



**HAL**  
open science

# Studies on Collaborative Transportation: Models and Metaheuristics

Elham Jelodari Mamaghani

► **To cite this version:**

Elham Jelodari Mamaghani. Studies on Collaborative Transportation: Models and Metaheuristics. Infrastructures de transport. Université de Technologie de Troyes, 2019. English. NNT: 2019TROY0004. tel-03614869

**HAL Id: tel-03614869**

**<https://theses.hal.science/tel-03614869>**

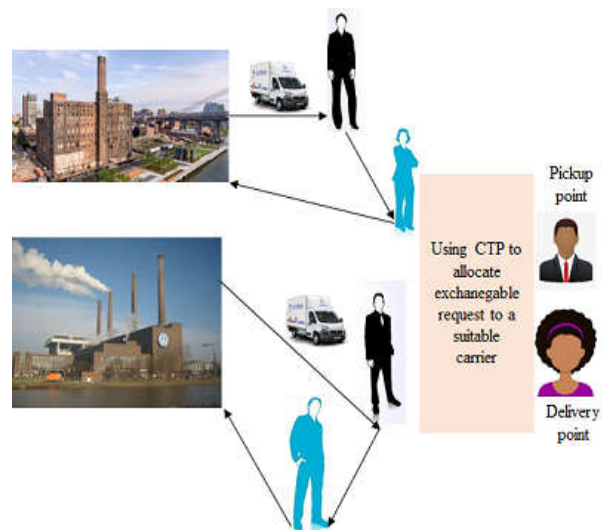
Submitted on 21 Mar 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**Elham JELODARI MAMAGHANI**

**Studies on Collaborative  
Transportation: Models and  
Metaheuristics**



**Champ disciplinaire :**  
Sciences pour l'Ingénieur

---

---

# THESE

*pour l'obtention du grade de*

## DOCTEUR

de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

EN SCIENCES POUR L'INGENIEUR

**Spécialité : OPTIMISATION ET SURETE DES SYSTEMES**

*présentée et soutenue par*

**Elham JELODARI MAMAGHANI**

*le 22 janvier 2019*

---

---

**Studies on Collaborative Transportation:  
Models and Metaheuristics**

---

---

## JURY

M. L. AMODEO  
Mme M.- A. MANIER  
M. M. IORI  
Mme W. RAMDANE CHERIF-KHETTAF  
M. H. CHEN  
M. C. PRINS

PROFESSEUR DES UNIVERSITES  
MAITRE DE CONFERENCE - HDR  
PROFESSORE ASSOCIATO  
MAITRE DE CONFERENCES  
PROFESSEUR DES UNIVERSITES  
PROFESSEUR DES UNIVERSITES

Président  
Rapporteure  
Rapporteur  
Examinatrice  
Directeur de thèse  
Directeur de thèse



کار نیست شنایی راز گل سخن  
کار مائید این است  
که در افون گل سخن شناور باشیم  
پشت دانایی اردو بزیم  
صبح باوقتی خورشید می آید متولد بشویم  
بیجان بار پرواز داریم  
باردانش را از دوش پرستو بر زمین بگذاریم  
نام را بازنشانییم از بر  
روی پای تریاران ببلندی محبت برویم  
در روی مشرد نور و گیاه و شمشیر باز کنیم  
کار مائید این است  
که میان گل نیلوفر و قرن  
پی آواز حقیقت برویم

*It is not our business to fathom the mystery of rose  
Perhaps our business is to float within the magic of the rose  
Camp behind wisdom  
And be born again when the sun rises in the mornings  
Let's allow our excitement to fly  
Let's lift down the burden of knowledge from the shoulders of the swallow  
Let's take back our name from the cloud  
From the plane tree, mosquito, summer  
Let's mount to the height of kindness of the wet feet of rain  
Let's open the door to mankind, light, plants and insects  
Our business is perhaps  
To run between the lotus flower and the  
Centrury after the sound of truth*

*Sohrab Sepehri*



## DEDICATION AND ACKNOWLEDGEMENTS

This work has been carried out within Laboratory of Industrial Systems Optimization (LOSI) at University of Technology of Troyes (UTT). It is funded by ANR (French Research Agent).

It is a genuine pleasure to express my deep sense of thanks and gratitude to my supervisors Prof. Haoxun Chen and Prof. Christian Prins for their highly valuable guidance and incessant support. I highly appreciate their patience during my study. The high knowledge given to me is the greatest reward of my endeavors.

I humbly extend my special thanks to Mme. MANIER and M.IORI for accepting review my PhD thesis, and Mme. Ramdane-CHERIF and M. AMODEO for agreeing to examine this thesis. Valuable remarks provided by the respectful experts in the field of collaborative logistics have helped me to improve the thesis' quality.

I would like to thank all researchers particularly Prof. YALAOUI and Prof. AMODEO Head of LOSI, secretaries and colleagues in LOSI, and the staffs of the doctoral school of UTT especially Mme. DENIS and Mme. LECLERCQ for their helps given to me in the past three years.

I would like to thank all my friends especially Nacef, Rezvan, Mohsen, Amer, Meisam, Elnaz, Fatemeh and other Iranian friends, Jin Rui, and all PhD students in LOSI for their helpful suggestions and encouragements during my stay in France.

Above all, I would like to express my extreme sincere gratitude and appreciation to my mother, father and my brother, Ehssan, for their supports, encouragements, and endless loves given to me in all my life.





## ABSTRACT

In collaborative logistics, multiple carriers form an alliance to improve their transportation operations and profitability by exchanging their transportation requests. In this thesis, we focus on carrier collaboration in less-than-truckload (LTL) transportation. More precisely three sub-problems of collaborative transportation planning are considered. Centralized multi-carrier collaborative transportation planning with pickup and delivery, time windows, exchangeable requests and reserved requests, multi-period bid generation problem with pickup and delivery, time windows, profits, reserved requests, and multi period bid generation problem with consideration of both cost and delivery lead time.

These sub-problems play a vital role in collaborative transportation planning among carriers, but in the literature, there is no in-depth study on them. We have presented new mathematical programming models for these problems and developed efficient heuristics to obtain solutions close to their optimums in a reasonable computation time. The heuristics are more efficient than commercial solver, CPLEX, not only in terms of solution quality but also in terms of computation time.

*key words : collaborative transportation planning, combinatorial auction, vehicle routing problem with pickup and delivery, periodic vehicle routing problem, multi-objective vehicle routing problem*



## RÉSUMÉ

Dans la logistique collaborative, plusieurs transporteurs forment une alliance pour améliorer leurs opérations et leur rentabilité en échangeant leurs demandes de transport. Dans cette thèse, nous nous concentrons sur la collaboration entre transporteurs dans le transport de chargement partiel (LTL). Plus précisément, trois sous-problèmes de la planification collaborative entre transporteurs sont pris en compte: un problème de planification centralisée de multi-transporteurs avec ramassage et livraison, fenêtres de temps, demandes échangeables et demandes réservées, un problème de génération d'enchères à plusieurs périodes, avec ramassage et de livraison, fenêtres de temps, profits, demandes réservées, et un problème de génération d'enchères à plusieurs périodes avec prise en compte à la fois du coût et du délais de livraison.

Ces sous-problèmes jouent un rôle essentiel dans la planification collaborative de transport entre transporteurs, mais dans la littérature, aucune étude profonde n'a été effectuée sur eux. Nous avons présenté de nouveaux modèles de programmation mathématique pour ces problèmes et développé des heuristiques efficaces pour obtenir des solutions proches de leurs optimums dans un temps de calcul raisonnable. Ces heuristiques proposées sont plus efficaces que le solveur commercial, CPLEX, non seulement en termes de qualité de solution mais aussi en termes de temps de calcul.

*mots clés: planification collaborative de transport, enchère combinatoire, problème de tournée de véhicule avec ramassage et livraison, problème de tournée de véhicules multi-objectif, problème de tournée de véhicules périodique*



## TABLE OF CONTENTS

	<b>Page</b>
<b>List of Tables</b>	<b>xiii</b>
<b>List of Figures</b>	<b>xv</b>
<b>1 General Introduction</b>	<b>1</b>
1.1 Key issues in carrier collaboration . . . . .	1
1.2 The problems studied in this thesis and contributions . . . . .	3
1.3 Conclusion . . . . .	6
<b>2 Literature review</b>	<b>7</b>
2.1 Vehicle Routing Problem with Pickup and Delivery and Time Windows . . . . .	9
2.2 Collaborative logistics . . . . .	9
2.3 Centralized planning approaches for collaborative transportation . . . . .	10
2.4 Decentralized planning approaches for collaborative transportation . . . . .	11
2.4.1 Auction-based Mechanisms . . . . .	11
2.4.2 Bid generation problems (BGP) . . . . .	12
2.5 Conclusion . . . . .	12
<b>3 Multi-carriers Pickup and Delivery requests with Time Windows, Reserved Requests and Exchangeable Requests</b>	<b>13</b>
3.1 Literature review . . . . .	15
3.2 Formal definition and mathematical model . . . . .	17
3.3 Solution procedure . . . . .	19
3.3.1 Construction of initial solution . . . . .	19
3.3.2 ALNS algorithm . . . . .	20
3.3.3 Hybrid genetic algorithm with simulated annealing (GASA) . . . . .	26
3.3.4 Improved tabu search (ITS) . . . . .	33
3.4 Numerical experiments . . . . .	34
3.4.1 Instances . . . . .	34
3.4.2 Parameter settings . . . . .	35

TABLE OF CONTENTS

---

3.4.3	Experimental results . . . . .	35
3.5	Conclusion . . . . .	48
<b>4</b>	<b>Multi-Periods Bid Generation Problem with Pickup and Delivery, Time Windows, Profits, and Reserved Requests</b>	<b>49</b>
4.1	Literature review . . . . .	50
4.2	Formal definition and mathematical model . . . . .	52
4.3	Metaheuristic approaches to solve multi-periods bid generation problem . . . . .	55
4.3.1	Initial solution construction procedure . . . . .	55
4.3.2	Hybrid genetic algorithm with simulated annealing (GASA) . . . . .	55
4.3.3	Biogeography-based optimization algorithm (BBO) . . . . .	60
4.4	Experimental results . . . . .	63
4.4.1	Instances . . . . .	63
4.4.2	Parameter setting . . . . .	64
4.4.3	Test results and optimality gaps . . . . .	64
4.5	Conclusion . . . . .	76
<b>5</b>	<b>Multi-Period Bid Generation Problem with Pickup and Delivery, Time Windows, Profits, Reserved Requests, and Delivery Lead Time</b>	<b>77</b>
5.1	Literature review . . . . .	78
5.2	Problem description and mathematical model . . . . .	79
5.3	Metaheuristic approaches to solve a single objective multi-period bid generation problem with consideration of delivery lead time . . . . .	83
5.4	Metaheuristic approaches to solve a bi-objective multi-period bid generation problem with consideration of delivery lead time . . . . .	83
5.4.1	Non-dominated sorting and crowding distance . . . . .	84
5.4.2	NSGA-II (Non-dominated Sorting Genetic Algorithm-II) implementation . . . . .	85
5.4.3	NRGA (Non-dominated Ranked Genetic Algorithm) implementation . . . . .	87
5.5	Computational study . . . . .	88
5.5.1	Comparison Metrics for Multi-Objective Optimization Algorithms . . . . .	88
5.5.2	Algorithm parameter tuning . . . . .	89
5.5.3	Results and Analysis . . . . .	92
5.6	Conclusion . . . . .	102
<b>6</b>	<b>Conclusions and Perspectives</b>	<b>103</b>
6.1	Conclusion . . . . .	103
6.2	Perspectives . . . . .	105
<b>A</b>	<b>Appendix of Extended French Abstract</b>	<b>107</b>
A.1	Les contributions de la thèse . . . . .	109

A.2	Problème de ramassage et de livraison avec plusieurs transporteurs, fenêtres de temps, demandes réservées et demandes échangeables . . . . .	112
A.2.1	Description et Formulation du Problème . . . . .	113
A.3	Problème de génération d'enchères sur plusieurs périodes avec demandes de ramassage et de livraison, profits et demandes réservées . . . . .	116
A.3.1	Description et modèle mathématique . . . . .	117
A.4	Problème de génération d'enchères sur plusieurs périodes avec prise en compte des délais de livraison . . . . .	120
A.4.1	Description et modèle mathématique . . . . .	121
A.5	Conclusion . . . . .	125
A.6	Perspectives . . . . .	127
	<b>Bibliography</b>	<b>129</b>





## LIST OF TABLES

<b>TABLE</b>	<b>Page</b>
3.1 Indices, sets, parameters for the mathematical model. . . . .	16
3.2 Decision variables for the mathematical model. . . . .	17
3.3 The adjustment of scores. . . . .	22
3.4 The detailes information of simple example to implement GASA. . . . .	28
3.5 Parameter values of ALNS determined empirically . . . . .	36
3.6 Parameter values of GASA and ITS determined empirically . . . . .	36
3.7 The domain of candidated parameters of GASA and ITS for calibration. . . . .	36
3.8 The domain of candidated parameters of ALNS for calibration. . . . .	37
3.9 Parameter values of GASA determined by Taguchi. . . . .	37
3.10 Parameter values of ITS determined by Taguchi . . . . .	37
3.11 Criteria used for comparison of ALNS, ITS, GASA, and CPLEX. . . . .	40
3.12 Computational results of ALNS, ITS, GASA and CPLEX – part one . . . . .	41
3.13 Computational results of ALNS, ITS, GASA and CPLEX – computation time. . . . .	42
3.14 Gap of ALNS, ITS, GASA and CPLEX. . . . .	44
3.15 Computational results (RPD) of ALNS, ITS and GASA. . . . .	45
3.16 The results of ANOVA test over the RPD of objective functions of algorithms. . . . .	47
4.1 Data of a simple instance. . . . .	56
4.2 The data for selective and reserved requests served in the first day. . . . .	56
4.3 The data for selective and reserved requests served in the second day. . . . .	57
4.4 The comparison of GA and BBO. . . . .	61
4.5 Parameter values of GASA determined empirically. . . . .	64
4.6 The domain of calibrated parameters of GASA and BBO. . . . .	65
4.7 Parameter values of GASA determined by Taguchi. . . . .	65
4.8 Parameter values of BBO determined by Taguchi. . . . .	66
4.9 Criteria used for comparison of GASA, BBO and CPLEX. . . . .	67
4.10 Computational results of GASA, BBO, and CPLEX –part one. . . . .	68
4.11 Gap of GASA, BBO, and CPLEX. . . . .	70
4.12 Computational time of GASA, BBO and CPLEX. . . . .	72

4.13	Computational result (RPD) of GASA and BBO. . . . .	74
4.14	The results of ANOVA test over the presented RPD for GASA and BBO. . . . .	75
5.1	The parameters of the presented mathematical model. . . . .	79
5.2	The decision variables of the considered model. . . . .	80
5.3	NSGA-II and NRGa parameter ranges and the levels of factors. . . . .	90
5.4	Taguchi procedure for NSGA-II. . . . .	91
5.5	Taguchi procedure for NRGa. . . . .	91
5.6	Taguchi response for NSGA-II and NRGa. . . . .	92
5.7	Computational results of GASA and ITS. . . . .	93
5.8	Computational results (RPD) and computation time of GASA and ITS . . . . .	94
5.9	The results of ANOVA test over the presented single objective mathematical model. . . . .	95
5.10	Multi-objective metrics computed for NRGa . . . . .	95
5.11	Multi-objective metrics computed for NSGA-II . . . . .	97
5.12	The results of ANOVA test over the studied bi-objective mathematical model. . . . .	99
A.1	Indices, ensembles, paramètres du modèle mathématique. . . . .	114
A.2	Variables de décision pour le modèle mathématique. . . . .	115
A.3	Les paramètres du modèle mathématique présenté. . . . .	122
A.4	Les variables de décision du modèle considéré. . . . .	122

## LIST OF FIGURES

FIGURE	Page
3.1 Illustration of chromosome's structure. . . . .	28
3.2 The single point crossover on two $X$ vectors. . . . .	29
3.3 The crossover on two $Z$ vectors. . . . .	30
3.4 The mutation operator over vector $Z$ . . . . .	31
3.5 Outputs of Taguchi ratio for ITS. . . . .	37
3.6 Outputs of Taguchi ratio for GASA. . . . .	38
3.7 Outputs of Taguchi ratio for ALNS ( $3 \times 5, 3 \times 8$ ). . . . .	38
3.8 Outputs of Taguchi ratio for ALNS ( $3 \times 15$ ). . . . .	39
3.9 Outputs of Taguchi ratio for ALNS ( $3 \times 30$ ). . . . .	39
3.10 Outputs of Taguchi ratio for ALNS ( $3 \times 50$ ). . . . .	40
3.11 Graphical comparisons of RPD for both algorithms on all test problems. . . . .	47
4.1 Illustration of chromosome's structurer . . . . .	57
4.2 Model of immigration $\lambda$ and emigration $\mu$ probabilities [90]. . . . .	62
4.3 Output of Taguchi ration for BBO. . . . .	65
4.4 Output of Taguchi ration for GASA. . . . .	66
4.5 Output of comparing GASA and BBO with RPD indicator. . . . .	76
5.1 Cuboid along locally non-dominated frontier [70]. . . . .	85
5.2 The crowding distance computation procedure of all solutions in a non-dominated set [70]. . . . .	86
5.3 Crowding distance for individuals $X1$ and $X2$ [70]. . . . .	86
5.4 The spacing and diversity metrics in bi-objective problems [70]. . . . .	89
5.5 Outputs of Taguchi ratio for NSGA-II. . . . .	91
5.6 Outputs of Taguchi ratio for NRGGA. . . . .	92
5.7 Graphical comparisons of Diversity metric for NSGA-II and NRGGA algorithms on all test problems. . . . .	98
5.8 Graphical comparisons of MID metric for NSGA-II and NRGGA algorithms on all test problems. . . . .	99

5.9	Graphical comparisons of Time metric for NSGA-II and NPGA algorithms on all test problems. . . . .	100
5.10	Graphical comparisons of Spacing metric for NSGA-II and NPGA algorithms on all test problems. . . . .	100
5.11	Graphical comparisons of NOS metric for NSGA-II and NPGA algorithms on all test problems. . . . .	101

## GENERAL INTRODUCTION

In 21 century, companies compete with each other to satisfy customers' increasing expectations and they have to reduce costs and improve profitability to survive in business world. On the one hand, customers want to have products with low prices and on the hand, the increasing fuel costs and other costs have negative effects on the profit margin of logistics providers and the final prices of products [11]. Forming collaborative networks and sharing resources and tasks are primary strategies for the companies to gain competitive advantages and capture more business opportunities. For this reason, carrier collaboration has attracted a growing interest of these logistics companies and has become an effective strategy for small to medium-sized carriers to reduce empty vehicles and increase vehicle fill rates [9]. Some implemented projects in USA disclose that Collaborative Transportation Management (CTM) (including carrier collaboration) can reduce the distance traveled by empty vehicles by 15%, the waiting time of vehicles by 15%, the turnover of drivers by 15% and can improve the fill rate of vehicles by 33% [17].

### 1.1 Key issues in carrier collaboration

Forming an alliance among multi-carriers at the same level of logistics operations is called carrier collaboration. In the carrier collaboration, multi-carriers in an alliance exchange their transportation requests to discover better assignment of requests among them and improve the profit of each carrier.

Such re-assignment of requests plays an important role in improving routing planning, decreasing empty back-hauls of vehicles and consequently reducing transportation costs. The optimal re-allocation of requests among carriers is one major problem in carrier collaboration. The objective of this problem is to maximize the total profit of all carriers in the alliance. Another problem in carrier collaboration is fair allocation of the profits obtained through the collaboration

among all carriers. This allocation of profits is usually made after collaborative transportation planning among carriers, which is called post-collaboration profit allocation. A fair allocation of such profits is necessary to guarantee the sustainability of the alliance [17].

These two problems or issues will be introduced in the following. In this thesis, we focus on the first problem which is also referred to as collaborative transportation planning (CTP) problem [56], [101].

***The requests re-assignment problem***

Carrier collaboration is usually realized by exchanging requests among multi-carriers. Through collaboration, small to medium-sized LTL carriers can increase asset utilization (reduce unused vehicle capacity) and strengthen their market positions. The challenge for such a collaborative effort is in finding a good balance between the requests assigned to each LTL carrier and its available transportation capacity to serve the requests. To ensure that such collaboration is sustainable, all participating carriers must benefit from it [46]. There are two different approaches in carrier collaboration: Decentralized planning approach and centralized planning approach.

In centralized planning approach, the collaboration among carriers is realized by exchanging some requests (called exchangeable requests) among carriers. In centralized planning approach, a central coordinator allocates all transportation exchangeable requests to minimize the post-collaboration transportation cost of each collaborative carrier. In addition to the selection of requests for each carrier, optimal routes to serve the requests must be determined. To put it more precisely, in a centralized planning approach, a central coordinator is responsible for optimally re-assigning all exchangeable requests among all carriers to maximize their total profit. In a transportation network, each vehicle tour of a carrier is defined by a circuit that starts with and ends at the depot node of the carrier. The combinatorial nature of requests re-assignment problem is driven the difficulty problem. In addition to the selection of requests for each carrier, optimal routes to serve the requests must be determined.

In decentralized planning approach, some protocols like auctions are utilized to the request allocation. Each carrier may organize the protocols by participating in various protocols to maximize its post-collaboration profit. In this approach, a central coordinator is similar to an auctioneer who organizes the protocols to maximize the re-assignment profit among coalition carriers. Auction-based approaches are the most important methods for CTP. An auction is a process of buying and selling goods or services by offering them up for bid, taking bids, and then selling the item to the winning bidder. As important category of auctions, combinatorial auctions allow bidders to bid on combinations of objects, tend to lead to more efficient allocations. The bid generation problem (BGP) is a key decision problem for auction-based decentralized planning approaches in CTP, which is considered from the perspective of each carrier. The BGP can also be considered as the request selection problem. The combinatorial nature of requests re-assignment problem is driven the difficulty problem.

Comparing the two approaches, the decentralized approach can preserve the decision-making

autonomy and the business data of each carrier confidentiality, whereas the total profit of all carriers generated by the centralized approach is usually higher than that generated by the decentralized approach, because of the global optimization nature of the centralized approach. The main concern of using a centralized approach for carrier collaboration is to keep the business data exchanged confidentiality among carriers. In recent years, with the appearance and development of Fourth-Party Logistics (4PL) providers who act as an integrator coordinating the transportation planning of multi-carriers without direct exchange of data among them, this concern is much eased. Moreover, it may be worthy for a carrier to reveal some information to the 4PLs in exchange of gaining a higher profit. Such a centralized planning approach is also used in collaborative transportation service trading in B2B e-commerce logistics [106].

In addition to the selection of requests for each carrier, optimal routes to serve the requests must be determined; since it is a type of the well-known NP-hard vehicle routing problems [101].

### ***The profit allocation problem***

After the requests re-assignment, the profits are allocated to all carriers in a fair way. Thorough this allocation, the post collaboration profit of each carrier is no less than the profit generated before collaboration and all carriers stay in the alliance. The profit allocation problem is a well known issue in the game theory. The big problem in this topic is, allocating the accepted profits for the all the carriers in alliance.

Inasmuch as there is not a unique definition for the fairness, the profit allocation problem is not unique allocation procedure. Thus, the design of an allocation approach is a difficult mechanism.

## **1.2 The problems studied in this thesis and contributions**

In collaborative logistics, multi-carriers form an alliance to improve their transportation operations and profitability by exchanging their transportation requests. In this thesis, we focus on carrier collaboration in less-than-truckload (LTL) transportation. More precisely, we study some sub-problems appeared in collaborative transportation planning realized by a centralized approach and an auction-based decentralized approach.

Firstly, we focus on centralized collaborative transportation planning (CTP) among carriers in a less-than-truckload transportation environment, where each request is a pickup and delivery request with time window for both pickup and delivery operations. Each carrier may have exchangeable requests and reserved requests, which must be self-served by the carrier and the collaboration among carriers is realized by exchanging unreserved requests among them. The problem becomes a multi-carriers pickup and delivery problem with time windows, two different types of requests, exchangeable and reserved requests, which is a new variant of NP-hard vehicle routing problems [97]. In spite of orienting problem which there is not compulsory to serve all re-

quests [10], in our problem all exchangeable requests have to be served. After presenting a mixed integer linear programming model to the problem, we propose an Adaptive Large Neighborhood Search (ALNS) algorithm, hybrid algorithm of Genetic and simulated Annealing algorithms (GASA) and Improved Tabu Search (ITS) to solve it. In [65], an ALNS algorithm is applied to solve a pickup and delivery problem with time windows, profits, reserved requests, and involving a single carrier with a single vehicle depot. In that problem which there is a single carrier, it is not compulsory to serve all exchangeable requests. The collaborative transportation considered in this thesis is a new problem involving multi- carriers with multi-depots. Furthermore, each reserved request must be served by its own carrier and all exchangeable and reserved requests must be served, the existing ALNS algorithms cannot be directly applied to our problem and some adaptations are necessary in order to solve our problem. The ALNS proposed in this thesis uses a variety of destroy/ repair operators. Its search procedure consists of multiple phases, where different phases use different policies for the choice of requests to destroy (delete) or repair (re-insert). The basic idea is to alter the behavior of destroy/repair operators over successive phase of the ALNS so it can explore different regions. In addition, local search based post-optimization is applied at the end of each phase in order to further improve the quality of the final solution obtained by the ALNS. The GASA takes advantages of genetic algorithm and simulated annealing at the same time (on the one hand, population-based criterion adds diversification and on the other hand, local search characteristic by applying SA prevents to be trapped in the local optimum). The Improved TS (ITS) adopts a mutation operator and keeps multiple solutions during the search process in order to get a better solution more quickly (Chapter 3).

Secondly, we study the multi-periods bid generation problem (BGP) for a carrier appeared in a multi-periods combinatorial auction for carrier collaboration. In real world applications, carriers usually plan their operations in a rolling horizon way. It means that each carrier determines which transportation requests to bid and serve in each period. Additionally, for exchangeable requests there is not fixed day to serve. Each request is allowed to be served within a period window consisting of multiple consecutive days. For example, goods ordered on-line by a customer on Monday in e-commerce is asked to deliver to the home of the customer within three days from Tuesday to Thursday. This gives rise to a multi-periods combinatorial auction (CA) problem. As a result, the bid generation problem of each carrier in such auction also involves multi-periods, where a bid is a combination of transportation requests over multi-periods [62]. Hence, in the problem, there are two different types of requests, reserved requests of the carrier and selective requests that are offered by other carriers and open for bid by the carrier. In selective requests, each pickup and delivery request is associated with two time windows. One is the period window which specifies the earliest period and the latest period the request must be served, whereas the other is a time window defining the earliest and latest service time of the request in each period. For each reserved request, there is predetermined period. It means that, each reserved request must be served in the fixed period. Moreover, both of the pickup and delivery request in reserved



request have their own time window to serve. A carrier must make two important decisions in the BGP: Which requests are chosen to bid and serve in their period window and how the routes are constructed to maximize its total profit. This leads to a new periodic pickup and delivery problem with time windows, profits and reserved requests with widespread use in the real application. A hybrid Genetic and Simulated Annealing algorithm and a Biogeography-Based Optimization algorithm are developed to solve the problem after presenting a MILP model. Hence, a hybrid approach that combines genetic algorithm and simulated annealing (GASA) is proposed to solve the problem. The GASA adopts a problem-specific chromosome structure, crossover and mutation operators well adapted to the chromosome structure. For the purpose of comparison, we also utilize a Biogeography-Based Optimization algorithm (BBO) that is a population-based algorithm and its specifications are close to the GA and its operation is over the population of solutions to search the best solution for the problem. By comparing GA and BBO, each chromosome in GA, which is considered as an individual with fitness value, is habitat with suitability index that is called HSI (Habitat Suitability Index) (Chapter 4). In reality, besides minimizing costs, shippers also pay attention to the reduction of the delivery lead time. Shorter delivery lead time makes orders can be delivered to customers faster. Therefore, to make bids competitive, carriers also need to consider delivery lead time as a quality specification when they construct their bids. For the auctioneer, it has two ways to evaluate each bid from carriers. One is evaluating a bid according to a score function that is defined as cost and delivery lead time as in a multi-attributes score auction. The other is evaluating a bid according to its multi-attributes (in our case, one attribute is price, the other attribute is the delivery lead time). Accordingly, each carrier has also two ways to submit its bid, one is submitting a score (cost and delivery lead time) and the other is submitting both attributes of a bid (cost and delivery lead time). Thus, to solve the problem, the multi-periods BGP becomes a bi-objective problem. There are two methods to deal with a bi-objective problem: The first approach is a weighted objective method, that is, we assign a weight to each objective and then transform two objectives into a single weighted objective. The other method is a bi-objective method, which is treated with all objective functions separately without transforming them into a single objective function. Consequently, two methods of treatment with the two assumed objectives lead to two problems: Single objective multi-periods BGP with delivery lead time and bi-objective multi-periods BGP with delivery lead time. After presenting MILP models to the both problems, to solve the single objective multi-periods BGP with pickup and delivery problem with time windows, profit, and delivery lead time, we apply GASA and ITS and to solve a bi-objective mathematical model formulation, two algorithms NSGAI and NRG are proposed to solve the model. The proposed algorithm to solve the bi-objective problem are compared based on a number of well-known metrics, as in Chapter 5 we refer to them.

### **1.3 Conclusion**

Collaborative logistics has attracted many attentions both in academic community and industrial practitioners in recent years because of its great significance in saving transportation costs. In this thesis, a carrier collaboration problem in pickup and delivery service with time windows and LTL transportation is studied. We develop a series of mathematical models and multiple solutions approaches for the problem. Chapter 1 mainly introduces our research background and generally describes the problems studied in this thesis. Chapter 2 provides an overview of current-art of collaborative transportation planning. A general review is given firstly to the field of collaborative logistics. Then the literature review focuses on significant contributions and important review papers on the centralized and decentralized planning approaches, especially the bid generation problem for carrier collaboration. Chapter 3 is devoted to solve the centralized collaborative transportation with pickup and delivery services, time windows and reserved requests. Chapter 4 focuses on multi-periods bid generation problem. Chapter 5 solves the Single objective and Bi-objective multi-periods BGP with delivery lead time concept. At last, Chapter 6 concludes this thesis and present the perspectives of future works opened in the field of collaborative transportation planning.

## LITERATURE REVIEW

The specialize of the freight logistics is in the cargo forwarding among places. In the last decades, which the use of e-commerce has increased dramatically, freight forwarding plays a vital role in daily economic activities. Hence, the freight logistics rapidly developments leads an intense competition among freight carriers. Furthermore, diminishing the profit margins of carriers, pivots on the increasing the fuel costs, enhances customer satisfaction and labor prices and reduces the life cycle of products [22]. So, for carriers, it is a real challenge to how to survive in such fierce competition environments. Collaborative Logistics (CL) or Collaborative Transportation Management (CTM) are developed to overcome these challenges. The horizontal cooperation that is active at the same level of SCM, can be defined among carriers/shippers [97]. Two types of horizontal collaborative logistics are specified in the literature: Shipper collaboration and carrier collaboration. The shipper collaboration contemplates on a single carrier and multi-shippers. In this type of collaboration, shippers consolidate all transportation requests to be afforded by a carrier. In the collaboration, asset repositioning costs are diminished by the shippers. Nevertheless, in the last years, most researchers have focused on the carrier collaboration. Actually, in carrier collaborations, there are opportunities for carriers to exploit the alliance among their transportation requests as on the one hand, reduce lead time and cost and on the other hand, improve the overall service levels [33]. The differences between carrier collaboration and shipper collaboration are explained in details in [77], [45]. Consequently, horizontal cooperation drives CL or CTM by either sharing transport capacities or transportation orders by making use of information technology, which has been developed in recent years. Thus, all involved actors can enhance their profitability by eliminating empty back-hauls; so the vehicles usage rates augments [23].

In the realistic logistics services, truckload (TL) transportation and less-than truckload (LTL)

transportation are the two types of logistics services provided to the customers. The movement of large amounts of homogeneous cargoes between two places is called TL shipping, while LTL refers to the small freights shipping among multiple places. For LTL shipment, customers pay according to the amount of occupied space of truck capacity, whereas in TL transportation, customers pay to the entire truck. Accordingly, in the case of shipping small freights, customers make use of this privilege and share the space of a truck with other customers. For small to medium sized enterprises without ability to provide TL shipping, LTL shipment is an ideal option. Transportation tools such as vans or vehicles are referred as truck in this thesis and henceforth the terms can be exchangeable. Exchanging (re-allocating) requests among carriers to minimize costs is a main issue in the carrier collaboration. To this reason, in the last years, industrial and academic centers take attention to the collaborative transportation planning (CTP) [101], [100], [102], [56]. Fairly allocate the post-collaboration profit among carriers through CTP, is the other issue to provide the sustainability. Although in this thesis we address the first issue in LTL mode, the readers can refer to the recent review in this issue [41].

Centralized and decentralized planning approaches are the two techniques for CTP in LTL. In centralized planning approaches, there is a central coordinator who is responsible to re-allocate (exchange) the transportation orders optimally among carriers inasmuch as their cost is minimized. In contrast, in decentralized planning approaches, there is no central coordinator to organize all plans of the carriers and each carrier performs like an independent agent. Totally, the main differences between centralized and decentralized planning approaches return to clarify all information of the orders of a carrier to other carriers in coalition. In the centralized planning approaches, all order information of each carrier is shared with other carriers in alliance, whereas in decentralized planning approaches only limited information is shared among coalition carriers. In realistic applications, although centralized approaches are often superior to decentralized methods regarding to minimize total cost, carriers are not only partners and competitors may be choose to conceal the information of customers' orders; so decentralized approaches are more practical [97].

The rest of this chapter will be organized as follows: A brief literature review on vehicle routing problems with pickup and delivery and time windows that are related to the studied subject in the thesis will be given in Section 2.1. A general literature review on collaborative logistics will be considered in Section 2.2. Since the two sub-problems studied in this thesis are related to the class of decentralized planning approaches and just one sub-problem applies the centralized approach, in Section 2.3 brief review on centralized planning approaches is given and a general review on decentralized planning approaches will be given in Section 2.4.

## 2.1 Vehicle Routing Problem with Pickup and Delivery and Time Windows

The latest studies in vehicle routing problems (VRP) and various specifications are considered in the book [40]. In that review chapter, the authors are presented variety of problems like heterogeneous vehicles and time window. They proposed mathematical models and heuristic approaches to solve the different types of vehicle routing problems. Vehicle routing problems often are known by some properties to take into account the specificities of real applications. Heuristics and meta-heuristic algorithms for Multi-Attributes Vehicle Routing Problems (MAVRP) are presented in [98] as a review study. One part of the studied problem in this thesis is close to the multi-depots vehicle routing problem with pickup and delivery requests and time windows (multi-depots VRPPDTW). Our problem is different from classical multi-depots VRPPDTW as in the assumed problem in this thesis, two requests are considered and it will be included the aspects of CTP. Therefore, it is beneficial to study some works in the multi-depots VRPPDTW. The mathematical model for multi- depots vehicle routing problems (multi-depots VRP) with time windows is suggested in [6]. In the paper, the mathematical models are applied in the delivery and in the installation of electronic equipment. The authors present mixed integer linear mathematical model (MILP) to minimize total costs. They solve the model by a genetic algorithm. The efficiency of the proposed solution is proved by reaching near-optimal solutions. A new type of multi-depots VRPPD with time windows is studied in [53]. In this type of VRP, the depot, where the vehicles return to it is flexible. An integer mathematical model to minimize the total travel cost is presented to the problem and a hybrid genetic algorithm with adaptive local search is proposed to solve. In the VRPPDTW problem studies by [84], the starting depot and destination depot of each request do not need to be the same. Metaheuristic algorithm to solve the problem is ALNS and the authors show the efficiency of the algorithm. In truth, ALNS algorithm is presented by [81] and it is used in the different fields of VRP because of its efficiency in acting on the neighborhood.

