equates to survival of the fittest;
  - **Crossover** which represents mating between individuals;
  - **Mutation** which introduces random modifications.

To execute a GA, we start with an initial population (which may be generated randomly or seeded by other heuristics), select parents from this population for mating. Apply crossover and mutation operators on the parents to generate new off-springs. Finally, those off-springs with high quality replace the existing individuals in the population. The procedure is shown in figure 4.1. A generalized pseudo-code for a basic GA is explained in algorithm 4.

4.2 Related Work

A lot of researches have been done to solve VRP variants, especially focused on the practical usage of the real world problems. Results from these researches show that compared to other metaheuristic algorithms, GA has advantages in both aspect of performance and final result on time constraints and limited compute ability. There exist some other metaheuristics able to find better solution than

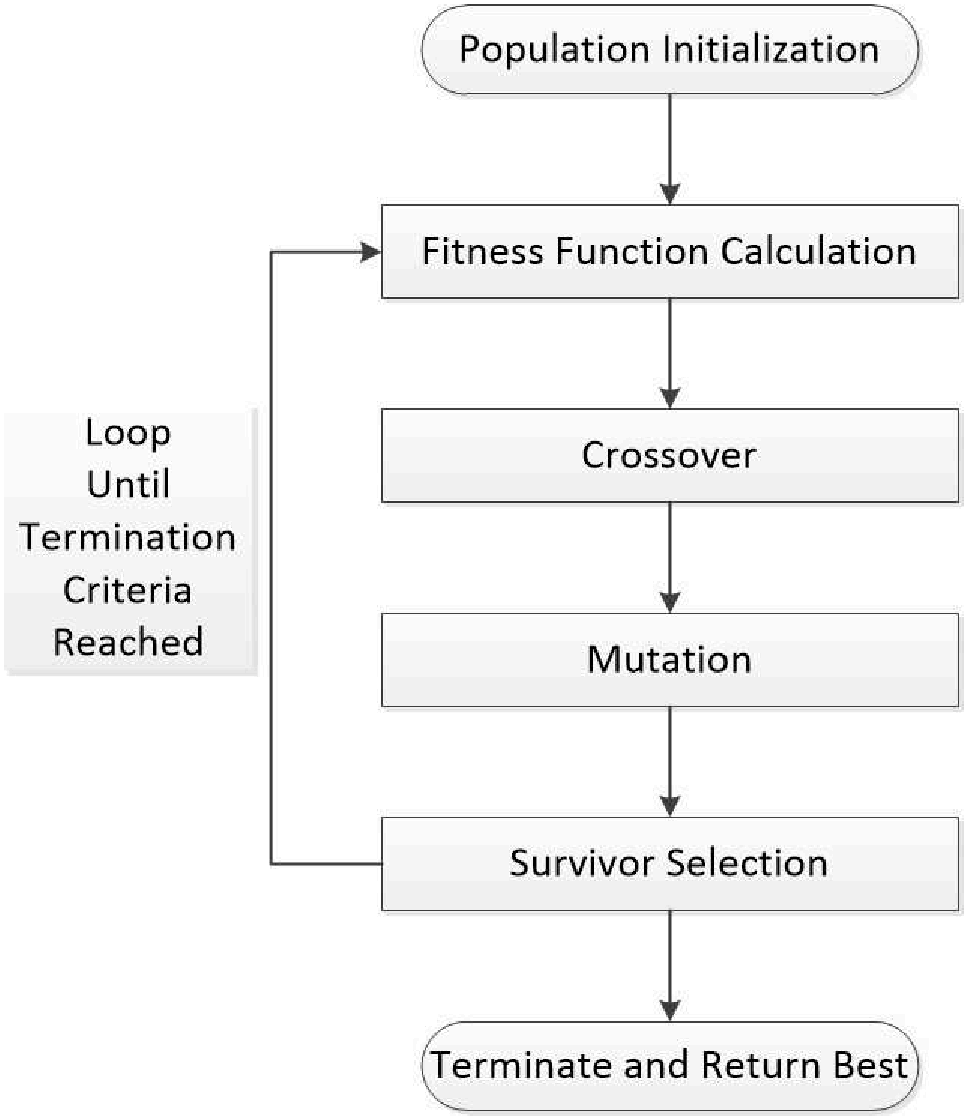

Figure 4.1: Procedure of Basic Genetic Algorithm

GA. However, GA make a balance between the solution quality and computing time.

(Vaira & Kurasova, 2014) proposed a genetic algorithm based on insertion heuristics for the vehicle routing problem with constraints. The author uses random insertion heuristic to get initial solutions and to reconstruct the existing ones. (Tasan & Gen, 2012) proposes a GA based approach to solve the VRP with simultaneous pick-up and deliveries. A genetic algorithm for the multi-compartment vehicle routing problem with continously flexible compartment sizes

Algorithm 4 Basic Genetic Algorithm

```
Initialize population
Find fitness of population
while termination criteria is not reached do
    Select parent solutions P1 and P2
    Create offspring C from P1 and P2 (crossover) with probability pc
    Mutate C with probability pm
    Fitness calculation
    Survivor selection
    Find best
end while
Return best solution
```

is proposed in (Koch *et al.*, 2016). The paper (Lau *et al.*, 2010) deals with the optimization of VRP of multi-depot, multi-customers and multi-products. They propose a stochastic search techinique to dynamically adjust the crossover rate and mutation rate after a certain generations.

VRPTW is one of the most studied VRP variant. Time window is a relative strong constraints in VRP. Genetic algorithm is frequently used to solve these sort of VRP thanks to its ability to always find feasible solution. A comparison of various GA for solving VRPTW before the year 2001 (Bräysy & Gendreau, 2001) is presented. Genetic algorithms for multi-objective VRPTW are discussed in(Ombuki *et al.*, 2006) and (Ghoseiri & Ghannadpour, 2010). The former takes advantage of the Pareto ranking techinique for evaluating the solution in a balanced way. The latter presents a new model and combined goal programming with genetic algorithm.

An other variants of VRP: multi-depot VRP is solved by a lot of adapted GA. (Karakatič & Podgorelec, 2015) presents a survey of GA designed for solving multi-depot vehicle routing problem by evaluating the efficiency of different existing genetic methods and between GA and other approaches, including both exact algorithms and heuristics for solve benchmark problems. (Thangiah) describe a GA heuristic for solving VRPTW called GIDEON who consists of a global customer clustering method and a local post-optimization method.

4.3 Genetic Algorithm for MDMPVRPTW

As mentioned earlier, GA has already been used to solve various VRP. However, the existed algorithms can not be applied to solve our problem. Firstly, this problem is a problem with multi-period and multi-depot, it is important to find a suitable chromosome representation. The coding structure plays a crucial role in the GA and consequently, this may have a profound impact on the performance. Secondly, the determination of initial population and search space is worth of discussion. Because the constraints in our problem are very strong, it is impossible to use random initial solutions for searching. At last, the design of crossover operator is a key to improve the genetic search quality.

We proposed a new genetic algorithm to our addressed problem in this dissertation. This section gives a description of the proposed genetic search for the problem. Then we provide details of the initial population, solution representation, fitness evaluation and the operators used in the GA.

4.3.1 Overview of the Proposed Genetic Search Algorithm

The framework of the method of proposed GA is shown in Algorithm 5. The initial population is constituted with feasible solution. The survivors to the new generation has three sources:(1) elite member of original population (called elitism), (2) offspring obtained with crossover and (3) new solutions to add diversity to the new population. The ratio of each part is determined by the parameters:

Table 4.1: Parameters for Selection of Survivors

Parameters	Behavior
Popsize	number of individuals of new population
Elitnumber	number of elite individuals to new population
CrossoverProportion	proportion of individuals obtained by crossover
CrossoverNumber	number of individuals obtained by crossover
MutationProbability	rate of mutation
NewSolutionNumber	number of new generated solutions to new population

Popsize is the size of population; *Elitnumber* for the number of elite members and $CrossoverNumber$ for the number of crossover offspring and the rest are the new solutions. $CrossoverProportion$ is the proportion of crossover offspring in the new population and $CrossoverNumber = Popsize * CrossoverProportion$. The terminate criteria may be a running time, a certain number of generations which the best solution does not improve or a maximum number of iteration, etc.

Algorithm 5 Genetic Algorithm for MDMPVRPTW

Initialize population with feasible solution: *pop*
Find fitness for all individuals in *pop*
while *Terminate criteria is not met* **do**
Elitism
Find *Elitenumber* best solutions of *pop*
Insert the selected best solutions to new population: *newpop*
Crossover
$number\ of\ crossover = 0$
for $number\ of\ crossover\ < Cnum$ **do**
Select parent solutions P_1 and P_2
Create offspring C from P_1 and P_2 by crossover operator
Generate a random number in $(0, 1)$
if $random(0, 1) < MutationProbability$ **then**
Mutate C
end if
if C is feasible **then**
Insert C to *newpop*
$number\ of\ crossover\ ++$
end if
end for
New Solutions
for $number\ of\ new\ solution\ < NewSolutionNumber$ **do**
$newSolution \leftarrow$ generate new feasible solution
Insert $newSolution$ to *newpop*
$number\ of\ new\ solution ++$
end for
end while
Return solution with best fitness *Best Solution*

4.3.2 Search Space

There are two situations for the determination of search space. One is to explore only the set of feasible solutions that satisfy the constraints; the other is to allow a controlled exploration of infeasible solutions that may violent some of the constraints. Researches (Glover & Hao, 2011) show that the latter may enhance the performance of the search which may more easily transit between structurally different feasible solutions. However, in our problem, the constraints are too strong that if unfeasible solutions are allowed in the population, we need spend much time to repair it. Therefore, we decide that only feasible solutions are permitted to the population. Most of existed methods for a multi-period VRP divide the problem into two problems: assign all customers to each period then solve the mono-period VRP. Customers are distributed to a certain period or a certain vehicle before the optimization. Then the problem can be seen as a multi-traveling salesman problems. The disadvantage of this method is obvious. The final solution depends largely on the initial clustering. As a consequence, the range of searching is very limited. In this dissertation, we propose a method of global search. The routes of all vehicles and all periods are seen as a entirety. We have introduced our method of a global construction for initial solutions based on best insertion in chapter 3. These solutions are served as initial solutions to the genetic search.

In summary, in the genetic algorithm, we define the search space $\mathcal{S}$ as a set of feasible solution $s \in \mathcal{S}$. Let s be a solution found in the search space. $\mathcal{R}(s)$ represents the set of routes making up s. Each route $r(d,p) \in \mathcal{R}(s)$ represents a tour of a (depot-period) $= (d,p)$, starting from a depot $v_0^r \in O$, visiting a sequence of s_r customers $v_1^r, v_2^r, ..., v_{s_r}^r \in N$ then returning to the same depot o^r. A resource $r(d,p)$ can be seen as a potential route for the routing problem. It is available during the optimization process.

Figure 4.2 shows a simple graphical model of a solution. In this solution, there are three depots and two periods which makes it six routes to complete the plan. There are six resources $r(1,1)$, $r(2,1)$, $r(3,1)$, $r(1,2)$, $r(2,2)$ and $r(3,2)$.

Let $g^r = (g_0^r, ..., g_{v_{s+1}}^r)$ be the arrival times at each customers', $b^r = (b_0^r, ..., b_{v_{s+1}}^r$ be the begin time of service of each customer request. A route implicitly specifies the earliest possible arrival time g_v^i, as well as the earliest possible service time $b_{v_i}^r$

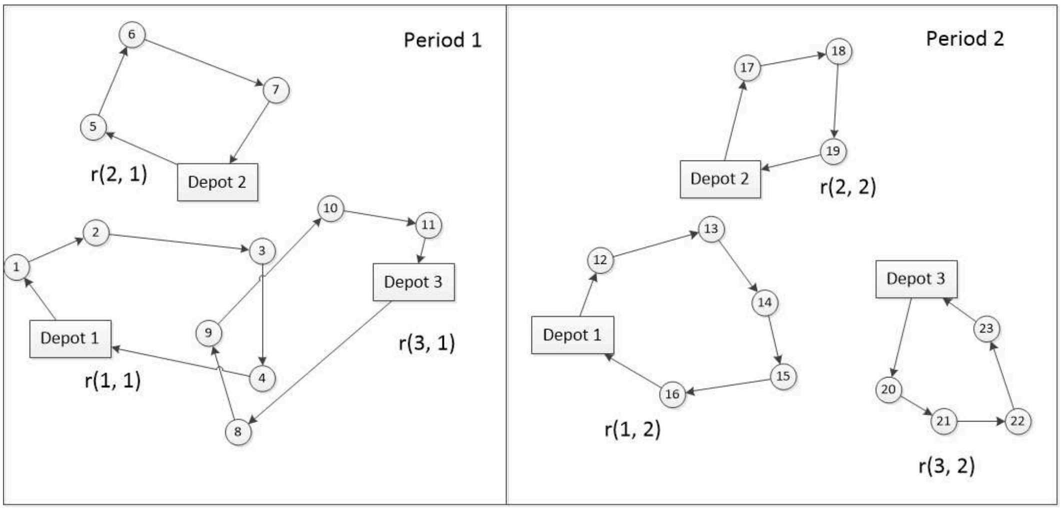

Figure 4.2: Example of a Routing Solution of VRP with Multi-depot and Multi-period

of each customer i. These values are computed using simple recursive equations.

$$b^r_{v_i} = \begin{cases} g^r_{v_i} & if \;\; g > e_{v_i} \\ e_{v_i} & if \;\; g \geq e_{v_i} \end{cases} \qquad\qquad g^r_{v_{i+1}} = b^r_{v_i} + \sigma^r_{v_i} + \tau^r_{v_i v_{i+1}} \tag{4.1}$$

If the vehicle arrives at a customer earlier than the lower bound of its time window ($g_{v_i} < e_{v_i}$), there will become a waiting time during which the vehicle cannot do other jobs but waiting. This waiting time is given by $\tau\omega_{v_i} = max\{e_{v_i} - g_{v_i}, 0\}$.

4.3.3 Chromosome Representation

The scheme for genetic representation of the solution albeit the chromosome coding structure plans a crucial role in the GA. Consequently, this may have a profound impact on the algorithm's performance.

In order to apply the genetic algorithm to a particular problem, the solution should be represented as an internal string. The choice of this component is one of the critical aspects to the success of the algorithm. Study compares three common representations of vehicle routing problem (Xu *et al.*, 2005).

Our problem is particular on account of its multi-period and multi-depot character. A representation of two-chromosome individual without trip delimiters is adopted by (Vidal *et al.*, 2013) for the solution of vehicle routing problem with time windows. The individual chromosome is shown in Figure 4.3. The representation of the routes out of the same period and depot as a giant tour provides the means to use simple and efficient crossover procedures working on permutations, but requires an algorithm to find the optimal segmentation of the tour into routes and, thus, retrieve both the solution and its cost. To obtain a full solution from this representation, it needs a *Split* algorithm and the shortest path problem of each route can be solved in $\mathcal{O}(mn^2)$ time.

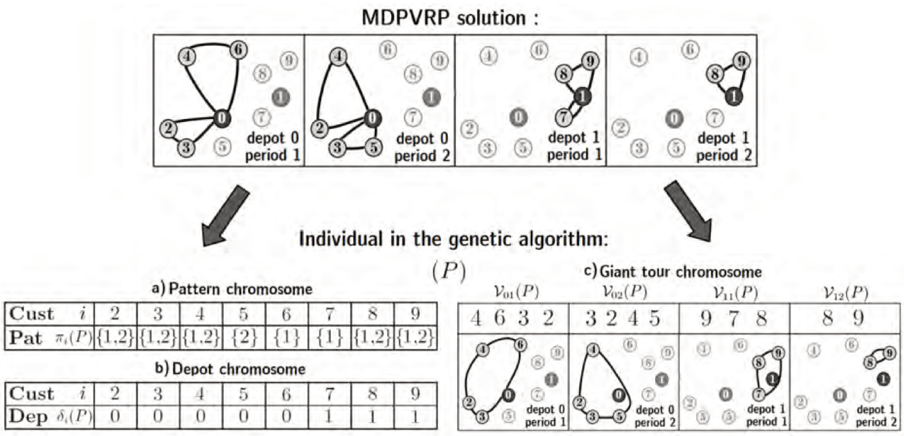

Figure 4.3: Individual chromosome representation of Vidal *et al.* (2013)

In this thesis, for a problem with d depots and p periods, p sections chromosomes are used to give a representation genetic of each solution. An example is given in figure 4.4. There are 11 customer requests numbered from 0 to 10. The codes d1 and d2 represent the two depots. First section of chromosome represents visit sequence of all vehicles on period 1. Requests coded as 6 and 10 of section (d1,p2) are distributed to the vehicle d1 in period 1. In a generalized way, the numbers after a depot gene d_1 is the visit sequence of vehicle d_1 before meeting another gene of depot.

This form of representation is intuitive. It is divided into sections of different period which makes it possible to insert customer requests only into location in their compatible period during initialization and evolution of the genetic search.

4.3.4 Population Initialization

The initial population is frequently randomly generated, and this initial 'seed' cannot, however, contain invalid individuals, which means that we must find a way

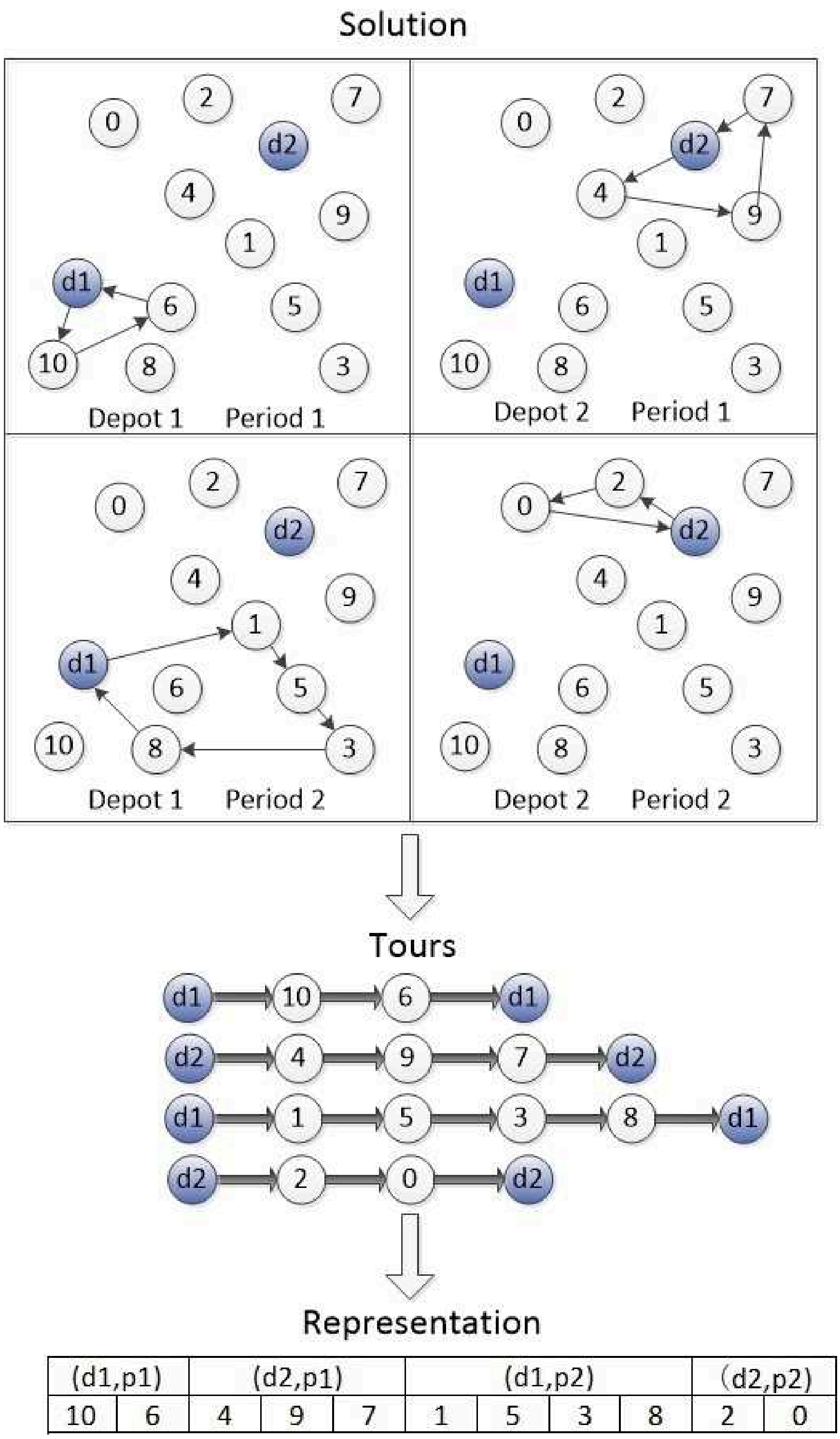

Figure 4.4: Chromosome representation of a problem with 2 depots and 2 periods

to generate a solution which is both random and compliant with the constraint.

Our problem is complicated because of there are a number of constraints, especially on the aspect of time. The constraints on time window, duration and valid period are too strong. Solutions randomly generated are hardly feasible. It takes much computing time to repair them. The only way to solve this is to generate the feasible solutions in the first step. In this way, the initial solution are feasible or approximately feasible. This initial solutions with high quality improve the possibility to find good solutions with high efficiency of the genetic search.

In our genetic algorithm, we generate the initial solutions with the method proposed in chapter 3. We accept only the feasible solutions. Infeasible solutions are abandoned and continue to execute again the adapted best insertion method to construct a new solution until the population size is reached.

4.3.5 Evaluation and Selection Based on Fitness

The fitness function allows us to identify the value of each individual. The individual with the best fitness value is treated as the best currently known solution in the population. In our proposed GA: the travel time is in direct proportion to the travel cost; the infeasible solutions are not accepted to the population; the number of vehicles is fixed; the objective of optimization is consequently unique: to minimize the travel cost. Therefore, the fitness function is defined as 4.2, the sum of the travel cost of all vehicles during the whole horizon.

$$f(s) = \sum_{v_i \in V} \sum_{v_j \in V} \sum_{k=1}^{m} \sum_{l=1}^{w} c_{ij} x_{ij}^{kl} \tag{4.2}$$

Parent selection is performed through a binary tournament, which twice randomly (with uniform probability) picks two individuals from the complete population and keeps the one with the best fitness.

In tournament selection, a number *Tour* of individuals is chosen randomly from the population and the best individual from this group is selected as parent. This process is repeated as often as individuals must be chosen. These selected parents produce uniform at random offspring. The parameter for tournament selection is the tournament size $Toursize$. $Toursize$ takes values ranging from 2 to $PopulationSize$. The loss of diversity is defined as (Blickle & Thiele, 1995):

$$LossDiv(Tour) = Tour^{\frac{-1}{Tour-1}} - Tour^{\frac{-Tour}{Tour-1}} \tag{4.3}$$

According to this formula, about 50% of the population diversity are lost if the tournament size $Toursize = 5$.

Apart from the crossover offspring, we keep the elite members of previous population to the new population. By using this technique, the elite of the current population is transferred to the next generation. The elite population includes a number $N_e < N$ of individuals with high fitness that are not involved in reproduction. The exploration-exploitation trade-off is extremely sensitive to the elite size N_e. Moreover, N_3 has to be chosen with care for each application, since a general rule does not exist. It is wise to keep at least the best encountered individual in the elite, for it could be the global optimum itself(Stefanoiu *et al.*, 2014).

4.3.6 Crossover

The performance of a GA very depends on the crossover operator. It is analogous to reproduction and biological crossover. More than on parent is selected and one or more off-springs are produced using the genetic material of the parents. It is usually applied in a GA with a high probability. The meaning of crossover is to preserve the genotype of good parents and to improve the possibility of obtaining good off-springs in next generation. Many crossover operators are proposed to the VRP, such as *Partially Matching Crossover*(PMX), *Cycle Crossover*(CX), *Order Crossover*(OX1), *Order Based Crossover*(OX2), *Position Based Crossover*(POX), *Alternating Position Crossover* (APX), etc (Potvin & Bengio, 1996).

These crossovers are operators to a simple chromosome representation. For a problem of multi-period and multi-depot, the crossover could be more complicated. We propose two crossover operators. One is crossover with insertions, the other one is partially matching crossover based on resources. The chromosome is represented by a number of segments for each vehicle-period. During the crossover, it is important to ensure the maximum guarantee of feasibility and inherit genotype characteristics from both parents. We proposed a crossover methods dedicated to routing problems with multi-depot and multi-period. It is designed to transmit good sequences of visits, while enabling the recombinations between customers and depots and periods. We aimed for a versatile crossover, which would allow for both a wide exploration of the search space and small refinements of good solutions. The possibility for the offspring to inherit genetic material from its parents in nearly equal proportions. It avoids copying most genotype of one parent and small part of genes of another. To ensure that the crossover rules are not determined a priori on how much genetic material the offspring inherits from each parent, the cut-off points are randomly selected.

Crossover Operator PIX

The tours are divided into three sets: one set mainly inherits from parent P_1, one set mainly inherits from P_2 and the other set inherits both parents with a certain rule. The procedure of crossover is shown in algorithm 6. First of all, the $|K|*|L|$ tours are numbered from 0 to $|K|*|L|-1$. n_1 and n_2 are random numbers that $n_1, n_2 \in [0, |K|*|L|-1]$ and $n_1 < n_2$. Randomly select n_1 tours to form a set X_1, n_2 tours to form a set X_2 and put the remaining tours into a set X_3. Tours in X_1 will transmit from parent P_1 to child C. Tours in X_2 will transmit from parent P_2 to child C. Tours in X_3 will be a mixed tour transmit from both parent P_1 and P_2 to child C. The whole crossover have four steps: inherit from P_1, inherit from P_2, combine genotype of P_1 and P_2 and complete the offspring.

- Inherit from P_1: all tours in X_1 are copied to the child C in the tour of the same (vehicle, period).

- Inherit from P_2: all tours in X_2 are copied to the child C in the tour of the same (vehicle, period). It is important to skip the customers that already exist in the child.

- Combine genotype of P_1 and P_2: for each tour in X_3, a delimiter is generated randomly to divide the tours into two segments. The first segment inherits from P_1, the second segment inherits from P_2. It is important to skip the customers that already exist in the child.

- Complete the offspring: examine if all customers are contained in the child. If not, insert the customer with the method best feasible insertion.

An example is presented in figure 4.5. In this example, there are 14 customers coded from 1 to 14. $X_1 = \{(d1, p1)\}$, $X_2 = \{(d1, p2)\}$, $X_3 = \{(d2, p1), (d2, p2)\}$. In step 1, genes in tour $(1, 1)$ of parent P_1 is copied to tour $(1, 1)$ of child. In step 2 targets the parent P_2 and we want to copy all tours in X_2 from parent P_2 to child. The sequence of tour $(d2, p2)$ is {9,11,13,4}. Since customer 9 is already exist in the child (tour (d1,p1)), we skip this customer and continue to copy next customers 11,13 and 4. In step 3, the tours are dealt in a random order. We assume that we started from tour (d1,p2). The delimiter generated is 2, we therefore copy the first two customers to child after examined their existences. The delimiter of tour (d2,p1) is randomly selected as 1. The first customer of parent P_1 is 4. However it has already been in the child. As a result, none of genotype of tour (d2,p1) inherits from parent P_1. The tour (d2,p1) of P_2 yields only the subsequence {10} out of {3,8,10} to child, because a visit to customer 8 was copied during the first step. Step 4 find out all customers that are not in child and insert it to the offspring with feasible best insertion method.