## 2.2 Collaborative logistics

In [22], a large-scale review on the opportunities and obstacles of horizontal collaboration in logistics is described. Generally, the advantages of horizontal collaboration on the quality of logistics services are strongly acceptable by logistics service providers (LSPs). Nevertheless, one of the expected obstacles in horizontal collaboration is a fair allocation of profit and it is considered as a hardest challenge. In [55], existing approaches to model and solve collaborative transportation and different challenges of the issue are provided. The problem is an expanded problem of vehicle routing problems as freights can be supplied by 4th party logistics. The mentioned survey paper indicates why there are gaps between theory and practice. Another prominent survey in the field of collaborative logistics is studied by [25]. At first, the opportunities

in the CTP are outlined and then the key issues in the coalitions like sharing resources, profits, information are discussed. Finally, the authors depicted the approaches of information protection and decisions technologies. Moreover, in the article, some business cases have been dedicated to support the problem. Another literature review in horizontal collaborative logistics is presented in [97]. In this study, the horizontal logistics collaboration is categorized into two methods: Order sharing and capacity sharing. For two methods, solution approaches are discussed in details. By applying order sharing approaches and property repositioning abilities, the transportation costs are reduced and the efficiency of transportation planning is enhanced. According to the existing studies, there are different approaches to order sharing and re-allocation of requests to optimize the logistics system. In the paper, authors classify different techniques in the order sharing: Joint route planning, auction-based mechanisms, bilateral lane exchanges, load swapping, and shipment dispatching approaches. Another method in collaborative transportation planning is sharing customer orders, which is provided by carrier collaboration and horizontal collaboration through sharing vehicles capacity. By making use of this way, on the one hand, empty backhauls is reduced and on the other hand, the capital expenses can be shared among all coalition carriers. Two general approaches are provided by previous studies to establish the most efficient way to share vehicle capacities. One of the approaches is mathematical programming and another is based on the negotiation protocols.

### **2.3 Centralized planning approaches for collaborative transportation**

For centralized planning approaches in collaborative logistics with TL transportation, the authors in [30] are proposed a Lane Covering Problem (LCP), which arises in the context of shipper collaboration. The object function of the model aims to find a minimum cost of directed cycles set (truck tours), not necessarily disjoint, covering a given subset of arcs (lanes) in a complete euclidean digraph. Effective algorithms and efficient implementations for solving the LCP are also developed in this paper. In later studies, the authors are considered some constrained variants of LCP, such as the cardinality constrained LCP [32], the length constrained LCP [30], the dispatch windows constrained LCP and driver restriction LCP [31]. The potential collaborative opportunities among carriers in TL transportation is investigated in [76]. In the proposed optimization models, carriers' repositioning costs may be reduced by exchanging transportation requests. Various exchange mechanisms are designed and evaluated, differing in terms of information sharing requirements and side payment options, which allow carriers to realize potential costs savings opportunities.

For centralized planning approaches in collaborative logistics with LTL transportation, a single-carrier collaboration problem (SCCP) is studied in [45], in which a carrier seeks to collaborate with other carriers in order to acquire extra transport capacity to service excess demand. The

SCCP is considered from a static planning perspective to gain insights into the potential benefits of the collaboration concept for carriers, and its ability to mitigate the consumption of fuel. The collaborative strategies are evaluated by computing the relative benefits of the collaboration over the non-collaboration situation. Single and multiple-products SCCPs are both formulated as binary (0-1) multi-commodities minimum cost flow problems, and the models are solved by a branch-and-cut algorithm.

## 2.4 Decentralized planning approaches for collaborative transportation

Auction-based methods are applied for decentralized planning in collaborative logistics. A protocol which is called auction, allows the agents to specify their interests in the resources and utilize their interest to assign the resources and payments among the agents. The winners of the bids are known as a NP-hard problem and combinatorial auction problem (CAP) is employed to determine the winners of the bids. Hence, combinatorial auction is specifically probed in a decentralized planning issue. To implement combinatorial auction in the collaborative transportation, there is a central coordinator who is responsible to all auctions taken place during the collaboration planning process.

In [59], a multi-rounds combinatorial auction technique is suggested in truckload transportation, where shippers permit to bids on the requests. In fact, in each round, a bid generation problem (BGP) is solved by each carrier to find the most profitable pack of requests. In this case, the winner determination problem (WDP) is solved by the auctioneer to allocate the requests to the carriers [92], [93], [63]. In CTP, the bid generation problem (BGP) is an important decision approach for auction-based decentralized planning issues and is considered from each carrier's viewpoint. To add to this fact, the BGP can also be used as a request selection strategy. Our study in chapter 4 and chapter 5 are related to the request selection problem, so the BGP will be reviewed in details in section 2.4.2.

### 2.4.1 Auction-based Mechanisms

The key strategies for CTP are the Auction-based approaches. The process of buying and selling products by proposing them for bid and selling the items to the winner bidder is called Auction. According to the literature, auctions are classified to four groups: Single-object auctions, multiple-objects auctions, combinatorial auctions (CA), and exchanges [89], [58]. Combinatorial auctions are the most popular and important strategy that allow bidders to bid on the combination of products and allocate them efficiently.

However, maximization revenue and determining the winners in CTP by combinatorial auctions is NP-hard problem [85].

### 2.4.2 Bid generation problems (BGP)

The BGP in carrier collaboration can be considered as deterministic BGPs and stochastic BGPs. In this thesis we focus on deterministic information; so deterministic BGPs will be exploited.

In [103], the concept of BGP in carrier collaboration transportation service is addressed. In the paper, the principle of bidder's optimality in combinatorial bids is discussed. A fleet assignment model and the nearest insertion method are the two different heuristics are developed as solution methods. The BGP in collaborative transportation is suggested in [59]. In the problem, carriers by using VRP, operate BGP on the sets of lanes based on the actual routes. To solve the model, column generation and Lagrangian based methods are used. In [12], two heuristic techniques and an exact strategy are developed to solve the bidding on the subsets of requests. The two heuristic bidding approaches determine combination of promising request, use the capacitated  $p$ -median problem and save values by utilizing cooperation. The suggested heuristic techniques can assist a carrier to raise the chance of winning and can reduce the computational challenges in participation among carriers. About the exact bidding approach, it is a technique based on the combinations of elementary request. In the article, it is shown bidding on each elementary request combination is sufficient.

## 2.5 Conclusion

In this chapter, we, at first, present the background of Collaborative Logistics (CL) or Collaborative Transportation Management (CTM). Then we describe two types of horizontal collaborative logistics: Shipper collaboration and carrier collaboration. Furthermore, two types of transportation service are discussed. Truckload (TL) transportation and less-than-truckload (LTL) transportation are the two types of transportation service. In this thesis, we focus on carrier collaboration in LTL transportation, where two core sub-problems should be addressed. These sub-problems are: Collaborative transportation planning (CTP) problem and profit allocation problem. We mainly concentrate on CTP in this thesis. At last, a general literature review on CL and centralized planning approaches of CTP, and a detailed literature review on decentralized planning approaches of CTP are given to close the chapter.



## MULTI-CARRIERS PICKUP AND DELIVERY REQUESTS WITH TIME WINDOWS, RESERVED REQUESTS AND EXCHANGEABLE REQUESTS

In this chapter, we study centralized planning approach for multi-carriers collaboration in a less-than-truckload transportation environment, where each carrier has a finite number of vehicles and each request is a pickup and delivery request with time windows. In this problem, each carrier may have reserved requests, which must be self-served by the carrier. In fact, two types of actors exist in our centralized planning approaches: Central coordinator and carriers. Multi-carriers come to an agreement for constituting a collaborative alliance with a central coordinator in charge of making collaborative transportation plans for them. The coordinator is a virtual actor who re-allocates the transportation requests of the alliance to all carriers. By exchanging some requests among them, such collaboration can help them to reduce transportation costs and increase profits. It can be realized via a Fourth-Party Logistics (4PL) provider who acts as an integrator coordinating the transportation planning among carriers.

In the carrier collaboration, several carriers form an alliance and exchange some transportation requests. Each carrier has a set of *reserved requests* (i.e. not proposed for exchange in CA/CE) and can serve additional requests (*exchangeable requests*) acquired from other carriers. Each request is a pickup and delivery request associated with an origin, a destination, a quantity, two time windows (pickup time window and delivery time window), and a price (revenue) for serving the request paid by its corresponding shipper (customer). In spite of orienting problem which there is not compulsory to serve all requests [10], in our problem all exchangeable requests have to be served. Such a problem raises a new variant pickup and delivery requests with time windows (PDPTW), i.e. Multi-Carriers Pickup and Delivery requests with Time Windows and Reserved requests (MCPDPTWR). To the best of our knowledge centralized collaborative transportation planning in LTL among multi-carriers, where each request is a pickup and de-

livery request with time Window for both pickup and delivery requests and reserved requests has not been studied in the literature. So, in this chapter, we try to fill this gap by presenting a mixed integer linear programming model to the problem and developing effective heuristic algorithms. We propose an Adaptive Large Neighborhood Search (ALNS) algorithm to solve which is successfully applied to solve various vehicle routing problems [84]. In [65], an ALNS algorithm is applied to solve a pickup and delivery problem with time windows, profits, reserved requests, and involving a single carrier with a single vehicle depot. In that problem, it is not compulsory to serve all selective requests. The vehicle routing problem considered in this thesis is a new problem involving multi-carriers with multi- depots vehicle routing problem. Furthermore, each reserved request must be served by its own carrier and all requests must be served by all carriers. It means that, all exchangeable requests must be served by a carrier. Therefore, the existing ALNS algorithms cannot be directly applied to our problem and some adaptations of them are necessary in order to solve our problem. The ALNS proposed in this work uses a variety of destroy/ repair operators. Its search procedure consists of multiple phases, where different phases use different policies for the choice of requests to destroy (delete) or repair (re-insert). The basic idea is to alter the behavior of destroy/repair operators over successive phase of the ALNS; so it can explore diverse regions. In addition, local search based post-optimization is applied at the end of each phase in order to further improve the quality of the final solution obtained by the ALNS. For the purpose of comparison, we also develop a Hybrid Genetic Algorithm combined with Simulated Annealing (GASA). The GASA adopts a problem-specific chromosome structure, crossover and mutation operators well adapted to the chromosome structure. We also develop an Improved Tabu search (ITS) approach. This approach improves classical Tabu search by keeping multiple solutions during the search process, collecting and using the information about them in order to improve the current solution more quickly [75]. Moreover, we use a mutation tactic which is another technique to improve the performance of this algorithm [52]. The performance of the proposed algorithms are evaluated by comparing them with commercial solver CPLEX on 50 benchmark instances in [17]. Numerical results show that ALNS, GASA, and ITS algorithms can obtain a better solution in a computation time much less than CPLEX commercial optimization solver. CPLEX could not even find a feasible solution after a limited running time for medium and large instances. Furthermore, the results show the cost obtained by ALNS and its running time are better than the cost and the running time achieved by GASA and ITS. This chapter is organized as follows. In Section 3.1, literature review on collaborative logistics and vehicle routing problems related to our work is given. The description of the problem and a mathematical model are provided in Section 3.2. Section 3.3 is devoted to the description of the applied ALNS, GASA, and ITS algorithms. In section 3.4, numerical results of comparing ALNS, GASA, and ITS algorithms with CPLEX solver on benchmark instances are reported and analyzed. Finally, Section 3.5 closes this chapter.

### 3.1 Literature review

Various types of cooperative relationships in the field of transportation and logistics have been discussed in both professional and academic literature. Both vertical cooperation and horizontal cooperation in supply chains are the focus of various research efforts over the last two decades. In recent years, horizontal collaboration which takes place among multiple firms at the same level of supply chain emerged as a promising framework to implement collaborative logistics [57], [102], [23]. Due to the increase in fuel costs, transportation companies are forced to use their vehicles more efficiently, which is given rise to CL or Collaborative Transportation Management (CTM). Driven by cost reduction requirements due to fierce competition in the transportation market and increased environmental concerns, improving operational efficiency through resource sharing and collaborative transportation planning is becoming a new business model for transportation industry [25]. This is why collaborative logistics has received much attention in recent years. As mentioned earlier, carrier collaboration in LTL can be realized by using two different types of approach, centralized and decentralized. For a centralized planning approach used in LTL collaborative logistics, a coalition multi-commodities game is studied in [68]. Players have a capacity and are the nodes of a given transportation network. A capacity defines the maximum flow permitted through the node. The optimal flow is obtained by solving a centralized planning problem and the total revenue is allocated to players in a fair way. A time-dependent centralized multi-carriers collaboration problem is studied in [46], where a central entity seeks for minimizing the total costs of all carriers. They formulate the problem as a binary multi-commodities minimum cost flow model, and solve it by a branch and cut algorithm. The obtained results confirm the attractiveness of the centralized cooperation approach in the increase of vehicle capacity utilization and the decrease of empty-haul trips for carriers. In decentralized planning approaches, there is a central coordinator takes charge of all auction activities taking place in an auction-based collaborative planning process. Where an auction is a protocol and allows agents to indicate their interests in one or more resources, uses these indications of interest to determine both an allocation of resources and an allocation of payments among the agents [23]. For combinatorial auction applied to a collaboration among independent freight forwarding entities, a decentralized combinatorial auction model is proposed in [57]. The problem studied in this thesis is related to multi-depots vehicle routing problems with pickup and delivery requests and time windows (multi-depots VRPPDTW). There are not various papers in this field and most VRP articles are concentrated on VRP with each specification. In [91] Capacitated vehicle routing problem with pickup and delivery is presented. The authors focus on alternative delivery node and after presenting mathematical model, apply hybrid heuristic approach to solve the model. Our problem is different from them and from classical multi-depots VRPPDTW because of a new feature, i.e., the reserved requests of each carrier must be assigned to its own depot since these requests must be served by the carrier itself [53]. In this type of VRP, the depot, where the vehicles return to it is flexible. An integer mathematical model to minimize the total travel cost is presented to

the problem under constraints of VRP. A hybrid genetic algorithm with adaptive local search is proposed to solve. In their problem, all customers have their own time windows, all routes start and finish at the same depot. Some articles in VRPPDTW are considered in the following. A mathematical model is studied in [84] for the problem, whereas the starting depot and destination depot of each request do not need to be the same. Metaheuristic algorithm for the model is ALNS and the authors prove that the efficiency of algorithm is very high. ALNS algorithm is presented by [84] and it is used in different fields of the VRP because of its efficiency in acting on the neighborhood.

**Table 3.1:** Indices, sets, parameters for the mathematical model.

<b>Symbols</b>	<b>Description</b>
$G = (V, E)$	Complete directed graph G on which the studied problem is defined
$V$	Set of all nodes
$E$	Set of all edges
$n$	Number of requests
$cr$	Maximum number of carriers
$M$	Set of carriers
$DC$	Set of depot nodes of carriers
$p = \{1, 2, \dots, n\}$	Set of pickup nodes, where the index of each pickup node $i$ is the same as the index of its corresponding request $i$ , $i = 1, \dots, n$
$D = \{n + 1, \dots, 2n\}$	Set of delivery nodes of all requests, where $n$ is the number of requests
$W$	Set of nodes excluding the depot nodes of all carriers
$R = \{1, \dots, n\}$	Set of requests that includes the subset of reserved requests $R_r$ and the subset of exchangeable requests $R_s$
$R_r^m$	Set of reserved requests of carrier $m$
$R_s$	Set of exchangeable requests
$Q$	Vehicle capacity
$c_{ij}$	Traveling cost from node $i$ to node $j$
$s_i$	Service time of node $i$
$K_m = \{1, \dots, VK_m\}$	Set of vehicles of carrier $m$ , where $VK_m$ is the maximum number of vehicles of carrier $m$
$d_{n+i} = -d_i$	Demand of the pickup node and the delivery node of request $i$ , where $d_i$ is the demand of the pickup node and $d_{n+i}$ is the demand of the delivery node, $d_{n+i}$ is negative
$[e_i, l_i]$	Time window of node $i$ , where $e_i$ is the lower bound of time window and $l_i$ is the upper bound of the time window
$t_{i,j}$	Traveling time from node $i$ to node $j$ , it is assumed equal to traveling time
$T$	Time duration of each route
$BM_{i,j} = l_j - e_i$	Used in time window constraints
$CV_i = Q + d_i$	Used in vehicle capacity constraints

**Table 3.2:** Decision variables for the mathematical model.

Variables	Description
$x_{ijkm}$	Binary variable equal to 1 if and only if vehicle $k$ of carrier $m$ uses edge $(i, j)$
$y_{ikm}$	Binary variable equal to 1 if and only if request $i$ is served by vehicle $k$ of carrier $m$
$U_{ikm}$	Arriving time of vehicle $k$ of carrier $m$ at node $i$
$L_{ikm}$	Load of vehicle $k$ of carrier $m$ when leaving from node $i$

### 3.2 Formal definition and mathematical model

In our problem, there are multi-carriers with two types of requests: Reserved requests and exchangeable requests. The reserved requests of each carrier must be served by the carrier itself, whereas its exchangeable requests can be served by other carriers. The collaboration among carriers is realized by exchanging requests among them. Each carrier has a limited number of vehicles initially located at a depot. It is assumed that all vehicles are homogeneous. Each transportation request is a pickup and delivery request characterized by a pickup location with a time window, a delivery location with a time window too, a demand and an amount of revenue that can be generated for serving the request. The revenue of a request is the price paid by a customer (shipper) to a carrier serving the request. The objective of the problem is to optimally re-allocated all exchangeable requests among carriers to minimize their total transportation cost to serve all reserved requests and exchangeable requests. We adopt a centralized approach to solve the problem. To do so, we first formulate this problem as a MILP. Table 3.1 and Table 3.2 list the indices, parameters, sets, parameters and decision variables used in the model sequentially. The problem can be formulated as the following mixed integer-programming model:

$$\min \sum_{m \in M} \sum_{i \in V} \sum_{j \in V} \sum_{k \in K_m} c_{ij} x_{ijkm}$$

subject to:

$$\sum_{j \in V, j \neq i} x_{jikm} - \sum_{j \in V, j \neq i} x_{ijkm} = 0 \quad \forall i \in W, \forall k \in K_m, \forall m \in M \quad (3.1)$$

$$\sum_{m \in M} \sum_{j \in V, j \neq i} \sum_{k \in K_m} x_{ijkm} = 1 \quad \forall i \in W \quad (3.2)$$

$$\sum_{j \in P, j \neq i} x_{ijkm} \leq 1 \quad \forall k \in K_m, \forall i = m \in M \quad (3.3)$$

$$\sum_{j \in P, j \neq i} x_{ijkm} = 0 \quad \forall k \in K_m, \forall i, m \in M, i \notin \{m\} \quad (3.4)$$

$$\sum_{j \in D, j \neq i} x_{ijkm} \leq 1 \quad \forall k \in K_m, \forall i = m \in M \quad (3.5)$$

$$\sum_{j \in D; j \neq i} x_{ijkm} = 0 \quad \forall k \in K_m, \forall i, m \in M, i \notin \{m\} \quad (3.6)$$

$$\sum_{k \in K_m} y_{ikm} = 1 \quad \forall i \in R_m, \forall i, m \in M \quad (3.7)$$

$$\sum_{k \in K_m} y_{ikm} = 0 \quad \forall i \in R, i \notin R_r^m, \forall i, m \in M \quad (3.8)$$

$$\sum_{m \in K_m} \sum_{m \in M} y_{ikm} = 1 \quad \forall i \in R_s \quad (3.9)$$

$$\sum_{j \in V, j \neq i, m'} x_{ijkm} = y_{ikm} \quad \forall i \in P, k \in K_m, \forall m \in M \quad (3.10)$$

$$\sum_{j \in V, j \neq i, m'} x_{j(i+n)km} = y_{ikm} \quad \forall i \in P, k \in K_m, \forall m \in M \quad (3.11)$$

$$U_{ikm} + t_{i(n+i)} + s_i \leq U_{(n+i)km} \quad \forall i \in P, k \in K_m, \forall m \in M \quad (3.12)$$

$$U_{jkm} \geq U_{ikm} + t_{ij} + s_i - BM_{ij}(1 - x_{ijkm}) \quad \forall i \in V, k \in K_m, \forall m \in M, \forall j \in W \quad (3.13)$$

$$e_i \leq U_{ikm} \leq l_i \quad \forall m \in M, \forall i \in W \quad (3.14)$$

$$U_{ikm} + t_{ij} + s_i - BM_{ij}(1 - x_{ijkm}) \leq T \quad \forall i \in W, k \in K_m, \forall m \in M, \forall j \in M \quad (3.15)$$

$$L_{ikm} + d_i - Q(1 - x_{ijkm}) \leq L_{jkm} \quad \forall i \in V, k \in K_m, \forall m \in M, \forall j \in V \quad (3.16)$$

$$\max\{0, d_i\} \leq L_{ikm} \leq \min\{Q, CV_i\} \quad \forall i \in V, k \in K_m, \forall m \in M \quad (3.17)$$

$$\sum_{j \in W} \sum_{k=1} x_{jikm} \leq K_m \quad \forall i \in M, \forall m \in M \quad (3.18)$$

$$x_{ijkm} \in \{0, 1\} \quad \forall i, j \in V, \forall k \in K_m, \forall m \in M \quad (3.19)$$

$$y_{ikm} \in \{0, 1\} \quad \forall i \in R, \forall k \in K_m, \forall m \in M \quad (3.20)$$

$$U_{ikm} \geq 0 \quad \forall i \in V, \forall k \in K_m, \forall m \in M \quad (3.21)$$

$$L_{ikm} \geq 0 \quad \forall i \in V, \forall k \in K_m, \forall m \in M \quad (3.22)$$

The goal is to minimize the total transportation cost of all carriers. The constraint (3.1) is the flow preservation equation and describes each vehicle arrives at a customer node must leave it. Equation (3.2) describes each customer node is visited exactly once by a vehicle. Constraints (3.3)-(3.6) indicate for each carrier when each vehicle leaves its depot should return to the same depot. Constraints (3.7)-(3.9) are about the reserved and exchangeable requests. Equations (3.7)-(3.8) ensure that all reserved requests of each carrier must be served by the carrier itself, whereas equation (3.9) implies each exchangeable request can be served by any carrier. Constraints (3.10) and (3.11) satisfy if a request is served, there must be a vehicle leaving its pickup node and arriving at its paired delivery node. The next constraint (3.12) is time windows constraint for pickup and delivery nodes. Constraint (3.13) shows time feasibility, i.e., vehicle  $k$  cannot start serving node  $j$  before completing its service at node  $i$  and traveling from node  $i$  to node  $j$ . Time window restriction at each customer node is imposed by constraint (3.14). Equations (3.15), (3.16) and (3.17) ensure the maximum duration time of each route, the respect of vehicle capacity and the maximum number of vehicles for each carrier. The rest of the constraints define the variables.

### 3.3 Solution procedure

#### 3.3.1 Construction of initial solution

To speed up the convergence, construct a good initial solution is important with some modifications to provide service for all exchangeable requests. The most important hypothesis is allocating all reserved requests and exchangeable requests to all carriers. Therefore, in the implementation of heuristic initial solution this specification should be considered. In the proposed heuristic, firstly, each carrier self-served to reserved requests. After that, each exchangeable request is inserted into an existing route of a carrier according to a chosen policy. Before assigning requests to the routes of any carrier, all reserved requests and all exchangeable requests are placed respectively in two lists, ordered according to their profits. Both reserved requests and exchangeable requests are assigned one by one in terms of order in their list. After assigning all reserved requests to the created routes of their own carriers, exchangeable requests are assigned to the existing routes of all carriers via the following two policies:

*Policy one:* After assigning all reserved requests of each carrier to its own routes, each exchangeable request is assigned to a carrier that both pickup and delivery nodes of the exchangeable request have a minimum distance to the depot of a carrier.

*Policy two:* After assigning all reserved requests of each carrier, each exchangeable request is allocated to a route of the best carrier with the lowest insertion cost.

Although, the first policy is a basic rule to assign exchangeable requests, there is a probability to set in local optimum. In comparison with first policy, the second policy is better to allocate exchangeable requests. An exchangeable request is assigned to a depot of a carrier according

to the priority of a carrier. Total distance of pickup and delivery nodes of each exchangeable request to each carrier is calculated. The carrier with the minimum distance has first priority to allocation; second priority is related to a carrier with the distance larger than the minimum distance to the exchangeable request and so on. Each exchangeable request is allocated to the carriers according to their priorities. In fact, by allocating each exchangeable request to all carriers, there is a chance to select the best carrier. After assigning, we calculate the value of objective function, total cost, due to the constraints. The carrier with minimum cost is the exchangeable request carrier.

Globally, the first policy is not better than the second policy, but it gives a chance to first possible position to reach a better solution in a reduced time. While policy 2 is more acceptable to reach a better solution, this policy needs more time.

### 3.3.2 ALNS algorithm

The quality of a local search heuristic depends on how the neighborhood of each solution is defined. The larger neighborhood and the higher quality of the final solution may be found by the heuristic. When a problem has tight constraints, search a small neighborhood may fail to explore the solution space thoroughly, because in this case the local search may be trapped in a local optimum of poor quality [1]. Large Neighborhood Search (LNS), which proposed in [88] for vehicle routing problem with time windows, one ruin operator and one repair operator are alternatively selected in each iteration. Each ruin operator removes some customers from the routes of current solution. In contrast, the role of repair operators is transforming a partial solution into a complete solution by re-inserting the removed customers into the routes of the partial solution. ALNS has been extended LNS by adding an adaptation layer. This layer adaptively chooses in each iteration a ruin operator and an insertion operator among a set of operators to intensify the search. Moreover, simulated annealing is used in the high layer of ALNS and allows a worse solution is accepted under some conditions to diversify the search. Thus, ALNS can be considered an improved version of LNS.

At each iteration, a pair of ruin and repair operators are selected according to the historical performances of all operators, where a score, which is updated, or keeps unchanged in each iteration, measures the performance of each operator. In other words, the probability to select an operator in each iteration depends on its score. Although the choice of this pair of operators is randomized, the historical performances help to choose the most effective pair of ruin and repair operators, because a level (layer) of adaptation is added over the local search level of ALNS in this procedure. The ALNS algorithm proposed in this thesis can be considered an adapted version of ALNS in [65], [84]. The procedure of the ALNS algorithm is outlined with pseudo-code in *Algorithm 1*.

In the current section, we described the ALNS procedure and how it is adopted to our problem. Firstly, multiple ruin and repair operators are designed by considering reserved requests



---

**Algorithm 1.** Pseudo-code of the ALNS algorithm.

---

```

1  In each nruns runs (nruns is the maximum number of runs allowed)
2    Construct an initial solution
3    Set the best solution  $S_{best}$  to the initial solution
4    Initialize starting temperature  $T_0$ 
5    Initialize the weights and scores of all ruin/repair operators
6    In each nseg segments; nseg is the maximum number of segments allowed
7      Choose the polices for the selection of requests
8      In each niters iteration(niters is the maximum number of iterations allowed)
9        If there is not improved solution found,apply operators of ALNS
10       Use SA rule
11       Update the performance scores of the selected operators
12       Update temperature of SA ( $T_{in} \leftarrow T_{in} \times co$ )
13     End of iteration
14     Update the weights and scores
15     Call LS(Section 4.6)
16   End of segment
17   Update global best solution
18 End of runing

```

---

and exchangeable requests. Secondly, we use heuristic initial solution by multiple restarts of ANLS to diversify the search in order to obtain a high quality solution. Finally, to enhance the intensification, local search with four moves is applied at the end of each phase to further improvement of current solution. All parts of the algorithm are described in the following sections.

### 3.3.2.1 Adaptive choice of ruin/repair operators

The ALNS chooses one ruin and one repair operator in each iteration according to the past performances of all operators. It gives a chance to all possible ruin/repair operators to be selected by the algorithm, even for operators of low performance, by using the roulette wheel method. Suppose there are in total  $c$  operators numbered from 1 to  $c$ . Each operator  $i = (1, 2, \dots, c)$  is associated with a weight  $W_i$ . The weight reflects the performance of this operator in the past. Thus, the probability of choosing each operator  $i$  is  $\frac{W_i}{\sum_{j=1}^c W_j}$ .

At the beginning of algorithm, the weight  $W_i$  of each operator  $i$  is set to one. It is adjusted at the end of each phase according to its performance score. For all scores, the weights of all operators are updated at the end of each phase in the following way. At the beginning of each phase, the scores of all operators are set to zero. By this setting, all operators including inefficient operators in the previous phase have a chance to be selected. Four scores denoted by  $s_1, s_2, s_3$  and  $s_4$ , which correspond to four different situations are defined and their update in different situations are described in Table 3.3. In the first one, a new best solution is found; this is the most favorable case. About second and third situations, both them are also favorable because

the obtained solution improves the current solution. Comparing second and third situation, the second one is more attractive since it explores a new region of the solution space. In fact, in the second situation, the algorithm finds a solution with new features. To know our algorithm has found a new solution, storing some attributes of the solutions previously found is compulsory. These attributes can be the number of requests served by the available vehicles, the number of vehicles used and the total profit. Hence, there is a list to store all profits in an increasing order and dichotomy search is used to check a solution with a given cost. Finally, fourth type of score brings some diversification to the search as it prevents from using the same operators frequently. This score depends on the acceptance criteria of SA used in the algorithm. Let  $W_{i,se}$  be the weight of operator  $i$  in  $se$  phase,  $tc_{i,se}$  be the number of times that operator is selected by the algorithm,  $fc_{i,se}$  is the final score of operator  $i$  at the end of phase  $se$ .  $rt \in (0,1)$  is a reaction factor and indicates which degree of weight in each operator reacts to its performance. If  $rt = 1$ , the current weight of each operator is disregarded and its new weight only depends on its achieved score in the last phase. If  $rt = 0$ , the current weight of each operator is preserved while its achieved score is overlooked.

**Table 3.3:** The adjustment of scores.

Score	Condition for score increment
$s_1$	Finding a new best solution.
$s_2$	Finding a new solution which improves the current solution.
$s_3$	The solution found is not new but improves the current solution.
$s_4$	The solution found is accepted by SA although it is worse than the current solution.

By mentioned four scores at the end of each phase, the weights of all operators are updated in each phase according to the following equation (3.23).

$$w_{i,se+1} = 1 - rt \times w_{i,se} + rt \times \frac{fc_{i,se}}{tc_{i,se}} \quad (3.23)$$

### 3.3.2.2 The rules of selecting requests by ruin/repair operators in each phase

As mentioned above, in ALNS, different phases use different policies for the selection of requests to remove (ruin) and the requests to re-insert (repair). In our problem, because all exchangeable requests must be served, the rules of applying ruin/repair operators for requests are the two following suitable policies.

#### *Request selection policies for ruin operators*

1. All requests are removable.

2. Only exchangeable requests are removable.

*Request selection policies for repair operators*

1. There is no difference between exchangeable requests and reserved requests when choosing a request to re-insert by a repair operator.
2. The priority of reserved requests is higher than that of exchangeable requests when choosing a request to reinsert by a repair operator.

The first policy is denoted by policy 1 and the second policy is called policy 2. The two policies are compatible with the nature of our problem. In the following, we describe two policies and their implementation.

If policy 1 is chosen, there is no difference between two types of requests when they are removed or re-inserted. In other words, any request can be served as long as all the constraints of the problem are satisfied and the corresponding ruin/repair operator is feasible. With time windows and vehicle capacity constraints in our problem, a ruin or repair operator may lead to an infeasible solution. Such situation may happen when some reserved requests are not served. Generally, the role of the first policy is to expand the search space by possibly selecting all requests. In this case, some reserved requests may be not served because of violating some constraints particularly vehicle capacity and time windows constraints. On the other hand, the role of the second policy is to generate a feasible solution. In the first phase of ALNS, the probability of choosing policy 1 is the highest and the probability of choosing policy 2 is the lowest. When the algorithm progresses, the first probability becomes lower and the second probability becomes higher. At the last phase, the probability of choosing policy 1 is the lowest and the probability of choosing policy 2 is the highest. Let  $P_1$  denotes the probability of choosing policy 1 and  $P_2$  denotes the probability of choosing policy 2 with  $P_1 + P_2 = 1$ .  $P_1$  and  $P_2$  are calculated by  $P_2 = \frac{se}{nse}$ ,  $P_1 = 1 - P_2$ , where  $se$  is the number of phases executed so far and  $nse$  is the total number of phases prespecified by the algorithm.

### 3.3.2.3 Ruin/repair operators of ALNS

Contrary to [65], where selective (exchangeable) requests are not obligatory to be served, our problem in this thesis has two types of requests and both must be served by all carriers while each reserved request must be served by its own carrier. In the design of ruin/repair operators, we have to consider the specific features of our problem. This means that the operators must consider serving all requests. Furthermore, all reserved requests must be served before serving exchangeable requests because it is not compulsory for each carrier to serve their exchangeable requests. Actually, each exchangeable request must be allocated to the most appropriate carrier according to the policies in previous section. When the ruin operators are applied, if a route has

just one request, it will be deleted. Additionally, all of the nonempty routes are considered to insert a request when an insertion operator is used.

---

**Algorithm 2.** Least profit removal operator.

---

- 1 **while** (number of remove <  $\lceil fr.rn \rceil$ )
  - 2     sort all requests in terms of profit decreasing and put in array  $L$
  - 3     choose a request randomly (if  $j$  is its index)  $j = \lceil (r^{U(0,1)})^{100.fr} |L| \rceil$
  - 4     remove the selected requests
  - 5     number of remove increases
- 

**Ruin operators**

In this thesis, we utilize six ruin operators. In the using of ruin operators, a fraction of the solutions is removed. It means that if  $(rn)$  is the total number of requests and  $fr \in [0,1]$  is a removal fraction, the number of removing requests is  $\lceil fr.rn \rceil$ .

*Random Removal.* This operator randomly chooses a subset of requests. In fact, the Random Removal operator has a strong effect in diversification.

*Least profit removal.* If  $f(so)$  is the cost of current solution and  $f(so')$  is obtained cost if request  $i$  is removed, the profit of the request  $i$  in solution  $so$  is described as  $f(so)-f(so')$ . *Algorithm 2*, is the algorithm of least profit removal operator and the number of low profit requests are  $\lceil fr.rn \rceil$ . They are removed according to *Algorithm 2*, because low profit requests can be reinserted in positions that are more profitable. The operator is performed randomly to avoid removing the same selected requests.

*Least paid removal.* This algorithm is the same as least profit removal operator with some differences. In this operator, the list  $L$  stores the prices of requests in increasing order. When the operator removes the requests with low prices and reinserts them in other positions, better solution may be achieved.

*most expensive removal operator.* The operator is similar to least paid removal operator; but in most expensive removal operator all requests are sorted in descending order of their cost. Moreover, we define the difference of transportation cost with and without a request.

*Shaw removal operator.* This operator is introduced in [88] and is used in [84], [65]. Primarily, it is necessary to choose a seed request. After that, requests which are similar to the seed request based on distance, time and demand are removed. When the criterion is distance, two nodes of each request must be closed to the seed request and in choosing criterion based on time, the service starting time of pickup and delivery nodes of each request must be as same as the

seed request. The significant criterion of *Shaw removal operator* is to have low probability of constraints' violation particularly, in using time windows and vehicle capacity constraints. Finally, this operator is randomized as *Algorithm 2* and applied  $[fr.rn]$  times.

*Price similarity removal.* This operator removes the requests with the same prices. After that, a repair operator exchanges the location of the selected requests or abandons them to decrease total cost. In this operator, we use price dissimilarities between two requests. This dissimilarity measures value growing with the price differences between two requests.