Algorithm 6 Crossover PIX

Pick two random numbers $n_1, n_2 \in [0, m * w]$ according to a uniform distribution. Let $n_1 < n_2$
Randomly select n_1 tours to form a set of tours X_1;
Randomly select $n_2 - n_1$ resources in the remaining tours to form a set of X_2;
The remaining tours form the set X_3.
Inheritance from P_1
for Each tour (k,l) belonging to set X_1 **do**
    Copy the sequence of customer visits from $T_(k,l)(P_1)$ to $T_(k,l)(C)$
end for
for Each tour (k,l) belonging to set X_2 **do**
    Randomly select two chromosome-cutting points α_{kl}, copy the genes before α_{kl} from $T_(k,l)(P_1)$ to $T_(k,l)(C)$
end for
Inheritance from P_2
for Each tour (k,l) belonging to set X_2 **do**
    for Each customer v in $T_(k,l)$ **do**
        if v does not exist in C **then**
            Insert v into $T_(k,l)(C)$
        end if
    end for
end for
for Each tour (k,l) belonging to set X_3 **do**
    for Each customer of index $(0, \alpha_{kl})$ and (β_{kl}, end) **do**
        if v does not exist in C **then**
            Copy v to from $T_(k,l)(P_2)$ to $T_(k,l)(C)$ with respect to the sequence
        end if
    end for
end for
Complete the child soluton
for each $v \in V$ **do**
    if v does not exist in C **then**
        Insert v to C with best insertion
    end if
end for

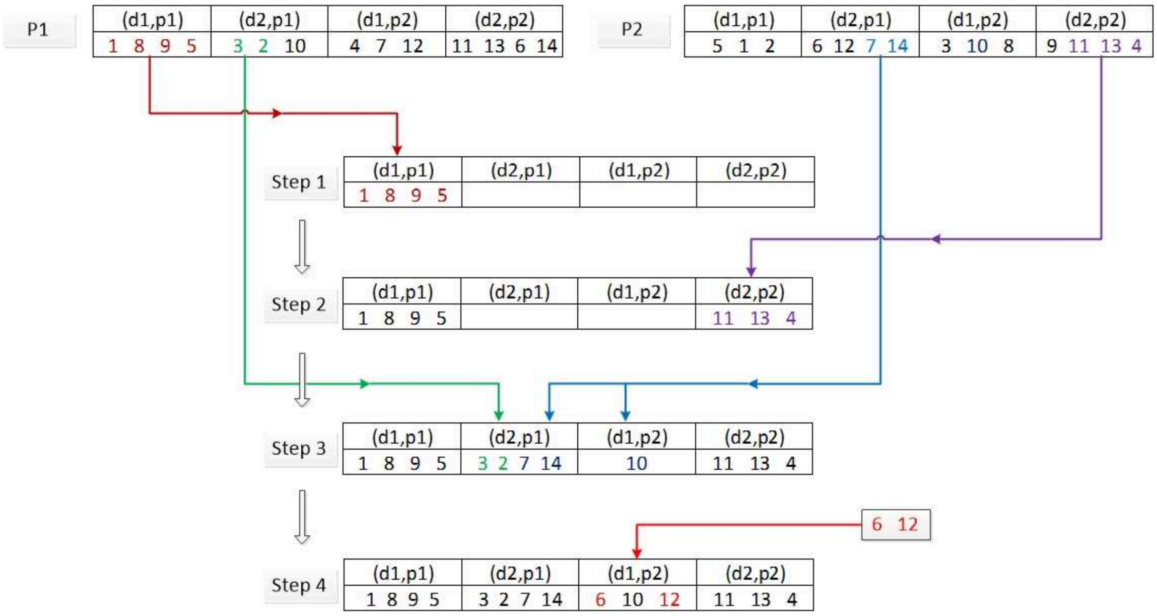

Figure 4.5: Crossover PIX

Crossover Operator 1

Unlike the above method, in order to maximize the inheritance of the characteristics of excellent parents, this crossover divides the tours into two part. The two parents involved in the crossover process are P_1 and P_2. The $|K| * |L|$ tours are numbered from 0 to $|K| * |L| - 1$. For each tour, we assign it randomly to the set X_1 or X_2. Tours in X_1 will transmit from parent P_1 to child C; tours in X_2 will transmit from parent P_2 to child C. Let $l_1 = |X_1|$ and $l_2 = |X_2|$. The steps of crossover are as follows:

- The tours in the two set are inherited in turn. The sequence of customers is copied from parent to child if the customer does not exist in the child.

- Repeat the first step until the tours of X_1 or X_2 is completed.

- Inherit tours in the other set.

- Complete the offspring: examine if all customers are contained in the child. If not, insert the customer with the method best feasible insertion.

Algorithm 7 Crossover 1

```
Pick a random number n_1 ∈ [0, m ∗ w] according to a uniform distribution. Let n_2 = m ∗ w − n_1
Randomly select n_1 tours to form a set of tours X_1;
The remaining tours form the set X_2.
Let n = minn_1, n_2
for The first n tours of X_1 and X_2 do
    for Each tour (k_1, l_1) belonging to set X_1 and (k_2, l_2) belonging to set X_2 do
        Copy the sequence of customer visits from T(k_1, l_1)(P_1) to T(k_1, l_1)(C) by eliminating customers already exist in child C
        Copy the sequence of customer visits from T(k_2, l_2)(P_2) to T(k_2, l_2)(C) by eliminating customers already exist in child C
    end for
end for
for Each tour (k, l) in the remaining tours in X_1 or X_2 do
    Copy the sequence of customer visits from T(k, l)(P_1) or T(k, l)(P_2) to T(k, l)(C) by eliminating customers already exist in child C
end for
Complete the child solution
for each v ∈ V do
    if v does not exist in C then
        Insert v to C with best insertion
    end if
end for
```

The example in figure **??** show the process of crossover2. There are 14 customers numbered from 1 to 14. Suppose that $X_1 = (d1, p1, d2, p2)$, $X_2 = (d2, p1), (d1, p2)$. In the first step, genes in tour $(d1, p1) \in X_1$ of parents P_1 is copied to tour $(d1, p1)$ of child. In step 2, tour $(d2, p1) \in X_2$ is copied from P_2 to child. In step 3, tour $(d2, p2) \in X_1$ of P_1 is copied from P_1 to child C by eliminating customer 6 and 14 because they already exist in child C. In step 4, tour $(d1, p2) \in X_2$ is inherit in the same way. At last, customer 2 and 4 are inserted to the chromosome to complete the solution.

For a chromosome with multiple tours, this crossover method can balanced the genotype inherited from the two parents. It avoids the case that after we copy from one parent, a large part of customers in tours to inherit of the other parent have already been positioned in the child. The gene sequence of the latter parent will be shattered. As a result, characteristics of the first parent will be

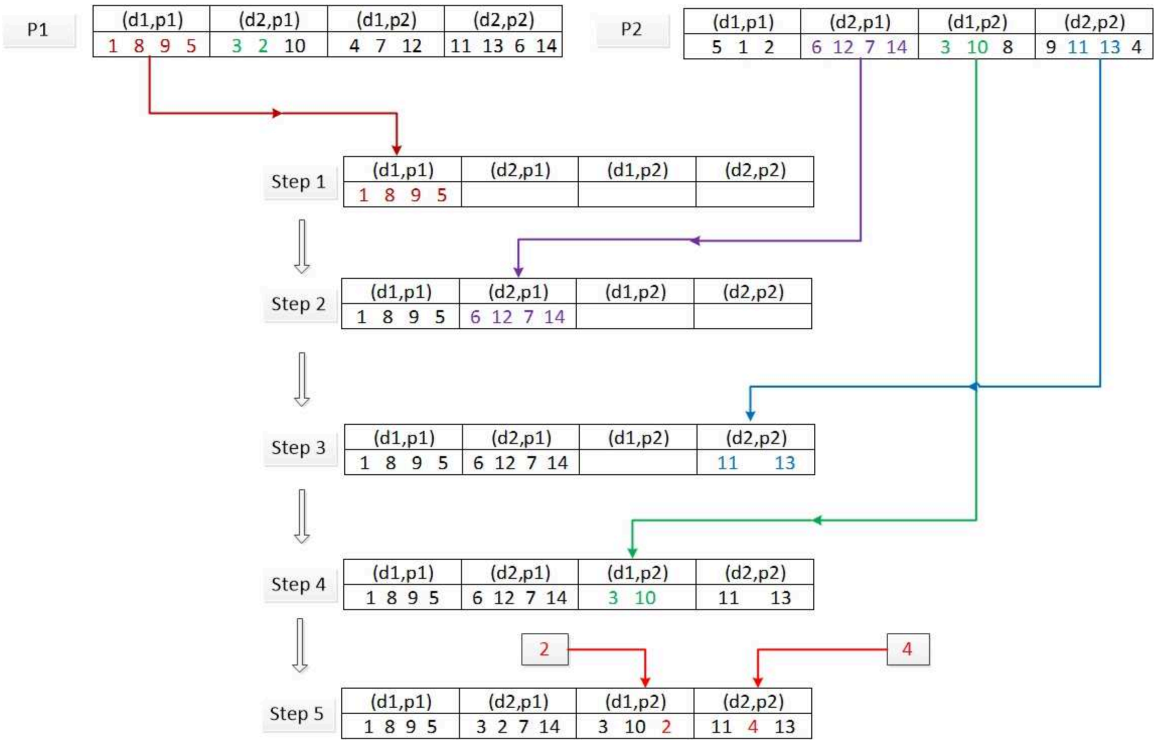

Figure 4.6: Crossover 1

overwhelming in the child.

The difference between crossover 2 and crossover 1 is that at the last step, crossover 2 completes the solution with random feasible insertion instead of best insertion. This could reduce the computing time of the algorithm. For a chromosome with multiple tours, this crossover method can balanced the genotype inherited from two parents. It avoids the case that after we copy from one parent, a large part of customers in tours which should be inherited from the other parent have already been positioned in the child. The gene sequence of the latter parent will be shattered. As a result, characteristics of the first parent will be overwhelming in the child. For both the two crossover operators, there is the possibility that the child of crossover is not feasible.

4.3.7 Mutation

Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next. It is analogous to biological mutation. The purpose of mutation in GA is preserving and introducing diversity. Mutation should allow the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other, thus stop or even stopping evolution. For a GA of TSP, there are some kinds of methods to mutate chromosomes. We adopted several mutation methods randomly applied to the chromosome to increase the diversity of population. We will present the mutations used in our problem.

- **Exchange Mutation.** Two random positions of the string are chosen and the gene corresponding to these positions are interchanged.
  $0\ 1\ \underline{2}\ 3\ 4\ 5\ \underline{6}\ 7$
  We simply choose two genes at random ('2' and '6') and exchange them:
  $0\ 1\ \underline{6}\ 3\ 4\ 5\ \underline{2}\ 7$

- **Scramble Mutation.** A subset of genes is randomly picked and then randomly rearrange them.
  $0\ 1\ 2\ 3\ \underline{4\ 5\ 6\ 7}$
  Choose the subset (position 5 to 8) and scramble the genes:
  $0\ 1\ 2\ 3\ \underline{6\ 5\ 7\ 4}$

- **Displacement Mutation.** Pick a subset of genes and move it as a group to a random position from its original position:
  $0\ \underline{1\ 2\ 3}\ 4\ 5\ 6\ 7$
  $0\ 4\ 5\ \underline{1\ 2\ 3}\ 6\ 7$

- **Insertion Mutation.** On gene is selected. Remove this gene and insert it back into the chromosome at a random position. This mutation operator has been shown to be very effective.
  $0\ 1\ \underline{2}\ 3\ 4\ 5\ 6\ 7$
  The gene '2' is selected. Take the '2' out and reinsert it.
  $0\ 1\ 3\ 4\ 5\ 6\ \underline{2}\ 7$

- **Inversion Mutation.** Pick two alleles at random and then invert the substring between them. It preserves most adjacency information and only breaks two links but it leads to the disruption of order information.
  $0\ 1\ 2\ \underline{3\ 4\ 5\ 6}\ 7$
  Two alleles (position 4 and 7) are picked at random:
  $0\ 1\ 2\ \underline{6\ 5\ 4\ 3}\ 7$

- **Displaced Inversion Mutation.** This method combines inversion mutation and displacement mutation. Select two random alleles, reverse the gene order between them, and them displace them somewhere along the length of the original chromosome.
  0 1 2 <u>3 4 5 6</u> 7
  Position 4 and position 7 are selected.
  0 <u>6 5 4 3</u> 1 2 7

The mutation is applied to the child obtained after the crossover with a probability P_m. The mutation operator is chosen randomly from the above six types. Various mutation operators contribute to the diversity of population and avoid the individuals to be similar.

In this section, we present a genetic algorithm for solving the multi-depot and multi-period field service routing problem with time window. In the following section, we introduce diversity control to the GA.

4.4 Genetic Algorithm with Diversity Control for MDMPVRPTW

A major problem in evolutionary algorithms is that simple EAs have a tendency to converge to local optima. This premature convergence is caused by several algorithmic features. First, a high selection pressure will quickly fill the population with clones of the better fit individuals, simply because their survival probability is too high compared to intermediate fit solutions. Diversity declines after a short while, and, because the population consists of similar individuals, the algorithm will have difficulties escaping the local optimum represented by the population. However, lowering the selection pressure is rarely an option because this will often lead to an unacceptable slow convergence speed. Second, high gene flow is often determined by the population structure. In simple EAs any individual can mate with any other individual. Consequently, genes spread fast throughout the population and the diversity drops quickly with fitness stagnation as a prevalent outcome.

Diversity is undoubtedly closely related to the performance of evolutionary algorithms, especially when attempts are made to overcome the problems of avoiding premature convergence and escaping local optima. Maintaining high diversity is particularly important for optimization of dynamic and multi-objective problems. Diversity measures are traditionally used to analyze evolutionary algorithms rather than guide them. However, diversity measures have been used to control EAs in at least three studies. The diversity control oriented genetic algorithm use a diversity measure based on Hamming distance to calculate a survival probability for the individuals. A low Hamming distance between the individual and the current best individual is translated into a low survival probability. Hence, diversity is preserved through the selection procedure. Another approach is the shifting-balance genetic algorithm. It calculates a containment factor between two subpopulations, which is based on Hamming distances between all members of the two populations. The third approach is the Forking GA, which used specialized diversity measures to turn a subset of the population into a subpopulation.

Amount of researchers announced that diversity measure and control can influence the efficiency of convergence. Various strategies are applied to maintain or increase the population diversity. Ursem (2002) presented a diversity-guided evolutionary algorithm (DGEA) using the distance-to-average-point measure to alternate between mutation and recombination. The method showed remarkable results on a set of benchmark problems by saving a substantial amount of fitness evaluations compared to a simple EA. Dual-population genetic algorithm (DPGA) is introduced in Park & Ryu (2010). This is a type of multipopulation GA uses an additional population as a reservoir of diversity. The DPGA adjusts the distance dynamically to achieve an appropriate balance between exploration and exploitation. Laumanns *et al.* (2002) defined the concept ϵ-dominance and the corresponding ϵ-Pareto-optimal set as well as the new selection algorithms. It proposed archiving/selection strategies that guarantee at the same time progress towards the Pareto-optimal set and a covering of the whole range of the non-dominated solutions. The diversity-control-oriented genetic algorithm in Shimodaira (1999) used a diversity measure based on Hamming distance to calculate a survival probability for the individuals. A low Hamming distance between the individual and the current best individual is translated into a low survival probability. Hence, diversity is preserved through the selection procedure. Oppacher & Wineberg (1999) proposed the shifting-balance genetic algorithm. This approach calculates a so-called containment factor between two subpopulations. The distance is calculated between each member of first population and all members of a second population. The factor determines the ration between individuals selected on fitness and individuals selected to increase the distance between the two populations. Forking genetic algorithm in Tsutsui *et al.* (1997) used specialized diversity measures to turn a subset of the population into a subpopulation.

On the basis of the GA we proposed in the previous section, we propose a genetic algorithm with diversity control for multi-depot and multi-period field service routing problem with time window.

4.4.1 Overview of the Genetic Algorithm with Diversity Control

This method evolves feasible and infeasible solutions in the population. Genetic operators are iteratively applied to select two parents from the population and combine them into an offspring. If the offspring is infeasible, it will undergo a local search-based repair procedure. If the offspring is feasible, it will be inserted directly into the new population.

4.4.2 Search Space

The determination of the search space allowing is important to the search efficiency. A controlled exploration of infeasible solutions may enhance the performance of the search, which may more easily transition between structurally different feasible solutions (Vidal *et al.*, 2012). The infeasible solutions can be obtained by relaxing some constraints. In this approach, the limits on the time window and the total duration of a vehicle are released to enrich the diversity of solutions. The constraints on the compatibility between period and customer request are still respected.