### Repair operators

In [65], two repair operators are applied: *Basic greedy insertion heuristic* and *two-regret insertion*. In the current thesis, in addition to the mentioned repair operators, we also use *three-regret insertion heuristic* to improve the quality of the solution repaired. How to select a request to repair with two policies was discussed in previous section. Each repair operation must ensure the repaired solution is feasible, i.e., each reserved request must be served by its own carrier, all requests must be served by all carriers, vehicle capacity and time windows constraints are satisfied.

*Basic greedy insertion heuristic.* This heuristic inserts one by one completely removed requests to achieve the highest increase in total profit.

#### *2-Regret and 3-Regret insertion heuristic*

An obvious problem with the basic greedy heuristic is to postpone handling those hard requests that lead to construct unreasonable route at the end. The regret heuristic tries to overcome the problem by incorporating a type of look-ahead information. This means, if we do not insert the request in the current iteration, in the next iteration we may pay a lot for not having chosen it. Regret heuristic has a grade: Each time we choose insertion of a request that holds the largest difference between its best position and second best position, which is named as *2-Regret insertion heuristic*. In *3-Regret insertion heuristic*, each time a request is chosen to insertion that holds the largest difference between its best position, second best, and third best positions. After each insertion, the regrets must be recomputed because some insertion positions are unavailable.

#### 3.3.2.4 Diversification of search by simulated annealing

The goal of using simulated annealing algorithm is to prevent to be trapped in local search. The general schema of SA has been illustrated in the structure of ALNS. In comparison with the general heuristic that only accepts the improved solution, SA accepts a worse solution  $so \rightarrow so'$  with a probability  $e^{-\frac{f(so)-f(so')}{T_{in}}}$ , where  $f(so') < f(so)$ . The possibility is related to the problem maximization. The probability of profit decreasing depends on two factors. The first one is profit disparity, which is described by  $-(f(so) - f(so'))$  equation and second factor is temperature  $T_{in}$ .

At the beginning of each run,  $T_{in}$  is allocated to  $T_0$ .  $T_{in}$  is computed and if a solution has a profit 30% worse than initial solution, it will be acceptable with a ( $pr$ ) probability.  $T_{in}$  decreases after each iteration in each phase, by multiplying cooling factor  $co \in (0, 1)$ . The cooling factor must be closed to one to avoid a premature convergence.

### 3.3.2.5 Local search approach

The conspicuous role of local search is to improve the results of heuristic and metaheuristic algorithms. Accordingly, at the end of each phase, we use four types of local search to achieve better results. The moves operators should assume specifications of our problem and serve all exchangeable requests. The description of neighborhoods is as follows. All feasible moves of neighborhoods are tested. The move is acceptable if an improvement is obtained; otherwise, the search continues with the next neighborhood. If there is no progress in the last results, the local search is terminated.

1. One pickup/delivery node is relocated in another place of the same route.
2. One pickup/delivery node of one exchangeable request is relocated in another place of the same route.
3. One request is relocated in another place of the same route.
4. One exchangeable request is removed from a route of a carrier and reinserted into a route of another carrier.

### 3.3.3 Hybrid genetic algorithm with simulated annealing (GASA)

Genetic algorithm (GA) is firstly proposed in [47] and it is one of the well-known metaheuristic algorithms to solve vehicle routing problems. Simulated annealing (SA), which accepts a worse solution with a probability that diminishes as the search progresses, allows the diversification of search space. The combination of GA with SA can prevent GA from premature convergence. Hence, we propose a hybrid GA combined with SA to solve our problem. GA is inspired from nature. In GA, each solution is represented by a chromosome and the population in each iteration is composed of multiple chromosomes correspond to a set of solutions. The objective function value of the problem is evaluating the fitness of each chromosome. The reproduction process of GA is similar to that in the real world, where the next generation (population) is generated by unifying the genes of two parents from the current generation via a crossover operation. Additionally, the genes of a chromosome may be altered like gene mutation in nature. To choose parents and individuals to transfer to the next generation, the fitness of each chromosome (solution) plays a decisive role. In this process, the offspring with lower fitness are removed from the population. In the rest of this section, all elements of GASA for multi-carriers CTP will be presented and described. The initial solutions of this algorithm, i.e., the solutions in the initial population

are constructed in the same way as the insertion heuristic used in ALNS (section 3.3.1) but introducing random factors in the heuristic. It means that we have initial solution to the size of population. In each iteration of GASA, the minimum cost is updated according to the achieved cost of the latest generation. After creating new individuals by mutation and crossover operators and mixing with the current population, each solution is selected with a probability based on a simulated annealing procedure.

### 3.3.3.1 Solution representation

The structure of chromosome plays a vital role in order to get good results for any GA-based algorithm. The chromosome of multi-carriers CTP is defined by three vectors  $X, Y$ , and  $Z$ .

Vector  $X$  consists of all pickup and delivery nodes and its size (dimension) is  $|P \cup D|$ . The size of vector  $Y$  is equal to the number of all requests and each component of vector  $Y$  that corresponds to a request indicates the carrier assigned to the request. The dimension of vector  $Z$  is equal to the number of all requests. In vector  $Z$ , each component indicates the index of the vehicle serving a request. It should be noted that the index of each request is the same as its pickup node index in the three vectors. The procedure for constructing a solution from a chromosome represented by the three vectors is given as follows:

Firstly, pickup and delivery nodes are chosen from vector  $X$  one by one from its first gene to its last gene. After choosing a pickup node and its corresponding request, the carrier and the vehicle (route) to serve the request are determined from vector  $Y$  and vector  $Z$  successively. When a pickup node is assigned to a route of a carrier, its paired delivery node is also assigned to the same route of the same carrier and the precedence relation between the two nodes must be satisfied. A new route will be created if the insertion of a request (a pickup node plus its paired delivery node) into the current route leads to an infeasible solution. This means that the insertion violates time windows constraints or vehicle capacity constraint and the request cannot be served by the current route.

The coding of a solution by the three vectors mentioned above is depicted in Fig. 3.1 for a simple example of multi-carriers CTP with 2 carriers and 4 requests. The dimension of  $X, Z$  and  $Y$  are 8, 4 and 4, respectively. Note that for each request  $i$ , its pickup node is indexed by  $i$  and its delivery node is indexed by  $n + i$ , where  $n = 4$  is the number of all requests. Vector  $Z$  indicates that request 1 is served by vehicle 1, request 2 is served by vehicle 2, request 3 is served by vehicle 4, and request 4 is served by vehicle 3. According to vector  $Y$ , request 1 and 2 are allocated to carrier 1, and request 3 and 4 are assigned to carrier 2. The detailed information of this simple example for each carrier is given in Table 3.4.

**Table 3.4:** The detailes information of simple example to implement GASA.

Carrier1	Carrier2
the index of the unique reserved request: 1	the index of the unique reserved request: 3
the pickup node index of the reserved request: 1	the pickup node index of the reserved request: 3
the delivery node index of the reserved request: 5	the delivery node index of the reserved request: 7
the index of the unique exchangeable request: 2	the index of the unique exchangeable request: 4
the pickup node index of the exchangeable request: 2	the pickup node index of the exchangeable request: 4
the delivery node index of the exchangeable : 6 request	the delivery node index of the exchangeable request: 8



Vector X



Vector Z



Vector Y

**Figure 3.1:** Illustration of chromosome's structure.

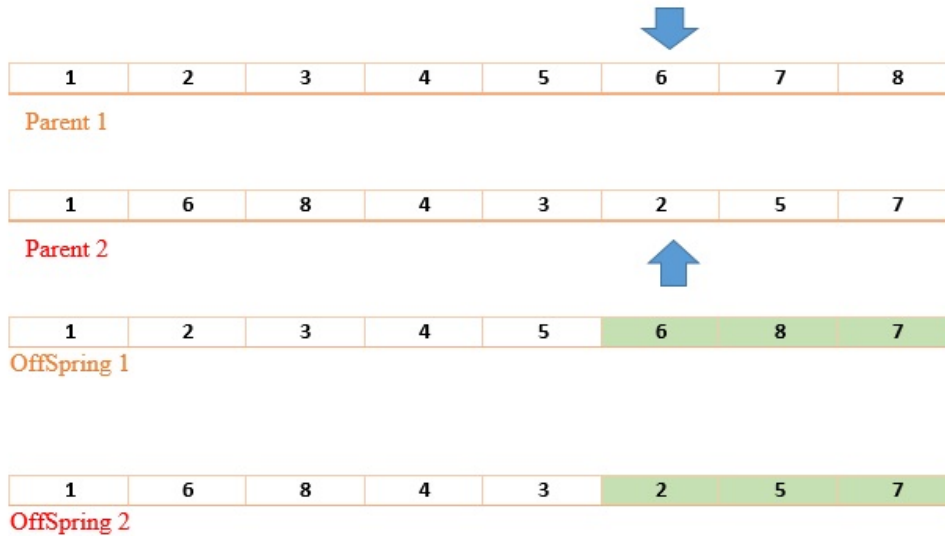
### 3.3.3.2 GASA agorithm operators

According to the chromosome structure defined for the multi-carriers CTP, we propose two crossover operators and two mutation operators for the solutions considered.

#### *Crossover operator on vector X*

Single point crossover is a suitable operator for two chromosomes of  $X$ , which has a permutation structure. By using this crossover operator, no need to apply an extra operation to make the resulted solution a feasible solution. By using this type of crossover operator in the resulted





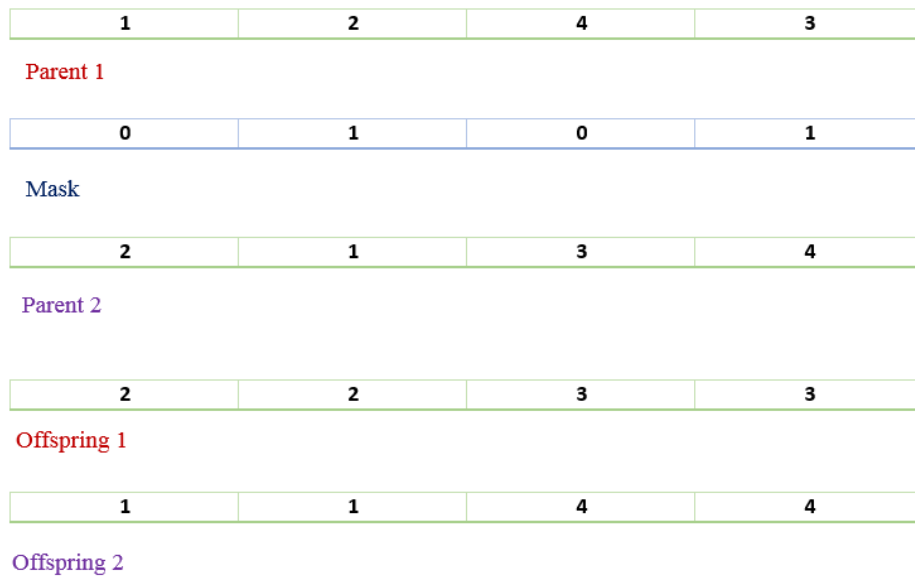
**Figure 3.2:** The single point crossover on two  $X$  vectors.

chromosome, there will be no gene repetition. To make this crossover operation, two parents chromosomes are chosen from the current population by applying the roulette wheel approach, while the crossover point is randomly chosen from all genes of vector  $X$ . The crossover operator on vector  $X$  works as follows:

To generate the first offspring, all genes of the first parent before the crossover point are transferred respectively to produce the first part of the first offspring's chromosome. To generate the second part of the first offspring after the crossover point, all genes of the second parent are compared with the genes of the first offspring already created. After ignoring repetitive genes, all non-repetitive genes are transferred to create the second part after the crossover point. The second offspring is generated in the same way as the first offspring but the roles of the two parents are swapped in the above mentioned procedure. In Fig. 3.2, the crossover operator is applied on two  $X$  vectors and their generated two offspring are given. In addition, the two arrows in Fig. 3.2 point to the crossover point.

#### *Crossover operator on vector $Y$ and $Z$*

Uniform crossover is a suitable crossover operator for vector  $Z$  and  $Y$ . In this type of crossover operator creating a mask vector with one and zero genes, as illustrated in Fig. 3.3, is necessary. The mask vector has the same size (dimension) as that of vector  $Z$  (vector  $Y$ ). A gene of an offspring is transferred from which parent depends on whether the value of the corresponding gene of the mask vector is one or zero. As illustrated in Fig 3.3, to generate the first offspring, a gene of the first parent is transferred to the first offspring if the corresponding gene of the mask vector is zero otherwise the gene of the first offspring will be transferred from the second parent.



**Figure 3.3:** The crossover on two  $Z$  vectors.

To generate the second offspring the same operation is performed but the roles of the two parents are swapped. Since the structure of vector  $Y$  is the same as that of vector  $Z$ , the same crossover operator procedure is used.

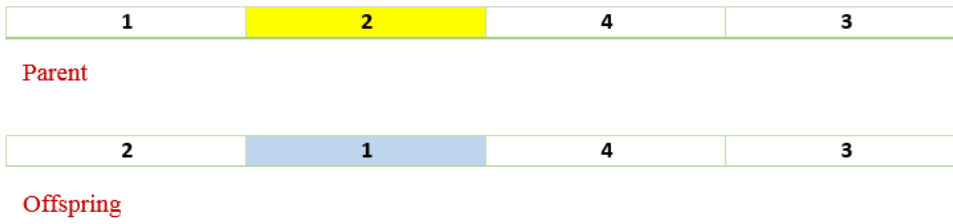
*Mutation on vector  $X$*

For vector  $X$  with a permutation structure, the mutation on vector  $X$  is made in the following way: Firstly, two distinct genes of vector  $X$  are chosen randomly. After that, one of the relocation, swap and reversion operators is chosen and applied to the two genes. This mutation operator can be applied to produce a diversified feasible solution.

1. *Swap*: this operator selects two components of vector  $X$  and swap their positions in the vector.
2. *reverse*: this operator selects two components of vector  $X$  and reverses the order of the components between the selected components.
3. *relocate*: this operator selects two components in vector  $X$  and relocates one of them to the front of another component.

*Mutation on vectors  $Z$  and  $Y$*

The mutation applied to vector  $Z$  and  $Y$  is similar to that applied to  $X$  with some differences mentioned hereafter. The steps of the mutation over vector  $Z$  as illustrated in Fig. 3.4 are given in the following:



**Figure 3.4:** The mutation operator over vector  $Z$ .

1. Determine randomly the number of genes to mutation. This number, which is denoted by  $l$  is obtained by the following procedure.
  - a) Generate randomly an integer number,  $h$ , between 1 and  $\dim[Z]$ , where  $\dim[Z]$  is the dimension of vector  $Z$ .
  - b) Calculate  $hr$  by multiplying  $h$  with a mutation rate  $r$ .
  - c) Round  $hr$  to the least integer number larger than or equal to it.
2. Choose randomly  $l$  genes of vector  $Z$ .
3. Generate randomly  $l$  integer numbers between 1 and the number of all vehicles,  $VK$ .
4. Replace the  $l$  genes of vector  $Z$  obtained in Step 2 by the  $l$  integer numbers obtained in Step 3.

For the mutation on vector  $Y$ , the same procedure is applied except that in step 4, each gene selected in step 2 is changed with a number randomly generated between 1 and the number of all carriers,  $M$ . In Fig. 3.4, the number of mutated genes in vector  $Z$  is one and the second gene with yellow color is selected to mutate. The chromosome of the offspring is transferred from its parent's chromosome with a new value in the mutated gene.

In each iteration of GA, after sorting the new chromosomes (solutions) generated by the above mentioned GA operations, they are merged with the current population to generate the next generation. In GASA, the rule of SA is applied when transferring each offspring to an individual in the next generation. This means that, the current solutions and the solutions obtained by crossover and mutation operations are compared based on SA rule. In this way, each solution has a chance to be transferred to the next generation. To use simulated annealing in the framework of our genetic algorithm, we use the same approach of SA used previously in the ALNS algorithm we developed. At the end of each iteration of the genetic algorithm, the temperature of SA is reduced like what is done in any simulated annealing algorithm.

**Algorithm 3** Pseudo-code of the ITS algorithm.

---

- 1: Set the maximum number of iterations as  $mni$ , the number of search points as  $nsp$ , the size of each tabu list  $stl$  and the number of initial solutions to be generated as  $nis$ , where  $mni, nsp, stl$ , and  $nis$  are three positive integers with  $nis \geq nsp$ .
  - 2: Calculate the number of possible operations  $no\_X$ ,  $no\_Y$  and  $no\_Z$  for vector  $X$ ,  $Y$  and  $Z$  respectively and set the sizes of the three tabu lists for the three vectors respectively as  $stl\_X \times no\_X$ ,  $stl\_Y \times no\_Y$  and  $stl\_Z \times no\_Z$ , where  $stl\_X$ ,  $stl\_Y$  and  $stl\_Z$  are three fractions between 0 and 1.
  - 3: Initialize the three tabu lists mentioned above, which are introduced to forbidden revisiting previously explored solutions in the tabu search
  - 4: Calculate the number of possible operations for vector  $X, Y$  and  $Z$
  - 5: Construct  $nis$  initial solutions
  - 6: Sort the initial solutions in a non-increasing order of their cost value
  - 7: Choose the  $nsp$  best solutions from the sorted list of initial solutions
  - 8:  $BestSol$  = the cost of the best solution obtained so far
  - 9: **While** the number of iterations  $< mni$
  - 10:     **While** the number of solutions (points)  $< nsp$
  - 11:         (the cost of the best solution in the neighborhood of the current solution) =  $Inf$
  - 12:         **While** the number of operations examined  $<$  the number of possible operations for vector  $X$
  - 13:             Choose an unexamined operation that moves the current solution to a new solution
  - 14:              $NewSol$  = the cost of the new solution
  - 15:             **If**( $NewSol < BestNewSol$ ) and the operation is not in the tabu list of vector  $X$
  - 16:                  $BestNewSol = NewSol$  and put the operation in the tabu list of vector  $X$
  - 17:             **End if**
  - 18:         **End While**
  - 19:         **While** the number of operations examined  $<$  the number of possible operations for vector  $Y(Z)$
  - 20:             Choose an unexamined operation that moves the current solution to a new solution
  - 21:              $NewSol$  = the cost of the new solution
  - 22:             **If**( $NewSol < BestNewSol$ ) and the operation is not in the tabu list of vector  $Y(Z)$
  - 23:                  $BestNewSol = NewSol$  and put the operation in the tabu list of vector  $Y(Z)$
  - 24:             **End if**
  - 25:         **End While**
  - 26:         **If** the number randomly generated from  $[0,1] <$  the mutation probability
  - 27:             Execute the mutation operation
  - 28:         **End While**
  - 29:         **If**  $BestNewSol < BestSol$ ,  $BestSol = BestNewSol$
  - 30:     **End While**
-

### 3.3.4 Improved tabu search (ITS)

Tabu search is a local search metaheuristic propose in [39]. Briefly, the method explores the solution space by moving at each iteration from a solution to the best solution in a subset of its neighborhood  $N(s)$ . Contrary to the classical descent method, the solution may deteriorate from one iteration to the next. Thus, to avoid any loop, some attributes of recently explored solutions are temporarily declared tabu or forbidden, even though there are better solutions in the neighbor of the current solution. The tabu list is updated in each iteration, adding a diversification in the search of the solution space. Actually, by using the tabu list the parts of solution space that were previously explored will be not considered with various techniques often employed to diversify or to intensify the search process [104]. In this section, we propose an improved tabu search (ITS) algorithm that keeps multiple solutions during the search process and uses a mutation operation in order to get a better solution quickly. On the one hand, multi-points tabu search algorithms are usually more efficient than single point ones [75], and on the other hand, some mutation operators can improve the performance of tabu search algorithm [52]. The procedure of our Improved TS (ITS) algorithm is given in the following *Algorithm 3*, where three tabu lists containing three subsets of attributes of the solutions to be forbidden from being revisited in the tabu search are introduced and updated in each iteration. The three subsets of attributes are denoted by three vectors  $X, Y$ , and  $Z$  respectively used to represent a solution (see Section 3.3.3.1 for a description of the three vectors).

#### 3.3.4.1 The local search operations (moves) on three vectors $X, Y$ and $Z$

In tabu search based algorithms, the local search operations (moves) made in the neighborhood of a solution have a great impact on the quality of the obtained solution. In the following, we describe the local search operations made on each of the three vectors  $X, Y$  and  $Z$  representing a solution.

##### *Operations (Moves) on vector $X$*

The vector  $X$  has a permutation structure. The operations applied to  $X$  include swap, insertion or reversion and relocation.

##### *Operations (Moves) on vector $Y$ and vector $Z$*

The vector  $Y$  and vector  $Z$  have the same structure. For this reason, the operation applied to vector  $Y$  is the same as the operation applied to vector  $Z$ . In the two vectors, a suitable change (move) is to replace the value of a component. In other words, the value of the component is changed by a number randomly generated between 1 and the number of carriers for vector  $Y$  and between 1 and the maximum number of vehicles for vector  $Z$ .

### 3.3.4.2 Tabu lists in the ITS

A tabu list represents a short-term memory in a tabu search algorithm. It consists of a list of previous solutions that must be avoided to revisit or a list of local search operations (moves) that must be forbidden in moving to the next solution. According to the proposed solution representation for the ITS algorithm, three tabu lists corresponding to vector  $X, Y$ , and  $Z$  respectively are introduced and updated in each iteration. Each tabu list contains a set of local search operations (moves) previously made on the corresponding vector ( $X, Y$  or  $Z$ ). These operations will be forbidden when moving to the next solution in the neighborhood of the current solution. The size of each tabu list is set by considering the number of possible operations that could make on the corresponding vector. In *Algorithm 3*, an operation made on  $X, Y$  or  $Z$  that moves the current solution to a neighborhood solution is acceptable if it is not in the corresponding tabu list. Otherwise, the operation is not valid. If the operation is not in the tabu list and leads to a better neighborhood solution, the best solution is updated, the operation will be put in the corresponding tabu list, and the size of the tabu list is increased. In *Algorithm 3*, the sizes of three tabu lists are set as predefined parameters.

## 3.4 Numerical experiments

To evaluate the performances of our ALNS, ITS, and GASA algorithm, we applied them to solve 5 sets of instances of small to large sizes taken from the literature and compared them with the MILP solver of CPLEX 12.6 in terms of cost and computation time. The three algorithms are also compared to each other on the instances in terms of RPD. We do not compare the three algorithms with CPLEX on medium and large instances, because the latter could not even find a feasible solution after a long running time for these instances.

In the following sections, we present these instances and the comparison results. The results demonstrate the ALNS, ITS, and GASA algorithm outperform CPLEX, while the ALNS algorithm outperforms the GASA and ITS algorithms.

### 3.4.1 Instances

The instances are taken from [17], with the reserved requests and the exchangeable requests for each carrier determined in the same way. These instances are grouped into 5 sets according to their problem sizes. The number of carriers are fixed to 3 for all instances, whereas the number of requests per carrier vary from 5 to 50 and the number of vehicles for each carrier vary from 2 to 8 for the five sets of instances. Each instance in the first set has five requests and two vehicles for each carrier. Each instance in the second set also has 2 vehicles for each carrier but has eight requests for each carrier. In the third set, there are 15 requests and 4 vehicles for each carrier. The fourth set is composed of the instances with 30 requests and 6 vehicles for each carrier. Each instance in the fifth set comprises 50 requests and 8 vehicles for each carrier. We solved the MILP

model of our problem by CPLEX 12.6 called in C++ with Visual studio 12 on a laptop with an Intel core i5 CPU and 4.00 GB RAM. Note that for the medium size instances with 30 requests for each carrier and 4 vehicles and large size instances with 50 requests per carrier and 8 vehicles, it is impossible to solve the model optimally by CPLEX after 2 hours and 3 hours of running, respectively.

### 3.4.2 Parameter settings

The performance of the ALNS algorithm, ITS and the GASA algorithm depends on the quality of their parameters. Some of the parameters are determined empirically and are given in Table 3.5 and Table 3.6. Other parameters are tuned by using Taguchi method, which is a popular and efficient parameter tuning technique based on fractional factorial experiments (FFE) [96]. Variety of approaches, such as response surface method [79], [73] and Taguchi approach [71], [42] are already been applied for calibration of algorithm's parameters. In this thesis, Taguchi methodology is addressed to tune the parameters of algorithms. This approach employs orthogonal arrays to handle and adapt experiences in the presence of a group of decision variables or factors [80]. The methodology tries to minimize the impact of noise and to achieve the optimal level of a signal factors. Since the nature of response is minimization, the smaller response is better. The following formulation expresses the signal-to-noise ratio ( $\frac{S}{N}$ ):

$$\frac{S}{N} = -10 \times \log\left(\frac{S(Y'^2)}{n}\right) \quad (3.24)$$

$Y'$  indicates response value,  $n$  represents the number of the orthogonal arrays, and  $S(Y')$  is the indicative objective function value.

The selected parameters of GASA and ITS to calibration are given in Table 3.7 and the parameters calibrated values are in Table 3.10 and Table 3.11 for GASA and ITS sequentially. The selected parameters of ALNS to tune is in Table 3.8 for different size of instances and their value after calibration by Taguchi is in Table 3.9. Actually, the optimal level of each parameter in each algorithm is the highest signal-to-noise ratio. Fig. 3.5 to Fig. 3.10 show clearly the appropriate level of each parameter of the considered algorithms.

### 3.4.3 Experimental results

After the parameter calibration, we executed the ALNS, ITS, GASA, and CPLEX on all instances. Since ALNS, ITS, and GASA algorithms contain random factors, to reduce the impacts of these random factors on their performance evaluation, each algorithm is executed three times for each instance. Then, the average cost and average computation time (in seconds) obtained by the three executions are reported. For CPLEX, since the considered carrier collaboration problem is NP-hard [66], it is very time consuming to solve optimally medium and large size instances. For this reason, we set a maximum running time for CPLEX to solve medium and large instances.

**Table 3.5:** Parameter values of ALNS determined empirically

Parameter	Description	Value
$nruns$	Number of runs	12
$Q_1$	Operator score increment case 1	10
$Q_2$	Operator score increment case 2	5
$Q_3$	Operator score increment case 3	3
$Q_4$	Operator score increment case 4	1
$rf$	Score reaction factor	0.8
$nseg$	Number of phases	3000
$fr$	Removal fraction	0.2

**Table 3.6:** Parameter values of GASA and ITS determined empirically

Parameter	Description	Value
$SubIt$	Number of iteration of SA	120
$\mu_m$	Mutation rate	0.26
$nsp$	Number of solution points	25

**Table 3.7:** The domain of candidated parameters of GASA and ITS for calibration.

Methodology	Parameter	Range	Low	Medium	High
GASA	$Npop$ (Size of population)	100-200	100	150	200
	$nIt$ (Maximum number of iterations)	200-600	200	400	600
	$Pc$ (Crossover probability)	0.4-0.8	0.4	0.6	0.8
	$Pm$ (Mutation probability)	0.05-0.3	0.05	0.175	0.3
ITS	<b>Tabu lists</b>				
	$stl\_X$	0.2-0.5	0.2	0.35	0.5
	$stl\_Y$	0.1-0.4	0.1	0.25	0.4
	$stl\_Z$	0.1-0.5	0.2	0.35	0.5
	$mni$ (Num of iteration)	50-100	50	75	100

The time limitation is 3 hours for medium instances with 30 requests and 4 vehicles, and large instances with 50 requests and 8 vehicles. The three proposed algorithms are compared with each other and with CPLEX based on all criteria defined in Table 3.10, where  $Cost_{ALNS}$ ,  $Cost_{GASA}$ , and  $Cost_{ITS}$  are the costs of the studied problem obtained by the three algorithms, respectively;  $UB_{MILP}$  and  $LB_{MILP}$  are the upper bound and the lower bound of the objective value of the problem obtained by CPLEX in a preset computation time.

The computation results are given in Table 3.12, Table 3.13, and Table 3.14. From Table



**Table 3.8:** The domain of candidated parameters of ALNS for calibration.

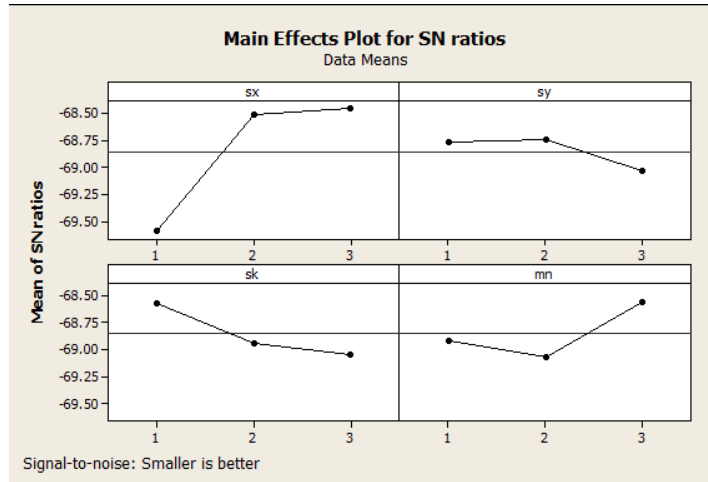
Methodology	Parameter	Range	Low(1)	Medium(2)	High(3)
ALNS	$w$ (Start temperature control)	0.5-0.7	0.5	0.6	0.7
	$c$ (SA cooling factor)	0.9996-0.9998	0.9996	0.9997	0.9998
	$nit$	500-700	500	600	700

**Table 3.9:** Parameter values of GASA determined by Taguchi.

parameters	npop	P_c	P_m	Nit
Value	150	0.6	0.05	800

**Table 3.10:** Parameter values of ITS determined by Taguchi

parameters	stl_X	stl_Y	stl_Z	mni
Value	0.5	0.25	0.2	100

**Figure 3.5:** Outputs of Taguchi ratio for ITS.

3.11, we can see, for small  $3 \times 5 \times 2$  instances, CPLEX, ALNS, ITS and GASA all could find an optimal solution. However, for some of small  $3 \times 8 \times 2$  instances and most medium to large size instances, CPLEX could not even find a feasible solution in a preset computation time, whereas our algorithms could find a feasible solution for all instances. Actually, for most instances with the number of requests more than 8, no feasible solution was obtained by CPLEX, so we compare the solutions obtained by our algorithms based on their relative gaps with the lower bound obtained by CPLEX, i.e., using the above-mentioned criteria. Our algorithms can find an optimal solution

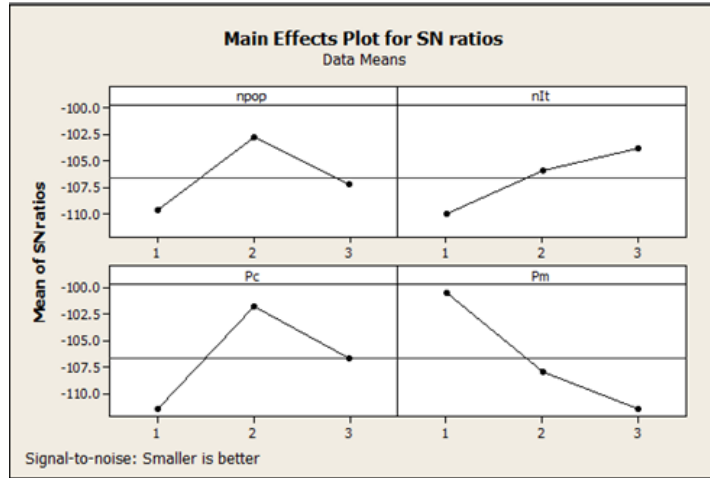


Figure 3.6: Outputs of Taguchi ratio for GASA.

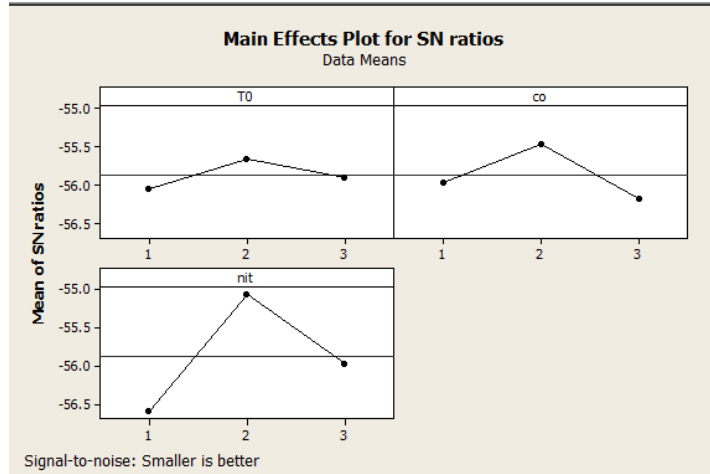
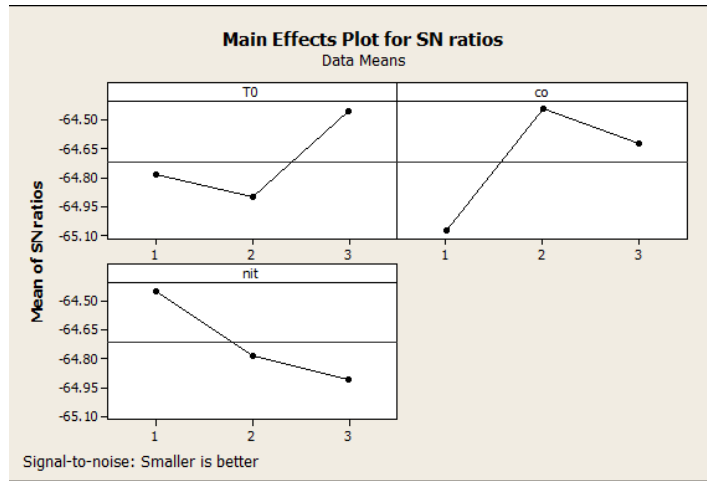
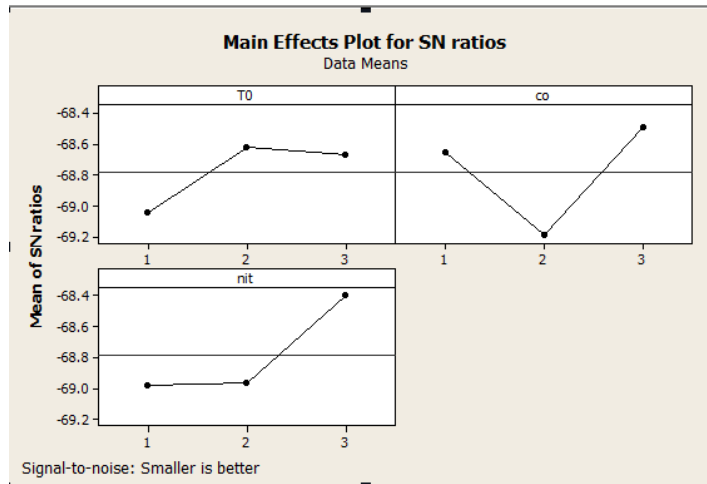


Figure 3.7: Outputs of Taguchi ratio for ALNS ( $3 \times 5, 3 \times 8$ ).

for all small  $3 \times 5 \times 2$  instances and can find a solution with the relative cost gap smaller than 3.6% for  $3 \times 8 \times 2$  instances. For medium  $3 \times 15 \times 4$  instances, our ALNS algorithm can find a solution with the gap smaller than 6%, whereas our ITS algorithm can find a solution with the gap smaller than 9.9% and GASA can find a solution with the gap smaller than 7.58%. For  $3 \times 30 \times 6$  instances, our ALNS algorithm can find a solution with the gap smaller than 12%, whereas our ITS algorithm can find a solution with the gap smaller than 14% and our GASA algorithm can find a solution with the gap smaller than 12.6%. For large  $3 \times 50 \times 8$  instances, our ALNS algorithm can find a solution with the gap smaller than 17%, whereas our ITS algorithm can find a solution with the gap smaller than 18% and our GASA algorithm can find a solution with the gap smaller than 17.4%. Moreover, our proposed algorithms are much better than CPLEX in terms of running time for medium and large instances, although the running times of the two proposed algorithms increase naturally with the size of instance and the computation time of CPLEX is smaller than



**Figure 3.8:** Outputs of Taguchi ratio for ALNS ( $3 \times 15$ ).



**Figure 3.9:** Outputs of Taguchi ratio for ALNS ( $3 \times 30$ ).

those of the algorithms for small  $3 \times 5 \times 2$  instances. According to Table 3.12, Table 3.14, and Fig. 3.11 ALNS has cost and computation time smaller than those of ITS and GASA respectively for all instances, therefore ALNS outperforms ITS and GASA. This may be because various ruin/repair operators, adaptive mechanism, and SA are used in ALNS, which makes it able to search diversified regions in the solution space.