If the beginning time of customer is later than the upper bound of its time window ($b_{v_i} > l_{v_i}$), there will become a violating time of time window: $tv_{v_i v_{i+1}} = b_{v_i} - l_{v_i}$. In a route r, the incurred violating time of customer request v_i^r is given by $tv_{v_i} = max\{b_{v_i}^r - l_{v_i}^r, 0\}$. For a determined route, we can calculate its characteristics:

Algorithm 8 Genetic Algorithm with Diversity Control

```
Initialize population with feasible solutions and infeasible solutions: pop
Find fitness for all individuals in pop
while number of iterations without improvement < It_{N1}, and time < T_{max} do
    Elitism
    Find Elitenumber best solutions of pop
    Insert the selected best solutions to new population: newpop
    Crossover
    number of crossover = 0
    for number of crossover < Cnum do
        Select parent solutions P_1 and P_2
        Create offspring C from P_1 and P_2 by crossover operator
        Generate a random number in (0, 1)
        if random(0, 1) < MutationProbability then
            Mutate C
        end if
        if C infeasible then
            Repair C
            Insert C into the new population
        end if
        if C feasible then
            Insert C into the new population
        end if
    end for
    if best solution non improved for It_{div} iterations then
        Diversify population
    end if
    Adjust penalty parameters for infeasible solutions
end while
Return best feasible solution
```

- travel time

$$\tau(r) = \sum_{i=0}^{s_r} \tau_{v_i^r v_{i+1}^r} \tag{4.4}$$

- violating time of time window

$$tw(r) = \sum_{i=0,1,\ldots,s_r} tw_{v_i} \tag{4.5}$$

- total duration of the vehicle

$$dt(r) = g^r_{s_r+1} - g^r_o \tag{4.6}$$

- waiting time

$$\tau\omega(r) = \sum_{i=0,1,\ldots,s_r} \tau\omega_{v_i} \tag{4.7}$$

These characteristics will help us to evaluate the solution. A solution is composed of a set of $m \times w$ routes. We use *penalized cost* $\phi(s)$ of solution s as the evaluation function of solution. First, we will defined the penalized cost of one single route r. It is the total travel time plus the weighted sum of its excess duration and time window (function 4.8).

$$\phi(r) = \tau(r) + \omega^D max\{0, dt(r) - D\} + \omega^{TW} \times tw(r) \tag{4.8}$$

ω^D and ω^{TW} are the penalty factors and D is defined as the stated total working duration each day. The penalized cost of solution s $\phi(s)$ is given in function 4.9 by the sum of the penalized costs of all routes it contains.

$$\phi(s) = \sum_{r \in \mathcal{R}(s)} \phi(r) \tag{4.9}$$

4.4.3 Population Diversity Measure

In different kinds of combinatorial problems, diversity measures are tightly problems depended. There are two different ways to measure individual diversity in combinatorial problem. One is measure the difference between two genotypes while the other is a structural difference measure based on mathematic foundation. Let one of individual chromosomes as the best fitness chromosome denotes X, and the others chromosomes denote set Y and L is the length of a chromosome. There are four different diversity measure approaches proposed in the literature and they are listed as follows.

Hamming distance

The hamming distance is used between two strings of equal length which is the number of positions for which the corresponding symbols are different. We use the representation as a permutation way in GA to solving the combination problem, such as TSP or scheduling problem. To measure the individual diversity, we compare each gene with the best fitness chromosome and others chromosome. I is an indicate function which is defined as the total number of positions where $x_i \neq y_i$.

Definition: There are two kinds of diversity measure denote $D(X, Y)$ by hamming distance below:

$$D(X,Y) = \frac{1}{L}, \quad I = \sum_{j=0}^{L} I_j, \qquad I_j = \begin{cases} x_j = y_j, & 0 \\ x_j \neq y_j, & 1 \end{cases} \tag{4.10}$$

$$D(X,Y) = 1 - \frac{\sum_{i=0}^{L} x_i - y_i}{M}, \qquad where\ M = \begin{cases} (L^2-1)/2, \ if\ L\ is\ odd \\ L^2/2, \ if\ L\ is\ even \end{cases} \tag{4.11}$$

Euclidien distance

Euclidean distance is used to a real encoding; the concept is the same with hamming distance in permutation encoding

$$D(X,Y) = \sqrt{\sum_{i=1}^{N} (x_i + y_i)^2} \tag{4.12}$$

By connection matrix

Considering a TSP problem, each tour represents as a permutation way in GA. Therefore, the diversity measures by hamming distance cannot reflex a true touring situation in TSP, and the connection matrix is considered the sequence in a tour. Although each tour represents as a permutation way differently in GA, there are still some chances thos touring sequence are the same in connection matrix

$$A = \begin{bmatrix} a_0 0 & a_0 1 & ... & a_0(n-1) \\ a_1 0 & a_1 1 & & ... \\ & & ... & \\ a_(n-1)0 & ... & & a(n-1)(n-1) \end{bmatrix}$$

where A is connection matrix of a tour, n is the number of customers.Let a similarity function $S(X,Y)$ measure the similarity

$$S(X,Y) = \sum_{ij}(x_{ij}|x_{ij} = y_{ij})/n \tag{4.13}$$

The diversity measure could be defined as follows:

$$D(X,Y) = 1 - S(X,Y) \tag{4.14}$$

By information entropy

In information theory, the Shannon entropy or information entropy is a measure of the uncertainty associated with a random variable. It quantifies the information contained in a message, usually in bits or bits/symbol. The locus diversity H_i of the ith locus ($i = 1...n$) is defined as follows:

$$H_i = -\sum_{c\in C} pr_{ic} \ln pr_{ic}, \quad where\ pr_{ic} = \frac{na_{ic}}{pop\_size} \tag{4.15}$$

where na_{ic}: the number of appearance of city c at locus I, C is the number of customers should be visited. We need to translate the individual diversity to a single colony index to measure the population diversity is low or high. There are two kinds of method to measure it. The first is arithmetic average defined by $PD = \frac{\sum D(X,Y)}{N}$. The other is linear scale measure defined by $PD = \frac{\overline{d}-d_{min}}{d_{max}-dmin}$ where $\overline{d}$ is the average diversity, d_{max} is the maximum diversity and d_{min} is the minimum diversity of the archive.

4.4.4 Evaluation

The individual-evaluation function in population-based meta-heuristic aims to determine for each individual a relative value with respect to the entire population. It is often based on the value of objective function of the problem which is called fitness. This fitness is used to the selections of mating parents or the individuals to survive to the next generation. However, the fitness is just a one-sided evaluation to a meta-heuristic problem. The diversity is also a critical performance factor. We therefore take the diversity into consideration on the evaluation function. Both the cost of an individual and its diversity contribution are evaluation criterion. Pareto ranking scheme is used to computed the integrated fitness of each individual.

The fitness is calculated based on the cost with penalty. Their are several factors to consider:

- The number of missed requests.
- The number of period-incompatible requests.
- The number of time-window violating requests.
- The total waiting time.

The fitness of individual can be defined as:

$$fitness(P) = f(P) + \omega^m * mc(P) + \omega^c * pc(P) + \omega^w * twv(P) + \omega^t * \tau\omega(P) \quad (4.16)$$

In the equation 4.16, $f(P)$ is the travel cost of a solution defined in 4.2. ω^m is the penalty factor of missed customer of a solution P and $mc(P)$ is the number of missed requests of P; ω^c is the penalty factor of period-incompatible requests $pc(P)$; ω^w is the penalty factor of time-window violation requests and $twv(P)$ is the number of requests that don not respect their time windows; $\tau\omega(P)$ is the penalized by a factor ω^t.

A solution s represents an individual P in a population of genetic search. The fitness of P is defined as the penalized cost of the solution s: $\phi(s)$. Its diversity is measured by a normalized Hamming distance. The Hamming distance between two individuals P_1 and P_2, noted as $\delta^H(P_1, P_2)$ is based on the differences between the service period and the depot assignments of the two individuals.

$$\delta^H(P_1, P_2) = \frac{1}{2n} \sum_{i=1,\dots,n} (\mathbf{1}(\pi_i(P_1) \neq \pi_i(P_2)) + \mathbf{1}(\delta_i(P_1) \neq \delta_i(P_2)) \quad (4.17)$$

Equation 4.17 gives the method to compute the hamming distance of this problem. $\pi_i(P)$ is the period of the service to customer v_i in individual P and $\delta_i(P)$ represents the depot of customer v_i. $\mathbf{1}((cond))$ is a valuation function that returns 1 if the condition *cond* is true and 0 otherwise.

The diversity contribution of an individual P in a population Π of size n_Π is computed according to Equation 4.18.

$$\Delta(P) = \frac{1}{n_\Pi} \sum_{P' \in \Pi} \delta^H(P, P') \tag{4.18}$$

to evaluate an individual in a population, we use biased fitness Vidal *et al.* (2013) $BF(P)$ defined in 4.19. This is a diversity and cost objective that involves both the rank $fit(P)$ of P in the population with regards to solution cost $\phi(P)$, and its rank $dc(P)$ in terms of diversity contribution $\Delta(P)$. $BF(P)$ depends upon the actual number of individuals in the subpopulation $nbIndiv$, and a parameter $nbElit$ ensuring elitism properties during survivor selection. This trade off between diversity and elitism is critical for a thorough and efficient search.

$$BF(P) = fit(P) + (1 - \frac{nbElit}{nbIndiv})dc(P) \tag{4.19}$$

4.4.5 Repair Procedure

An infeasible offspring resulting from the crossover operator undergoes an repair operator based on neighbourhood search. The repair is essential for a fast progression toward high-quality solutions.

When infeasible solutions are used, evaluation moves implies to compute the change in total arc costs, as well as the variation of duration and time-window infeasibility of the routes. Calculation of cost and load variation is straight forward to perform in amortized $O(1)$ for moves based on a constant number of arc exchanges. A method is proposed to compute infeasibility in $O(1)$ for some neighbourhoods, including $2 - opt*$, inter-route swaps, and inter-route inserts. We introduce an approach to evaluate combined duration and time-window infeasibility which can be applicable to various neighbourhood based on a constant number of arc exchanges or sequence relocations.

For measuring the feasibility of a solution, we define a score of feasibility. This is the sum of penalized terms of an individual.

$$S(P) = \omega^m * mc(P) + \omega^c * pc(P) + \omega^w * twv(P) + \omega^t * \tau\omega(P) \tag{4.20}$$

The repair procedure is a set of nine heuristic. The nine heuristics have been described in chapter 3. The route improvement phase iterates, in random order, over each vertex v_1 and v_2. Moves are examined in random order, the first yielding an improvement of feasibility score being implemented. The route improvement phase stops when the solution becomes feasible or all possible moves have been successively tried without success.

4.4.6 Population Management

The population management mechanism complements the selection, crossover, and repair operators in identifying and propagating the characteristics of good solutions, enhancing the population diversity, and providing the means for a thorough and efficient search. For a population with μ individuals, if the infeasible individuals exceed $\rho * \mu$, the population needs a 'refresh'. ρ is the maximum permitted rate of infeasible solutions. Firstly, we delete some infeasible individuals to the proper number and generate new feasible solutions to insert in the population. The penalty parameters are dynamically adjusted during the execution of the algorithm, to favor the generation of naturally-feasible individuals. Let R be a target proportion of naturally-feasible in individuals, Let Σ^X be the proportion in the last generated N_{new} individuals of solutions that is feasible to one respect. The X could be M, C, W and T, represents respectively missed city, period compatibility, time window and waiting time. The following adjustment is performed every N iterations.

- if $\Sigma^X \leq R - 0.05$, then $\omega^X = \omega^X * \lambda$;
- if $\Sigma^X \geq R + 0.05$, then $\omega^X = \omega^X * \gamma$;

In the above formula, λ and γ are coefficients to adjust the penalty parameters. λ is a number greater than 1 and γ is a number in $(0, 1)$. In our algorithm, λ is fixed as 1.2 and γ is fixed as 0.85.

4.5 Computational results

In this section, we present the results of computational experiments of our proposed algorithm on technician routing problem. The proposed algorithm is coded in Java language and implemented on a personal computer with a procesor Inter Core i5 1.8 GHz. The genetic algorithm is tested with instances of Tricoire (2007).

As we described in previous chapter, the instances of C1 contain 100 customers and instances of C2 contain 180 customers. The time horizon is consisted of 5 periods of 8 hours to simulate the 5 workdays during a week. The number of vehicles is 3.

Firstly, we compare the results of our proposed GA and the results of Cplex on small size multi-depot and multi-period field service routing problem with time window. Secondly, proposed algorithms are executed on the instances of C1 and C2. The results are analyzed and a sensitivity analysis is done. Thirdly, we test the performance of genetic algorithm with diversity control on instances C1 and C2.

4.5.1 Computational experiments of proposed GA

4.5.1.1 Results on small size instances

We solve the instances with the proposed GA. The results and the computing times are shown in Table 4.2. The column Cost GA is the average of the best solutions for each instances with 10 runs. The column Time GA is the average of the computing time with proposed genetic algorithm. The column Cost Cplex is the average of best solutions computed with Cplex. The column Time Cplex is the computing time of Cplex. The value of column Gap is calculated by $Gap = \frac{Cost\ GA - Cost\ Cplex}{Cost\ Cplex}$. For the instance size $N = 20$ and $N = 30$, the results are average values of the 10 instances in C1 and C2. While for instances of size $N = 40$, the results are calculated from the three instances C1_4, C_4 and C2_5.

The gap is 0.06% for instances of size $N = 20$. The differences of computing time between two methods are relatively small. As the size of instance becomes larger, the gap becomes larger but still in a reasonable range. However, the computing time has undergone great changes. For instance size of $N = 30$, the two times are 27s and 1347s for the two methods; for instance size of $N = 40$, the computing time of Cplex is up to 18h which is not any more suitable for the real world application. It can be concluded that for instances more than 40 customers, exact method can not solve the problem.

For instances of larger size, the mentioned computer has not enough memory to solve the problem with exact method. We will investigate the large scale

Table 4.2: Results of GA and Cplex for small size instances

Instance size	Cost GA	Time GA	Cost Cplex	Time Cplex	Gap
20	5041.17	14s	5040.89	79s	0.06%
30	8486.26	27s	8465.65	1347s	0.24%
40	9203.21	183s	9154.96	18h	0.53%

Table 4.3: Parameter setting for crossover operators comparison

Parameter	Time	*popsize*	P_m	*elitenum*	R_c
Value	30min	100	0.05	20	0.7

problems from other aspect.

4.5.1.2 Investigation of crossover operator

This investigation is performed to test the performance of the new crossover methods proposed in our genetic algorithm. We compare our proposed the three crossover operators mentioned in this chapter: Crossover1, Crossover2 and PIX. The numerical results are computed after 10 independent runs with each crossover operator and each run is executed for a time of 3o minutes. We compare our two proposed crossover operators and the PIX crossover operator to determine whether the proposed operators are advantageous to produce offspring. For this reason, the three crossover operators mentioned are employed in our proposed GA with parameters shown in Table 4.3. These parameters are fixed during the whole computing. The computational results of the comparison of the three crossover operators are shown in Table 4.4.