### 3.4.3.1 Relative percentage deviation (RPD) to compare algorithms

There are two ways to compare the algorithms. One is to classify the samples in small, medium, and large sizes and then compare the algorithms in each category. The second way is to remove the effect of problem size by utilizing an index. RPD is an indicator calculated by the difference between the best solution of an algorithm in terms of cost or profit and the best solution achieved

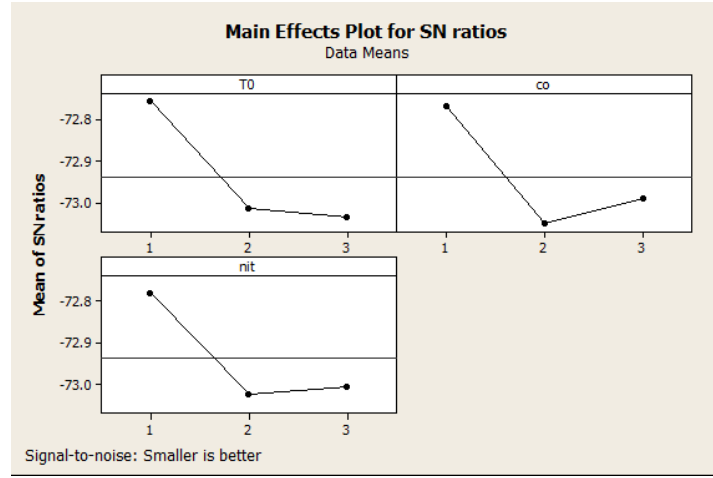


Figure 3.10: Outputs of Taguchi ratio for ALNS ( $3 \times 50$ ).

Table 3.11: Criteria used for comparison of ALNS, ITS, GASA, and CPLEX.

Criterion	Description
$Gap_{MILP}$	The relative gap between $LB_{MILP}$ and $UB_{MILP}$ defined as $\frac{(UB_{MILP} - LB_{MILP})}{LB_{MILP}}$
$Gap_{ALNS}$	The relative gap between $LB_{MILP}$ and $Cost_{ALNS}$ defined as $\frac{(Cost_{ALNS} - LB_{MILP})}{LB_{MILP}}$
$Gap_{GASA}$	The relative gap between $LB_{MILP}$ and $Cost_{GASA}$ defined as $\frac{(Cost_{GASA} - LB_{MILP})}{LB_{MILP}}$
$Gap_{ITS}$	The relative gap between $LB_{MILP}$ and $Cost_{ITS}$ defined as $\frac{(Cost_{ITS} - LB_{MILP})}{LB_{MILP}}$

by all algorithms, which are compared. An algorithm with minimum RPD value is supposed to be the most effective to solve the problem. RPD is formulated in the equation (3.25), where  $Criterion_{best}$  and  $Criterion_{alg}$  indicate the best value of a criterion obtained by all algorithms and the best value of the same criterion achieved by a considered algorithm correspondingly. In this thesis, we calculate the RPD of each algorithm after three executions. According to the formulation of RPD, an algorithm is efficient if it has small amount of RPD. Table 3.15 and Fig. 3.11 illustrate the results of RPD over ALNS, GASA, and ITS. According to the results, ALNS has better RPD and it outperforms the other two algorithms. Additionally, GASA has better RPD in comparison with ITS. Their statistical analysis is illustrated in Table 3.16. According to Table 3.16 there is not significant difference between GASA and ITS. Finally, the algorithms are also statistically analyzed according to obtained solutions via analysis of variance (ANOVA) tests [42].

The procedure of ANOVA including F-test value and also P-value on RPD is summarized in Table 3.16. The results are analyzed at 95% confidence level. According to the table, null hypothesis is rejected. The rejection of null hypothesis declares there are significant differences among ALNS and ITS, ALNS, and GASA algorithms.

$$RPD_{alg}(Criterion) = \left| \frac{Criterion_{best} - Criterion_{alg}}{Criterion_{best}} \right| \times 100 \quad (3.25)$$

**Table 3.12:** Computational results of ALNS, ITS, GASA and CPLEX – part one

Instance Number	$LB_{MILP}$	$UB_{MILP}$	$Cost_{ALNS}$	$Cost_{ITS}$	$Cost_{GASA}$
3×5×2 instances					
1	707.343	707.343	707.343	707.343	707.343
2	556.776	556.776	556.776	556.776	556.776
3	649.908	649.908	649.908	649.908	649.908
4	545.779	545.779	545.779	545.779	545.779
5	592.725	592.725	592.725	592.725	592.725
6	690.334	690.334	690.334	690.334	690.334
7	645.825	645.825	645.825	645.825	645.825
8	632.609	632.609	632.609	632.609	632.609
9	597.151	597.151	597.151	597.151	597.151
10	573.451	573.451	573.451	573.451	573.451
3×8×2 instances					
11	1006.084	1022.831	1009.361	1009.361	1009.361
12	949.505	1030.405	962.535	963.279	963.038
13	868.533	1018.795	875.608	875.676	875.670
14	1085.425	1156.400	1094.486	1095.472	1095.255
15	885.501	-	894.517	905.971	900.100
16	824.222	-	852.645	854.515	853.211
17	939.638	-	955.521	968.100	959.131
18	892.157	1024.313	902.816	910.393	909.308
19	968.684	-	990.001	1001.646	997.164
20	898.9721	1013.951	915.531	917.503	916.845
3×15×4 instances					
21	1721.373	2055.101	1755.073	1772.496	1770.922
22	1620.219	-	1693.407	1748.906	1723.664
23	1723.603	-	1791.902	1813.741	1808.142
24	1812.100	-	1885.538	1911.649	1891.535
25	1685.278	-	1758.043	1787.043	1779.101
26	1628.011	-	1702.668	1749.865	1726.273

CHAPTER 3. MULTI-CARRIERS PICKUP AND DELIVERY REQUESTS WITH TIME WINDOWS, RESERVED REQUESTS AND EXCHANGEABLE REQUESTS

27	1691.520	-	1761.116	1806.729	1793.915
28	1504.787	-	1580.933	1637.845	1614.407
29	1587.245	-	1656,240	1741.291	1711.633
30	1572.561	-	1643.800	1690.465	1680.281
3×30×6 instances					
31	2716.809	-	2909.774	3048.238	2965.753
32	2805.576	-	3030.573	3178.502	3133.935
33	2770.914	-	3068.606	3081.330	3068.697
34	2336.500	-	2574.208	2630.200	2594.390
35	2382.518	-	2622.581	2690.049	2660.315
36	2495.036	-	2741.556	2771.446	2751.543
37	2578.563	-	2719.427	2798.751	2735.300
38	2237.001	-	2485.519	2536.715	2503.416
39	2506.774	-	2753.443	2808.556	2773.703
40	2462.576	-	2684.044	2744.142	2742.871
3×50×8 instances					
41	4014.250	-	4515,789	4646.570	4574.916
42	3822.045	-	4321.190	4419.248	4357.647
43	3930.257	-	4408.724	4498.569	4441.602
44	3911.100	-	4381.453	4461.489	4424.935
45	3833.296	-	4385.240	4449.991	4415.816
46	3915.422	-	4521.686	4560.812	4568.101
47	3866.586	-	4282.326	4534.288	4490.595
48	3874.032	-	4511.115	4542.242	4496.716
49	3821.148	-	4429.470	4490.538	4462.221
50	3893.907	-	4353.763	4435.156	4404.549

**Table 3.13:** Computational results of ALNS, ITS, GASA and CPLEX – computation time.

<b>Instance Number</b>	$CPU_{CPLEX}(s)$	$CPU_{ALNS}$	$CPU_{ITS}$	$CPU_{GASA}$
3×5×2 instances				
1	75.478	83.369	94.346	88.456
2	68.521	84.109	96.893	89.825
3	76.107	83.674	98.833	92.380
4	68.733	84.223	96.021	87.958
5	76.112	84.973	96.861	86.727
6	75.207	84.123	97.362	89.033
7	74.800	84.490	96.735	90.359
8	75.564	85.274	96.587	90.874
9	68.806	83.432	95.922	86.922
10	68.225	83.549	91.278	87.224

---

3×8×2 instances				
11	480.114	144.590	213.215	166.923
12	588.731	154.627	207.690	184.461
13	595.695	151.556	219.803	183.274
14	7200	144.651	218.110	172.096
15	7200	155.264	214.196	169.102
16	7200	157.348	214.997	192.658
17	7200	155.255	213.730	188.100
18	7200	168.278	221.090	195.644
19	7200	150.410	219.766	176.025
20	7200	158.806	220.983	197.401
3×15×4 instances				
21	7200	386.118	513.426	433.114
22	7200	354.353	514.151	429.856
23	7200	373.197	553.588	450.709
24	7200	398.521	560.057	461.727
25	7200	374.573	513.180	435.622
26	7200	351.466	511.718	426.101
27	7200	346.740	501.621	439.258
28	7200	399.014	546.218	455.313
29	7200	359.055	519.400	418.634
30	7200	395.097	530.143	470.952
3×30×6 instances				
31	10800	647.292	918.072	747.309
32	10800	636.893	980.594	818.343
33	10800	665.402	977.191	822.658
34	10800	747.845	994.642	861.731
35	10800	727.486	997.247	854.911
36	10800	737.883	1005.626	832.947
37	10800	713.761	1011.710	875.204
38	10800	720.068	1038.242	893.365
39	10800	758.641	1039.987	896.501
40	10800	745.292	1041.907	882.113
3×50×8 instances				
41	10800	1855.576	3337.3249	2450.717
42	10800	1879.069	3361.1928	2518.162

---

CHAPTER 3. MULTI-CARRIERS PICKUP AND DELIVERY REQUESTS WITH TIME WINDOWS, RESERVED REQUESTS AND EXCHANGEABLE REQUESTS

---

3×50×8 instances				
43	10800	1863.512	3422.015	2676.481
44	10800	1861.487	3353.724	2713.277
45	10800	1941.217	3385.177	2679.195
46	10800	1908.095	3378.935	2785.726
47	10800	1952.140	3366.213	2753.205
48	10800	1975.549	3397.837	2749.458
49	10800	1960.266	3376.773	2770.116
50	10800	1953.358	3394.546	2788.315

**Table 3.14:** Gap of ALNS, ITS, GASA and CPLEX.

<b>Instance Number</b>	$Gap_{ALNS}$	$Gap_{ITS}$	$Gap_{MILP}$	$Gap_{GASA}$
3×5×2 instances				
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0
3×8×2 instances				
11	0.3256	0.325	1.664	0.325
12	1.372	1.450	8.520	1.425
13	0.814	0.822	17.300	0.821
14	0.834	0.925	6.538	0.905
15	1.018	2.311	-	1.648
16	3.448	3.553	-	3.517
17	1.690	3.029	-	2.074
18	1.194	2.043	14.814	1.922
19	2.200	3.402	-	2.940
20	1.841	2.061	12.790	1.988



---

3×15×4 instances				
21	1.957	2.969	19.387	2.878
22	4.517	7.942	27.410	6.384
23	3.904	5.229	-	4.904
24	3.868	5.493	-	4.714
25	4.319	6.038	-	5.567
26	4.585	7.484	-	6.035
27	3.917	6.814	-	4.279
28	5.060	8.842	-	7.284
29	4.346	9.705	-	7.836
30	4.530	7.497	-	6.849
3×30×6 instances				
31	7.102	12.199	-	9.1631
32	8.019	13.292	-	11.703
33	10.743	11.202	-	10.746
34	10.173	12.570	-	11.165
35	10.076	12.907	-	11.659
36	9.880	11.078	-	10.280
37	9.717	12.918	-	10.335
38	10.876	13.160	-	11.674
39	9.840	12.038	-	10.648
40	8.949	11.388	-	10.119
3×50×8 instances				
41	12.497	15.755	-	13.967
42	13.059	15.625	-	14.013
43	12.173	14.459	-	13.010
44	12.026	14.072	-	13.137
45	14.398	16.087	-	15.196
46	7.821	16.483	-	16.669
47	15.894	17.238	-	16.108
48	16.444	17.248	-	16.073
49	15.919	17.518	-	16.776
50	12.590	14.695	-	13.903

---

**Table 3.15:** Computational results (RPD) of ALNS, ITS and GASA.

---

<b>Instance Number</b>	$RPD_{ALNS}$	$RPD_{GASA}$	$RPD_{ITS}$
1	0	0	0
2	0	0	0
3	0	0	0

---

CHAPTER 3. MULTI-CARRIERS PICKUP AND DELIVERY REQUESTS WITH TIME  
WINDOWS, RESERVED REQUESTS AND EXCHANGEABLE REQUESTS

---

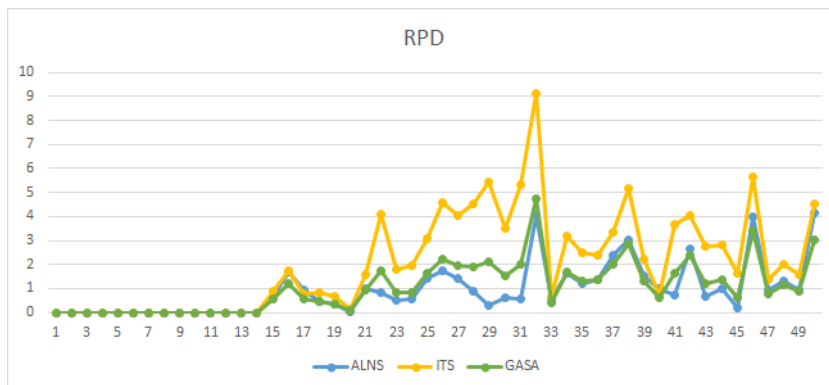
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	0	0	0
11	0.034	0.064	0.150
12	0.974	1.010	1.116
13	0.266	0.671	0.925
14	0.125	0.440	0.866
15	0.169	0.553	0.903
16	0.667	0.810	1.020
17	0.544	0.786	0.861
18	0.363	0.472	0.859
19	0.366	0.874	0.665
20	0.860	0.967	1.152
21	1.018	1.232	1.777
22	0.908	1.047	1.812
23	0.904	2.139	3.785
24	1.080	1.249	1.853
25	1.028	1.311	2.069
26	1.142	2.720	3.263
27	1.249	1.867	2.034
28	1.271	1.827	2.103
29	1.362	2.411	3.252
30	0.636	1.538	3.493
31	2.864	3.035	5.350
32	2.043	3.935	5.921
33	3.374	4.231	7.663
34	2.524	3.594	5.777
35	1.410	2.634	4.511
36	2.396	3.343	5.378
37	1.706	2.523	5.466
38	3.031	2.888	5.153
39	2.026	3.129	4.234

---

40	3.295	4.597	6.754
41	4.228	6.416	8.846
42	4.648	6.369	8.027
43	4.447	7.183	9.746
44	3.983	6.351	9.828
45	3.883	5.646	7.662
46	3.990	5.422	7.657
47	4.925	6.795	8.345
48	5.302	8.174	9.001
49	3.959	6.909	8.587
50	4.155	6.035	9.502

**Table 3.16:** The results of ANOVA test over the RPD of objective functions of algorithms.

Source	DF	SS	MS	F-test	P-value	Results
ALNS,ITS	1	32.52	32.52	11.96	0.001	Null hypothesis is rejected.
Error	98	266.38	2.72			
Total	99	298.90				
ALNS,GASA	1	25.44	25.44	9.43	0.003	Null hypothesis is rejected.
Error	98	264.26	2.70			
Total	99	289.70				
GASA,ITS	1	0.43	0.43	0.38	0.540	Null hypothesis is not rejected.
Error	98	112.20	1.14			
Total	99	112.63				



**Figure 3.11:** Graphical comparisons of RPD for both algorithms on all test problems.

### **3.5 Conclusion**

In this chapter, the collaborative transportation planning of multi-carriers with pickup and delivery requests with time windows has been studied. This problem has a new feature that each carrier has reserved requests, which must be self-served. We have proposed a mixed integer linear programming model for the problem. Because the model is NP-hard, it is impossible to get a feasible solution in a reasonable computation time for large instances. For this reason, we have proposed three metaheuristic algorithms, ALNS, ITS and, GASA, to solve the model. Numerical experiments on benchmark instances with RPD index show that the algorithms can obtain optimal solutions for small instances and ALNS can obtain better solutions for medium to large instances compared with GASA and ITS, whereas CPLEX cannot find even a feasible solution for medium and large instances in the allocated computation time. Moreover, we compared ITS and GASA with RPD index. Statistical analysis indicated GASA can obtain a better solution by comparing ITS in a smaller running time.

## MULTI-PERIODS BID GENERATION PROBLEM WITH PICKUP AND DELIVERY, TIME WINDOWS, PROFITS, AND RESERVED REQUESTS

In the real world applications, carriers usually plan their pickup and delivery operations and use of vehicle resources in advance (several days ago) in a rolling horizon [99], [44]. This requires that each carrier considers multi-periods (days) when it determines which transportation requests to bid and serve in each period (day). Moreover, requests open for bid (requests to be exchanged among carriers) may span across multi-periods (days). That is, instead of fixing a day to serve each selective request, each request is allowed to be served within a service day window consisting multi consecutive days. For example, goods ordered on-line by a customer on Monday in e-commerce is asked to deliver to the customer's home within three days from Tuesday to Thursday. This gives rise a multi-periods combinatorial auction (CA) problem. As a result, the bid generation problem of each carrier in such auction also involves multi-periods, where a bid is a combination of transportation requests over multi-periods [62].

In this chapter, a multi-periods Bid Generation Problem (BGP) for a carrier is considered. In the problem, there are two different types of requests, reserved requests of the carrier and selective requests. The carrier is committed by contracts with its shippers to serve all reserved requests by itself. The selective requests are offered by other carriers and are open for bid by the carrier. Each request has a pickup and delivery request with a pickup node, a delivery node, a pickup/delivery quantity, and two time windows. Period window specifies the earliest period and the latest period that each request must be served while time window shows the earliest and latest service time and is allocated to both pickup and delivery nodes. In addition, each selective request is associated with a profit that is the price for serving the request provided by a shipper. By including periodic concept in CA, the carrier can plan its transportation operations in advance and in a rolling-horizon. A carrier must make two important decisions in the BGP:

Which requests are chosen to bid and serve in their service period windows and how the routes are constructed to maximize its total profit. This leads to a new periodic pickup and delivery problem with time windows, profits, and reserved requests. So, this problem is a different problem from the problem presented in [65] that considers single period BGP. According to [101], the present problem is NP-hard and it is impossible to get feasible solutions for large instances by using a commercial solver like CPLEX. Hence, a hybrid approach that combines a genetic algorithm and simulated annealing (GASA) is proposed to solve the problem. The GASA adopts a problem-specific chromosome structure, crossover, and mutation operations well adapted to the chromosome structure. This algorithm utilizes the specifications of population-based algorithm and single point algorithm to diversification and intensification of the solution sequentially. For the purpose of comparison, we utilize Biogeography-Based Optimization algorithm (BBO) that is a population-based algorithm and its specifications are close to the GA and its operation is over the population of solutions to reach the best solution. By comparing GA and BBO, each chromosome in GA, which is considered as an individual with fitness value is habitat with suitability index that is called HSI (Habitat Suitability Index). Usually, a habitat with high HSI is acceptable and yields better solutions. The numerical results demonstrate the proposed algorithms can find a feasible solution in a reasonable computation time for large instances.

The rest of the chapter is organized as follows. Section 4.1 is devoted to literature review. A detailed description of a mathematical model for the studied problem is given in Section 4.2. In section 4.3, the GASA and BBO algorithms are described. In Section 4.4, detailed numerical results of solving the model by GASA, BBO and CPLEX solver on randomly generated instances are presented and compared. The final section, Section 4.5, concludes this chapter.

## 4.1 Literature review

The considered problem in this chapter is a bid generation problem with multi-periods in collaborative transportation. The bid generation problem (BGP) is a key decision problem for auction-based decentralized planning approaches in CTP, which is considered from the perspective of each carrier. The BGP can also be considered as the request selection problem. In [103] authors study a carrier's BGP in the context of TL transportation service procurement. In this paper, the focus is on the bundling method when an OR bidding language is used. They firstly define the bidder's optimality criterion of combinatorial bids. Then two heuristics are developed and compared, one is based on a fleet assignment model and the second is based on nearest insertion method. In [63] the carrier's optimal BGP is considered in combinatorial auctions for transportation procurement in TL transportation. Carriers employ vehicle routing models to identify the sets of lanes to bid for based on actual routes. Both column generation and Lagrangian based techniques are used for solving the carrier optimization model and promising results are reported. An exact strategy and two heuristic strategies have been proposed in [12] for bidding on subsets of requests. The

exact bidding strategy is based on the concept of elementary request combinations. The author shows that it is sufficient for a carrier to bid on each elementary request combination in order to guarantee the same result as bidding on each element of the power set of tendered requests. In [63] column generation and Lagrangian relaxation approaches are used in transportation problem to solve the BGP in combinatorial auction.

Our work in this chapter is related to the work in [65]. Both assume the BGP of a carrier, but the BGP considered in this thesis involves multi-periods. There are two interesting studies in multi-periods BGP [99], [44]. In these papers, each carrier considers multi-periods (days) when it determines which transportation requests to bid and serve in each period (day). Moreover, requests open for bid may span across multi-periods (days).

Other works related to ours include studies on the Oriented Problem (OP) and Team Orienting Problem (TOP). Selective TSP or TSP with profits is known as OP [61], [37]. A survey article in OP is presented in [34]. Multi-vehicle routing problem with profits is called Team Orienting Problem (TOP). In [13], [16] authors focus on TOP by considering multi-tours maximum collection problem and multi-tours VRP with profits. Full truckload VRP with profits is studied in [64]. In the article, all demands are assigned to outsource companies if they cannot be allocated to resources. To solve large instances, hybrid genetic algorithm is applied. An interesting study in VRP with profit is in [74]. In the paper, asymmetric prize collecting traveling salesman problem has been demonstrated. The objective of the problem is to minimize the cost of the routes and the penalty value of unserved customers.

The pickup and delivery problem with multi-vehicles and profit is suggested in [35]. In the article, there are multi-carriers and the customers with high profits are served. To solve the problem, general variable neighborhood search (GVNS) is compared with guided local search (GLS).

In [60] TOP with time windows is considered. The solution method in the paper is a granular variable neighborhood search. Multi-periods orienting problem is presented in [95]. In the problem different time windows are considered in terms of different serving days. Periodic VRP with unpaired pickup and delivery is proposed in [51]. In each period of the problem, the demand from a pickup node can be delivered to single or multiple delivery nodes. Therefore, the authors called the problem unpaired VRP with pickup and delivery. Different problems in periodic VRP is proposed in [19]. Different solutions are suggested in the paper to solve the problem. Multi-periods vehicle routing problem with due dates is investigated in [5]. In the paper, customers have to be served between delivery time and due date, otherwise they have to be paid a delay cost. Another type of periodic VRP is flexible periodic VRP is examined in [4]. In this type of VRP, each customer is served in different periods with the same demand. This type of periodic VRP is considered as a special case of inventory routing problem (IRP). In [14] periodic VRP with time windows is reported. In the paper, a mathematical model is presented and greedy randomized adaptive search is used to solve the model. There are little works in the multi-periods

VRP with time windows and the periodic VRP with pickup and delivery with time windows is not considered in the literature. An article in the multi-periods VRP with time windows is in [21] that the authors present general mathematical model to different VRP problems. The solution approach to solve the model is tabu search.

## 4.2 Formal definition and mathematical model

In this problem, we consider a carrier who wants to determine which requests to bid (select) among all requests open for bid (offered by all carriers) in a combinatorial auction to maximize its own profit by solving a bid generation problem. Since the carrier plans its transportation operations in advance and in a rolling horizon way as mentioned in the introduction, this bid generation problem involves multi-periods. We consider the problem in the less-than-truck load transportation, where each transportation request is a pickup and delivery request with time windows, reserved requests, and selective requests are involved as each request is associated with a profit which is the revenue to serve the request. Formally, the multi-periods bid generation problem with pickup and delivery requests, time windows, selective requests, reserved requests, and profits can be defined on a directed graph  $G = (V, E)$ , where  $V$  is the set of all nodes comprising all pickup nodes, delivery nodes, and the depot node of the carrier. In the graph,  $E$  is the set of all edges. The node set is defined as  $V = \{0, \dots, 2n + 1\}$ , where  $n$  represents the number of requests, 0 and  $2n+1$  denote the depot of the carrier,  $i$  and  $n + i$  represent the pickup and delivery nodes of request  $i$ ,  $i = 1, 2, \dots, n$ . Let  $W$  denotes the set of nodes excluding the depot node. As mentioned before the problem is multi-periods BGP with the number of periods denoted by  $H$ . In the problem, the carrier has a finite fleet of homogeneous vehicles whose index set is given by  $K = \{1, 2, \dots, VK\}$ , where  $VK$  is the maximum number of vehicles. The capacity of each vehicle is denoted by  $Q$  and the load of each vehicle cannot exceed its capacity. The set  $P = \{1, 2, \dots, n\}$  is the set of pickup nodes of all requests and the set of all delivery nodes is given by  $D = \{n + 1, \dots, 2n\}$ . In our model, each request  $i$  has its pickup node  $i$  and delivery node  $i + n$ . The demand of the pickup node of request  $i$  is denoted by  $d_i$ , while the demand of the delivery node of the same request is denoted by  $d_{i+n}$ , as  $d_{i+n} = -d_i$ . The delivery node of each request must be visited after its pickup node on the same route. The set of all requests is denoted by  $R$ , where  $R = (\bigcup_{l \in H} R_r^l) \cup R_s$ .  $R_r^l$  is the set of reserved requests that must be served in period  $l$  and  $R_s$  is the set of selective requests. Each selective request has a service period window and two time windows. The service period window determines in which periods the selective request can be served. The two time windows determine at which times in each period the pickup and delivery nodes of a request can be visited by a vehicle that serves the request. Both selective and reserved requests are associated with two types time windows, whereas only selective requests are associated with a service period window (the period in which each reserved request must be served is prespecified). The time window of pickup node  $i$  and delivery node  $i + n$  of request  $i$  are denoted by  $[e_i, l_i]$  and



$[e_{i+n}, l_{i+n}]$ , respectively. The service period window for each selective request  $i$  is represented by  $[E_i, L_i]$ . Each reserved request  $i \in R_r^l$  must be served in its prespecified period  $l$ ,  $l \in H$ . The traveling time and transportation cost between two nodes  $i$  and  $j$  are assumed to be same and they are denoted by  $t_{ij}$  and  $c_{ij}$  respectively. The maximum duration of each route  $i$  is limited by  $T$ . The multi-periods bid generation problem can be formulated as a mixed-integer linear programming model. In the model, parameters  $BM_{ij} = l_j - e_i$  is used to formulate linearly time windows constraints while  $C_i = Q + d_i$  is used in capacity constraints. The decision variables of the model include binary variables,  $x_{ijkh}$  and  $y_{ikh}$ . Two real variables  $U_{ikh}$  and  $CV_{ikh}$  are defined as follows:

$$x_{ijkh} = \begin{cases} 1 & \text{if and only if in period } h \text{ vehicle } k \text{ visits directly node } j \text{ after node } i \\ 0 & \text{Otherwise} \end{cases}$$

$$y_{ikh} = \begin{cases} 1 & \text{if and only if request } i \text{ is served by vehicle } k \text{ in period } h \\ 0 & \text{Otherwise} \end{cases}$$

$U_{ikh}$  = arrival time of vehicle  $k$  at node  $i$  in period  $h$

$CV_{ikh}$  = load of vehicle  $k$  when it leaves node  $i$  in period  $h$

$$\max(\sum_{i \in R} \sum_{h \in H} \sum_{k \in K} p_i y_{ikh} - \sum_{h \in H} \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{ij} x_{ijkh})$$

subject to:

$$\sum_{j \in V; j \neq i} x_{ijkh} - \sum_{j \in V; j \neq i} x_{ijkh} = 0 \quad \forall i \in W, \forall k \in K, \forall h \in H \quad (4.1)$$

$$\sum_{k \in K} \sum_{j \in P; j \neq 0} \sum_{h \in H} x_{0jkh} \geq 1 \quad (4.2)$$

$$\sum_{k \in K} \sum_{i \in D; i \neq (2n+1)} \sum_{h \in H} x_{i(2n+1)kh} \geq 1 \quad (4.3)$$

$$\sum_{k \in K} y_{ikh} = 1 \quad \forall l \in H, \forall i \in R_r^l \quad (4.4)$$

$$\sum_{h \in [E_i, L_i]} \sum_{k \in K} y_{ikh} \leq 1 \quad \forall i \in R_s \quad (4.5)$$

$$\sum_{j \in V; j \neq i, 0} x_{j(n+i)kh} = y_{ikh} \quad \forall i \in P, k \in K, \forall h \in H \quad (4.6)$$

$$\sum_{j \in V, j \neq i, 2n+1} x_{ijkh} = y_{ikh} \quad \forall i \in P, k \in K, \forall h \in H \quad (4.7)$$

$$U_{ikh} + t_{i,(n+i)} + s_i \leq U_{(n+i),kh} \quad \forall i \in P, k \in K, \forall h \in H \quad (4.8)$$

$$U_{jkh} \geq U_{ikh} + t_{ij} + s_i - BM_{ij}(1 - x_{ijkh}) \quad \forall i \in V, k \in K, \forall h \in H, \forall j \in V \quad (4.9)$$

$$e_i y_{ikh} \leq U_{ikh} \leq l_i + (1 - y_{ikh}) \sum_{j \in V} BM_{ij} \quad \forall h \in H, \forall i \in V, \forall k \in K \quad (4.10)$$

$$U_{ikh} + t_{ij} + s_i - BM_{ij}(1 - x_{ijkh}) \leq T \quad \forall i \in V, k \in K, \forall h \in H, \forall j \in \{0\} \quad (4.11)$$

$$CV_{ikh} + d_i - C_i(1 - x_{ijkh}) \leq CV_{jkh} \quad \forall i \in V, k \in K, \forall h \in H, \forall j \in V \quad (4.12)$$

$$\max\{0, d_i\} \leq CV_{ikm} \leq \min\{Q, Q + d_i\} \quad \forall i \in V, k \in K, \forall h \in H \quad (4.13)$$

$$x_{ijkh} \in \{0, 1\} \quad \forall i, j \in V, \forall k \in K, \forall h \in H \quad (4.14)$$

$$U_{ikh} \geq 0 \quad \forall i \in V, \forall k \in K, \forall h \in H \quad (4.15)$$

$$CV_{ikh} \geq 0 \quad \forall i \in V, \forall k \in K, \forall h \in H \quad (4.16)$$

$$y_{ikh} = 0 \quad \text{for any } i \in R_s, \forall k \in K, \text{ for any } h \notin (E_i, L_i) \quad \text{and} \quad y_{ikh} = 0 \quad \text{for any } i \in R_{rl}, l \neq h \quad (4.17)$$

The objective function represents the total profit of the carrier, which is equal to the difference between the total payments of serving requests in all periods and the total transportation cost. Constraint (4.1) ensures when a vehicle arrives at a node in a period, it must leave from the node in the same period. Constraints (4.2) and (4.3) signify each vehicle leaves its depot in a period must return to the depot in the same period. Equation (4.4) implies each reserved request must be served in its pre-specified period. Equation (4.5) indicates each selective request can be served in a period within its service period window or not served. Constraints (4.6) and (4.7) guarantee if a request is served in a period, its delivery node must be visited after its pickup node with the same vehicle in the same period. Equations (4.8)-(4.11) specify time windows constraints on the pickup and delivery nodes of each request and the constraint on the maximum duration of each route. Constraints (4.12)-(4.13) ensure vehicle capacity constraints. Equations (4.14)-(4.17) describe the variables.

### 4.3 Metaheuristic approaches to solve multi-periods bid generation problem

In section 4.2, we have presented the MILP model to multi-periods BGP. Since existing commercial solvers are not powerful enough to solve the large size of PDPTW in a reasonable time, we apply GASA and BBO to solve this problem.

#### 4.3.1 Initial solution construction procedure

The presented initial solution for the problem in this chapter is close to the initial solution structure in Chapter 3 (section 3.3.1) with a small difference in step 3. In the proposed hybrid genetic algorithm with simulated annealing (GASA), the size of population determines the number of initial solutions to construct. As described in the previous sections, all reserved requests must be served in their predetermined periods while some selective requests may be not served when they are not profitable. Profitable selective requests may be served in the periods within their service period windows.

The three ways are given in the following:

1. Only the reserved requests are served.
2. All reserved requests are served firstly and then selective requests are served.
3. All reserved requests must be served first and selective requests are served only if its assigned period is not zero (it is explained in section 4.3.2.1). That is, if the period assigned to the request is zero, it will be not served.

If the probability of choosing each three ways to construct an initial solution is denoted by  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$ , respectively, then  $\alpha_1 + \alpha_2 + \alpha_3 = 1$ .

#### 4.3.2 Hybrid genetic algorithm with simulated annealing (GASA)

The applied GASA in this section is the same as GASA in Chapter 3. It means that in this algorithm, a chromosome represents each solution and the population in each iteration is composed of multiple chromosomes. All components of GASA for multi-periods BGP with pickup, delivery and time windows will be explained in the rest of this section. In each iteration of GASA, the maximum profit is updated thanks to the obtained profits in the latest generation. SA is applied to each solution of the present population. Each solution is selected with a probability based on a simulated annealing procedure.