From Table 4.4, we can clearly observe that the better solution quality is obtained under the crossover operator 2. The efficiency of crossover operator 1 is better than the PIX but is not as good as crossover operator 2. As a result, the proposed two crossover operators are both advantageous in generating better offspring. Compared to PIX, the two proposed crossover operators inherit more orderly and more respectable to the compatibility of period and time window. This makes them easier to get feasible offspring feasible. Only feasible offspring are accepted to the new generation. In this way, our operators have higher possibility to get a better offspring.

Table 4.4: Comparison between crossover operators.

Instance	Crossover 1	Crossover 2	PIX
C1_1	17556.27	**16899.08**	19183.93
C1_2	16333.30	**14641.58**	17280.72
C1_3	15439.22	**15013.20**	17714.03
C1_4	16076.60	**15010.54**	18893.53
C1_5	14559.87	**13461.81**	15617.30
C2_1	26766.25	**25954.63**	28989.23
C2_2	26029.15	**25276.24**	27408.71
C2_3	25946.78	**25434.10**	26868.58
C2_4	25818.01	**24777.15**	27570.77
C2_5	24526.74	**23939.51**	26127.61

Fig. 4.7 shows the search evolution of GA of crossover 1 and crossover 2 with iteration number. The searching efficiency per generation of crossover 1 is much more higher than crossover 2. However, during a running time of 1 hour, GA with crossover 1 searched 604 generations and the GA with crossover 2 searched up to 10^5 generations. The 'quantity' overwhelms the 'quality' in our experiments. That's why crossover 2 gets better solution than crossover 1 within a certain running time.

4.5.1.3 Investigation of initial population

The computational study is carried out to compare the algorithm with random generation for the initialization procedure (GA1) and the proposed algorithm with constructive heuristic based on best insertion (GA2). The performance is evaluated using the instances in C1. The parameters of the GA for this experiments are: population size = 100, mutation rate = 0.05, crossover rate = 0.7, elite number = 10. The running time is limited as 60 min. Table 4.5 shows the results of two algorithms. The GA based on best insertion get better results in all the instances. The only difference between the two experiments is the quality of initial solutions. To deep understand the performance of the two algorithms, we compare the evolution of travel cost with iteration number in Fig 4.8.

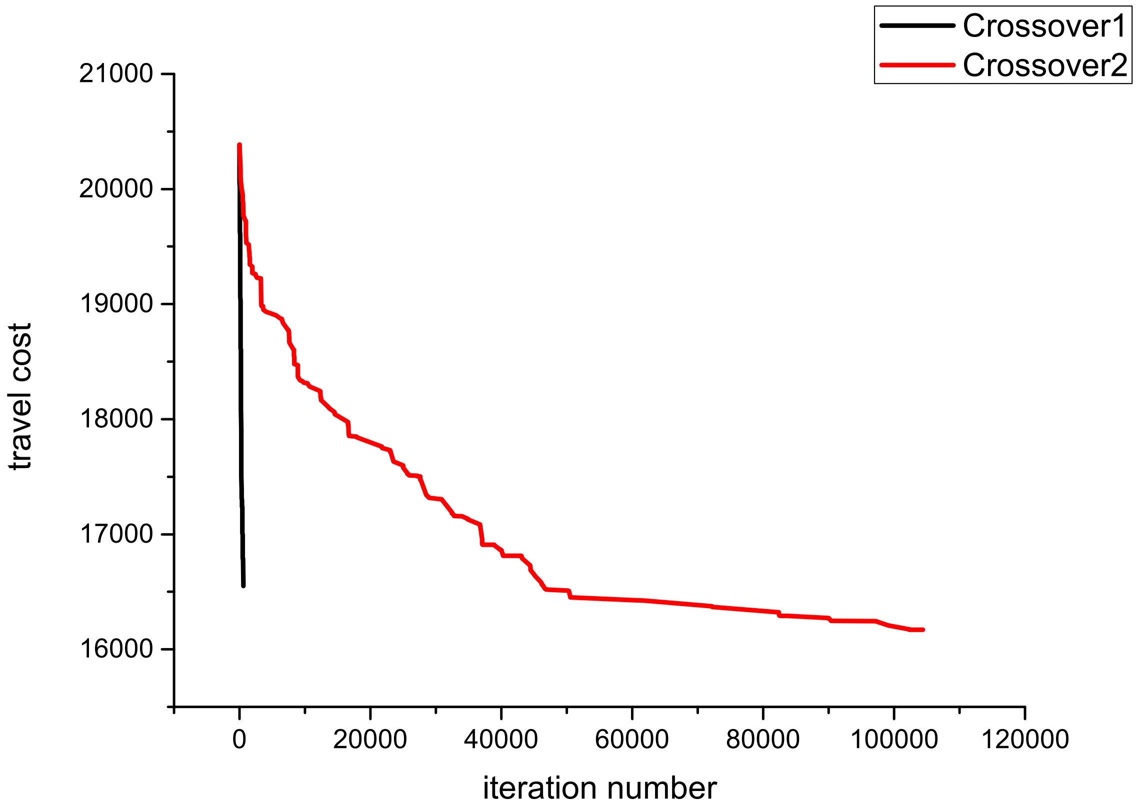

Figure 4.7: Evolution with iteration number of crossover 1 and crossover 2

In Fig.4.8, evolution of GA2 start at a lower point and always finds better solution than GA1 in every single iteration. GA1 converges quickly at the beginning. After it gets the level of the initial solutions of GA2, the convergence rate slows down. Though the convergence rate of GA2 is lower than GA1, it find better solutions than GA1 during the whole searching process. We conclude that initial population makes great influence to the search efficiency.

4.5.1.4 Sensitivity analysis of the parameters

To analyze the influence of the parameters, we performed many experiments with different settings. To find the relationship between the time consumption and the quality of solutions with different parameter settings, we applied the proposed algorithm on the instances with 10 runs. The parameters we investigate include rate of crossover, rate of mutation and elite number. In Fig 4.9, the processes of

Table 4.5: Influence of initial population

Instance	Random initialization	Construction based on best insertion
C1_1	17726.04	**15800.33**
C1_2	16849.62	**14916.66**
C1_3	16077.32	**15013.54**
C1_4	15916.82	**14593.60**
C1_5	15365.61	**14025.91**

genetic search with different crossover rates are shown. The three curves represent respectively the evolution of searching with *crossover rate* = 0.5, 0.7 and 0.9. We can tell from the figure that the higher the crossover rate is, the better the search quality is. Crossover rate of 0.9 gets the best solution among the three crossover rate value.

Fig 4.10 presents the influence of the elite number on the final results. The population size of the experiment is 100. We set the elite number as 5, 10 and 20. The results tells us that the elite number makes influence to the results. The search with elite number = 20 gives better results than the lower elite number.

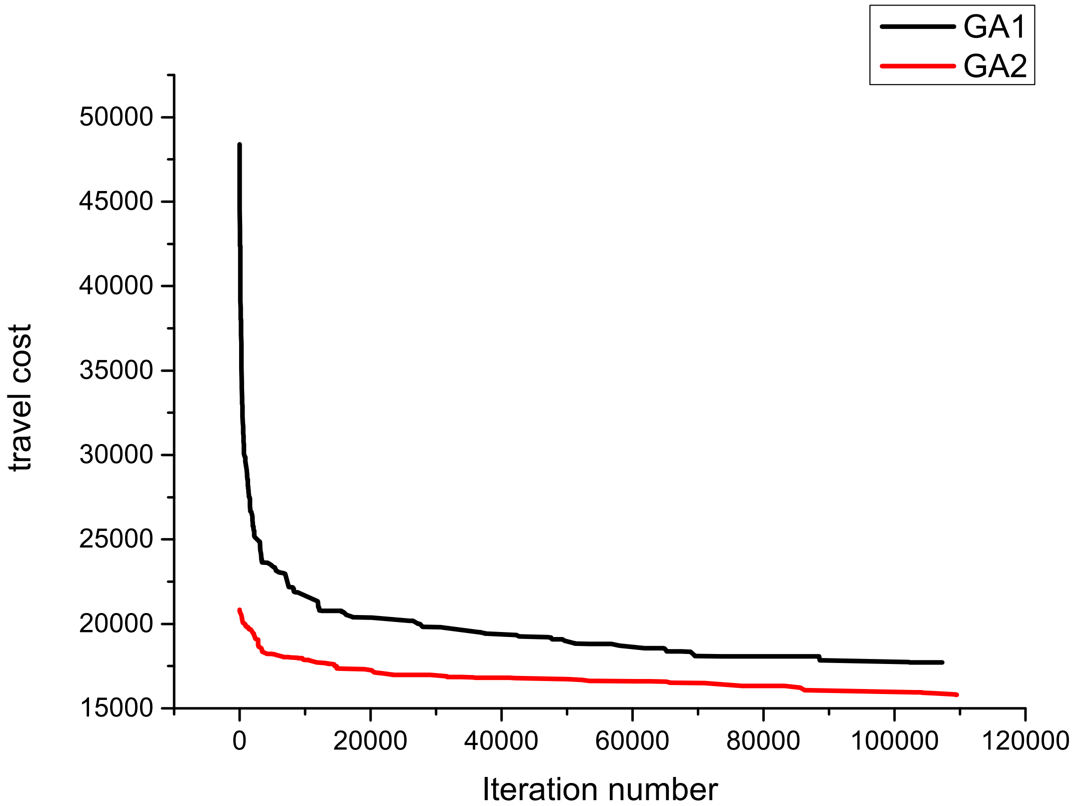

Figure 4.8: Compare initial population

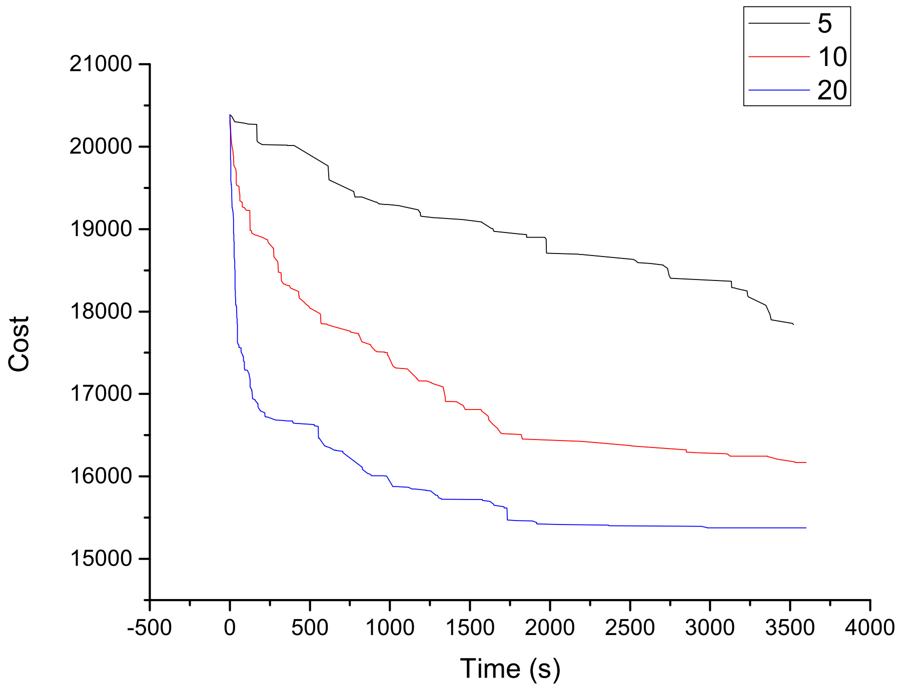

Figure 4.10: Results of different elite numbers

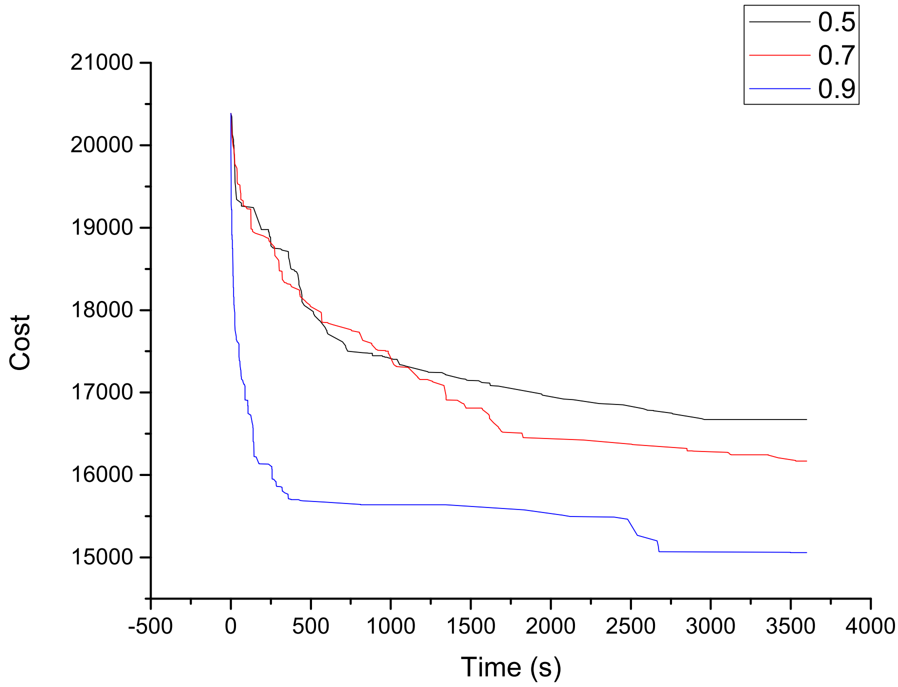

Figure 4.9: Results of different rates of crossover

Fig 4.11 presents the influence of the mutation rate on the final results. The influence of different mutation rates don not make as large difference to the results as the crossover rate and elite number.

These experiments shows that the three factors have more or less influence to the performance of proposed GA. Every new generation is composed with three parts: crossover offspring, elite members and new generated solutions. In Fig.4.9 and Fig.4.10, we observe that when the number of new generated solution increases, it is hard to find good solutions. New solutions bring diversity to the population while at the same time they reduce the quality of individuals. As a result, if the number of new generated solution is too large the searching efficiency gets worse.

4.5.2 Results of GA with Diversity Control

We conducted several sets of experiments to evaluate the performance of the method. Firstly, we employ the method to the instances in C1 and C2 mentioned in chapter 3.

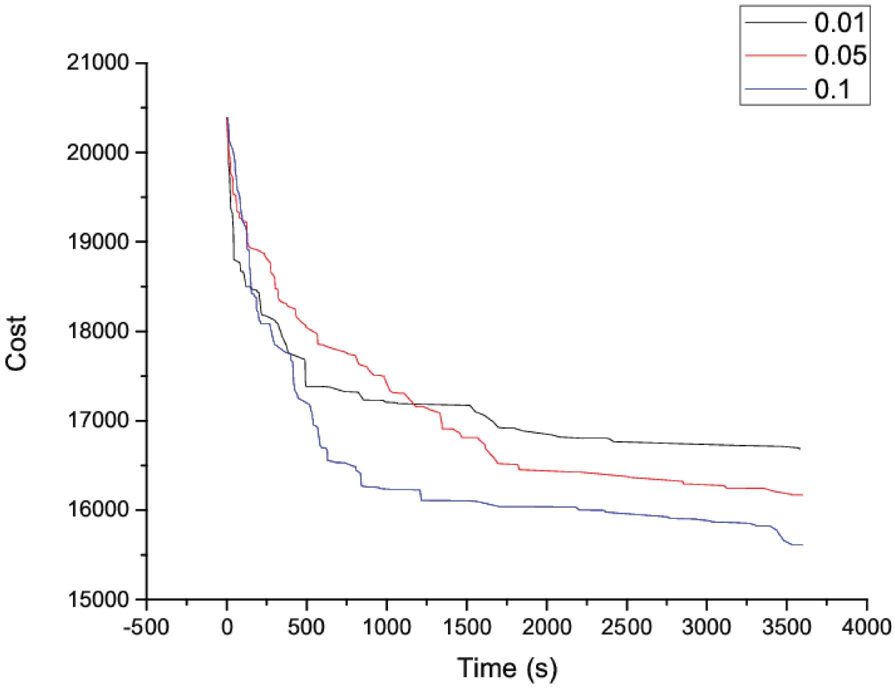

Figure 4.11: Results of different mutation rates

4.5.2.1 Parameter calibration

As for most meta-heuristics, GA relies on a set of correlated parameters and configuration choices for its key operators. In order to identify good parameter values, we adopted the meta-calibration approach which was shown to perform particularly well for GA calibration.

Meta-calibration involves solving the problem of parameter optimization by means of meta-heuristics. In this scope, any evaluation of a set of parameters implies launching automatically the algorithm to be calibrated on a restricted set of training instances and measuring its effectiveness.

4.5.2.2 Results on field service routing instances

We applied the algorithm to the instances in C1 and C2 and the results are shown in Table 4.7. For instances in C1, the results obtained are better than that use the genetic algorithm without diversity control. For instances in C2, the computing efficiency is very low when using GA without diversity control.

Table 4.6: Parameter Calibration

Parameter	Range	Final Setting
Population Size	[20,200]	100
Proportion of elite individuals	[0,1]	0.3
Reference proportion of feasible individuals	[0,1]	0.3
Granularity threshold	[0,1]	0.4
Rate of crossover	[0,1]	0.7
Rate of mutation	[0,1]	0.05

The GA proposed in this chapter relax the constraints and allowed the search in infeasible solutions. We are able to fine near-optimum results for a large instance in C2. The column GADC presents the best results found by genetic algorithm with diversity control. The column GAP is calculated by $GAP = \frac{GADC-GA}{GA}$, the gap of results of genetic algorithm with diversity and the previous genetic algorithm. The computing time of each instance is 3600s. The results are the best solutions during 10 runs of the genetic algorithm with diversity control for each instance. For instances of C1, the gaps between two results are small. For instances of C2 with large number of customers, the GADC performs better than GA. The reason is that for a more constrained problem in C2, it takes much time to find feasible solutions. As GADC searches the infeasible solution space and its searching speed is larger, it is more likely to find a feasible solution.

4.5.2.3 Sensitivity analysis on method components

This section analyses the role of several of these components. We measure the impact of the decomposition phases, the contribution of infeasible solutions to the search, which required new move evaluation procedures, and the diversity and cost objective. Table 4.8 compares the average results on 5 runs, as an average gap to the best results of the instances C1 and C2.

Table 4.7: Results of Instances C1 and C2

Instance	GADC	GA	GAP
C1_1	15792.09	15800.33	−0.05%
C1_2	14853.55	14641.58	1.4%
C1_3	14832.71	15013.20	−1.2%
C1_4	14433.23	14593.60	−1.1%
C1_5	13335.14	13461.81	−0.9%
C2_1	24800.93	25954.63	−4.2%
C2_2	24116.23	25276.24	−4.5%
C2_3	25013.08	25434.10	−1.7%
C2_4	23593.25	24777.15	−4.8%
C2_5	22025.05	23939.51	−8.0%

Table 4.8: Sensitivity Analysis on the Components

Instance	No Diversity Objective	No Infeasibility	Complete Algorithm
C1	+1.21%	+0.65%	+0.27%
C2	+3.94%	+1.71%	+0.61%

These experiments confirms the pertinence of the framework of the algorithm, as the diversity and infeasible search space contributes largely to the performance of the proposed method.

4.6 Conclusion

In this chapter, we studied the multi-period and multi-depot field service problem with time window. We proposed a genetic algorithm with a chromosome representation by resource and a new crossover operator for solving the problem. The computational experiments showed that the proposed algorithm is competitive in terms of the quality of the solutions found. Then we proposed a new genetic search method with diversity control to efficiently address several classes of multi-depot and multi-period field service routing problems, for which few efficient algorithms are currently available. We introduce several methodological contributions, in particular the repair of infeasible solutions, the individual evaluation procedure driven both by solution cost and contribution to population diversity and the adaptive population management mechanism that enhances diversity.

Conclusions and Perspectives

In this chapter, we will conclude the dissertation by summarizing the main results and then present some perspectives of future research to complete and improve this work.

Conclusions

The aim of this dissertation has been focused on optimizing the field service routing problem for realistic application. We have studied the multi-depot and multi-period field service routing problem of technicians, where we have determined daily technician schedules and routes in order to meet customer visit requirements over a time horizon. The problem has not been much studied in the literature. In this dissertation, we proposed several approaches to solve the problem.

- Chapter 1 explained the motivation of this thesis. A general introduction of the problem studied in this thesis is given.
- In chapter 2, a state of the art of the literature in the addressed problem is given. We introduced the different variants of VRP classified by the different characteristics and elements. Then the methodologies of VRP especially of the variants that similar to our problem is presented. The study in chapter 2 perform as the basis of the following research.
- In chapter 3, a formal statement of the addressed problem is given. Heuristics of construction and improvement are investigated to find optimized solutions of the multi-period and multi-depot field service routing with time window. These heuristics can obtained feasible solutions and these solutions constructed by constructive heuristic based on best insertion can

be improved by a series of improvement heuristics. Different heuristics are compared in terms of the results and running times (Liu & El Kamel, 2015).

- In chapter 4, we studied the addressed problem with genetic algorithm. First, we adapted the GA with a new chromosome representation and new crossover operators. Only feasible solutions are accepted in the population. Second, we proposed a GA with diversity control. Diversity contribution is added to the evaluation of the individuals which avoids the premature of the search. Experiments results show that these approaches have a approving performance on the instances simulated the real world data (Liu & El Kamel, 2017a), (Liu & El Kamel, 2017b).

Future Work

In order to meet the higher needs of the practical applications, there are still following aspects to be studied in the future on this addressed problem.

First of all, this dissertation has provided theoretical foundations on the field service routing problem. Due to lack of the instances that corresponding to the addressed problem, very limited numerical experiments have been done to the problem. It is necessary to modify some of the existed benchmark problems or create new instances to help the study of the problem.

Second, for a use of real world production, it is critical to consider the dynamic elements of the problem, such as emergency requests, stochastic travel time and service time, the break down of vehicles, etc, and therefore, the dynamic routing problem should be studied in the future.

Third, as the development of new technology, new informations could be involved for the resolution of the problem. For example, GPS can obtain the current location of each vehicle and telecommunication technology contributes to the real-time communication between customers and technicians. These informations are very useful to the real-time optimization of the field service routing problem.

Résumé Étendu en Français

Introduction

Aujourd'hui, avec le développement du secteur tertiaire, la performance logistique des entreprises et l'optimisation des transports sont devenus des enjeux économiques importants. Cela repose notamment, pour beaucoup d'entreprise, sur l'efficacité des tournées de véhicules réalisées quotidiennement. La planification de tournées de service est une activité consiste à organiser, sur plusieurs périodes de temps, les déplacements de personnels chez des clients pour effectuer des opérations techniques. Nous nous intéressons à un problme de tournées des service multi-période et multi-dépôt avec fenêtres de temps de visite chez les clients.

Cette thèse est composée de cinq chapitres organisés de la manière suivante:

- Le chapitre 1 donne la motivation de la recherche et une introduction générale au problème que nous avons étudié dans cette thèse.

- Le chapitre 2 examine la littérature sur le VRP et les problèmes similaires de notre problème.

- Au chapitre 3, nous avons formalisé le problème étudié avec le modèle methématique. Ensuite, les heuristiques de construction de d'amélioration sont proposées pour obtenir des solutions raisonnables.

- Le chapitre 4 a proposé un algorithme génétique pour le problème abordé. Ensuite nous avons discuté de l'algorithme génétique avec le contrôle de diversité.

Motivation

Au cours des dix dernières années, la logistique a finalement été reconnue comme un domaine qui était essentiel à la réussite globale de lentreprise. Il a une grande importance dans léconomie, dans l'industrie et dans la protection de l'environnement. Figure 4.12 montre qu'en union européenne, la logistique représentait environ 9% du GDP au cours des dernières décennies. Ce nombre est significativement plus élevé dans les pays en développement que dans les pays développés. En adoptant une approche de gestion de l'efficacité logistique, les coûts liés à la logistique en pourcentage des ventes diminuent de 4% à 7%. En outre, l'augmentation de la consommation d'énergie liée aux transports et ses effets négaifs sur l'environnement ont suscité de plus en plus de préoccupations mondiales. Table 4.9 montre le pourcentage d'émmisions totales liées aux transports à Ile-de-France. Nous pourrions dire que la pollution environnementale pourrait être réduite en optimisant l'acheminement de la logistique, car les camions sont habituellement utilisés à des fins de logistique.

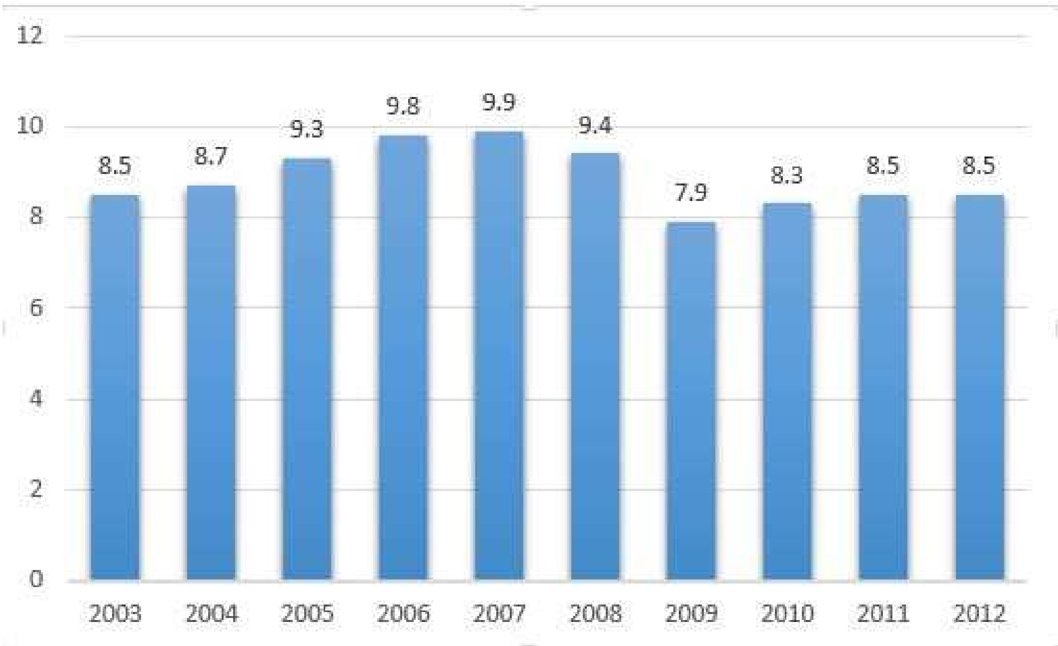

Figure 4.12: Coût de la logistique en pourcentage du GDP dans l'UE (Source: State of Logistics Report 2014/CSCMP)

Dans la littérature, la plupart des travaux traitent de problèmes impliquant des livraisons de marchandises. Cependant, des problèmes de tournées de ser-

Table 4.9: Pourcentage des émissions des camions dans des émissions totales de transport en Ile-de-France

	CO_2	SO_2	NO_X	PM_{10}
Camions	26%	43%	38%	59%

vice sont aussi importants que des problèmes de livraisons. Les tournées de service concernent l'organisation de déplacement de personnels vers des clients afin d'effectuer différents activités techniques ou commerciales. Les entreprises souhaitent organiser au moindre coût un meilleur service chez les clients. Avec le développement de l'économie, de plus en plus d'entreprises doivent fournir des services à faible coût et de haut niveau. Par conséquent, le problème de tournées de service est très digne d'étude.

Pésentation du Problème

Dans cette thèse, nous intéressons en cas des tournées en clientèle, qui sont des tournées de service. Il ne s'agit pas de livrer une marchandise, mais d'effectuer des réparations, des opérations de maintenance, des relevés, ou même des enquêtes. Les demandes sont séparées en deux catégories : les clients obligatoires avec fenêtres de temps et les clients optionnels sans fenêtres de temps.

Les problèmes des tournées de véhicules (Vehicle Routing Problem, VRP) constituent une famille de problèmes abondamment traités dans la littérature. Les problèmes étudiés dans le cadre de cette thèse présentent quelques différences majeures avec le cas classique:

- Les problématiques liées à la capacité sont absentes de ces problème; cependant, la durée totale de chaque tournée est bornée par la durée d'une journée de travail, et il s'agit en pratique d'une contrainte forte.

- Nous traitons le cas multi-périodes, c'est-à-dire avec un horizon de planification de plusieurs jours. Une période de validité est associée à chaque demande.

- Le nombre de véhicules est limité, et cette limite est une contrainte forte.

- Chaque véhicule dispose de points de départ et d'arrivée propres, et potentiellement différents.

- Les clients obligatoires sont considérés comme étant plus importants que les optionnels. La période de validité d'un client obligatoire est toujours limitée à une seule période.

L'objectif est de minimiser le coût total des tournées. Il n'existe pas de coût fixe associé à la création d'une tournée, et le coût est assimilé à la distance. L'objectif est donc de minimiser la distance totale de parcours.

Heuristiques de Construction et d'Amélioration

Nous présentons une famille d'heuristiques de construction et amélioration pour le problème de tournées de service multi-dépôt multi-périodes avec fenêtres de temps. Ces heuristiques permettent de produire des solutions réalisables, servant de point de départ pour les métaheuristiques.