##### 4.3.2.1 Solution representation in GASA

The chromosome of multi-periods BGP is defined by three vectors  $X$ ,  $Y$ , and  $Z$ . Vector  $X$  includes all pickup nodes and delivery nodes and its dimension is  $|P \cup D|$  as the index of each request

is the same as its pickup node index in the three vectors. What's more, each selective request must be served in a period within its service period window. Thus, an extra period is introduced to indicate a not served selective request. This extra period is referred to period 0. The size of vector  $Y$  is equal to the number of all requests and each component of vector  $Y$  indicates the period assigned to a request. Since a selective request is not necessarily to be served, it will not be served if it is not profitable. The dimension of vector  $Z$  is equal to the number of all requests. In vector  $Z$ , each component indicates the index of the vehicle serving a request. A simple instance (example) is given in the following to help the understanding of chromosome structure, where the data of this instance are given in Table 4.1 and a solution of the instance is given in Table 4.2 and Table 4.3, where four requests are served in day 1 and day 2 and no request is served in day 3. The chromosome represented by three vectors  $X, Y$ , and  $Z$  for this solution is given in Fig 4.1:

**Table 4.1:** Data of a simple instance.

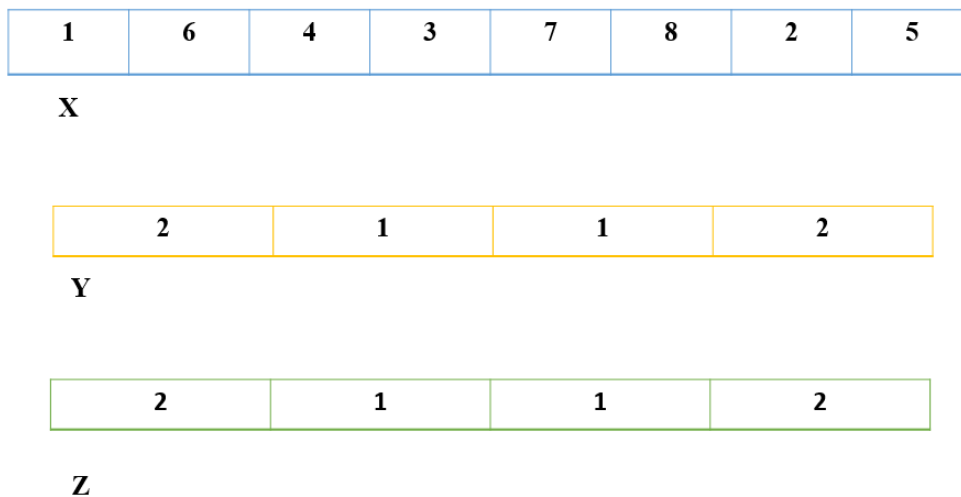
<b>Small instance Data</b>	<b>Value</b>
Number of periods	3 days
Number of requests	4 requests
Number of pickup nodes	4 nodes
Number of delivery nodes	4 nodes
Total number of customer nodes	8 nodes
Reserved requests	{3,4}
Selective requests	{1,2}
Prespecified period for serving each reserved request	3, day 1 and 4, day 2
Service period window for each selective request	1, day [2,3] and 2, day [1,3]
Index of pickup node of request $i$	As the same as its request
Index of delivery node of request $i$	$i + \text{number of requests}$
Number of vehicles available in each period	2 vehicles

**Table 4.2:** The data for selective and reserved requests served in the first day.

<b>reserved requests and selective requests served in day 1</b>	<b>Value</b>
Index of reserved request served	3
Pickup node index of reserved request	3
Delivery node index of reserved request	7
Index of vehicle used to serve the reserved request	1
Index of selective request served	2
Pickup node index of selective request	2
Delivery node index of selective request	6
Index of vehicle use to serve the selective request	1

**Table 4.3:** The data for selective and reserved requests served in the second day.

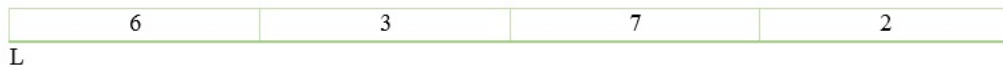
reserved requests and selective requests served in day 2	Value
Index of reserved request served	4
Pickup node index of reserved request	4
Delivery node index of reserved request	8
Index of vehicle used to serve the reserved request	2
Index of selective request served	1
Pickup node index of selective request	1
Delivery node index of selective request	5
Index of vehicle used to serve the selective request	2



**Figure 4.1:** Illustration of chromosome's structure

To transform a given chromosome into a feasible solution, we apply a schedule repair function to construct a sequence of visited pickup and delivery nodes in each feasible route. Actually, the most important thing in constructing a route in each period is to know which pickup and delivery nodes are assigned to the route. According to the chromosome defining a solution, we can determine all pickup and delivery nodes are visited by each route in each period. We put all these pickup and delivery nodes in a list we call it list  $L$ . To construct a route, a node is selected from the beginning of the list. If the node is a pickup node, it will be added to the route. Otherwise, if the node is a delivery node, it is added to a route only if its corresponding pickup node has been added to the route previously. Once a node from list  $L$  is added to the route, it will be deleted from the list. The following procedure explains the schedule repair function applied for generating routes in day 1.

1. In day 1 and route 1, there are 4 pickup/delivery nodes, which constitute the list  $L$ .



2. A selected node from the beginning of list  $L$  is inserted into a route  $R$  if it is a pickup node or its pickup node has been inserted in route  $R$  previously. For example,  $L(1) = 6$  is a delivery node because its index is larger than the number of requests. When it is chosen, all nodes of route  $R$  are checked. If its corresponding pickup node has been inserted previously into list  $R$ , the node 6 is inserted into the route as well. Otherwise, it will be ignored. The next node is  $L(2) = 3$ , which is a pickup node, so it is inserted into route  $R$ .



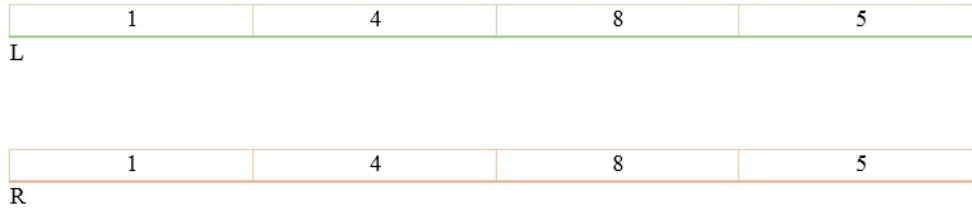
In the next step after removing node 3 from list  $L$ , it is checked from the beginning to choose the next node to insert into route  $R$ . The node 6 is a delivery node and its pickup node, node 2, is not in the route  $R$ . Consequently, the component which is chosen from the list  $L$  as a next node is node 7. The node 7 is a delivery node and its corresponding pickup node, node 3, has been inserted previously in route  $R$ . Therefore, node 7 is inserted to list  $R$  and deleted from list  $L$ . Generally, after deleting a node from the list  $L$  and inserting in the list  $R$ , the list  $L$  will be explored from the beginning to choose a node to insert into the route  $R$ .



After deleting node 7, the list  $L$  is checked from the beginning again. Since the pickup node of 6 has not been inserted previously into route  $R$ , exploring the list  $L$  continues. By exploring the list  $L$  from the beginning, there is a single node, node 6, which is the corresponding delivery node of node 2. Because its pickup node, node 2, has been added to the list  $R$  previously, so the node 6 is inserted in the route list  $R$ .



The same procedure is applied to construct the routes in other days. The list  $L$  and route  $R$  for first day are given below:



### 4.3.2.2 GASA algorithm operators

The GASA chromosome in this chapter and Chapter 3 has the same structure. In the previous chapter, there are multi-carriers, which have been illustrated by vector  $Y$ . In this chapter, there are multi-periods instead of multi-carriers and they can be illustrated by the same vector. The other two vectors with the same structure, vector  $X$  and  $Z$ , can be applied to the GASA in this chapter. In Chapter 3, each component of vector  $Y$  is the suitable carrier to serve each request. In this chapter, each component of vector  $Y$  is the suitable period to serve a request. According to the structure of each solution for the studied problem, two crossover operators and two mutation operators are proposed. In the rest of this section, these operators are summarized. The comprehensive and complete descriptions of operators are in Chapter 3.

#### *Crossover operator of vector $X$*

In the suggested chromosome structure to the GASA, vector  $X$  has a permutation structure and a single point crossover is applied. In fact, this crossover operator avoids node duplications in the offsprings.

#### *Crossover operator of vector $Y$ and $Z$*

The uniform crossover is a suitable crossover operator to the vectors  $Z/Y$ . In the uniform crossover, a mask vector with zero and one entries (genes) and the same size as  $K/Y$  is generated. The value of each offspring's gene is produced according to the value of the corresponding gene in the mask vector and the component of gene in the parent.

#### *Mutation operator of vector $X$*

To have a diversified feasible solution, a mutation operator with the following two steps is applied to vector  $X$  with the permutation structure.

1. Choose randomly two genes of vector  $X$ .
2. Choose randomly one of the insertion, swap, and reversion operations. Execute the selective operation on the selective genes.

#### *Mutation over vectors $Z$ and $Y$*

The mutation over vector  $Z$  is realized in three steps. In the first step, the number of mutated genes of vector  $Z$  is determined randomly and denoted by  $l$ . The number of mutated genes are obtained in the following way: An integer number  $h$  is randomly generated between 1 and  $\dim[Z]$ , where  $\dim[Z]$  is the number of genes in vector  $Z$  and  $VK$  is the total number of vehicles. This number is then multiplied by a mutation rate  $r$ , leading to  $hr$ . By rounding  $hr$  to the smallest integer number larger than or equal to  $hr$ , the number  $l$  is finally obtained. In the second step,  $l$  genes are randomly selected from  $Z$ . In the third step, for each selective gene of  $Z$ , an integer number is randomly generated between 1 and  $VK$ , and the gene in  $Z$  is changed to this value. The mutation over vector  $Y$  is the same procedure as the mutation over vector  $Z$ .

### 4.3.2.3 Simulated annealing algorithm in GASA

Most heuristic algorithms only accept an improved solution, whereas simulated annealing gives a chance to worse solutions, which accepts a worse new solution with a probability. The new solution acceptance probability is given by  $e^{-\frac{f(s)-f(s')}{T_0}}$ , where  $f(s')$  is the objective value of the new solution and  $f(s)$  is the objective value of the current solution and  $f(s') < f(s)$ . The acceptance probability depends on both the temperature parameter  $T_0$  and the profit decrease  $f(s) - f(s')$  while the temperature is decreased at the end of each iteration. The temperature reduction is performed by multiplying  $T_0$  with a cooling factor  $co \in (0, 1)$ . To attain a slow cooling, the cooling factor must be set close to one. At the beginning of the execution of the GASA algorithm the temperature parameter,  $T_0$ , is set to  $T_{in}$  and a solution with profit 30% lower than the initial solution is accepted with a given possibility  $pr$ . In GASA, to produce the next generation solutions, at the end of each GA iteration, the solutions generated by crossover and mutation are sorted and merged with the current population. GASA utilizes the SA rule to determine whether each solution in the sorted list becomes a solution (chromosome) in the next generation. In other words, to give an opportunity being selective to each solution, SA rule is used to compare the solutions in the current population and the solutions obtained by crossover and mutation. In the same way as in simulated annealing algorithm, the temperature of SA is decreased at the end of each iteration in our GASA algorithm.

### 4.3.3 Biogeography-based optimization algorithm (BBO)

BBO is a meta-heuristic algorithm inspired by Biogeography Science to explore the solution space. In [90], the theory of Biogeography is used to solve the optimization problems according to this fact that Biogeography studies the manner of organisms during time and space. Simon's proposed algorithm like GA is a population-based algorithm as its operation is over the population of solutions to search the best solution. By comparing GA and BBO, each chromosome in GA, which is considered as an individual with fitness value is habitat with suitability index that is called HSI (Habitat Suitability Index). Usually, a habitat with high HSI is acceptable and yields better solutions. In the algorithm, species migrate from the regions with high HSI to places with



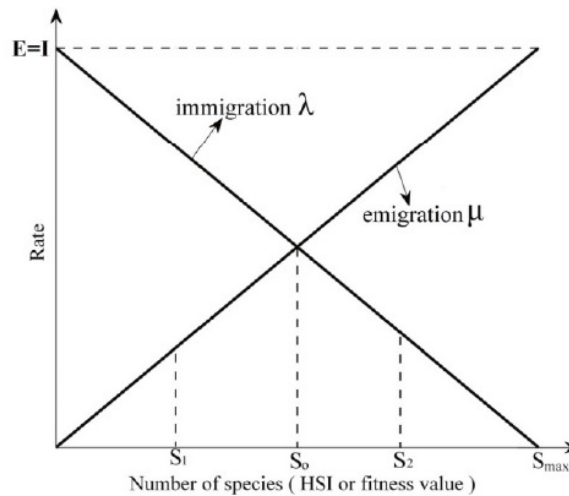
low HSI to upgrade these regions. Emigration and immigration are two operators to improve solutions and maximize the HSI, where the first operator is applied on the regions with high HSI and the second one is used in opposite side. Solution structure of BBO is called habitat and its representation is same as individual in GASA. In [82], two algorithms are compared and presented in Table 4.4.

**Table 4.4:** The comparison of GA and BBO.

<b>BBO</b>	<b>GA</b>
Population-based	Population-based
Habitat (individual)	Chromosome (individual)
SIV	Gene
Habitats consisted of SIV	Chromosomes consisted of genes
Mutation operator	Mutation operator
Migration operators (immigration and emigration)	Crossover operator
Good solution is characterized by high HSI	Good solution is characterized by high fitness
A good habitat is one which has more diversity and species	A good chromosome is the one which has more value of fitness function
No individual of initial population is discarded during iterations but it is modified	Initial individuals discarded by GA operators during iterations

#### 4.3.3.1 Selection strategy

The main difference between GASA and BBO is the modification of solutions in diverse iterations instead of their omission. Actually, the main goal of BBO in this step is determining the regions, which needs corrections along with highlighting solutions to use their attributions. Selection methods in BBO are divided to two parts; first migration operators, second mutation. The migration operator of BBO is like GASA crossover operator, but BBO migration operator is performed between immigrant habitat and emigrant one. In the current sections, operators are described. Like GASA, the most important part of BBO is solution representation.  $H_i$  is the solution structure of BBO, called habitat, is the same as an individual structure of GASA (explained as in section 4.3.2).



**Figure 4.2:** Model of immigration  $\lambda$  and emigration  $\mu$  probabilities [90].

1. Selection approach of migration operators

Overall, there are two decisions. Does the special place ( $H_i$ ) alter or not? Produced random number should be compared with immigration rate ( $\lambda_i$ ). Setting region ( $H_j$ ) as an origin of conducting attributes to  $H_i$  will be the next step. Roulette wheel is an appropriate technique that is implemented at the emigration rate ( $\mu_j$ ). In the current paper, fulfilling BBO mutation operator is the same as GASA mutation operator. In reality, the output of this phase determines whether the considered region has to be mutated or not.

2. Migration operator is used to the modification of solutions and its idea related to the theory of migration in species. In this process, each  $H_j$  solution is selected according to its immigration and emigration rate to achieve or share attributes, respectively. There is necessary to mention, SIV in BBO algorithm is the equivalent with a gene in chromosome structure of GASA. If SIV is a solution attribute, the following inequality shows the mentioned possible relation  $H_i(SIV) \leftarrow H_j(SIV)$ . Figure 4.2 illustrates the mechanism in translation between both rates in relation to species plenty [90]. Indeed, interaction between high and low HSI as mentioned previously causes emigration rate to reduce due to species increasing in habitat; as a result of this increment, immigration rate decreases and the emigration rate rises. In Fig. 4.2, E and I demonstrate maximum immigration and emigration rates respectively in which most of the time their values are 1.  $S_{max}$  is the maximum number of species that a habitat can accept and  $S_0$  is the balance point, where two rates are equal. Although in Fig. 4.2 rates show it can be used for the linear model, the non-linear model also can be discussed [90].

Accordingly, after calculating HSI, immigration and emigration rates are calculated by

inequalities (4.18) and (4.19).  $K_i$  is the rank of each solution after sorting them based on HSI insofar as the better solution has high rank and consequently larger  $k_i$ .

$$\lambda_i = I(1 - \frac{k_i}{n}) \quad (4.18)$$

$$\mu_i = E(\frac{k_i}{n}) \quad (4.19)$$

In equations (4.18) and (4.19),  $I$  and  $E$  are two coefficients and their values are determined by parameter tuning. The other variables related to the concept of migration is “probability of existing  $S$  species” in a habitat. When species departure from time  $t$  to  $\Delta t$ , there are species in a habitat at  $t + \Delta t$ . In fact, it shows from each iteration to next iteration, one of the following states can be happened to each solution. Equation (4.20) defines these states.

1. At  $t$  there are  $S$  species; therefore, it is not needed to change in  $t + \Delta t$ .
2. At  $t$  there are  $S-1$  species and immigration to  $t + \Delta t$  is required.
3. At  $t$  there are  $S+1$  species and an emigration to  $t + \Delta t$  is required.

Equation (4.20) defines these states.

$$P_s(t + \Delta t) = P_s(t)(1 - \lambda_s \Delta t - \mu_s \Delta t) + P_{s-1} \lambda_{s-1} \Delta t + P_{s+1} \mu_{s+1} \Delta t \quad (4.20)$$

After solving equation (4.20) in a stable state, the underlying mathematical inequality yields  $P_i$  [90]. Subsequently, after calculating  $\lambda_i$ ,  $\mu_j$  and implementing selection operator, migration operator is applied.

## 4.4 Experimental results

To evaluate the performance of the proposed two metaheuristic algorithms, we applied them to solve 8 sets of instances of small to large sizes taken from the literature and compared them with the MILP solver of CPLEX 12.6 in terms of profit and computation time. In the following sections, we present these instances and the comparison results. The results display both GASA and BBO algorithm remarkably outperform CPLEX, while the GASA algorithm outperforms the BBO algorithm.

### 4.4.1 Instances

The instances are taken from [65] with the same reserved requests, selective requests, the coordinates of each node, the demand and time windows of each request. 56 instances are generated in small size ( $n \in \{6, 8, 10, 20\}$ ), medium size ( $n \in \{30, 40, 50\}$ ) and large size ( $n \in 100$ ).

The name of each instance has a format  $n - |R_r| - |R_s| - source$ . As an example, we explain instance 10-5-5-50a. There are 10 requests in total, including 5 reserved requests (1-5) and 5 selective requests (6-10). The code 50a means this instance is derived from the original one, prob50a : Only the 10 first requests appearing in prob50a are copied in instance 10-5-5-50a. To include the period specification, we consider there are 5 periods and each period has its own reserved requests according to the random function.

In [65], for each instance size (10, 20, 30, 40, 50 or 100 requests), authors build nine instances using nine PDPTW files. These nine instances can be decomposed in three types: Three with roughly one-third of reserved requests, three with 50% of reserved requests, and three with two-thirds of reserved requests. The fleet size in [65] is adjusted in accordance with the proportion of the number of requests extracted from the original instance.

The service price of each request is set according to the coordinates of its nodes. Take request  $i$  as an example, let  $d_{i,n+i}$  denotes the distance from its pickup node  $i$  to delivery node  $n+i$ , then this request is given a service price  $p_i = d_{i,n+i} \cdot \lambda, \lambda \in [3, 5]$ . This formula generates a large proportion of profitable requests and a small proportion of non-profitable requests [65].

The instances are executed in CPLEX12.6 on a laptop with an Intel core i5 CPU and 4.00 GB RAM. Note that for the medium instances and large instances, it is impossible to solve the MILP model optimally by CPLEX after 2 hours and 4 hours of running time, respectively.

#### 4.4.2 Parameter setting

The performance of GASA algorithm and BBO algorithm depends on the quality of their parameters setting. The values of some parameters of the two algorithms are determined empirically and are given in Table 4.5. The values of their other parameters are tuned by using Taguchi method, which is a popular and efficient parameter tuning technique based on fractional experiments (FFE) [87]. The parameters of the two algorithms selected to tune are in Table 4.6, and their values are given in Table 4.7 and Table 4.8 according to Fig. 4.3 and Fig. 4.4 by utilizing Taguchi approach previously discussed in Chapter 3.

**Table 4.5:** Parameter values of GASA determined empirically.

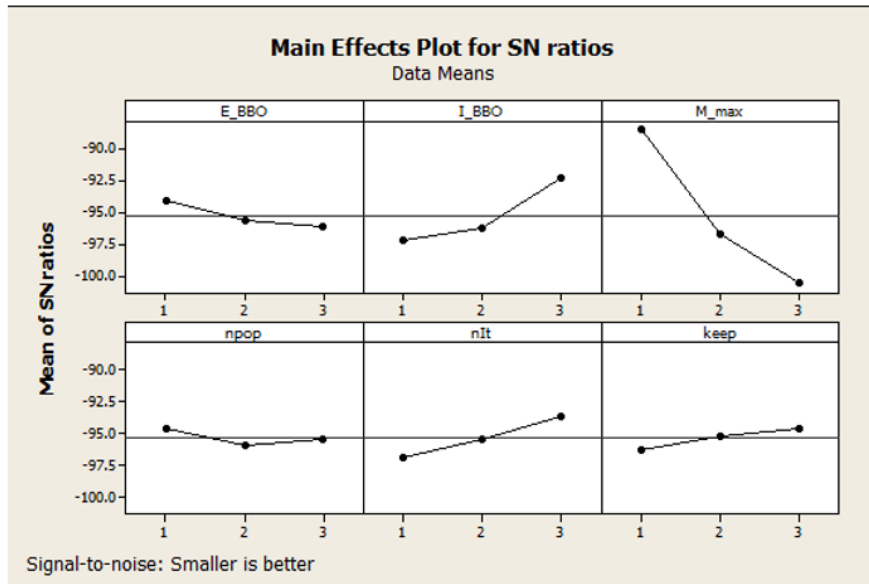
Parameter	Description	Value
$SubIt$	Number of iteration of SA	110
$\mu_m$	Mutation rate	0.26

#### 4.4.3 Test results and optimality gaps

After the parameter calibration, we execute GASA algorithm, BBO algorithm and CPLEX on all instances. Since both GASA and BBO algorithms contain random factors to reduce the impacts

**Table 4.6:** The domain of calibrated parameters of GASA and BBO.

Methodology	Parameter	Range	Low(1)	Medium(2)	High(3)
GASA	Npop	400-600	400	500	600
	nIt	450-800	450	625	800
	Pc	0.4-0.8	0.4	0.6	0.8
	Pm	0.05-0.3	0.05	0.175	0.3
BBO	E	1-2	1	1.5	2
	I	1-2	1	1.5	2
	Mmax	0.05-0.15	0.05	0.1	0.15
	Npop	200-300	200	250	300
	nIt	450-800	450	625	800
	Keep	0.15-0.25	0.15	0.2	0.25

**Figure 4.3:** Output of Taguchi ratio for BBO.**Table 4.7:** Parameter values of GASA determined by Taguchi.

Methodology	Parameter	Description	Value
GA	Npop	Population size	150
	nIt	Number of iterations	800
	$P_c$	Crossover probability	0.6
	$P_m$	Mutation probability	0.05

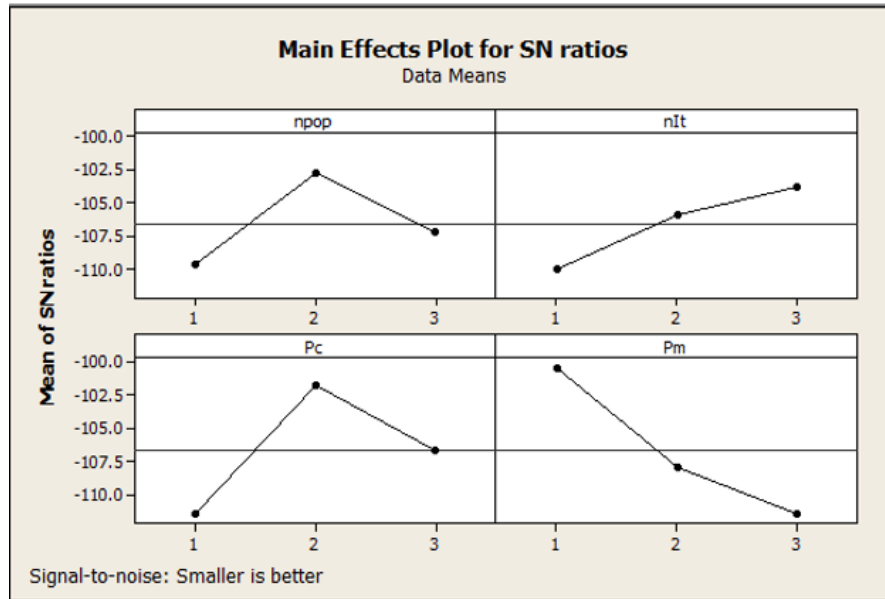


Figure 4.4: Output of Taguchi ration for GASA.

Table 4.8: Parameter values of BBO determined by Taguchi.

Methodology	Parameter	Description	Value
BBO	E	Emigration rate	1
	I	Immigration rate	2
	$M_{max}$	Mutation probability	0.05
	npop	Population size	200
	nIt	Number of iterations	800
	Keep	Elitism parameter	0.25

of these factors on their performance evaluation, each algorithm is executed three times for each instance. The average profit and the average running time (in seconds) obtained by the three executions are reported. For CPLEX, since the considered carrier collaboration problem is NP-hard [66], it is very time consuming to solve optimally medium size and large size instances. For this reason, we set a maximum running time for CPLEX to solve each medium or large size instance. The time limitation is 2 hours for each medium size instance with 20 requests and 30 requests 4 hours for each large size instance with 40, 50 and 100 requests. Our two proposed algorithms are compared with each other and with CPLEX based on criteria defined in Table 4.9, where  $Obj_{GASA}$  and  $Obj_{BBO}$  are the profits of the studied problem obtained by the two algorithms, respectively;  $UB_{MILP}$  and  $LB_{MILP}$  are the upper bound and the lower bound of the objective value of the problem obtained by CPLEX in the preset running time.

The computation results are given in Table 4.10, Table 4.11, and Table 4.12. From Table 4.10 and Table 4.11, we can see, for small 6 requests instances, CPLEX, GASA, and BBO all can find

**Table 4.9:** Criteria used for comparison of GASA, BBO and CPLEX.

Criterion	Description
$Gap_{MILP}$	The relative gap between $LB_{MILP}$ and $UB_{MILP}$ , defined as $\frac{(UB_{MILP}-LB_{MILP})}{UB_{MILP}}$
$Gap_{BBO}$	The relative gap between $UB_{MILP}$ and $Obj_{BBO}$ , defined as $\frac{(UB_{MILP}-Obj_{BBO})}{UB_{MILP}}$
$Gap_{GASA}$	The relative gap between $LB_{MILP}$ and $Obj_{GASA}$ , defined as $\frac{(UB_{MILP}-Obj_{GASA})}{UB_{MILP}}$

an optimal solution. However, for some of small instances with 8 requests and most medium to large size instances, CPLEX could not even find a feasible solution in a preset computation time, whereas both algorithms could find a feasible solution for all instances. Actually, for most instances with the number of requests more than 8, no feasible solution is obtained by CPLEX, so we compare the solutions obtained by our two algorithms based on their relative gaps with the upper bound obtained by CPLEX, i.e., using the above-mentioned criteria. The applied algorithms can find an optimal solution for all small instances with 6 requests. They can find a solution with the relative profit gap smaller than 0.34% for 8 requests instances and can find a solution with the relative profit gap smaller than 4.50% for 10 requests instances. For medium instances, our GASA algorithm can find a solution with the gap smaller than 5.6%, whereas our BBO algorithm can find a solution with the gap smaller than 6.117%. For instances with 30 requests, GASA algorithm can find a solution with the gap smaller than 6.98%, while our BBO algorithm can find a solution with the gap smaller than 7.74%. For instances with 30 requests, our GASA algorithm can find a solution with the gap smaller than 10.31%, whereas our BBO algorithm can find a solution with the gap smaller than 11.60%. For instances with 50 requests, our GASA algorithm can find a solution with the gap smaller than 13.2%, whereas our BBO algorithm can find a solution with the gap smaller than 13.6%. In the instances with 100 requests, our GASA algorithm can find a solution with the gap smaller than 16.4% and BBO algorithm can find a solution with the gap smaller than 18.2%. Moreover, our proposed algorithms are much better than CPLEX, Table 4.12, in terms of running time for medium and large instances, although the running time of the two proposed algorithms increases naturally with the size of instance and the computation time of CPLEX is smaller than those of the algorithms for small instances with 6 requests. To compare the two algorithms we utilize RPD that has been mentioned in the previous chapter. After calculating RPD, statistical analysis is done as we mentioned in Chapter 3. According to Table 4.13, Table 4.14, and Fig. 4.5,  $H_0$  is rejected. GASA has better profit, small RPD, and running time smaller than those of BBO respectively for all instances; therefore GASA

outperforms BBO. This may be because using diversification and intensification operators by using SA in GASA, which makes it able to search diversified regions in the solution space.

**Table 4.10:** Computational results of GASA, BBO, and CPLEX –part one.

<b>Instance Number</b>	$LB_{MILP}$	$UB_{MILP}$	$Obj_{GASA}$	$Obj_{BBO}$
6 request instances (small instance)				
6-3-3	256.525	265.525	265.525	256.525
8 request instances (small instances)				
8-4-4	512.015	514.731	514.682	513.373
10 request instances (small instances)				
10-5-5a	909.810	984.660	979.614	972.828
10-5-5b	968.6747	1011.463	1008.415	991.473
10-5-5c	974.001	1028.626	1019.693	1009.930
10-3-7d	1125.153	1157.164	1148.267	1138.231
10-3-7e	828.254	903.828	897.786	893.741
10-3-7f	-	1210.611	1183.689	1179.442
10-7-3g	917.419	936.1774	929.350	926.464
10-7-3h	1266.383	1298.205	1286.358	1281.631
10-7-3i	-	1375.914	1328.921	1314.160
20 request instances (medium instances)				
20-10-10a	-	2619.785	2473.626	2461.335
20-10-10b	-	2286.16	2194.358	2173.123
20-10-10c	-	3815.445	3651.975	3609.450
20-5-15d	1832.804	2015.517	1991.431	1988.117
20-5-15e	-	2568.903	2462.240	2406.062
20-5-15f	-	2819.162	2698.150	2673.227
20-15-5g	-	3765.529	3577.579	3573.118
20-15-5h	-	3324.128	3153.251	3131.662
20-15-5i	-	4514.717	4278.305	4238.713
30 request (medium instances)				
30-15-15c	-	7432.469	7003.483	6991.804
30-10-20d	-	6629.284	6263.020	6150.135
30-10-20e	-	8803.686	8249.580	8183.466
30-10-20f	-	8217.720	7783.125	7694.472
30-20-10g	-	11214.849	10452.390	10425.340
30-20-10h	-	12717.674	11991.118	11940.267
30-20-10i	-	11591.228	10783.710	10694.481

---



---

40 request (large instances)			
40-20-20a	-	12065.100	11103.219 10983.392
40-20-20b	-	12891.558	11864.083 11612.618
40-20-20c	-	13494.750	12394.446 12293.720
40-15-25d	-	12205.416	10948.519 10860.099
40-15-25e	-	10611.560	9712.119 9559.921
40-15-25f	-	13563.683	12297.303 12073.077
40-25-15g	-	11115.574	9979.391 9828.174
40-25-15h	-	14780.456	13421.144 13183.012
40-25-15i	-	12737.218	11462.503 11274.236
50 request (large instances)			
50-25-25a	-	23781.930	21105.171 20549.460
50-25-25b	-	20231.195	18171.659 17810.230
50-25-25c	-	17400.241	15318.582 15159.107
50-20-30d	-	22083.155	19809.005 19206.231
50-20-30e	-	24757.404	22108.796 21593.364
50-20-30f	-	20495.213	18004.147 17954.968
50-30-20g	-	18370.485	16482.312 16472.686
50-30-20h	-	22756.810	20018.565 19784.818
50-30-20i	-	17531.044	15435.438 15289.753
100 request (large instance)			
100-50-50a	-	88406.782	75606.060 73984.573
100-50-50b	-	90722.467	78110.724 77302.100
100-50-50c	-	103462.656	87843.157 85268.202
100-25-75d	-	112103.589	93850.223 92412.151
100-25-75e	-	81434.327	69715.008 67737.466
100-25-75f	-	90686.991	75815.239 74337.463
100-75-25g	-	93663.471	78945.834 76826.505
100-75-25h	-	82124.016	70237.949 68772.828
100-75-25i	-	100133.674	86774.130 85564.521

---

**Table 4.11:** Gap of GASA, BBO, and CPLEX.

<b>Instance Number</b>	$Gap_{GASA}$	$Gap_{BBO}$	$Gap_{MILP}$
6 request instances (small instance)			
6-3-3	0	0	0
8 request instances (small instance)			
8-4-4	0.009	0.263	0.333
10 request instances (small instances)			
10-5-5a	0.512	1.201	7.601
10-5-5b	0.301	1.976	4.230
10-5-5c	0.869	1.818	5.311
10-3-7d	0.769	1.636	2.766
10-3-7e	0.668	1.116	8.361
10-3-7f	2.223	2.574	-
10-7-3g	0.729	1.037	2.003
10-7-3h	0.912	1.276	2.451
10-7-3i	3.415	4.488	-
20 request instances (medium instances)			
20-10-10a	5.579	6.048	-
20-10-10b	4.015	4.944	-
20-10-10c	4.284	5.398	-
20-5-15d	1.195	1.359	9.065
20-5-15e	4.1520	6.338	-
20-5-15f	4.152	6.295	-
20-15-5g	4.991	5.109	-
20-15-5h	5.140	5.789	-
20-15-5i	5.236	6.113	-

---

30 request (medium instances)			
30-15-15c	5.771	5.928	-
30-10-20d	5.524	7.227	-
30-10-20e	6.294	7.045	-
30-10-20f	5.288	6.367	-
30-20-10g	6.797	7.039	-
30-20-10h	5.712	6.113	-
30-20-10i	6.967	7.774	-
40 request (large instances)			
40-20-20a	7.973	8.965	-
40-20-20b	8.048	9.920	-
40-20-20c	8.153	8.899	-
40-15-25d	10.297	11.022	-
40-15-25e	8.476	9.910	-
40-15-25f	9.336	10.989	-
40-25-15g	10.221	11.581	-
40-25-15h	9.196	10.807	-
40-25-15i	10.007	11.485	-
50 request (large instances)			
50-25-25a	11.255	13.592	-
50-25-25b	10.180	11.966	-
50-25-25c	11.963	12.879	-
50-20-30d	13.027	13.540	-
50-20-30e	10.698	12.780	-
50-20-30f	12.154	12.394	-
50-30-20g	10.278	10.330	-
50-30-20h	12.032	13.059	-
50-30-20i	11.953	12.784	-
100 request (large instance)			
100-50-50a	14.479	16.313	-
100-50-50b	13.901	14.792	-
100-50-50c	15.096	17.585	-
100-25-75d	16.282	17.565	-
100-25-75e	14.391	16.819	-
100-25-75f	16.398	18.028	-
100-75-25g	15.713	17.976	-
100-75-25h	14.473	16.257	-
100-75-25i	13.341	14.549	-

---

**Table 4.12:** Computational time of GASA, BBO and CPLEX.

<b>Instance Number</b>	$CPU_{MILP}$	$CPU_{GASA}$	$CPU_{BBO}$
6 request instances (small instances)			
6-3-3	67.649	85.117	89.216
8 request instances (small instances)			
8-4-4	98.365	100.794	123.675
10 request instances (small instances)			
10-5-5a	500	184.342	215.429
10-5-5b	500	160.105	204.767
10-5-5c	500	173.110	217.353
10-3-7d	500	186.185	253.419
10-3-7e	500	190.856	229.022
10-3-7f	500	185.647	232.109
10-7-3g	500	177.961	211.360
10-7-3h	500	201.128	249.029
10-7-3i	500	198.403	253.715
20 request instances (medium instances)			
20-10-10a	3600	331.511	395.389
20-10-10b	3600	343.992	416.356
20-10-10c	3600	351.836	407.354
20-5-15d	3600	326.537	402.911
20-5-15e	3600	322.419	373.902
20-5-15f	3600	335.395	411.363
20-15-5g	3600	339.014	402.742
20-15-5h	3600	346.191	413.808
20-15-5i	3600	333.587	421.066