Pour des problèmes multi-période, généralement, les chercheurs décomposent le problème en deux sous-problèmes: affectation des visites aux jours et résolution d'un VRP par période de l'horizon. Nous proposons une méthode de construction globale pour les problèmes de tournées basée sur *best insertion*. Dans best insertion, on essaie d'insérer chaque demande dans chaque tournée afin de déterminer l'insertion de coût minimal. L'incompatibilité entre certaines demandes et certaines tournées est donc une contrainte forte.

Avant chaque insertion, il faut vérifier la faisabilité de cette insertion; c'est-à-dire, si cette insertion est exécutée, la solution devrait être faisable. Il y a trois critères à vérifier: (1) tous les clients sont visités pendant les périodes compatibles; (2) pas de violation de fenêtres de temps; (3) la durée de chaque véhicule pour chaque période est inférieure au temps de travail maximal. Les procédures sont exécutées deux fois, d'abord pour des clients obligatoires et ensuite pour des clients optionnels.

La méthode de construction nous offre les solutions faisables comme un point départ des heuristiques d'amélioration. Nous avons proposé neuf voisinages différents, chacun associé à un type d'amélioration, puis nous avons les utilisé pour produire des heuristiques d'amélioration.

Algorithme Génétique

Les algorithmes génétiques sont des heuristiques de recherche locale qui imitent le processus de la sélection naturelle. Ils sont été appliqués dans de nombreux de domaines. Nous avons proposé deux algorithmes génétiques pour résoudre le multi-dépôt multi-période problème de tournées de service avec fenêtre de temps. En raison de la nature aléatoire, les solutions trouvées par algorithme génétique peuvent être bonnes, mauvaises ou irréalisables. Il est de grande importance de trouver des solutions faisables.

Comme mentionné précédemment, l'algorithme génétique a déja été utilisé pour résoudre divers VRP. Cependant, les algorithmes existants ne peuvent pas être utilisés pour résoudre notre problème. Tout d'abord, ce probème est un problème avec multi-période et multi-dépôt, il est important de trouver une représentation de chromosome appropriée. Deuxièmement, la détermination de la population initiale et de l'espace de recherche mérite d'être discutée car les contraintes de notre problème sont très fortes. Enfin, la conception de l'opérateur de croisement est une clé pour améliorer la qualité de la recherche génétique. Nous avons proposé un nouvel algorithme génétique pour résoudre le problème abordé dans cette thèse.

Nous avons implémenté l'algorithme en trois catégories des instances: de petite taille (<40 clients), de moyenne taille (100 clients) et de grande taille (180 clients). En comparant avec les solutions obtenues avec le solveur Cplex, pour les instances de petite taille, la solution optimale est obtenue dans un délai raisonnable et les résultats obtenus par nos algorithmes génétiques sont assez proches de la solution optimale. Pour les instances de moyenne taille et de grande taille, Cplex ne peut plus obtenir des solutions optimales dans un délai acceptable et nos algorithmes peuvent identifier de meilleures solutions rapidement. Pourtant, la performance de cet algorithme en instances de grande taille n' est pas aussi satisfaisante que celle en instances de moyenne taille. Donc nous avons proposé un algorithme génétique avec controle de diversité basé sur l'algorithme ci-dessus.

Cette méthode développe des solutions faisables et infaisables dans la population. La contribution de diversité d'un individu est définie par le Hamming distance dans la population. Elle constitue un facteur dans l'évaluation des chromosomes. Nous avons proposé une méthode pour réparer les progénitures pour enlever la possibilité d'avoir des solutions faisables. Les résultats de cet algorithme sont comparés avec l'algorithme avant en instances de moyenne taille et de grande taille. Pour les instances de moyenne taille, les écarts entre deux algorithmes sont faibles. Pour les instances de grande taille, l'algorithme génétique avec contrôle de diversité fonctionne mieux que l'autre.

Contributions Principales

Cette thèse considère l'optimisation du probème de tournées de service dans le monde réel. L'objectif est de trouver un plan de tournées optimisé pour les entreprises afin dóffrir un service de qualité avec un coût minimum.

Les principales contributions de cette dissertation sont résumées comme suit:

Le problème de tournées de véhicules de service est un problème important dans l'industrie moderne. Pourtant, la recheche menée sur ce problème du monde réel est très limitée. Les problèmes réalistes sont généralement plus contraint et compliqués. Le problème étudié dans cette thèse est modélisé pour une enquête plus approfondie. Diverses contraintes ont été prises en considération pour simuler le problème réaliste afin de résoudre le problème réel.

Les méthodes exactes et les méthodes méta-heuristiques sont largement utilisées pour la VRP générale. Cependant, il n'y en a pas beaucoup qui peuvent être applicables au problème abordé. Nous avons adopté un algorithme génétique pour résoudre le problème avec la nouvelle représentation chromosomique conçue et les opérateurs pour s'adapter au problème. Ces composants forment un nouvel algorithme génétique adapté au problème abordé. Cette méthode résout le problème en obtenant l'effet souhaité.

Les algorithmes *évolutionnaires rencontrent souvent le problème de prématuré. Pour faire face à ce problème, un nouvel algorithme génétique permettant l'exploration de la solution infaisable est proposé. Une procédure de contrôle de la diversité aide à éviter les prématurés. Les solutions infaisables sont réparées par un opérateur de réparation. Cette méthode peut résoudre les instances de grande taille avec une bonne efficacité.

Perspective

Afin de répondre aux besoins plus élevés des applications pratiques, il reste encore des aspects à étudier à l'avenir sur ce problème adressé.

Tout d'abord, cette thèse a fourni des bases théoriques sur le problème de tournées de véhicules. En raison du manque de cas qui correspondent au problème, des expérimentation numériques sont limitées. Il est nécessaire de modifier certains des problèmes de référence existants ou de créer de nouvelles instances pour faciliter l'étude du problème.

Deuxièmement, pour l'utilisation de la production du monde réel, il est essentiel de considérer les éléments dynamiques du problème, tels que les demandes d'urgence, le temps de déplacement stochastique et le temps de service, l'indisponibilité des véhicules, etc. Par conséquent, le problème de tournées dynamique devrait être étudié dans le futur.

Troisièmement, en tant que développement de nouvelles technologies, nouvelles informations pourraient être impliquées dans la résolution du problème. Par exemple, le GPS peut obtenir l'emplacement actuel de chaque véhicule et la technologie de télécommunication contribue à la communication en temps réel entre les clients et les techniciens. Ces informations sont très utiles pour l'optimisation en temps réel du problème de tournées de véhicules de service.

Bibliography

ALEGRE, J., LAGUNA, M. & PACHECO, J. (2007). Optimizing the periodic pick-up of raw materials for a manufacturer of auto parts. *European Journal of Operational Research*, **179**, 736–746.

ALONSO, F., ALVAREZ, M. & BEASLEY, J.E. (2008). A tabu search algorithm for the periodic vehicle routing problem with multiple vehicle trips and accessibility restrictions. *Journal of the Operational Research Society*, **59**, 963–976.

ARAS, N., AKSEN, D. & TEKIN, M.T. (2011). Selective multi-depot vehicle routing problem with pricing. *Transportation Research Part C: Emerging Technologies*, **19**, 866–884.

BALDACCI, R., CHRISTOFIDES, N. & MINGOZZI, A. (2008). An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, **115**, 351–385.

BALDACCI, R., BARTOLINI, E., MINGOZZI, A. & VALLETTA, A. (2011). An exact algorithm for the period routing problem. *Operations research*, **59**, 228–241.

BAÑOS, R., ORTEGA, J., GIL, C., FERNÁNDEZ, A. & DE TORO, F. (2013). A simulated annealing-based parallel multi-objective approach to vehicle routing problems with time windows. *Expert Systems with Applications*, **40**, 1696–1707.

BARCELÓ, J., GRZYBOWSKA, H. & PARDO, S. (2007). Vehicle routing and scheduling models, simulation and city logistics. In *Dynamic Fleet Management*, 163–195, Springer.

BASU, S. (2012). Tabu search implementation on traveling salesman problem and its variations: a literature survey.

Beaudry, A., Laporte, G., Melo, T. & Nickel, S. (2010). Dynamic transportation of patients in hospitals. *OR spectrum*, **32**, 77–107.

Belhaiza, S., Hansen, P. & Laporte, G. (2014). A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. *Computers & Operations Research*, **52**, 269–281.

Benavent, E. & Martínez, A. (2013). Multi-depot multiple tsp: a polyhedral study and computational results. *Annals of Operations Research*, **207**, 7–25.

Bianchi, L., Dorigo, M., Gambardella, L.M. & Gutjahr, W.J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, **8**, 239–287.

Blickle, T. & Thiele, L. (1995). A comparison of selection schemes used in genetic algorithms.

Blum, C. & Roli, A. (2003). Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Computing Surveys*, **35**, 189–213.

Borenstein, Y., Shah, N., Tsang, E., Dorne, R., Alsheddy, A. & Voudouris, C. (2010). On the partitioning of dynamic workforce scheduling problems. *Journal of Scheduling*, **13**, 411–425.

Bostel, N., Dejax, P., Guez, P. & Tricoire, F. (2008). Multiperiod planning and routing on a rolling horizon for field force optimization logistics. In *The vehicle routing problem: latest advances and new challenges*, 503–525, Springer.

Braekers, K., Caris, A. & Janssens, G.K. (2014). Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. *Transportation Research Part B: Methodological*, **67**, 166–186.

Braekers, K., Ramaekers, K. & Van Nieuwenhuyse, I. (2015). The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*.

Braekers, K., Ramaekers, K. & Van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, **99**, 300–313.

Bräysy, O. & Gendreau, M. (2001). Genetic algorithms for the vehicle routing problem with time windows. *Arpakannus,(1)*, 33–38.

Campbell, A.M. & Savelsbergh, M. (2004). Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation science*, **38**, 369–378.

Campbell, A.M. & Savelsbergh, M.W. (2005). Decision support for consumer direct grocery initiatives. *Transportation Science*, **39**, 313–327.

Cattaruzza, D., Absi, N., Feillet, D. & Vidal, T. (2014). A memetic algorithm for the multi trip vehicle routing problem. *European Journal of Operational Research*, **236**, 833–848.

Chen, C.H. & Ting, C.J. (2006). An improved ant colony system algorithm for the vehicle routing problem. *Journal of the Chinese institute of industrial engineers*, **23**, 115–126.

Chiang, W.C. & Russell, R.A. (1996). Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research*, **63**, 3–27.

Chiu, H.N., Lee, Y.S. & Chang, J.H. (2006). Two approaches to solving the multi-depot vehicle routing problem with time windows in a time-based logistics environment. *Production Planning & Control*, **17**, 480–493.

Clarke, G. & Wright, J.W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, **12**, 568–581.

Coelho, L.C., Cordeau, J.F. & Laporte, G. (2013). Thirty years of inventory routing. *Transportation Science*, **48**, 1–19.

Consulting, C. (2016). 2016 state of third-party logistics study.

Contardo, C. & Martinelli, R. (2014). A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optimization*, **12**, 129–146.

Cordeau, J.F. & Maischberger, M. (2012). A parallel iterated tabu search heuristic for vehicle routing problems. *Computers & Operations Research*, **39**, 2033–2050.

CORDEAU, J.F., LAPORTE, G. & MERCIER, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational research society*, **52**, 928–936.

DABLANC, L. (2013). Commercial goods transport, paris, france.

DANTZIG, G.B. & RAMSER, J.H. (1959). The truck dispatching problem. *Management science*, **6**, 80–91.

DAYARIAN, I., CRAINIC, T.G., GENDREAU, M. & REI, W. (2015). A branch-and-price approach for a multi-period vehicle routing problem. *Computers & Operations Research*, **55**, 167–184.

DESROCHERS, M. & LAPORTE, G. (1991). Improvements and extensions to the miller-tucker-zemlin subtour elimination constraints. *Operations Research Letters*, **10**, 27–36.

DIRAC, P. (1953). The lorentz transformation and absolute time. *Physica*, **19**, 888–896.

DU, L. & HE, R. (2012). Combining nearest neighbor search with tabu search for large-scale vehicle routing problem. *Physics Procedia*, **25**, 1536–1546.

FEYNMAN, R. & VERNON JR., F. (1963). The theory of a general quantum system interacting with a linear dissipative system. *Annals of Physics*, **24**, 118–173.

FISHER, M.L. & JAIKUMAR, R. (1981). A generalized assignment heuristic for vehicle routing. *Networks*, **11**, 109–124.

FRANCIS, P.M., SMILOWITZ, K.R. & TZUR, M. (2008). The period vehicle routing problem and its extensions. In *The vehicle routing problem: latest advances and new challenges*, 73–102, Springer.

FUKASAWA, R., LONGO, H., LYSGAARD, J., DE ARAGÃO, M.P., REIS, M., UCHOA, E. & WERNECK, R.F. (2006). Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical programming*, **106**, 491–511.

GAJPAL, Y. & ABAD, P.L. (2009). Multi-ant colony system (macs) for a vehicle routing problem with backhauls. *European Journal of Operational Research*, **196**, 102–117.

GARCIA-NAJERA, A. & BULLINARIA, J.A. (2011). An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows. *Computers & Operations Research*, **38**, 287–300.

GENDREAU, M. & TARANTILIS, C.D. (2010). *Solving large-scale vehicle routing problems with time windows: The state-of-the-art*. Cirrelt.

GHOSEIRI, K. & GHANNADPOUR, S.F. (2010). Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. *Applied Soft Computing*, **30**, 286–296.

GLOVER, F. (1989). Tabu searchâĂŤpart i. *ORSA Journal on computing*, **1**, 190–206.

GLOVER, F. & HAO, J.K. (2011). The case for strategic oscillation. *Annals of Operations Research*, **183**, 163–173.

GOLDEN, B.L., RAGHAVAN, S. & WASIL, E.A. (2008). *The vehicle routing problem: latest advances and new challenges*, vol. 43. Springer Science & Business Media.

GULCZYNSKI, D., GOLDEN, B. & WASIL, E. (2011). The period vehicle routing problem: New heuristics and real-world variants. *Transportation Research Part E: Logistics and Transportation Review*, **47**, 648–668.

HASLE, G., LIE, K.A. & QUAK, E. (2007). *Geometric Modelling, Numerical Simulation, and Optimization*. Springer.

HEMMELMAYR, V.C., DOERNER, K.F. & HARTL, R.F. (2009). A variable neighborhood search heuristic for periodic routing problems. *European Journal of Operational Research*, **195**, 791–802.

HO, W., HO, G.T., JI, P. & LAU, H.C. (2008). A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence*, **21**, 548–557.

HWANG, H.S. (2000). Integrated ga-vrp solver for multi-depot system. In *Int. Jnl. Of Computers and Industrial Engineering. 27th Conference Proceeding*, 1–6.