---

30 request instances (medium instances)			
30-15-15c	3600	547.992	618.982
30-10-20d	3600	513.14	626.153
30-10-20e	3600	525.082	644.195
30-10-20f	3600	546.232	634.247
30-20-10g	3600	536.779	639.816
30-20-10h	3600	538.278	630.252
30-20-10i	3600	521.305	646.795
40 request instances (large instances)			
40-20-20a	7200	825.753	917.153
40-20-20b	7200	819.124	884.221
40-20-20c	7200	814.638	892.439
40-15-25d	7200	844.225	903.678
40-15-25e	7200	826.439	920.164
40-15-25f	7200	856.030	929.082
40-25-15g	7200	819.471	982.464
40-25-15h	7200	837.366	907.201
40-25-15i	7200	840.476	916.676
50 request instances (large instances)			
50-25-25a	7200	1719.124	1984.866
50-25-25b	7200	1725.753	1871.513
50-25-25c	7200	1630.225	1846.887
50-20-30d	7200	1696.439	1881.504
50-20-30e	7200	1756.309	1972.163
50-20-30f	7200	1690.491	2082.744
50-30-20g	7200	1655.366	1869.513
50-30-20h	7200	1740.672	1944.025
50-30-20i	7200	1842.201	2030.828
100 request instances (large instances)			
100-50-50a	10800	2725.955	3171.128
100-50-50b	10800	2728.801	3205.662
100-50-50c	10800	2814.200	3173.351
100-25-75d	10800	2868.371	3121.081
100-25-75e	10800	2872.335	3090.196
100-25-75f	10800	2759.854	3234.307
100-75-25g	10800	2916.492	3255.168
100-75-25h	10800	2891.837	3170.435
100-75-25i	10800	2939.072	3245.277

---

**Table 4.13:** Computational result (RPD) of GASA and BBO.

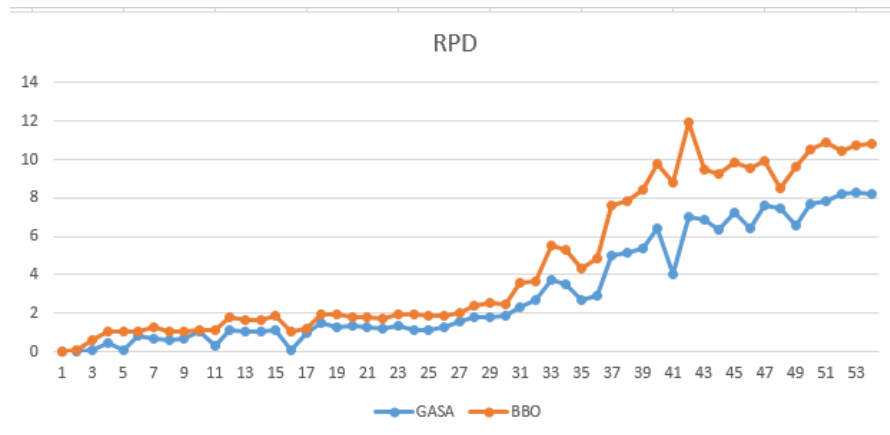
<b>Instance Number</b>	$RPD_{GASA}$	$RPD_{BBO}$
6-3-3	0	0
8-4-4	0.002	0.047
10-5-5a	0.115	0.631
10-5-5b	0.481	1.079
10-5-5c	0.111	1.054
10-3-7d	0.854	1.069
10-3-7e	0.679	1.238
10-3-7f	0.582	1.032
10-7-3g	0.704	1.081
10-7-3h	0.617	1.151
10-7-3i	0.329	1.141
20-10-10a	1.089	1.827
20-10-10b	1.015	1.619
20-10-10c	1.049	1.665
20-5-15d	1.128	1.866
20-5-15e	0.114	1.030
20-5-15f	0.938	1.169
20-15-5g	1.477	1.981
20-15-5h	1.283	1.961
20-15-5i	1.359	1.809
30-15-15c	1.284	1.802
30-10-20d	1.216	1.732
30-10-20e	1.346	1.910
30-10-20f	1.103	1.937
30-20-10g	1.117	1.885
30-20-10h	1.246	1.862
30-20-10i	1.599	2.033

---

40-20-20a	1.804	2.420
40-20-20b	1.822	2.553
40-20-20c	1.837	2.483
40-15-25d	2.351	3.583
40-15-25e	2.678	3.666
40-15-25f	3.701	5.561
40-25-15g	3.495	5.333
40-25-15h	2.660	4.316
40-25-15i	2.910	4.874
50-25-25a	4.998	7.648
50-25-25b	5.173	7.823
50-25-25c	5.371	8.442
50-20-30d	6.441	9.759
50-20-30e	4.016	8.833
50-20-30f	7.013	11.975
50-30-20g	6.849	9.495
50-30-20h	6.337	9.287
50-30-20i	7.217	9.864
100-50-50a	6.441	9.568
100-50-50b	7.591	9.933
100-50-50c	7.492	8.499
100-25-75d	6.539	9.653
100-25-75e	7.682	10.558
100-25-75f	7.810	10.889
100-75-25g	8.176	10.420
100-75-25h	8.309	10.781
100-75-25i	8.239	10.827

**Table 4.14:** The results of ANOVA test over the presented RPD for GASA and BBO.

<i>Source</i>	<i>DF</i>	<i>SS</i>	<i>MS</i>	<i>F-test</i>	<i>P-value</i>	<i>Results</i>
Algorithm	1	51.75	59.71	10.14	0.001	Null hypothesis is rejected
Error	40	235.63	5.89			
Total	77	311.15				



**Figure 4.5:** Output of comparing GASA and BBO with RPD indicator.

## 4.5 Conclusion

In this chapter, a new vehicle routing problem appeared in carrier collaboration via a combinatorial auction (CA) is studied. A carrier with reserved requests wants to determine within a time horizon of multi-periods (days) which pickup and delivery requests to serve among the requests open for bid of the auction to maximize its own profit subject to vehicle capacity, time windows constraints, and the maximum duration of each route. It is a multi-Periods Pickup and Delivery Problem with Time Windows, Profits, and both Reserved and Selective Requests. In the problem, each selective request is associated with a profit, which is the price for serving the request provided by a shipper and associated with a period window that specifies the earliest period and the latest period the request must be served. In each period, the carrier has a set of reserved requests and must be served by the carrier itself. Each request has a pickup node and a delivery node, and each node has a time window which specifies the earliest time and the latest time at which the node must be visited by a vehicle in each period. By including periodic concept in CA, the carrier can plan its transportation operations in advance and in a rolling-horizon way. The objective of the decision problem for the carrier is to determine which selective requests to serve in each period in addition of reserved requests and feasible routes to serve the reserved and selected requests to maximize its total profit. For this NP-hard problem, a mixed-Integer Linear Programming (MILP) model is formulated and two metaheuristic algorithms are proposed to solve the model: One is Genetic algorithm combined with Simulated Annealing (GASA) and the other is Biogeography-Based Optimization algorithm (BBO). The GASA and BBO are evaluated on 30 instances with 6-100 requests. The computational results show the two algorithms significantly outperform CPLEX solver, not only in solution quality but also in computation time, and the performance of GASA is better than BBO in terms of the two criterions.



## MULTI-PERIOD BID GENERATION PROBLEM WITH PICKUP AND DELIVERY, TIME WINDOWS, PROFITS, RESERVED REQUESTS, AND DELIVERY LEAD TIME

Management in the transport sector occurs along traditional lines rather than following a contemporary integrated approach [69]. Many manufacturers are seeking optimized logistics solutions to reduce transportation costs. Optimized logistics solutions that improve the transportation planning by one percent would save millions of euros. Taking the multi-billion transportation market into account, innovative ideas, and effective logistics solutions are demanded and valued [48]. There is however, evidence supporting a move towards more partnerships and horizontal cooperation [86]. In fact, during the last two decades collaboration within the supply chain management receives a great deal of attention [8].

For companies, besides minimizing all costs, shippers are interested in reducing the delivery lead time. Reducing delivery lead time has two benefits. At first, shorter delivery lead time makes supply chain management more agile, since orders are processed faster. Thus, it improves the ability of shippers to compete in the industry. Another benefit of shorter delivery lead time is choosing the carriers according to geographic factors [48]. In many procurements of transportation services, although cost is an important consideration in evaluating bids, shippers care about delivery lead time when evaluating bids submitted by carriers [26].

Although the delivery lead time is an important criterion in the procurement of a transportation service, its consideration in BGP has not been well addressed in the literature. Therefore, we focus on this topic in this chapter. Actually, in this chapter we consider a bid generation problem in a combinatorial auction for carrier collaboration, where each carrier determines the requests to bid for solving a BGP. After receiving the bids from all carriers, the auctioneer (an auction platform) determines the winning carriers and their winning bids. The auctioneer has

two ways to evaluate each bid from carriers. The first approach is a weighted score method, that is, we assign a weight to each attribute (cost and delivery lead time) of a bid and then transform two attributes into a single score. It means that evaluating a bid is according to a score function combining cost and delivery lead time, as in the multi-attributes score auction [26]. The other method is a bi-objective method which is treated with all attributes (objectives) separately without transforming them into a single score function. In this approach, a bid is evaluated based on its both attributes (cost and delivery lead time) as two different objective functions.

To solve the problem, we present mathematical programming models to both cases. For the single objective multi-period BGP with pickup and delivery requests, time windows, profits, and delivery lead time, we apply GASA and ITS. After that, we compare two algorithms with CPLEX. To solve bi-objective multi-period BGP, two well-known multi-objective algorithms, NSGA-II and NRG are presented. NSGA-II and NRG algorithms are compared based on standard metrics.

## 5.1 Literature review

This section is devoted to review articles related to bi-objective multi-period bid generation problem with pickup and delivery requests, time windows, profits, and reserved requests. In Chapter 3 and Chapter 4, the previous works related to collaborative transportation planning with pickup and delivery requests and multi-period bid generation problem have been analyzed. In this section, the works related to the transportation service with the consideration of delivery lead time are studied.

In [38] authors express the impact of collaborative transportation on transportation lead-time through empirically investigating on. In [48] a bi-objective integer programming model is proposed to formulate the problem. The authors find that many shippers take attention to have short transit time besides costs minimization. A bi-objective branch-and-bound algorithm that finds all non-dominated solutions is developed to solve the problem. Mathematical expressions for the transit time of cargo is presented in [29]. The authors propose two new trans-shipment policies: Backward trans-shipment and one-to-many trans-shipment, and mathematically calculate the corresponding connection times. Similar problems also are found in passengers transportation. In [15], a genetic algorithm is used to the optimization of transfer times in a bus transit system. The algorithm for this transfer time optimization problem tries to find the best feasible solution by shifting existing timetables.

As the problem presented in this chapter is close to multi-objective vehicle routing problems, a few works on multi-objective vehicle routing problems are cited. In [7] the authors consider two objective functions for a vehicle routing problem with time windows: The minimization of costs and the minimization of workload imbalance in terms of distances traveled by the vehicles. The suggested algorithm to solve the problem is a Pareto-based hybrid algorithm that integrates evolutionary computation and simulated annealing. In addition to the minimization of total

distance, time is also an important factor in the transportation system [36]. Consequently, total travel time minimization is considered as well. An improved evolutionary algorithm is suggested to solve a vehicle routing problem with time windows and two objectives. For these two objectives an improved evolutionary algorithm is proposed to solve the multi-objective vehicle routing problem with time windows.

**Table 5.1:** The parameters of the presented mathematical model.

Symbols	Explanation
$V = \{0, \dots, 2n + 1\}$	Set of all nodes which comprises all pickup, delivery nodes and depot node of the carrier. $n$ is the number of requests, 0 and $2n + 1$ denote the depot of the carrier.
$P = \{1, 2, \dots, n\}$	Set of pickup nodes of all requests.
$D = \{n + 1, \dots, 2n\}$	Set of delivery nodes of all requests.
$W$	Set of nodes excluding the depot node of the carrier.
$H$	Set of period planning horizon.
$K = \{1, 2, \dots, VK\}$	Finite set of homogeneous vehicles where $VK$ is the maximum number of vehicles.
$Q$	Capacity of each vehicle that can be used.
$R_l^r$	Reserved requests that must be served in period $l$ , $l \in H$ .
$R_s = (\cup_{l \in H} R_l^s)$	Set of selective requests
$R = (\cup_{l \in H} R_l^r)$	Set of all requests.
$R_l^s$	Set of selective requests $s$ in period $l$ .
$[e_i, l_i]$	Time windows of pickup node $i$ .
$[e_{i+n}, l_{i+n}]$	Time windows of delivery node $i + n$ .
$[E_i, L_i]$	Period window of selective request $i$ .
$t_{ij}$	Traveling time between two nodes $i$ and $j$ .
$c_{ij}$	Traveling cost between two nodes $i$ and $j$ .
$T$	Maximum duration of each route.
$BM_{ij} = l_j - e_i$	Parameter used to formulate linearly the time window constraints, where $l_j$ is the upper bound of time window for node $j$ and $e_i$ is the lower bound of time window for node $i$ .
$CV_i = Q + d_i$	Parameter used to formulate linearly the vehicle capacity constraints.
$d_i$	Demand of the pickup node of request $i$ .
$d_{i+n} = -d_i$	Demand of delivery node $i + n$ if the corresponding pickup point's demand is $d_i$ .
$\alpha_i$	Weight of delivery lead time for request $i$ .

## 5.2 Problem description and mathematical model

Our considered problem in this chapter is a transportation-planning problem with pickup and delivery requests, hard time windows defined on a directed graph  $G = (V, E)$ , where  $V$  is the set of all nodes and  $E$  is the set of edges with the following specifications. Each request  $i$  is associated

**Table 5.2:** The decision variables of the considered model.

Symbols	Explanation
$x_{ijkh} = 1$	If and only if vehicle $k$ visits directly node $j$ after node $i$ in period $h$ .
$y_{ikh} = 1$	If and only if request $i$ is served by vehicle $k$ in period $h$ .
$U_{ikh}$	Arriving time of vehicle $k$ at node $i$ in period $h$ .
$CV_{ikh}$	Load of vehicle $k$ when it leaves node $i$ in period $h$ .
$y'_{ikh}$	An auxiliary variable used in the linearization of model.

with two nodes, a pickup node  $i$  and a delivery node  $n + i$ , where  $n$  represents the number of requests and the demand of a pickup node is equal to the demand of its corresponding delivery node. The delivery node of each request must be visited after its pickup node on the same route. The number of customers, their locations, the capacity of each vehicle, and the fleet of vehicles are predefined. There are other assumptions to this problem like the quantity of pickup/ delivery is smaller than vehicle capacity and all vehicles are homogeneous. The cost of satisfying customers' demands depends on the vehicles' total travel distances. Each selective request has a service period window and two time windows for pickup and delivery respectively. The service period window defines the periods when each selective request can be served and the two time windows determine at which times in each period the pickup node and the delivery node of the request can be visited by a vehicle that serves the request, for instance morning and afternoon windows separated by a lunch break. Both selective and reserved requests are associated with two time windows, whereas only selective requests are associated with a service period window (the period in which each reserved request must be served is prespecified). If a selective request is served, the most appropriate period to serve it should be determined. In addition, each request  $i$  is associated with a weight  $\alpha_i$ , which can be interpreted as the penalty cost for one time unit longer of delivery lead time.

The multi-period bid generation problem can be formulated as a mixed-integer linear programming model. The aim of the problem is to serve all reserved requests in their predefined period and to determine which selective requests should be served to improve the total profit. The parameters of the problem are listed in Table 5.1, while the decision variables employed are in Table 5.2. Hence, the single objective mathematical model of the multi-period BGP is given as follows:

$$\max(\sum_{i \in R} \sum_{h \in H} \sum_{k \in K} p_i y_{ikh} - \sum_{h \in H} \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{ij} x_{ijkh} - \sum_{j \in R_s} \alpha_j (\sum_{h \in [E_i, L_i]} \sum_{k \in K} y'_{jkh}))$$

subject to:

$$\sum_{j \in V; j \neq i} x_{jikh} - \sum_{j \in V; j \neq i} x_{ijk} = 0 \quad \forall i \in W, \forall k \in K, \forall h \in H \quad (5.1)$$

$$\sum_{k \in K} \sum_{j \in P; j \neq 0} \sum_{h \in H} x_{0jkh} \geq 1 \quad (5.2)$$

$$\sum_{k \in K} \sum_{i \in D; i \neq (2n+1)} \sum_{h \in H} x_{i(2n+1)kh} \geq 1 \quad (5.3)$$

$$\sum_{k \in K} y_{ikh} = 1 \quad \forall h \in H, \forall i \in R_l^r \quad (5.4)$$

$$\sum_{h \in [E_i, L_i]} \sum_{k \in K} y_{ikh} \leq 1 \quad \forall i \in R_s \quad (5.5)$$

$$\sum_{j \in V, j \neq i, 0} x_{j(n+i)kh} = y_{ikh} \quad \forall i \in P, k \in K, \forall h \in H \quad (5.6)$$

$$\sum_{j \in V, j \neq i, 2n+1} x_{ijkh} = y_{ikh} \quad \forall i \in P, k \in K, \forall h \in H \quad (5.7)$$

$$U_{ikh} + t_{i,(n+i)} + s_i \leq U_{(n+i),kh} \quad \forall i \in P, k \in K, \forall h \in H \quad (5.8)$$

$$U_{jkh} \geq U_{ikh} + t_{ij} + s_i - BM_{ij}(1 - x_{ijkh}) \quad \forall i \in V, k \in K, \forall h \in H, \forall j \in V \quad (5.9)$$

$$e_i y_{ikh} \leq U_{ikh} \leq l_i + (1 - y_{ikh}) \sum_{j \in V} BM_{ij} \quad \forall h \in H, \forall i \in V, \forall k \in K \quad (5.10)$$

$$U_{ikh} + t_{ij} + s_i - BM_{ij}(1 - x_{ijkh}) \leq T \quad \forall i \in V, k \in K, \forall h \in H, \forall j \in \{0\} \quad (5.11)$$

$$CV_{ikh} + d_i - Q(1 - x_{ijkh}) \leq CV_{jkh} \quad \forall i \in V, k \in K, \forall h \in H, \forall j \in V \quad (5.12)$$

$$\max\{0, d_i\} \leq CV_{ikm} \leq \min\{Q, d_i\} \quad \forall i \in V, k \in K, \forall h \in H \quad (5.13)$$

$$x_{ijkh} \in \{0, 1\} \quad \forall i, j \in V, \forall k \in K, \forall h \in H \quad (5.14)$$

$$U_{ikh} \geq 0 \quad \forall i \in V, \forall k \in K, \forall h \in H \quad (5.15)$$

$$CV_{ikh} \geq 0 \quad \forall i \in V, \forall k \in K, \forall h \in H \quad (5.16)$$

$$y_{ikh} = 0 \quad \text{for any } i \in R_s, \forall k \in K, \text{ for any } h \notin (E_i, L_i) \quad \text{and} \quad y_{ikh} = 0 \quad \text{for any } i \in R_{rl}, l \neq h \quad (5.17)$$

$$y'_{jkh} \geq h \times y_{jkh} - E_j \quad \forall j \in V, \forall k \in K, \forall h \in H \quad (5.18)$$

$$y'_{jkh} \geq 0 \quad \forall j \in V, \forall k \in K, \forall h \in H \quad (5.19)$$

The objective function corresponds to the score of a bid of the carrier, which is equal to the difference between the total payments of serving requests in all periods, the total transportation cost and the penalty costs of delivery lead-time. The considered score, is formulated as the single objective function, in this model. Constraint (5.1) ensures that when a vehicle arrives at a node in a period, it must leave in the same period. Constraints (5.2) and (5.3) signify that each vehicle leaves its depot of carrier in a period must return to it in the same period. Equation (5.4) implies that each reserved request must be served in its predetermined period. Equation (5.5) indicates that each selective request can be either served in a period within its service period window or not served. Constraints (5.6) and (5.7) guarantee if a request is served in a period, its delivery node must be visited after its pickup node with the same vehicle in the same period. Equations (5.8)-(5.11) specify time windows constraints on the pickup and delivery nodes of each request and the constraint on the maximum duration of each route. Constraints (5.12) and (5.13) check vehicle capacity. Equations (5.14)-(5.17) define all variables. To linearize the last term of the objective function, two constraints (5.18) and (5.19) are added.

Except for costs, shippers are also concerned with transit time (delivery lead time); shorter transit time represents better transportation service. It is possible to formulate the multi-period BGP as a bi-objective problem, on the one hand, the profit is maximized and on the other hand the delivery lead time is minimized. It means that one of the objective functions is the maximization of the total profit and the other one is the minimization of weighted delivery lead time for all requests. Consequently, in the previous single objective model, the weighted delivery lead time of all requests will be considered as a second objective function.

First objective function:

$$\max(\sum_{i \in R} \sum_{h \in H} \sum_{k \in K} P_i y_{ikh} - \sum_{h \in H} \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} C_{ij} x_{ijkm})$$

Second objective function:

$$\min \sum_{j \in R_s} \alpha_j (\sum_{h \in [E_i, L_i]} \sum_{k \in K} y'_{jkh})$$

### **5.3 Metaheuristic approaches to solve a single objective multi-period bid generation problem with consideration of delivery lead time**

In the literature, two approaches are usually used to solve complicated multi-objective optimization problems. In the first approach, a multi-objective problem is transformed into a single-objective optimization using some multi-criteria decision making (MCDM) techniques presented in [49]. In the second approach, multi-objective algorithms such as non-dominated sorting genetic algorithm (NSGA-II) and non-dominated ranking genetic algorithm (NRGA) are directly employed to find a set of optimal solutions called Pareto optimal solutions in a single run [3].

The multi-period bid generation problem with pickup and delivery requests, time windows, profits, reserved requests, and delivery lead time is a NP-hard vehicle routing problem [65]. Consequently, it cannot be solved optimally for large instances, and metaheuristic algorithms are required to solve these types of problems. In this chapter, we solve the problem in both single objective perspective and bi-objective perspective. Firstly, the single-objective problem is solved by using GASA and ITS which are described in Chapter 3. Secondly, the multi-objective problem is solved using NSGA-II and NRGA.

For the single objective multi-period bid problem with the consideration of delivery lead time, GASA and ITS used to solve it are similar to those presented in Chapter 4, except that the delivery lead time of each request is considered in the objective function of the problem. For this reason, we will not present the two algorithms again in this chapter. Besides, NSGA-II and NRGA that are used to solve the bi-objective problem are derived from Multi-Objective Genetic Algorithms, so their chromosome is the same as chromosome of GASA which is described in Chapter 4.

### **5.4 Metaheuristic approaches to solve a bi-objective multi-period bid generation problem with consideration of delivery lead time**

Multi-objective evolutionary algorithms (MOEAs) are usually fast to find Pareto fronts in a single run. So, in this section we utilize them to solve our bi-objective bid generation problem. Among MOEAs, NSGA-II due to its efficiency, its ability to solve similar problems, and its ease of use is chosen. Moreover, as there is no benchmark existing in the literature to verify the results obtained by NSGA-II, another GA-based MOEA called NRGA is used as well [78]. In this section we first describe bi-objective decision problems.

A single objective optimization algorithms aim at obtaining an optimal or near optimal solution. For bi-objective optimization problems, decision makers are not satisfied with only one

solution returned, because they want to make different compromises among multiple objectives by finding a set of non-dominated solutions. In the following, we first provide some basic definitions about a multi-objective minimization problem with  $n$  decision variables and  $m$  objectives ( $m > 1$ ) [94]:

$$\text{Minimize } y = f(x) = (f_1(x), f_2(x), \dots, f_m(x))$$

where  $x \in R^n$  and  $x \in R^m$ .

**Definition 1.** A solution  $b$  is dominated by another solution  $a$  if and only if:

$$(1) f_i(a) \leq f_i(b) \quad \forall i \in \{1, 2, \dots, m\}$$

$$(2) f_i(a) < f_i(b) \quad \exists i \in \{1, 2, \dots, m\}$$

Non-dominated solutions dominate the others but do not dominate themselves. The local optimality explained as follows [94]:

**Definition 2.** A solution  $a$  is a local optimum solution in the Pareto sense if there exists a real  $\varepsilon > 0$  such that there is no solution  $b$  dominating  $a$  with  $b \in R^n \cap B(a, \varepsilon)$ , where  $B(a, \varepsilon)$  is a  $n$ -dimensional sphere with center  $a$  and radius  $\varepsilon$ .

Different techniques are developed to solve multi-objective optimization problems. These algorithms can be classified into meta-heuristic, decision-aided, interactive, fuzzy and scalar ones [18]. For a detailed description of the multi-objective optimization, readers may refer to [20]. In the following, two popular metaheuristic algorithms, NSGA-II (Non-dominated Sorting Genetic Algorithm-II based on sorting the population by non-dominance) and NPGA (Non-dominated Ranked Genetic Algorithm) to solve multi-objective optimization problems are utilized. Before presenting the two algorithms, some important concepts of multi-objective optimization are introduced.

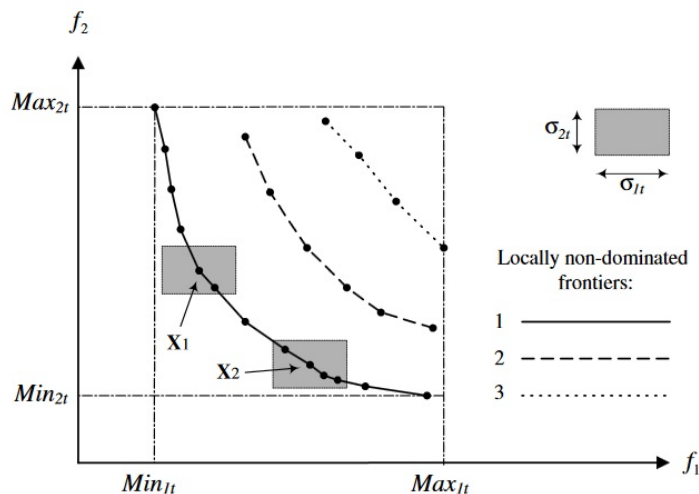
### 5.4.1 Non-dominated sorting and crowding distance

To sort a population with size  $N$  according to non-domination, each solution must be compared with every other solution in the population to check if it is dominated. At this step, all individuals in the first non-dominated level are obtained. In order to determine the individuals of the next level, the solutions of the first level are deduced and the procedure to sort and compose the non-dominated level is executed again. This method is repeated to find a successive front [70].

#### *Diversity mechanism*

In addition to the convergence to the Pareto-optimal set, a good multi-objective optimization algorithm should be able to obtain solutions that maintain a good diversity. In [67], the concept of cuboid is used to measure diversity in the population. A cuboid is a rectangular centered on an individual (a solution) in the front. Figure 5.1 illustrates such a cuboid for a bi-objective problem,





**Figure 5.1:** Cuboid along locally non-dominated frontier [70].

where two cuboids are depicted for two randomly chosen individuals  $X_1$  and  $X_2$ . Inasmuch as each cuboid has the same size as others, the density of solutions in each cuboid can be easily measured by the number of individuals included in the cuboid. According to the cuboid technique, the cuboid of  $X_1$  and  $X_2$  has the same size. Since it is desired to have a spread of solutions in the front, the individual  $X_1$  should be selected to survive in the next generation [70].

To get an estimation of the density of solutions surrounding a particular solution,  $i$ , in the population, we calculate the average distance of two points on either side of this point,  $i$ , along each of the objectives. This quantity  $i_{distance}$  serves as an estimate of the perimeter of the cuboid formed by using the nearest neighbors as the vertices (call this ‘crowding distance’) [28]. Furthermore, the total crowding distance value is calculated as the sum of individual distances value corresponding to each objective. In Fig. 5.2, the computation procedure of crowding distance of all solutions in a non-dominated set  $L$  is depicted [28]. In that figure,  $L[i]_m$  addresses the value of  $m^{th}$  objective function of the  $i^{th}$  individual in the set  $L$  [70].

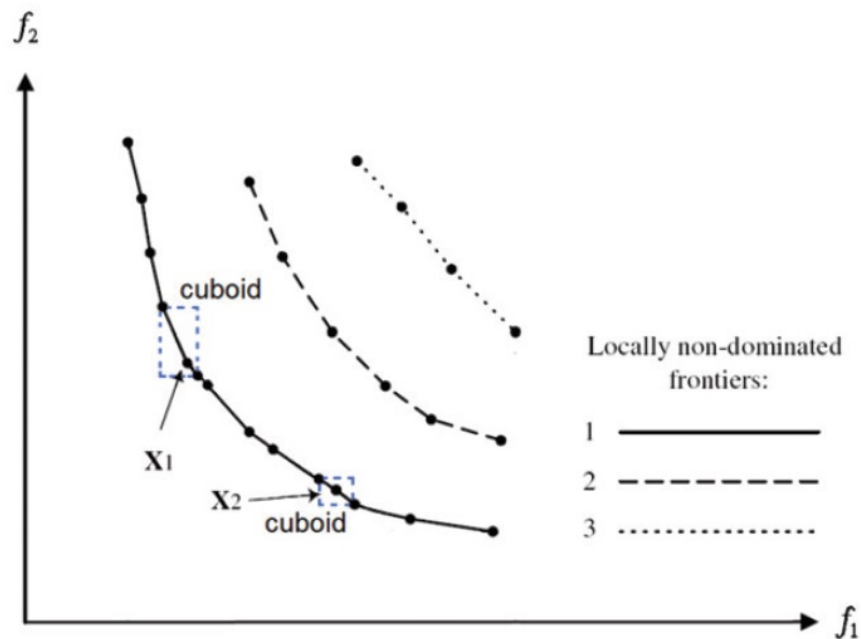
According to Fig. 5.3, the crowding distance for individual  $X_1$  is larger than  $X_2$ , hence, individual  $X_1$  has more probability to be chosen as a parent.

### 5.4.2 NSGA-II (Non-dominated Sorting Genetic Algorithm-II) implementation

The Non-dominated Sorting Genetic Algorithm is a Multi Objective Optimization (MOO) algorithm and is an instance of an Evolutionary Algorithm from the field of Evolutionary Computation. NSGA-II is an extension of the Genetic Algorithm for multi objective function optimization. Actually, one of the well-known multi-objective optimization algorithms is NSGA-II, which has three special features. The significant features of NSGA-II are fast non-dominated sorting, fast crowded

Crowding-distance-assignment ( $L$ )	
$l =  L $	number of solutions in $L$
for each $i$ , set $L[i]_{\text{distance}} = 0$	
for each objective $m$	
$L = \text{sort}(L, m)$	sort using each objective value
$L[1]_{\text{distance}} = L[l]_{\text{distance}} = \infty$	boundary points are always selected for all other points
for $i = 2$ to $(l = 1)$	
$L[i]_{\text{distance}} = L[i]_{\text{distance}} + (L[i+1]_{.m} - L[i-1]_{.m})$	

**Figure 5.2:** The crowding distance computation procedure of all solutions in a non-dominated set [70].



**Figure 5.3:** Crowding distance for individuals  $X1$  and  $X2$  [70].

distance evaluation and simple crowding comparison [105]. Different test problems from previous studies applying NSGA-II are compared in [28], which claims that this approach outperforms some algorithms like PAES and SPEA in obtaining a more diverse set of solutions [54]. Overall, NSGA-II can be summarized by the following steps [105]:

**Step 1:** Initialize the population

Initialize the population using a solution generation procedure. For the bi-objective problem studied, we apply the initial solution procedure in Chapter 4 to initialize the population. The considered problem in this chapter is multi-period bid generation problem with delivery lead time. The chromosome structure is the same as chromosome structure in Chapter 4, which is composed of three vectors,  $X$ ,  $Y$ , and  $Z$  indicating pickup and delivery nodes, period allocation for each request and assigning a vehicle to each request.

**Step 2:** Non dominated sort

Sort the population based on non-domination.

**Step 3:** Calculation of crowding distances

After sorting the population, the crowding distance value is calculated as the sum of individual distance values corresponding to each objective.

**Step 4:** Selection

The individuals are chosen based on the rank of their front in the sorting and on the crowding distance. The selection of individuals is performed by tournament selection approach with crowding comparison operator.

**Step 5:** Genetic Operations

Offspring generated by the operators previously described for the GASA algorithm. The crossover operator over vector  $X$  is single point crossover and the crossover operator over vector  $Y$  and  $Z$  are uniform crossover which both of them are described in Chapter 4. The mutation operators over the three vectors are also introduced in Chapter 4.

**Step 6:** Combination and selection

The population of offspring and current generation are combined and the individuals of the next generation are determined by selection. The new generation is composed by each front subsequently until the population size exceeds the current population size [105].

**Step 7:** Stop criterion

The algorithm stops when a prespecified maximum number of iterations is reached.

### 5.4.3 NPGA (Non-dominated Ranked Genetic Algorithm) implementation

NPGA is also a multi-objective genetic algorithm to find Pareto optimal solutions. NPGA is similar to NSGA-II with a difference wherein the selection operation a roulette wheel strategy is employed [72]. In NPGA, a fitness value representing rank is assigned to each individual of the population. In this regard, two ranked based roulette wheel selection strategies including: (I) select the fronts and (II) choose solutions from the fronts, are used [72], [50], [83], [2].

$$P_f = \frac{2 \times rank_f}{(NF) \times ((NF) + 1)} \quad f = 1, \dots, NF \quad (5.20)$$

$$P_{fs} = \frac{2 \times rank_{fs}}{(NS)_f \times ((NS)_f + 1)} \quad f = 1, \dots, NF, s = 1, \dots, NS \quad (5.21)$$

In the equations (5.20) and (5.21),  $NF$  and  $(NS)_f$  are the number of fronts and the number of solutions in front  $f$ , respectively. Based on equation (5.20), a front with highest rank has the highest probability to be selected. According to equation (5.21), solutions with more crowding distance have higher selection probability. The roulette wheel selection is applied until a desired number of solutions is reached. The algorithm stops when a predetermined number of iterations

is obtained. Hence, our preference is the solution located in the higher front rank. If both solutions are embedded in the same front, the solution with a higher crowding rank distance is preferred [70].

## 5.5 Computational study

To assess the performance of GASA, ITS, NSGA-II, and NPGA algorithms, we use the instances in [65], which their explanations are in Chapter 4. We executed all algorithms on the instances. The characteristics of the instances include the number of requests, the demand of each request, time windows of pickup and delivery nodes, the number of vehicles, and the capacity of each vehicle have been explained in Chapter 4. To include the period concept in the instances, we assume the number of periods is 5 and each day has predetermined reserved requests. The results unveil that both GASA and ITS algorithm for the single-objective bid generation problem studied remarkably outperform CPLEX, while the GASA algorithm outperforms the ITS algorithm. To compare two bi-objective algorithms, NSGA-II and NPGA, we use different metrics as defined in the next subsection.

### 5.5.1 Comparison Metrics for Multi-Objective Optimization Algorithms

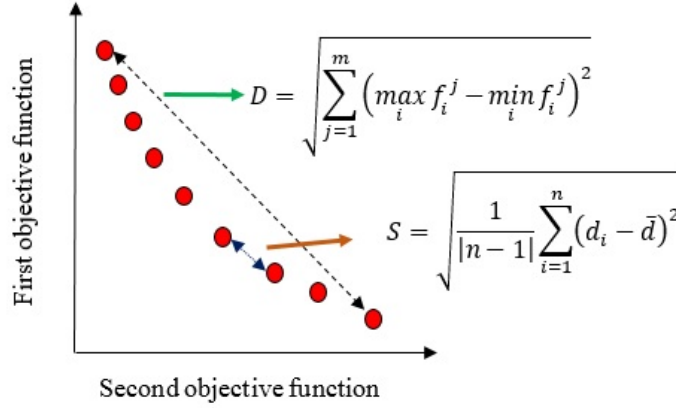
In order to analyze the performance of each algorithm, two criteria must be evaluated: Efficiency and effectiveness. In the evaluation of a single objective algorithm, the value of the objective function corresponds to the effectiveness and the computation time represents the efficiency of the algorithm. We solved the mathematical programming model of our problem by CPLEX 12.6 called in C++ with Visual studio 12 on a laptop with an Intel core i5 CPU and 4.00 GB RAM. Note that for the instances with 30 requests and the instances more than 30 requests, it is impossible to solve the model optimally by CPLEX after 2 hours and 3 hours of running, respectively. However, to evaluate the efficiency and effectiveness of a multi-objective algorithm, there are different metrics explained below. These metrics can be used to get a comprehensive picture of the algorithm's ability [43]. Overall, for a multi-objective optimization algorithm, the diversity of Pareto solutions found and also its convergence are two important factors [27]. Computational time can still be applied as a suitable metric to evaluate the efficiency of an algorithm.

*Diversification metric (D or diversity).*

The diversification metric measures the spread of the solutions found and is defined in [107]. In Fig. 5.4, diversity metric is illustrated.

*Spacing metric (S)*

The standard deviation of the solutions distances in the Pareto front is defined as the spacing metric. In Fig. 5.4, the spacing metric is illustrated where  $d_i = \min_{k \in n, k \neq i} \{ \sum_{m=1}^M |f_m^i - f_m^k| \}$ ,  $f_m^{(i)}$



**Figure 5.4:** The spacing and diversity metrics in bi-objective problems [70].

is  $i^{th}$  objective function value of the  $m^{th}$  solution in the Pareto front, and  $\bar{d}$  is the mean value of the above distance measure  $\bar{d} = \sum_{i=1}^{|n|} \frac{d_i}{|n|}$  [42].

*The number of Pareto solutions (NOS)*

This metric enumerates the number of Pareto solutions in the optimal front.

*Mean Ideal distance (MID).*

This metric measures the proximity of solutions from the ideal point that is (0, 0) on the Pareto front appraisal [83]. In the following equation (5.22),  $f_{ji}$  is the value of objective function in the  $i^{th}$  solution of optimal front.

$$MID = \frac{1}{NOS} \sum_{i=1}^{NOS} C_i \quad \text{where} \quad C_i = \sqrt{\sum_{j=1}^2 f_{ji}^2} \quad (5.22)$$

*Time.*

Time is an important criterion in comparing two algorithms.

Whereas in terms of diversity and NOS metrics, larger values are desirable, for the spacing metric, MID, and CPU time, smaller values are desired [42].

### 5.5.2 Algorithm parameter tuning

To apply the taguchi approach to parameter calibration of a bi-objective optimization algorithm, criteria for comparing two solutions must be chosen [79]. Two important criteria are 1) convergence and 2) diversity of Pareto solutions. As we mentioned previously, there are different metrics

for comparing multi-objective optimization algorithms. Among them, computational time and mean ideal distance (MID) are two metrics for evaluating the convergence of the algorithms while other metrics are applied for evaluating the diversity of the Pareto solutions. In the suggested metrics, diversity and MID are integrated through the following equation (5.23). The two criteria are considered at the same time. Accordingly, by applying this integrated metric as the response of the Taguchi approach, a combination of main signals can be proposed. So, we can gain precise outputs. This metric is called the multi-objective coefficient of variation (MOCV) [42], [79]:

$$MOCV = \frac{MID}{Diversity} \quad (5.23)$$

When applying Taguchi method, the level of each factor taken is reported in Table 5.3. For GASA and ITS algorithms, we use the same parameters which have been determined in Chapter 3. By numerical experiments using Taguchi method, we find that the predetermined values of all parameters in Chapter 3 are appropriate. For each algorithm, three levels (i.e. low, medium and high) are defined for each factor. By using Minitab Software, L9 design is exploited for NSGA-II and NRGGA. The orthogonal arrays of these designs and the gained responses are illustrated in Table 5.4 and Table 5.5 for NSGA-II and NRGGA respectively. For each algorithm, the effect plots for S/N ratio are presented in Fig. 5.5 and Fig. 5.6. By using these results for each algorithm, the appropriate values of all parameters are set and reported in Table 5.6.

**Table 5.3:** NSGA-II and NRGGA parameter ranges and the levels of factors.

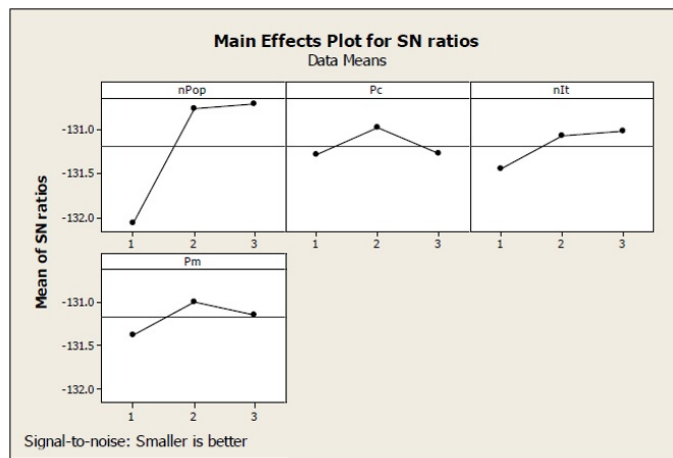
<b>Algorithm</b>	<b>Parameter</b>	<b>Description</b>	<b>Range</b>	<b>Low(1)</b>	<b>Medium(2)</b>	<b>High(3)</b>
	Npop	Population size	50-200	50	125	200
NRGA and	nIt	Number of iterations	200-500	200	350	500
NSGA-II	$P_c$	Crossover probability	0.3-0.7	0.3	0.5	0.7
	$P_m$	Mutation probability	0.1-0.5	0.1	0.3	0.5

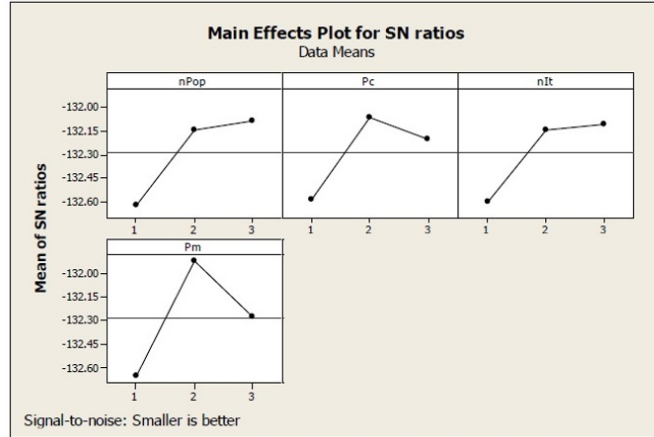
**Table 5.4:** Taguchi procedure for NSGA-II.

Run Order	Npop	$P_c$	$P_m$	nIt	Response
1	1	1	1	1	9.5048
2	1	2	2	2	7.909
3	1	3	3	3	10.206
4	2	1	2	3	3.607
5	2	2	3	1	3.586
6	2	3	1	2	4.486
7	3	1	3	2	8.280
8	3	2	1	3	5.399
9	3	3	2	1	13.877

**Table 5.5:** Taguchi procedure for NPGA.

Run Order	Npop	$P_c$	$P_m$	nIt	Response
1	1	1	1	1	4.820
2	1	2	2	2	1.470
3	1	3	3	3	5.559
4	2	1	2	3	8.603
5	2	2	3	1	7.367
6	2	3	1	2	5.767
7	3	1	3	2	2.193
8	3	2	1	3	13.993
9	3	3	2	1	4.565

**Figure 5.5:** Outputs of Taguchi ratio for NSGA-II.



**Figure 5.6:** Outputs of Taguchi ratio for NRGa.

**Table 5.6:** Taguchi response for NSGA-II and NRGa.

Algorithm	Npop	$P_c$	$P_m$	nIt
NSGA-II	3 (200)	2 (0.5)	2 (0.3)	3 (500)
NRGA	3 (200)	2 (0.5)	2 (0.3)	3 (500)

### 5.5.3 Results and Analysis

RPD is one of the well-developed approaches for measuring the efficiency of an algorithm with respect to another algorithm. In Chapter 3, we described completely RPD indicator in equation (3.25). In this subsection, we apply it to compare algorithms. Table 5.7 illustrates the results of GASA and ITS over the instances of Chapter 4 while Table 5.8 shows their results in terms of RPD. Their statistical analysis is given in Table 5.9. The results of NRGa and NSGA-II in each metric are indicated in Table 5.10 and Table 5.11. We calculate the RPD of each metric after three execution of each algorithm. Two gaps,  $Gap_{ITS}$  and  $Gap_{GASA}$ , are defined in Table 4.9. All gaps defined in Table 5.7 are in percentage (%).



**Table 5.7:** Computational results of GASA and ITS.

<b>Instance Number</b>	$Obj_{GASA}$	$Obj_{ITS}$	$LB_{MILP}$	$UB_{MILP}$	$Gap_{MILP}$	$Gap_{GASA}$	$Gap_{ITS}$
20-10-10a	1356.511	1344.763	-	1429.288	-	5.091	5.913
20-10-10b	1087.414	1070.291	967.511	1140.983	15.203	4.694	6.195
20-10-10c	2592.257	2564.118	1964.131	2705.007	27.389	4.168	5.208
20-5-15d	968.265	971.701	-	1017.989	-	4.884	4.547
20-5-15e	996.798	985.357	-	1038.841	-	4.047	5.148
20-5-15f	2572.020	2537.411	2214.212	2666.525	16.962	3.544	4.842
20-15-5g	2271.493	2249.102	-	2349.265	-	3.310	4.263
20-15-5h	1423.878	1420.825	-	1511.192	-	5.777	5.979
20-15-5i	3778.519	3732.614	-	3967.039	-	4.752	5.909
30-15-15c	5312.654	5197.674	-	5610.102	-	5.302	7.351
30-15-15d	4985.130	4889.179	-	5311.148	-	6.138	7.944
30-15-15e	6217.237	6055.472	-	6569.163	20.431	5.357	7.819
30-15-15f	6373.191	6262.774	-	6728.334	-	5.278	6.919
30-20-10g	8628.015	8644.162	-	9164.702	-	5.856	5.679
30-20-10h	10415.620	10180.660	-	11040.900	-	5.663	7.791
30-20-10i	9529.182	9337.791	-	10038.770	-	5.076	6.982
40-20-20a	9632.193	9477.823	-	10520.107	-	8.440	9.908
40-20-20b	10774.450	10851.256	-	11868.040	-	9.214	8.567
40-20-20c	10616.947	10617.322	-	11697.518	-	9.237	9.234
40-15-25d	8541.138	8386.904	-	9301.346	-	8.173	9.831
40-15-25e	8046.854	7900.667	-	8821.121	-	8.777	10.434
40-15-25f	10176.106	10091.119	-	11266.065	-	9.675	10.429
40-25-15g	8590.802	8429.175	-	9386.584	-	8.477	10.198
40-25-15h	11471.169	11209.378	-	12497.361	-	8.211	10.306
40-25-15i	9781.714	9496.332	-	10755.698	-	9.055	11.708
50-25-25a	18047.940	17663.351	-	20192.817	-	10.626	12.526
50-25-25b	15050.833	14672.267	-	16743.635	-	10.110	12.371
50-25-25c	12601.148	12512.386	-	14175.432	-	11.105	11.732
50-20-30d	15131.713	14887.265	-	16932.874	-	10.637	12.081
50-20-30e	16886.925	16838.288	-	19169.190	-	11.906	12.160
50-20-30f	14833.207	14600.436	-	16867.276	-	12.059	13.440
50-30-20g	11594.018	11517.593	-	13008.477	-	10.873	11.461
50-30-20h	16011.674	15594.412	-	17814.653	-	10.121	12.463
50-30-20i	11920.136	11876.153	-	13497.751	-	11.687	12.013

CHAPTER 5. MULTI-PERIOD BID GENERATION PROBLEM WITH PICKUP AND DELIVERY,  
TIME WINDOWS, PROFITS, RESERVED REQUESTS, AND DELIVERY LEAD TIME

---

100-50-50a	48721.136	47276.470	-	56925.861	-	14.413	16.950
100-50-50b	56073.317	53670.838	-	65258.020	-	14.074	17.755
100-50-50c	65932.449	65221.674	-	78329.939	-	15.827	16.734
100-25-75d	72511.185	70001.090	-	85135.573	-	14.828	17.776
100-25-75e	42075.695	40863.368	-	50296.812	-	16.345	18.754
100-25-75f	54426.855	54003.893	-	64712.100	-	15.893	16.547
100-25-75f	51488.401	50353.712	-	60876.267	-	15.421	17.285
100-25-75g	51623.397	50577.439	-	61025.346	-	15.406	17.120
100-25-75h	71438.922	69927.386	-	85642.313	-	16.585	18.349

**Table 5.8:** Computational results (RPD) and computation time of GASA and ITS

<b>Instance Number</b>	$RPD_{GASA}$	$RPD_{ITS}$	$CPU_{GASA}$	$CPU_{ITS}$
20-10-10a	1.603	1.977	431.555	504.792
20-10-10b	0.528	1.256	429.222	503.326
20-10-10c	1.800	2.572	430.770	504.344
20-5-15d	1.523	2.837	428.317	504.120
20-5-15e	1.837	1.243	452.568	546.661
20-5-15f	0.185	1.569	459.527	544.775
20-15-5g	0.402	1.843	468.733	588.079
20-15-5h	1.594	3.265	464.176	585.230
20-15-5i	1.275	2.016	473.551	589.345
30-15-15c	1.082	2.896	503.737	611.280
30-10-15d	1.123	2.124	505.5621	596.051
30-10-15e	2.882	2.853	521.853	614.605
30-10-20f	1.786	2.980	536.869	628.074
30-20-10g	1.929	2.909	534.912	627.562
30-20-10g	1.720	3.020	538.414	641.857
30-20-10g	1.647	2.615	548.990	655.382
40-20-20a	2.833	4.190	840.096	969.553
40-20-20b	1.772	3.604	851.147	975.242
40-20-20c	2.088	4.657	837.539	973.140
40-15-25d	2.521	4.242	842.808	974.931
40-15-25e	2.438	3.895	856.658	982.990
40-15-25f	2.565	3.367	869.731	980.407
40-25-15g	2.652	4.615	872.548	983.625
40-25-15h	1.000	3.481	874.772	985.922
40-25-15i	2.442	4.927	882.646	986.164
50-25-25a	2.375	3.173	1718.932	1983.595
50-25-25b	2.563	5.710	1723.169	1984.144
50-25-25c	2.104	4.906	1808.709	2007.163
50-20-30d	3.861	5.986	1775.111	1940.328

50-25-25a	2.375	3.173	1791.665	1960.613
50-25-25b	2.563	5.710	1837.876	1995.127
50-25-25c	2.104	4.906	1845.172	2038.825
50-20-30d	3.861	5.986	1868.917	2033.805
50-20-30e	3.827	5.769	1885.421	2280.310
50-20-30f	2.636	6.526	1885.754	2253.416
50-30-20g	3.070	5.962	1876.965	2213.809
50-30-20h	2.920	5.811	1829.517	2259.317
50-30-20i	2.873	4.586	1883.001	2235.991
100-50-50a	5.826	7.740	2819.110	3340.961
100-50-50b	4.042	6.125	2859.675	3361.365
100-50-50c	6.965	8.295	2788.557	3293.458
100-25-75d	5.207	9.355	2852.249	3347.070
100-25-75e	7.888	8.020	2847.522	3349.463
100-25-75f	5.773	7.378	2855.983	3350.209
100-75-25g	7.358	8.882	2888.263	3215.119
100-75-25h	7.791	9.247	2859.822	3341.180
100-75-25i	6.544	9.270	2782.046	3294.168

**Table 5.9:** The results of ANOVA test over the presented single objective mathematical model.

<i>Source</i>	<i>DF</i>	<i>SS</i>	<i>MS</i>	<i>F-test</i>	<i>P-value</i>	<i>Results</i>
Algorithm	1	51.75	51.75	15.16	0.000	Null hypothesis is rejected
Error	76	259.40	3.41			
Total	77	311.15				

**Table 5.10:** Multi-objective metrics computed for NRGGA

<b>Instance Number</b>	<i>Diversity</i>	<i>NOS</i>	<i>MID</i>	<i>Spacing</i>	<i>Time</i>
6-3-3	14.787	2.765	0.039	4.022	14.677
8-4-4	24.915	30.377	0.030	13.414	1.312
10-5-5a	49.138	13.529	2.650	6.321	6.182
10-5-5b	47.287	13.905	0.001	10.975	3.664
10-5-5c	50.960	42	12.517	40.538	1.589
10-3-7d	19.603	6.529	15.652	28.847	6.821

CHAPTER 5. MULTI-PERIOD BID GENERATION PROBLEM WITH PICKUP AND DELIVERY,  
 TIME WINDOWS, PROFITS, RESERVED REQUESTS, AND DELIVERY LEAD TIME

---

10-3-7e	64.341	17.978	31.037	48.308	15.965
10-3-7f	69.703	4.410	8.986	35.001	12.477
10-7-3g	38.808	3.725	7.204	19.219	5.548
10-7-3h	126.638	12.982	4.006	1281.631	5.461
10-7-3i	43.553	12	17.598	59.675	16.737
20-10-10a	28.657	7.128	13.705	57.593	12.689
20-10-10b	24.230	3.823	17.232	47.522	16.339
20-10-10c	6.910	3.122	14.581	58.826	5.049
20-5-15d	35.218	26	4.149	47.820	31.132
20-5-15e	25.589	61	2.198	38.400	17.139
20-5-15f	26.405	11.956	3.272	19.488	4.466
20-15-5g	28.103	25	13.027	37.632	6.161
20-15-5h	31.286	25	19.216	45.092	0.027
20-15-5i	0.662	8.137	19.918	50.446	12.307
30-15-15c	26.407	14	2.631	28.895	0.322
30-10-20d	14.076	20	2.298	11.062	1.237
30-10-20e	12.496	18.888	26.888	85.147	0.587
30-10-20f	3.639	24	16.895	96.673	8.525
30-20-10g	25.422	62.5	17.672	94.593	4.711
30-20-10h	49.891	73.809	49.542	71.521	9.199
30-20-10i	26.192	22	55.671	34.511	2.748
40-20-20a	51.220	13.349	0.905	95.764	1.644
40-20-20b	45.278	13	4.833	90.275	0.365
40-20-20c	0.442	23.333	11.771	17.470	0.774
40-15-25d	15.492	21	11.793	11.645	0.709
40-15-25e	37.838	60	15.988	2.698	1.531
40-15-25f	97.434	42.857	11.142	6.306	1.682
40-25-15g	93.970	19.791	0.556	8.053	4.493
40-25-15h	59.595	4.255	15.003	0.112	0.559
40-25-15i	43.101	57.777	1.361	0.066	0.186
50-25-25a	2.542	3.106	6.586	13.095	0.579
50-25-25b	3.449	12.815	12.173	2.480	0.261
50-25-25c	76.508	11.195	4.839	2.422	5.174
50-20-30d	72.300	55.590	9.763	10.628	2.713
50-20-30e	77.514	21	14.834	3.800	0.323
50-20-30f	21.194	7.457	2.426	49.871	1.322
50-30-20g	6.768	7.777	2.8725	1.359	1.782
50-30-20h	94.558	5.425	4.960	3.341	1.463
50-30-20i	72.028	46	13.892	4.594	26.053

---

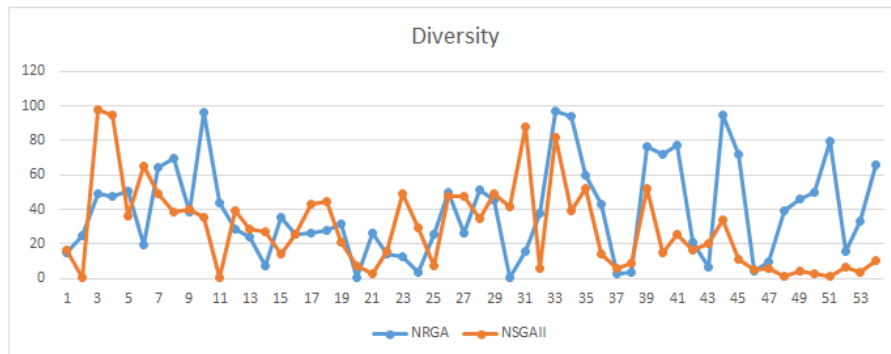
100-50-50a	3.911	27	11.364	2.025	42.655
100-50-50b	9.704	12.824	8.815	1.211	40.189
100-50-50c	38.900	4.462	7.251	1.679	34.338
100-25-75d	46.054	64.905	4.968	7.315	53.381
100-25-75e	50.174	5.294	4.891	2.365	41.264
100-25-75f	79.289	37	6.234	11.326	43.881
100-75-25g	15.451	23	12.844	7.475	0.460
100-75-25h	33.430	8.301	4.689	0.852	0.181
100-75-25i	65.983	6.808	19.928	0.512	1.211

**Table 5.11:** Multi-objective metrics computed for NSGA-II

<b>Instance Number</b>	<i>Diversity</i>	<i>NOS</i>	<i>MID</i>	<i>Spacing</i>	<i>Time</i>
6-3-3	16.628	7.439	1.260	2.298	22.114
8-4-4	0.508	0.003	0.003	10.365	0.207
10-5-5a	98.211	4.059	0.086	3.723	0.431
10-5-5b	94.525	4.882	0.523	3.658	0.338
10-5-5c	36.161	38	8.986	48.406	0.088
10-3-7d	64.842	45.652	17.384	31.538	0.288
10-3-7e	49.013	3.609	30.024	52.563	1.891
10-3-7f	38.134	2.167	46.29	40.808	0.643
10-7-3g	39.718	43	24.235	83.379	0.970
10-7-3h	35.543	26	98.635	23.976	0.827
10-7-3i	5.292	6	4.769	60.365	0.930
20-10-10a	39.163	3.301	29.787	49.504	0.968
20-10-10b	28.387	2.038	18.778	48.524	0.506
20-10-10c	27.198	27	0.280	26.573	0.347
20-5-15d	14.095	38	0.252	63.679	0.888
20-5-15e	25.717	22	1.299	18.680	0.412
20-5-15f	43.305	6	0.721	29.429	0.722
20-15-5g	44.338	6.310	9.687	88.088	23.104
20-15-5h	20.965	5.859	1.618	1.944	30.746
20-15-5i	6.966	25.336	0.751	11.283	0.395
30-15-15c	2.941	5.730	0.190	18.768	30.565
30-10-20d	35.602	9	1.656	17.545	29.959
30-10-20e	48.791	26.666	4.040	0.817	23.887
30-10-20f	29.322	34.444	0.906	0.462	30.480
30-20-10g	7.276	14.255	0.039	1.135	28.911

CHAPTER 5. MULTI-PERIOD BID GENERATION PROBLEM WITH PICKUP AND DELIVERY, TIME WINDOWS, PROFITS, RESERVED REQUESTS, AND DELIVERY LEAD TIME

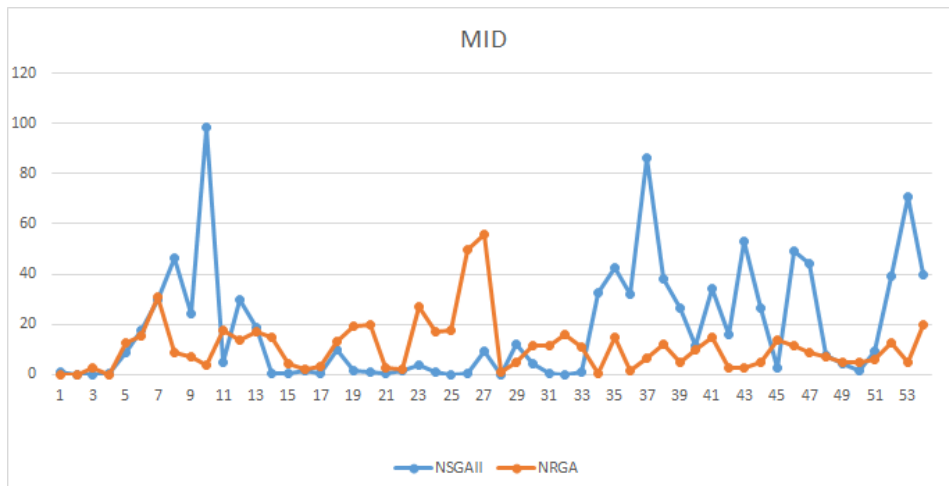
30-20-10h	47.891	25	0.536	0.820	31.702
30-20-10i	47.895	16.041	9.458	19.398	31.206
40-20-20a	35.066	16	0.017	0.017	26.333
40-20-20b	49.537	14	11.805	4.833	24.901
40-20-20c	41.496	21	4.367	11.771	25.614
40-15-25d	88.351	15	0.362	11.793	23.377
40-15-25e	5.422	6	0.059	15.988	24.512
40-15-25f	81.832	18	1.098	11.142	33.135
40-25-15g	39.212	5.356	32.840	0.556	36.765
40-25-15h	51.826	25	42.525	15.002	31.892
40-25-15i	14.234	18	31.844	1.361	28.974
50-25-25a	5.688	30.377	86.455	6.586	0.912
50-25-25b	8.790	13.529	38.306	12.173	1.562
50-25-25c	52.000	6.905	26.473	4.839	1.486
50-20-30d	14.618	17.978	11.573	9.763	0.519
50-20-30e	25.370	17.423	34.435	14.834	22.245
50-20-30f	16.740	8.137	16.045	2.426	25.246
50-30-20g	20.165	5.425	53.212	2.872	22.595
50-30-20h	34.277	2.824	26.259	4.960	31.545
50-30-20i	10.924	20.043	2.509	13.892	22.396
100-50-50a	4.864	4.808	49.212	11.364	27.815
100-50-50b	5.382	7.439	44.234	8.815	13.578
100-50-50c	1.068	2.167	7.376	7.251	39.281
100-25-75d	4.399	3.301	4.250	4.968	19.096
100-25-75e	2.408	5.119	1.704	4.891	4.722
100-25-75f	1.572	10.866	9.212	6.234	18.461
100-75-25g	6.825	8.921	39.171	12.844	20.314
100-75-25h	3.343	11.882	70.730	4.689	16.423
100-75-25i	10.574	8.124	39.968	19.928	17.435

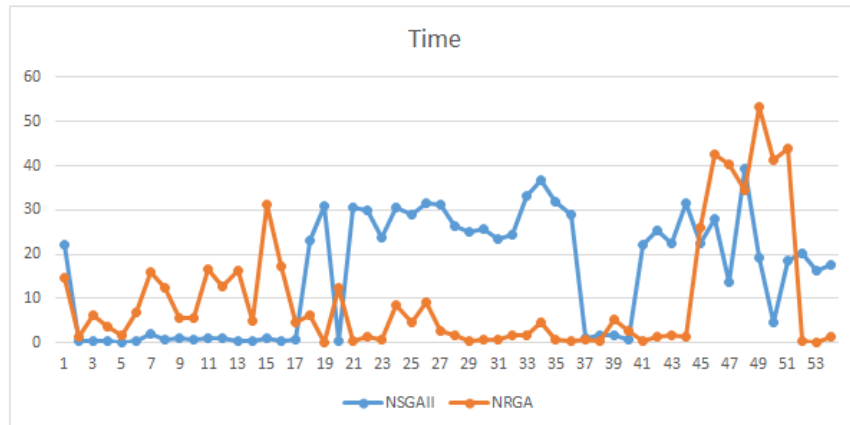


**Figure 5.7:** Graphical comparisons of Diversity metric for NSGA-II and NPGA algorithms on all test problems.

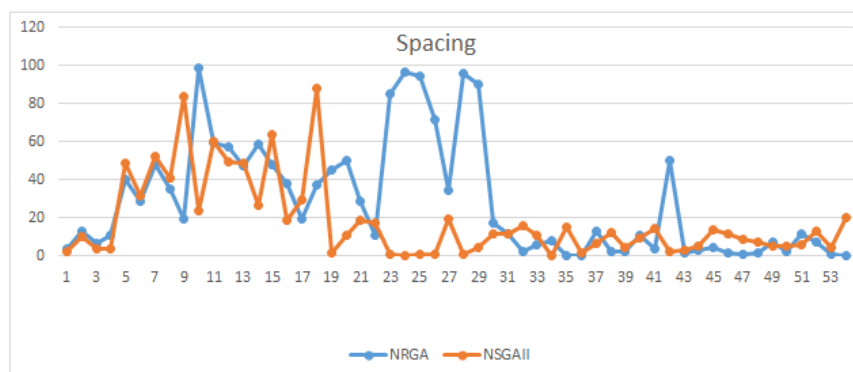
**Table 5.12:** The results of ANOVA test over the studied bi-objective mathematical model.

<i>Metrics</i>	<i>Source</i>	<i>DF</i>	<i>SS</i>	<i>MS</i>	<i>F-test</i>	<i>P-value</i>	<i>Results</i>
<i>MID</i>	Algorithm	1	1430	1430	4.40	0.038	Null hypothesis is rejected
	Error	106	34444	325			
	Total	107	35874				
<i>Time</i>	Algorithm	1	962	962	5.32	0.023	Null hypothesis is rejected
	Error	106	19162	181			
	Total	107	20124				
<i>Spacing</i>	Algorithm	1	28276	28276	1.86	0.175	Null hypothesis is not rejected
	Error	106	1611068	15199			
	Total	107	20124				
<i>NOS</i>	Algorithm	1	1200	1200	5.48	0.021	Null hypothesis is rejected
	Error	106	23198	219			
	Total	107	24398				
<i>Diversity</i>	Algorithm	1	26426	26426	1.81	0.182	Null hypothesis is not rejected
	Error	106	1550063	14623			
	Total	107	1576488				

**Figure 5.8:** Graphical comparisons of MID metric for NSGA-II and NPGA algorithms on all test problems.

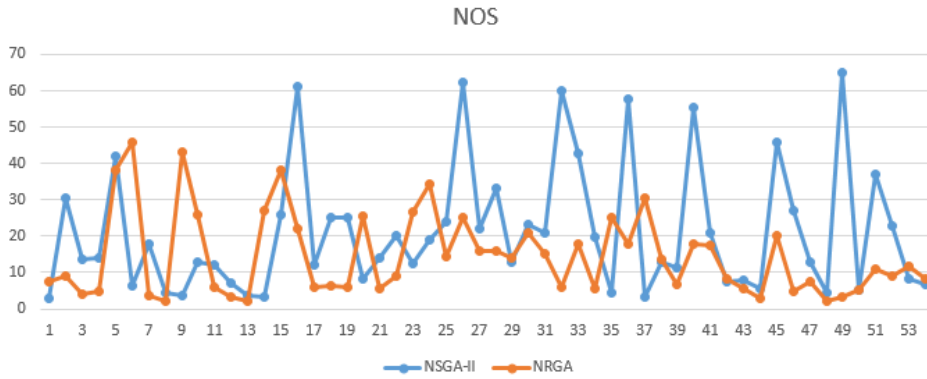


**Figure 5.9:** Graphical comparisons of Time metric for NSGA-II and NPGA algorithms on all test problems.



**Figure 5.10:** Graphical comparisons of Spacing metric for NSGA-II and NPGA algorithms on all test problems.





**Figure 5.11:** Graphical comparisons of NOS metric for NSGA-II and NPGA algorithms on all test problems.

In order to show the significant difference in terms of RPD (metrics for bi-objective problems algorithms), the algorithms are statistically analyzed according to obtained solutions via analysis of variance (ANOVA) tests [42]. The procedure of ANOVA including F-test value and also P-value on RPD (each metric for bi-objective algorithms) are summarized in Table 5.9 and Table 5.12. The results are analyzed at 95% confidence level. Therefore, for the single objective problem, there are significant differences and the average results of RPD for GASA is better than ITS. Additionally, the computation time of GASA is shorter than ITS according to Table 5.7. According to Table 5.7, for some instances when the number of requests are 20, CPLEX could not obtain a solution in a preset running time while ITS and GASA have obtained a solution. Indeed, for all instances with the number of requests 20, no feasible solution is obtained by CPLEX in the time limitation 1 hour, so we compare the solutions obtained by our algorithms based on their relative gaps with the upper bound obtained by CPLEX, i.e., using the above-mentioned criteria. Our algorithms can find an optimal solution for all 20 requests instances and can find a solution with the relative profit gap smaller than 5.92%. For medium 30 requests instances, our GASA algorithm can find a solution with the gap smaller than 6.2%, whereas our ITS algorithm can find a solution with the gap smaller than 7.96%. For 40 requests instances, our GASA algorithm can find a solution with the gap smaller than 9.69%, whereas our ITS algorithm can find a solution with the gap smaller than 11.72%. For large 50 requests instances, our GASA algorithm can find a solution with the gap smaller than 11.69%, whereas our ITS algorithm can find a solution with the gap smaller than 12.54%. For 100 requests instances, our GASA algorithm can find a solution with the gap smaller than 16.84%, whereas our ITS algorithm can find a solution with the gap smaller than 18.77%.

For the bi-objective problem, in terms of MID, Time, and NOS, there are significant differences between the two algorithms, whereas in MID metric, NSGA-II is better than NPGA, NPGA outperforms NSGA-II in NOS and time. It means that NSGA-II has a better convergence performance in MID, whereas in diversity metric, NPGA performs better. In the other metrics,

there are not significant differences between the two algorithms and the average results of the metrics indicates which algorithm outperforms another one. The average in the spacing and diversity metrics show NREGA operates better than NSGA-II. Finally, the results of NSGA-II and NREGA for each metric given in Fig. 5.6, Fig. 5.7, Fig. 5.8 and Fig. 5.9, which can confirm the statistical results.

## 5.6 Conclusion

In this chapter, we have presented two methods to evaluate each bid submitted by carriers to the auctioneer. One is evaluating a bid according to a score function which is defined as cost plus weighted delivery lead time as in a multi-attribute score auction. The other is evaluating a bid according to its multi-attributes. Accordingly, each carrier has two ways to determine its bid, one is minimizing a score (cost and delivery lead time) and the other is minimizing both attributes of a bid (cost plus weighted delivery lead time). At first, we focused on the first way by assuming a score function composed of cost and weighted delivery lead time and formulate the multi-period bid generation problem as a single objective mathematical model. After presenting the MILP model for the problem, we solved it by two metaheuristic algorithms GASA and ITS. Then, we have taken the second way by evaluating a bid according to two attributes. We have presented a bi-objective mathematical model for multi-period BGP. To solve the bi-objective problem, NSGA-II and NREGA have been applied. They are compared according to well-known metrics and the superiority of each algorithm in each metric has been determined. About the single objective problem, GASA outperforms the other algorithm in terms of profit and computation time. For the bi-objective problem, in MID, Time, and NOS there are significant differences between two algorithms, NSGA-II and NREGA, whereas in MID metric, NSGA-II is better than NREGA, NREGA algorithm overcomes NSGA-II in NOS and time. It means that NSGA-II has a better convergence performance in MID, whereas in diversity features, NREGA performs better. In the other metrics, there are not significant differences and the average results in the metrics indicate which outperforms another one. The average in the spacing and diversity metrics show NREGA performs better than NSGA-II.

## CONCLUSIONS AND PERSPECTIVES

### 6.1 Conclusion

In last years, collaborative logistics and especially carrier collaborative transportation have been appeared as a beneficial technique for small to medium-sized freight carriers to ameliorate profitability by diminishing empty back-hauls by vehicle reposition and improving vehicle fill rates. In this thesis, we discussed collaborative transportation planning (CTP) raised in carrier collaboration, particularly the carrier collaboration in less than- truckload (LTL) transportation.

Two key-subjects are often discussed for carrier collaboration in LTL transportation: The request re-assignment problem and the post-collaboration profit allocation problem. In this thesis, we concentrated on the requests re-assignment problem in centralized multi-carrier collaboration and suggested an auction-based carrier collaboration approach for CTP with profits. In this field, three main sub-problems have been considered: Multi-carrier centralized CTP with pickup and delivery requests with time windows and reserved requests, periodic bid generation problem (BGP) and periodic bid generation problem (BGP) with delivery lead time.

After studying the research background in Chapter 1 and reviewing the collaborative transportation planning state-of-the-art in Chapter 2, we suggested in Chapter 3, a multi-carrier centralized CTP with pickup and delivery requests and time windows with reserved requests (PDPTWR), a new vehicle routing problem developed in LTL carrier collaboration.