JIN, T., GUO, S., WANG, F. & LIM, A. (2004). One-stage search for multi-depot vehicle routing problem. In *Proceedings of the Conference on Intelligent Systems and Control*, 446–129.

KARAKATIČ, S. & PODGORELEC, V. (2015). A survey of genetic algorithms for solving multi depot vehicle routing problem. *Applied Soft Computing*, **27**, 519–532.

KHALIFA, I.H., EL KAMEL, A. & BARFETY, B. (2010). Real time indoor intelligent navigation system inside hypermarkets. *IFAC Proceedings Volumes*, **43**, 461–466.

KHALIFA, I.H., ZRIBI, N. & EL KAMEL, A. (2011). Itinerary optimisation approach inside hypermarkets. In *Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on*, 726–731, IEEE.

KHOUADJIA, M.R., SARASOLA, B., ALBA, E., JOURDAN, L. & TALBI, E.G. (2012). A comparative study between dynamic adapted pso and vns for the vehicle routing problem with dynamic requests. *Applied Soft Computing*, **12**, 1426–1439.

KINDERVATER, G.A. & SAVELSBERGH, M.W. (1997). Vehicle routing: handling edge exchanges. *Local search in combinatorial optimization*, 337–360.

KOCH, H., HENKE, T. & WÄSCHER, G. (2016). WORKING PAPER SERIES A Genetic Algorithm for the Multi-Compartment Vehicle Routing Problem with Flexible Compartment Sizes.

LABADI, N., PRINS, C. & REGHIOUI, M. (2008). A memetic algorithm for the vehicle routing problem with time windows. *RAIRO-Operations research*, **42**, 415–431.

LAPORTE, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European journal of operational research*, **59**, 345–358.

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

Lau, H.C.W., Chan, T.M., Tsui, W.T. & Pang, W.K. (2010). Application of Genetic Algorithms to Solve the Multidepot Vehicle Routing Problem. **7**, 383–392.

Laumanns, M., Thiele, L., Deb, K. & Zitzler, E. (2002). Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary computation*, **10**, 263–282.

Lim, A. & Wang, F. (2005). Multi-depot vehicle routing problem: A one-stage approach. *IEEE transactions on Automation Science and Engineering*, **2**, 397–402.

Lim, A. & Zhu, W. (2006). A fast and effective insertion algorithm for multi-depot vehicle routing problem with fixed distribution of vehicles and a new simulated annealing approach. *Advances in Applied Artificial Intelligence*, 282–291.

Lin, S.W., Vincent, F.Y. & Lu, C.C. (2011). A simulated annealing heuristic for the truck and trailer routing problem with time windows. *Expert Systems with Applications*, **38**, 15244–15252.

Liu, R., Xie, X. & Garaix, T. (2014). Hybridization of tabu search with feasible and infeasible local searches for periodic home health care logistics. *Omega*, **47**, 17–32.

Liu, Y. & El Kamel, A. (2015). The multi-period and multi-depot dynamic vehicle routing problem with time windows. In *International Conference on Logistics Operations Management*, IEEE.

Liu, Y. & El Kamel, A. (2017a). Genetic algorithm for multi-period and multi-depot vehicle routing problem with time window based on best insertion. In *International Conference on Advanced Logistics and Transport*, IEEE.

Liu, Y. & El Kamel, A. (2017b). Optimization for multi-period and multi-depot field service routing problem with time window. In *International Conference on Smart Grid and Smart Cities*, IEEE.

Macedo, R., Alves, C., de Carvalho, J.V., Clautiaux, F. & Hanafi, S. (2011). Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model. *European Journal of Operational Research*, **214**, 536–545.

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

Montané, F.A.T. & Galvao, R.D. (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research*, **33**, 595–619.

Montoya-Torres, J.R., Franco, J.L., Isaza, S.N., Jiménez, H.F. & Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, **79**, 115–129.

Mourgaya, M. & Vanderbeck, F. (2007). Column generation based heuristic for tactical planning in multi-period vehicle routing. *European Journal of Operational Research*, **183**, 1028–1041.

Nagy, G. & Salhi, S. (2005). Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European journal of operational research*, **162**, 126–141.

Ngueveu, S.U., Prins, C. & Calvo, R.W. (2009). A hybrid tabu search for the m-peripatetic vehicle routing problem. In *Matheuristics*, 253–266, Springer.

Ngueveu, S.U., Prins, C. & Calvo, R.W. (2010). An effective memetic algorithm for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, **37**, 1877–1885.

Ombuki, B., Ross, B.J. & Hanshar, F. (2006). Multi-objective genetic algorithms for vehicle routing problem with time windows. *Applied Intelligence*, **24**, 17–30.

Oppacher, F. & Wineberg, M. (1999). The shifting balance genetic algorithm: Improving the ga in a dynamic environment. In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1*, 504–510, Morgan Kaufmann Publishers Inc.

Osman, I.H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of operations research*, **41**, 421–451.

Park, T. & Ryu, K.R. (2010). A dual-population genetic algorithm for adaptive diversity control. *IEEE transactions on evolutionary computation*, **14**, 865–884.

Pillac, V., Gendreau, M., Guéret, C. & Medaglia, A.L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, **225**, 1–11.

Polacek, M., Hartl, R.F., Doerner, K. & Reimann, M. (2004). A variable neighborhood search for the multi depot vehicle routing problem with time windows. *Journal of heuristics*, **10**, 613–627.

Pongcharoen, P., Khadwilard, A. & Hicks, C. (2008). A genetic algorithm with a new repair process for solving multi-stage, multi-machine, multi-product scheduling problems. *Industrial Engineering and Management System*, **7**, 204–213.

Potvin, J.Y. & Bengio, S. (1996). The vehicle routing problem with time windows part ii: genetic search. *INFORMS journal on Computing*, **8**, 165–172.

Prodhon, C. (2008). A metaheuristic for the periodic location-routing problem. In *Operations Research Proceedings 2007*, 159–164, Springer.

Prodhon, C. (2011). A hybrid evolutionary algorithm for the periodic location-routing problem. *European Journal of Operational Research*, **210**, 204–212.

Rizzoli, A.E., Montemanni, R., Lucibello, E. & Gambardella, L.M. (2007). Ant colony optimization for real-world vehicle routing problems. *Swarm Intelligence*, **1**, 135–151.

RODRIGUE, J.P., COMTOIS, C. & SLACK, B. (2013). *The geography of transport systems*. Routledge.

ROSENKRANTZ, D.J., STEARNS, R.E. & LEWIS, P.M., II (1977). An analysis of several heuristics for the traveling salesman problem. *SIAM journal on computing*, **6**, 563–581.

SALHI, S. & NAGY, G. (1999). A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *Journal of the operational Research Society*, **50**, 1034–1042.

SALHI, S. & SARI, M. (1997). A multi-level composite heuristic for the multi-depot vehicle fleet mix problem. *European Journal of Operational Research*, **103**, 95–112.

SAP, M.N.M. (2002). Sweep algorithm in vehicle routing problem for public transport. *Jurnal Antarabangsa (Teknologi Maklumat)*, **2**, 51–64.

SHIMODAIRA, H. (1999). A diversity control oriented genetic algorithm (dcga): development and experimental results. In *Proceedings of GECCO'99*, vol. 1, 603–611.

SMOLIC-ROCAK, N., BOGDAN, S., KOVACIC, Z. & PETROVIC, T. (2010). Time windows based dynamic routing in multi-agv systems. *IEEE Transactions on Automation Science and Engineering*, **7**, 151–155.

SOLOMON, M.M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, **35**, 254–265.

STADLER, W. (2013). *Multicriteria Optimization in Engineering and in the Sciences*, vol. 37. Springer Science & Business Media.

STEFANOIU, D., BORNE, P., POPESCU, D., FILIP, F. & EL KAMEL, A. (2014). Optimization in engineering sciences. metaheuristics, stochastic method and decision support, iste.

TAGMOUTI, M., GENDREAU, M. & POTVIN, J.Y. (2011). A dynamic capacitated arc routing problem with time-dependent service costs. *Transportation Research Part C: Emerging Technologies*, **19**, 20–28.

Taş, D., Dellaert, N., Van Woensel, T. & De Kok, T. (2013). Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Computers & Operations Research*, **40**, 214–224.

Taş, D., Gendreau, M., Dellaert, N., Van Woensel, T. & De Kok, A. (2014). Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach. *European Journal of Operational Research*, **236**, 789–799.

Tasan, A.S. & Gen, M. (2012). A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries. *Computers & Industrial Engineering*, **62**, 755–761.

Tavakkoli-Moghaddam, R., Gazanfari, M., Alinaghian, M., Salamatbakhsh, A. & Norouzi, N. (2011). A new mathematical model for a competitive vehicle routing problem with time windows solved by simulated annealing. *Journal of manufacturing systems*, **30**, 83–92.

Thangiah, S.R. (????). Vehicle Routing with Time Windows using Genetic Algorithms. 1–24.

Toth, P. & Vigo, D. (2014). *Vehicle routing: problems, methods, and applications*. SIAM.

Tricoire, F. (2006). *Optimisation de tournées de véhicules et de personnels de maintenance: application à la distribution et au traitement des eaux*. Ph.D. thesis, Université de Nantes.

Tricoire, F. (2007). Vehicle and personnel routing optimization in the service sector: application to water distribution and treatment. *4OR*, **5**, 165.

Tsutsui, S., Fujimoto, Y. & Ghosh, A. (1997). Forking genetic algorithms: Gas with search space division schemes. *Evolutionary computation*, **5**, 61–80.

Ursem, R.K. (2002). Diversity-guided evolutionary algorithms. In *International Conference on Parallel Problem Solving from Nature*, 462–471, Springer.

Vaira, G. & Kurasova, O. (2014). Genetic algorithm for vrp with constraints based on feasible insertion. *Informatica*, **25**, 155–184.

Van Breedam, A. (1994). *An Analysis of the Behavior of Heuristics for the Vehicle Routing Problem for a Selectrion of Problems with Vehicle-related, Customer-related, and Time-related Constraints*. RUCA.

Vidal, T., Crainic, T.G., Gendreau, M., Lahrichi, N. & Rei, W. (2012). A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, **60**, 611–624.

Vidal, T., Crainic, T.G., Gendreau, M. & Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & operations research*, **40**, 475–489.

Villegas, J.G., Prins, C., Prodhon, C., Medaglia, A.L. & Velasco, N. (2010). Grasp/vnd and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, **23**, 780–794.

Vincent, F.Y., Lin, S.W., Lee, W. & Ting, C.J. (2010). A simulated annealing heuristic for the capacitated location routing problem. *Computers & Industrial Engineering*, **58**, 288–299.

Wen, M., Cordeau, J.F., Laporte, G. & Larsen, J. (2010). The dynamic multi-period vehicle routing problem. *Computers & Operations Research*, **37**, 1615–1623.

Wu, T.H., Low, C. & Bai, J.W. (2002). Heuristic solutions to multi-depot location-routing problems. *Computers & Operations Research*, **29**, 1393–1415.

Xiang, Z., Chu, C. & Chen, H. (2008). The study of a dynamic dial-a-ride problem under time-dependent and stochastic environments. *European Journal of Operational Research*, **185**, 534–551.

Xu, Y.L., Lim, M.H. & Er, M.J. (2005). Investigation on genetic representations for vehicle routing problem. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, 3083–3088, IEEE.

YU, B. & YANG, Z.Z. (2011). An ant colony optimization model: the period vehicle routing problem with time windows. *Transportation Research Part E: Logistics and Transportation Review*, **47**, 166–181.

YU, B., YANG, Z.Z. & YAO, B. (2009). An improved ant colony optimization for vehicle routing problem. *European journal of operational research*, **196**, 171–176.

YU, B., YANG, Z. & XIE, J. (2011). A parallel improved ant colony optimization for multi-depot vehicle routing problem. *Journal of the Operational Research Society*, **62**, 183–188.

ZHENG, D.X., NG, S.T. & KUMARASWAMY, M.M. (2005). Applying pareto ranking and niche formation to genetic algorithm-based multiobjective time–cost optimization. *Journal of Construction Engineering and Management*, **131**, 81–91.

Optimisation de Problème Tournées de Véhicules de Service à Domicile

Résumé: La performance logistique des entreprises et l'optimisation des transports sont devenues un grand problème ces dernières années. La planification et l'optimisation de la force de service constituent un nouveau défi pour le secteur des services. Afin d'accroître la productivité et de réduire les coûts de la logistique, cette dissertation contribue à l'optimisation d'un problème de tournéns de service à domicile multi-dépôt, multi-période avec fenêtres de temps de vie réelle. Le problème vient du problème réaliste et est formulé comme un modèle en Mixed Integer Programming (MIP). Les résultats avec Cplex montrent que ce problme ne peut être résolu par des méthodes exactes dans un délai raisonnable pour une utilisation pratique. Par conséquent, nous étudierons les heuristiques. Premièrement, les heurstiques de recherche locales sont utilisées pour résoudre le problème. Les solutions réalisables initiales sont générées par une heuristique de construction et plusieurs heuristiques de recherche locales sont appliquées pour obtenir des solutions dans un temps de calcul assez court. Ensuite, nous proposons un algoritme génétique avec une nouvelle représentation du chromosome et de nouveaux opérateurs génétiques pour le problème abordé. Enfin, nous considérons un algorithme génétique avec le contrôle de la diversité pour problèmes à grande échelle. Les solutions infaisables sont prises en compte dans la population et la contribution à la diversité fait partie de l'évaluation afin d'éviter une recherche prématurée. Ces méthodes ont été mises en œuvre avec succès pour optimiser le problème de tournées.

Mots-clés: Problème de tournées des véhicules, Algorithme génétique, Multi-dépôt, Multi-période, Fenêtres de temps, Tournées de service, Heuristique.

Optimization of Vehicle Routing Problem for Field Service

Abstract: The logistics performance of the enterprises and the optimization of transportation have become a great issue in recent years. Field force planning and optimization is a new challenge for the service sector. In order to increase productivity and reduce cost of logistics, this dissertation contributes to the optimization of a real-life multi-depot multi-period field service routing problem with time window. The problem is abstracted from the realistic problem and formulated as a Mixed Integer Programming (MIP) model. Computational results with Cplex show that this problem cannot be solved by exact methods in reasonable time for practical use. First, local search heuristics are used for solving the problem. Initial feasible solutions are generated by a constructive heuristic and several local search heuristics are applied to obtain solutions in a very short computing time. Then we propose a genetic algorithm with new representation of chromosome and new genetic operators for the addressed problem. Finally we consider a genetic algorithm with diversity control to deal with large scale problems. Infeasible solutions are taken account in the population and the diversity contribution is part of the evaluation to avoid premature of search. These methods have been successfully implemented to the optimization of the routing problem.

Keywords: Vehicle routing problem, Genetic algorithm, Multi-depot, Multi-period, Time windows, Service routing, Heuristics.