The multi-carrier centralized PDPTWR involves exchangeable requests, which can be exchanged among carriers in a coalition, while each carrier keeps reserved requests to serve by the carrier itself. After presenting mathematical programming model, three metaheuristic algorithms have been developed as solution methods: An adaptive large neighborhood search (ALNS), a

hybrid Genetic Algorithm and Simulated Annealing (GASA) and an improved tabu search (ITS). Numerical experimental results show that our algorithms give promising results compared with commercial solver CPLEX. Besides, the ALNS algorithm outperforms the two other algorithms not only in finding better solutions but also its running time is better than others.

In Chapter 4, a multi-period Bid Generation Problem (BGP) for a carrier has been considered. In the problem, there are two different types of requests, reserved requests of the carrier and selective requests. Each request has a pickup and delivery request with two time windows. For each selective request there is a period window while each reserved request has a predetermined period to be served. A period window specifies the earliest period and the latest period the request must be served while a time window shows the earliest and latest service time and is allocated to both pickup point and the delivery point. In addition, each selective request is associated with a profit that is the price for serving the request provided by a shipper. A hybrid approach that combines a genetic algorithm and simulated annealing (GASA) has been proposed to solve the problem. For the purpose of comparison, we utilized Biogeography-Based Optimization algorithm (BBO) that is a population-based algorithm whose specifications are close to the GA. The numerical results demonstrated that in spite of CPLEX, the proposed algorithms can find a good feasible solution in a reasonable computation time for large instances.

In the companies, besides minimizing all costs, a shipper interested in reducing the delivery lead time. Reducing delivery lead time has two benefits. At first, shorter delivery lead time makes agile supply chain management and responding to the orders is executed faster. Thus, it improves the ability of shippers to compete in the industry. Second benefit of shorter delivery lead time is choosing the carriers according to the geographic factors. It means that shippers distribute freight volumes to a carrier geographically. Totally, by considering the short delivery lead time, the auctioneer gets involve also in other attributes not only costs. In many acquisition situations although cost is an important attribute to evaluate the orders, carriers care about the quality specifications when estimate the submitted orders by suppliers. To these points, in Chapter 5, we addressed delivery lead time in transportation service procurement, which is conducted by shippers and using auctions to purchase transportation service from carriers in the planning stage. Many shippers get involve with delivery lead time in practice; the transportation service is better when delivery lead time decreases. consequently, we considered two ways to comprise delivery lead time in the problem. Firstly, a bi-objective problem has been considered, where the first objective is the maximization of all profits while the other is the minimization of delivery lead time. After presenting the mathematical model, we solved it by NSGAI and NPGA and compared them by well-known metrics. Secondly, we assumed in a single objective problem delivery lead time included as an attribute in the objective function by using some multi-criteria decision making (MCDM) methods. GASA and ITS have been employed to solve the model. The results indicated both of them can get a feasible solution in large instances, on the contrary, CPLEX could not.

In summary, the principal contributions of this thesis is comprised three new concepts appeared in carrier collaboration, presenting mathematical models for them, and the development of efficient solution methods: 1) the centralized multi-carrier PDPTWR, a new vehicle routing problem with multi-carriers to allocate exchangeable requests to the suitable carriers. 2) the multi-period BGP: A new multi-period vehicle routing problem to identify profitable requests in a period. 3) the multi-period BGP with minimizing delivery lead time, a new bid generation problem appeared in LTL carrier collaboration in order to reduce the delivery lead time and increasing the profits.

## 6.2 Perspectives

Although we have suggested a complete scheme to address the requests re-allocation problem in LTL carrier collaboration, there are still other studies to be done to develop the models, the solution issues, and make them more suitable in the realistic carrier collaboration environments.

Firstly, for the multi-carrier centralized PDPTWR, we have assumed all requests are deterministic while in the realistic environment, requests can be stochastic. Thus, in the future study we can focus on the stochastic requests particularly exchangeable requests can be stochastic.

Secondly, for the multi-period PDPTWPR via BGP, it will be an interesting issue to deal with classified selective requests wherein each period if a selective request in each category cannot be served, serving all selective requests in the same category will not be possible. In the realistic collaboration environment, sometimes, ignoring a group of requests is reasonable to save costs.

Thirdly, we can add the effect of decreasing CO<sub>2</sub> emission to our model in the third problem. Although reducing the delivery lead time is important to meet the customer's satisfaction, CO<sub>2</sub> emission and increasing greenhouse gases in the environment is a critical issue in the last years as it has become the huge problem in the world and most of the well-developed countries make plans to reduce the effects of CO<sub>2</sub> emissions in the atmosphere. Consequently, in our model minimization of CO<sub>2</sub> emission could be included to take into account the environmental aspect besides cost and profit.

Finally, profit allocation approaches have not addressed in this thesis while this subject is necessary for collaborative logistics implementing. In the literature, the requests re-allocation problem and the profit allocation problem are often manipulated independently. It would be better if the two problems could be examined together, so better results can be achieved compared to observed them separately.





## APPENDIX OF EXTENDED FRENCH ABSTRACT

La collaboration entre les petites et moyennes entreprises (PME) est cruciale pour améliorer leur situation économique. En effet, cette collaboration au sein d'un réseau est une approche importante pour réaliser des bénéfices dans un environnement très concurrentiel. Le partage des demandes entre les transporteurs génère des profits pour toutes les entreprises, et assure la satisfaction des demandes des clients. Les raisons qui motivent les industriels et les chercheurs à mettre en œuvre la planification de la collaboration entre transporteurs sont la concurrence féroce sur les marchés mondiaux, la courte durée de vie des produits, l'augmentation du coût du carburant, le prix de la main-d'œuvre et l'effet de toutes ces raisons sur les profits des entreprises [22]. Par conséquent, la collaboration entre les transporteurs est une stratégie pertinente pour les chercheurs et les praticiens de l'industrie afin de réduire les retours à vide des véhicules et d'augmenter leurs taux de remplissage [9].

Une coalition entre plusieurs acteurs du même niveau d'opérations logistiques est définie comme une collaboration de transporteur. Grâce à cette collaboration, de nombreux transporteurs peuvent échanger leurs demandes de transport entre eux pour arriver à une meilleure complémentarité entre les demandes. Cet échange de demandes peut améliorer la planification de tournées des véhicules tout en réduisant les retours à vide et, par conséquent, en augmentant leurs profits de transport. Deux problèmes importants sont discutés dans le problème lié à la collaboration. Le premier est de savoir comment ré-attribuer de façon optimale les demandes de transport entre transporteurs pour maximiser le profit total. Le second est de définir comment répartir les bénéfices obtenus après une collaboration équitable entre les transporteurs au sein d'une coalition, afin d'assurer la pérennité de cette alliance [65].

Dans cette thèse, nous nous concentrons sur le premier problème qui est également considéré comme un problème de planification collaborative des transports (CTP) [101], [102], [99]. Dans cette thèse, nous étudions le problème de la collaboration entre les transporteurs dans le service de

ramassage et de livraison avec différentes spécifications (fenêtres de temps, bénéfices, demandes réservées, concept périodique et en supposant deux fonctions objectifs dans le Chapitre (5) où le transport de chargement partiel est pris en compte dans le cadre de l'initiative centralisée de transport de lots brisés et des approches décentralisées. En effet, le problème de collaboration entre transporteurs dans le service de ramassage et de livraison avec des fenêtres de temps a attiré beaucoup d'attention au cours des dernières années avec un attrait particulier à l'objectif de la planification opérationnelle dans les réseaux logistiques. Dans le logistique collaborative avec ramassage et livraison, chaque demande est définie par deux emplacements, la ramassage et la livraison, De plus, chaque point a une fenêtre de temps. Dans un premier temps, chaque transporteur est rémunéré par les expéditeurs (clients). Chaque transporteur établit un plan pour répondre aux demandes qu'il reçoit au moyen de ses propres véhicules et de la pré-collaboration.

Au final, le rapport coût/bénéfice peut être calculé. Certaines demandes (appelé aussi demandes spéciales) sont aussi livrées en collaboration avec d'autres transporteurs. En plus, les transporteurs peuvent aussi émettre des demandes d'échanges de livraison des biens (appelé demande d'échange). De ce fait, dans la collaboration, toutes les demandes partagées de tous les transporteurs sont cumulées. Par conséquent, deux approches notables sont utilisées pour résoudre le problème du logistique collaborative avec ramassage et livraison dans cette thèse:

La première approche consiste à réaffecter un ensemble de demandes échangeables au moyen d'une approche centralisée pour faire une planification collaborative des transports et les desservir ensuite par des tours à chaque transporteur en prenant en compte ses contraintes de disponibilité de ressources. Dans ce cas, le problème est d'abord formulé par le biais d'une approche centralisée en échangeant des demandes entre les transporteurs. Ensuite, en présentant le modèle mathématique, le problème est résolu par des algorithmes métaheuristiques.

La seconde est une approche de planification décentralisée qui nécessite l'utilisation de (BGP) pour sa résolution. Dans ce problème, il y a deux demandes différentes: les demandes réservées à chaque transporteur et les demandes sélectives. Chaque demande sélective est associée à un bénéfice qui est le prix de la demande fournie par un expéditeur. En incluant un concept périodique, le transporteur peut planifier ses opérations de transport à l'avance et dans l'horizon temporel.

En comparant les deux approches, l'approche décentralisée permet de préserver l'autonomie décisionnelle et les données métier ainsi que la confidentialité de chaque transporteur. Aussi, le bénéfice total de tous les transporteurs généré par le système centralisé est généralement plus élevé que celui généré par l'approche décentralisée, ceci est dû en raison de l'importance de l'approche décentralisée. La principale préoccupation de l'utilisation d'un système centralisé consiste à préserver la confidentialité des données commerciales échangées entre les transporteurs. Avec l'apparition et le développement ces dernières années de Fourth-Party Logistics (4PL), les fournisseurs agissent à titre d'intégrateur et coordonnent la planification du transport sans un échange direct de données. De ce fait, cette préoccupation de partage de données est évitée. De



plus, il est possible, digne pour un transporteur de révéler certaines informations aux (4PL) en échange d'une augmentation de profit. Une telle approche de planification centralisée a également été utilisée dans le transport collaboratif, notamment dans le commerce tel que le commerce de services dans le domaine de la logistique du commerce électronique inter-entreprises [79]. Enfin, l'objectif de la deuxième approche est de réduire le délai de livraison (temps de transit du transport) en plus de maximiser les profits. Cela signifie que le problème bi-objectif doit être résolu. Par conséquent, on a défini un modèle mathématique bi-objectif que l'on va résoudre avec des algorithmes bi-objectifs appropriés.

## A.1 Les contributions de la thèse

Dans cette thèse, deux types d'approches de planification concertée des transports sont proposés pour résoudre les problèmes cités précédemment. L'une est une approche décentralisée et utilisant le problème de la génération de soumissions, l'autre est l'approche de planification centralisée.

1. Dans l'ensemble, l'approche de planification centralisée peut permettre d'obtenir un meilleur rendement en matière de collaboration entre plusieurs transporteurs avec des lots brisés. Dans la planification collaborative centralisée des transports, où chaque transporteur dispose d'un nombre limité de véhicules et chaque demande est une demande de ramassage et de livraison dans les fenêtres de temps spécifiées. Dans ce cas, chaque transporteur peut réserver des demandes qui doivent être en libre-service. En échangeant certaines demandes entre eux, une telle collaboration peut les aider à réduire les coûts de transport et augmenter leurs profits. Cette approche peut être réalisée par l'intermédiaire d'une Quatrième Partie Prestataire logistiques (4PL) qui agit en tant qu'intégrateur coordonnant la planification du transport entre les transporteurs. Un modèle de programmation linéaire à nombres entiers mixtes (MILP) est formulé pour ce problème. Étant donné que le modèle est NP-difficile, une recherche adaptative de grand quartier (Adaptive Large Neighborhood Search- ALNS) l'algorithme génétique hybride avec recuit simulé (GASA) et un algorithme amélioré de La recherche Tabu (ITS) pour résoudre le modèle. Le ALNS utilise une variété de méthodes de destruction et d'élimination. Les opérateurs de réparation, les multiples phases de recherche et la post-optimisation est basée sur la recherche locale, le GASA profite des avantages de l'algorithme génétique et du recuit simulé (d'une part, le critère démographique ajoute la diversification et d'autre part, la caractéristique de recherche locale par l'application de l'AS permet d'éviter d'être piégé dans l'optimum local). Le TS amélioré (ITS) adopte un opérateur de mutation et conserve de multiples solutions pendant la durée de vie de l'équipement. Le processus de recherche afin d'obtenir une meilleure solution plus rapidement est présenté dans le (Chapitre 3).
2. Un problème de génération d'enchères sur plusieurs périodes (BGP) pour un transporteur peut être pris en compte dans la logistique collaborative avec ramassage et livraison.

Dans le problème, il existe deux types de demandes différentes, les demandes réservées du transporteur et des demandes sélectives. Le transporteur s'engage par contrat avec ses expéditeurs à servir tous les clients par lui-même. Les demandes sélectives sont proposées par d'autres transporteurs et elles sont ouvertes aux soumissions des transporteurs. Chaque demande comporte une demande de ramassage et de livraison avec un point de ramassage, un point de livraison, une quantité de ramassage et de livraison et deux fenêtres de temps. Le temps de livraison spécifie la période la plus rapprochée et la dernière période de traitement de la demande, tandis que l'intervalle de temps affiche l'heure de service la plus ancienne et la plus récente et elle est affectée aux deux points de ramassage et de livraison. En outre, un bénéfice est associé à chaque demande sélective qui est le prix de la demande fournie par un expéditeur. En incluant le concept périodique dans la CA (enchères combinatoires), le transporteur peut planifier ses opérations de transport à l'avance et dans un horizon mobile. De ce fait, un transporteur doit prendre deux décisions importantes dans le BGP: Quelles demandes sélectives sont choisies? La seconde est de déterminer l'intervalle de temps de chaque requête sélective sélectionnée pour construire les routes afin de maximiser le profit total. Il en résulte un nouveau problème de ramassage et de livraison périodiques avec les fenêtres de temps, les profits et les demandes réservées. Donc, ce problème est un problème différent de celui présenté dans [65] qui considère les BGP à période unique. Selon [24], le présent problème est NP-difficile et il est impossible d'obtenir des solutions réalisables ainsi qu'une solution optimale pour les grandes instances en utilisant un solveur commercial comme CPLEX. Ainsi, une approche hybride qui combine un algorithme génétique et un recuit simulé (dénommé « GASA») est proposée pour résoudre la problématique. Le GASA adopte une structure chromosomique spécifique à un problème. Un croisement et un des opérateurs de mutation bien adaptés à la structure chromosomique sont utilisés. Cette méthode utilise l'algorithme basé sur la population et l'algorithme du point unique à la diversification et l'intensification de la solution séquentiellement. A des fins de comparaison, nous utilisons également un algorithme d'optimisation basé sur la biogéographie (BBO) qui est un algorithme basé sur la population et dont ses spécifications sont proches de l'GA et son fonctionnement est sur la population de solutions pour trouver la meilleure solution au problème. En comparant GA et BBO, chaque chromosome dans l'GA, qui est considérée comme une personne ayant une valeur de condition physique, est un habitat convenable qui est appelé HSI (Indice de qualité de l'habitat, cet indice est présenté dans le Chapitre 4).

3. Dans les entreprises, en plus de minimiser tous les coûts, l'expéditeur est intéressé par la réduction des délais de livraison. La réduction des délais de livraison présente deux avantages: D'abord, des délais de livraison plus courts permettent une gestion agile de la chaîne d'approvisionnement et l'exécution des commandes est accélérée. Ainsi, elle améliore la capacité des expéditeurs à être concurrentiels dans l'industrie. Le deuxième avantage

de la réduction de délai de livraison est le choix des transporteurs en fonction de facteurs géographiques. Dans de nombreuses acquisitions, bien que le coût soit un facteur important dans l'évaluation des commandes, les transporteurs se soucient de ce qui suit sur les spécifications de qualité lors de l'estimation des commandes passées par les fournisseurs [48]. En effet, en considérant le délai de livraison court, le commissaire-priseur s'implique aussi dans d'autres facteurs que les coûts. Dans de nombreuses situations d'acquisition, bien que le coût soit un attribut important dans l'évaluation des commandes, les transporteurs se soucient des spécifications de qualité lors de l'estimation des commandes passées par les fournisseurs [26]. Dans le travail actuel, l'objet du service de transport est de maximiser les avantages totaux et de réduire au minimum le délai de livraison total présenté dans le (Chapitre 5). Cet objectif peut se répartir de la manière suivante:

- Un objectif unique: On peut noter dans la théorie traditionnelle de la passation de marchés standard des enchères que, le commissaire-priseur ne se soucie que du prix de l'objet, mais pas d'autres attributs. Toutefois, dans de nombreuses situations d'approvisionnement, l'acheteur se soucie de certains attributs autres que le prix lors de l'évaluation des offres soumises par les fournisseurs. On peut trouver parmi les contraintes non monétaires qui préoccupent les acheteurs la qualité ou le temps nécessaire à la livraison. En d'autres termes, les expéditeurs s'inquiètent aussi des délais de livraison. Ce délai indique un meilleur service de transport. Dans ce problème, les transporteurs recherchent des demandes de transport par le biais d'enchères combinatoires au cours de chaque période qui représente la fenêtre de demandes sélectives. En plus des demandes réservées pour maximiser le profit total et d'autre part, le délai de livraison doit être réduit. Chaque demande sélective a des fenêtre de temps qui décrivent les heures de ramassage et de livraison valides et la fenêtre de période indiquant la période la plus rapprochée et la période la plus éloignée pour répondre à la demande sélective, tandis que chaque demande réservée doit être délivrée dans un une période préétabli. L'objectif du problème est de décider quelles demandes sélectives dans quelles périodes sont servies en plus des demandes réservées afin de maximiser le profit total. Un modèle de programmation linéaire mixte (MILP) est proposé au problème. Le problème est NP-difficile et pour résoudre les grandes instances, deux algorithmes métaheuristiques sont suggérés, une combinaison d'Algorithme Génétique et de Recuit simulé (GASA) et recherche Tabu améliorée (ITS).
- Bi-objectif: En fait, problème de génération d'enchères à périodes multiples et à objectifs multiples maximiser les profits et réduire les délais de livraison (temps de transit) [48]. Les deux fonctions objectives visant à minimiser le temps de transit et à maximiser les profits sont contradictoires et sont donc considérées séparément. Résoudre un problème multi-objectifs signifie qu'on doit trouver un ensemble de solutions appelées frontière de Pareto. Depuis que les algorithmes évolutionnaires

travaillent sur la population de solutions, ils sont bien adaptés aux problèmes multi-objectifs. Lors de leur conception, deux objectifs sont pris en compte: ils doivent atteindre la frontière de Pareto, ils doivent aussi trouver toutes les solutions le long de la frontière. C'est un problème d'intensification et de diversification, et il existe certains mécanismes pour traiter les problèmes suivants ces objectifs. Par conséquent, deux algorithmes basés sur la population, NSGAI et NPGA, sont les solveurs appliqués pour résoudre le modèle mathématique bi-objectif suggéré pour les soumissions à périodes multiples.

## **A.2 Problème de ramassage et de livraison avec plusieurs transporteurs, fenêtres de temps, demandes réservées et demandes échangeables**

Dans le présent document, nous mettons l'accent sur la planification concertée du transport (CTP) entre les transporteurs dans un environnement de transport de lots brisés, où chaque demande est une demande de ramassage et de livraison avec fenêtre de temps pour les points de ramassage et de livraison. Chaque transporteur peut avoir des demandes réservées, qui doivent être en libre-service, et la collaboration entre les transporteurs est réalisée par l'échange de demandes sans réserve (appelées ci-après demandes échangeables) entre eux. Nous adoptons une approche de planification centralisée pour le problème et ce dernier devient un problème de ramassage et de livraison par plusieurs transporteurs avec des fenêtres de temps, deux types différents de demandes, des demandes échangeables et réservées, ce qui est une nouvelle variante du problème NP-difficile de tournées des véhicules [97]. Malgré le problème d'orientation où il n'y a pas d'obligation de répondre à toutes les demandes [10], dans notre problème toutes les demandes échangeables doivent être traitées. Après avoir présenté un modèle de programmation linéaire d'entiers mixtes au problème, nous proposons un algorithme ALNS (Adaptive Large Neighborhood Search) pour le résoudre. Le ALNS a été utilisé avec succès pour résoudre divers problèmes de routage de véhicules [81]. Dans [65], un algorithme ALNS est appliqué pour résoudre un problème de ramassage et de livraison avec des fenêtres de temps, des profits, des demandes réservées et impliquant un seul transporteur avec un seul dépôt de véhicules. Dans ce problème, il n'est pas obligatoire de répondre à toutes les demandes sélectives. Le problème de tournées de véhicules examiné dans le présent document est un nouveau problème qui concerne plusieurs transporteurs et plusieurs dépôts de véhicules. De plus, chaque demande réservée doit être servie par son propre transporteur et toutes les demandes doivent être servies par tous les transporteurs, les algorithmes ALNS existants ne peuvent pas être appliqués directement à notre problème, certaines adaptations sont nécessaires afin de résoudre notre problème. Le ALNS proposé dans le présent document fait appel à divers opérateurs de destruction/réparation. Sa procédure de recherche se compose de plusieurs phases, où différentes phases utilisent différentes

politiques pour le choix des demandes de destruction (suppression) ou de réparation (réinsertion). L'idée de base est de modifier le comportement des opérateurs de destruction/réparation au cours d'une phase successive de l'ALNS afin qu'il puisse explorer diverses régions. De plus, une post-optimisation basée sur la recherche locale est appliquée à la fin de chaque phase afin d'améliorer encore la qualité de la solution finale obtenue par l'ALNS. A des fins de comparaison, nous développons également une approche de recherche Tabu améliorée. Cette approche améliore la recherche tabu classique en conservant de multiples solutions pendant le processus de recherche, en collectant et en utilisant leurs informations afin d'améliorer plus rapidement la solution actuelle [75]. De plus, nous utilisons une tactique de mutation qui est une autre technique pour améliorer la performance de cet algorithme [52].

### A.2.1 Description et Formulation du Problème

Il y a plusieurs transporteurs avec deux types de demandes: les demandes réservées et les demandes échangeables. Les demandes réservées de chaque transporteur doivent être servies par le transporteur lui-même, tandis que ses demandes échangeables peuvent être servies par d'autres transporteurs. La collaboration entre les transporteurs est réalisée par l'échange de demandes entre eux. Chaque transporteur dispose d'un nombre limité de véhicules situés initialement dans un dépôt. On suppose que tous les véhicules sont homogènes. Chaque ordre de transport est un ordre de ramassage et de livraison caractérisé par un lieu de ramassage avec une fenêtre de temps, un lieu de livraison avec une fenêtre de temps également, une demande et un montant de recettes qui peut être généré pour répondre à l'ordre. Aussi, le revenu d'une demande est le prix payé par un client (expéditeur) à un transporteur qui répond à sa demande. L'objectif du problème est de réaffecter de façon optimale toutes les demandes échangeables entre les transporteurs afin de maximiser leur profit total ou, de façon équivalente, de minimiser leur coût total de transport pour répondre à toutes les demandes réservées et échangeables. Nous adoptons une approche centralisée pour résoudre ce problème. De ce fait, nous formulons d'abord ce problème comme un MILP. Le Tableau A.1 et Tableau A.2 énumère les indices, les paramètres, les ensembles, les variables de décision utilisés dans le modèle.

$$\min \sum_{m \in M} \sum_{i \in V} \sum_{j \in V} \sum_{k \in K_m} c_{ij} x_{ijkm}$$

tel que:

$$\sum_{j \in V, j \neq i} x_{jikm} - \sum_{j \in V, j \neq i} x_{ijkm} = 0 \quad \forall i \in W, \forall k \in K_m, \forall m \in M \quad (\text{A.1})$$

$$\sum_{m \in M} \sum_{j \in V, j \neq i} \sum_{k \in K_m} x_{ijkm} = 1 \quad \forall i \in W \quad (\text{A.2})$$

$$\sum_{j \in P, j \neq i} x_{ijkm} \leq 1 \quad \forall k \in K_m, \forall i = m \in M \quad (\text{A.3})$$

**Table A.1:** Indices, ensembles, paramètres du modèle mathématique.

Symbols	Description
$G = (V, E)$	Graphique $G$ dirigé est complet et a fait l'objet de l'étude.
$V$	Ensemble de tous les nœuds.
$E$	Ensemble de toutes les arcs.
$n$	Nombre de demandes.
$cr$	Nombre maximal de transporteurs.
$M$	Ensemble de transporteur.
$DC$	Ensemble de noeuds de dépôt de transporteurs.
$p = \{1, 2, \dots, n\}$	Ensemble de noeuds de ramassage, où l'index de chaque noeud de ramassage $i$ est le même que l'index de sa demandes correspondante $i$ , $i = 1, \dots, n$ , $n$ est le nombre de demandes.
$D = \{n + 1, \dots, 2n\}$	Ensemble de noeuds de livraisons, où $n$ est le nombre de demandes.
$W$	Ensemble de nœuds, à l'exception des nœuds de dépôt de tous les transporteurs.
$R = \{1, \dots, n\}$	Ensemble de demandes qui inclut le sous-ensemble de demandes réservées $R_r$ et le sous-ensemble de demandes échangeables $R_s$ .
$R_r^m$	Ensemble des demandes réservées du transporteur $m$ .
$R_s$	Ensemble de demandes échangeables.
$Q$	Capacité du véhicule.
$c_{ij}$	Coût de déplacement du nœud $i$ au nœud $j$ .
$s_i$	Temps de service du nœud $i$ .
$K_m = \{1, \dots, VK\}$	Ensemble de véhicules du transporteur $m$ , où $VK$ est le nombre maximum de véhicules.
$d_{n+i} = -d_i$	Demande du nœud de ramassage est égale à la demande du nœud de livraison $d_i$ est la demande du nœud de ramassage et $d_{n+i}$ est la demande du nœud de livraison, $d_{n+i}$ est négatif
$[e_i, l_i]$	Fenêtre de temps du nœud $i$ , où $e_i$ est la limite inférieure de la fenêtre de temps et $l_i$ est la limite supérieure de la fenêtre de temps.
$t_{i,j}$	Temps de déplacement du nœud $i$ au nœud $j$ , il est supposé égal au coût de déplacement du nœud $i$ au nœud $j$ .
$T$	Durée de chaque route.
$BM_{i,j} = l_j - e_i$	Utilisé dans les contraintes de fenêtres de temps.
$CV_i = Q + d_i$	Utilisé dans les contraintes de capacité des véhicules.

$$\sum_{j \in P; j \neq i} x_{ijkm} = 0 \quad \forall k \in K_m, \forall i, m \in M, i \notin \{m\} \quad (\text{A.4})$$

$$\sum_{j \in D; j \neq i} x_{ijkm} \leq 1 \quad \forall k \in K_m, \forall i = m \in M \quad (\text{A.5})$$

$$\sum_{j \in D; j \neq i} x_{ijkm} = 0 \quad \forall k \in K_m, \forall i, m \in M, i \notin \{m\} \quad (\text{A.6})$$

A.2. PROBLÈME DE RAMASSAGE ET DE LIVRAISON AVEC PLUSIEURS TRANSPORTEURS, FENÊTRES DE TEMPS, DEMANDES RÉSERVÉES ET DEMANDES ÉCHANGEABLES

**Table A.2:** Variables de décision pour le modèle mathématique.

Variables	Description
$x_{ijkm}$	Variable binaire égale à 1 si et seulement si le véhicule $k$ du transporteur $m$ utilise le arc $(i, j)$ .
$y_{ikm}$	Variable binaire égale à 1 si et seulement si la demandes $i$ est servie par le véhicule $k$ du transporteur $m$ .
$U_{ikm}$	Heure à laquelle le véhicule $k$ du transporteur $m$ commence le service au nœud $i$ .
$L_{ikm}$	Charge du véhicule $k$ du transporteur $m$ au départ du nœud $i$ .

$$\sum_{k \in K_m} y_{ikm} = 1 \quad \forall i \in R_m, \forall i, m \in M \quad (\text{A.7})$$

$$\sum_{k \in K_m} y_{ikm} = 0 \quad \forall i \in R, i \notin R_r^m, \forall i, m \in M \quad (\text{A.8})$$

$$\sum_{m \in K_m} \sum_{m \in M} y_{ikm} = 1 \quad \forall i \in R_s \quad (\text{A.9})$$

$$\sum_{j \in V, j \neq i, m'} x_{ijkm} = y_{ikm} \quad \forall i \in P, k \in K_m, \forall m \in M \quad (\text{A.10})$$

$$\sum_{j \in V, j \neq i, m'} x_{j(i+n)km} = y_{ikm} \quad \forall i \in P, k \in K_m, \forall m \in M \quad (\text{A.11})$$

$$U_{ikm} + t_{i(n+i)} + s_i \leq U_{(n+i)km} \quad \forall i \in P, k \in K_m, \forall m \in M \quad (\text{A.12})$$

$$U_{jkm} \geq U_{ikm} + t_{ij} + s_i - BM_{ij}(1 - x_{ijkm}) \quad \forall i \in V, k \in K_m, \forall m \in M, \forall j \in W \quad (\text{A.13})$$

$$e_i \leq U_{ikm} \leq l_i \quad \forall m \in M, \forall i \in W \quad (\text{A.14})$$

$$U_{ikm} + t_{ij} + s_i - BM_{ij}(1 - x_{ijkm}) \leq T \quad \forall i \in W, k \in K_m, \forall m \in M, \forall j \in M \quad (\text{A.15})$$

$$L_{ikm} + d_i - Q(1 - x_{ijkm}) \leq L_{jkm} \quad \forall i \in V, k \in K_m, \forall m \in M, \forall j \in V \quad (\text{A.16})$$

$$\max\{0, d_i\} \leq L_{ikm} \leq \min\{Q, CV_i\} \quad \forall i \in V, k \in K_m, \forall m \in M \quad (\text{A.17})$$

$$\sum_{j \in W} \sum_{k=1} x_{jikm} \leq K_m \quad \forall i \in M, \forall m \in M \quad (\text{A.18})$$

$$x_{ijkm} \in \{0, 1\} \quad \forall i, j \in V, \forall k \in K_m, \forall m \in M \quad (\text{A.19})$$

$$y_{ikm} \in \{0, 1\} \quad \forall i \in R, \forall k \in K_m, \forall m \in M \quad (\text{A.20})$$

$$U_{ikm} \geq 0 \quad \forall i \in V, \forall k \in K_m, \forall m \in M \quad (\text{A.21})$$

$$L_{ikm} \geq 0 \quad \forall i \in V, \forall k \in K_m, \forall m \in M \quad (\text{A.22})$$

L'objectif est de minimiser le coût total de transport de tous les transporteurs. La contrainte (A.1) est l'équation de préservation du flux et décrit chaque véhicule qui arrive à un nœud client et doit le quitter. L'équation (A.2) décrit que chaque nœud client est visité exactement une fois par les véhicules. Les contraintes (A.3)-(A.6) indiquent pour chaque transporteur le moment où chaque véhicule quitte son dépôt et doit retourner au même dépôt. Les contraintes (A.7)-(A.9) concernent les demandes réservées et échangeables. Les équations (A.7) à (A.8) garantissent que toutes les demandes réservées de chaque transporteur doivent être servies par le transporteur lui-même, tandis que l'équation (A.9) implique que chaque demande échangeable peut être servie par tout transporteur. Les contraintes (A.10) et (A.11) satisfont si une demande est servie, il doit y avoir un véhicule quittant son point de ramassage et arrivant à son point de livraison jumelé. La contrainte suivante (A.12) est la contrainte de fenêtre de temps pour les nœuds de ramassage et de livraison. La contrainte (A.13) montre la faisabilité dans le temps, c'est-à-dire que le véhicule  $k$  ne peut pas commencer à desservir le nœud  $j$  avant d'avoir terminé son service au nœud  $i$  et de se déplacer du nœud  $i$  au nœud  $j$ . La contrainte (A.14) impose une restriction sur la fenêtre de temps à chaque nœud client. Les équations (A.15), (A.16) et (A.17) garantissent la durée maximale de chaque trajet, le respect de la capacité des véhicules et le nombre maximal de véhicules pour chaque transporteur. Les autres contraintes définissent les variables.

### **A.3 Problème de génération d'enchères sur plusieurs périodes avec demandes de ramassage et de livraison, profits et demandes réservées**

Dans ce section, un problème de génération d'enchères sur plusieurs périodes (BGP) pour un transporteur est pris en compte. Dans le problème, il y a deux types de demandes différentes, les demandes réservées du transporteur et les demandes sélectives. Le transporteur s'engage par contrat avec ses expéditeurs à répondre seul à toutes les demandes réservées. Les demandes sélectives sont proposées par d'autres transporteurs et peuvent faire l'objet d'une offre par le transporteur. Chaque demande comporte une demande de ramassage et de livraison avec un point de ramassage, un point de livraison, une quantité de ramassage et de livraison et deux fenêtres de temps. La fenêtre de période spécifie l'intervalle dans lequel la demande doit être



### A.3. PROBLÈME DE GÉNÉRATION D'ENCHÈRES SUR PLUSIEURS PÉRIODES AVEC DEMANDES DE RAMASSAGE ET DE LIVRAISON, PROFITS ET DEMANDES RÉSERVÉES

---

exécutée, tandis que la fenêtre de temps affiche l'intervalle horaire affecté à la fois au point de ramassage et au point de livraison. De plus, chaque demande sélective est associée à un profit qui est le prix de service de la demande fournie par un expéditeur. En incluant le concept périodique dans l'AC, le transporteur peut planifier ses opérations de transport à l'avance et de façon horizontale. Un transporteur doit prendre deux décisions importantes dans le BGP: quelles demandes sont choisies pour enchère et servir dans leurs fenêtres de période de service et comment les routes sont construites pour maximiser son profit total. Cela entraîne un nouveau problème de ramassage et de livraison périodiques avec les fenêtres de temps, les profits et les demandes réservées. Ce problème est donc différent de celui présenté dans [65] qui considère les BGP à période unique. Selon [101], le problème actuel est NP-difficile et il est impossible d'obtenir une solution optimale pour les grandes instances en utilisant un solveur commercial comme CPLEX. Une approche hybride combinant un algorithme génétique et un recuit simulé (GASA) est donc proposée pour résoudre ce problème. Le GASA adopte une structure chromosomique spécifique au problème, des opérateurs de croisement et de mutation bien adaptés à la structure chromosomique. A des fins de comparaison, nous utilisons un algorithme d'optimisation basé sur la biogéographie (BBO) qui est un algorithme basé sur la population, ses spécifications sont proches de le GA et son fonctionnement pour rechercher la meilleure solution dans une population est similaire. En comparant le GA et le BBO, chaque chromosome du GA, qui est considéré comme un individu ayant une valeur de forme physique, est un habitat dont l'indice de convenance est appelé Indice de qualité de l'Habitat (Habitat Suitability Index). Habituellement, un habitat à HSI élevé est acceptable et offre de meilleures solutions. Les résultats numériques démontrent que l'algorithme proposé peut trouver une bonne solution réalisable dans un temps de calcul raisonnable pour des cas importants.

#### A.3.1 Description et modèle mathématique

Dans ce problème, nous considérons un transporteur qui veut soumissionner à des demandes parmi celles ouvertes à enchère (offertes par tous les transporteurs) dans une enchère combinatoire afin de maximiser son propre profit en résolvant un problème de génération de soumission. Étant donné que le transporteur planifie ses opérations de transport à l'avance et selon un horizon mobile, comme il est mentionné dans l'introduction, ce problème de génération d'offres implique plusieurs périodes. Nous considérons le problème dans le cas du le transport de chargement partiel, où chaque demande de transport est une demande de ramassage et de livraison avec fenêtres de temps, deux types de demandes, les demandes réservées et les demandes sélectives, sont impliquées, et chaque demande est associée à un profit qui est le revenu pour servir la demande. Formellement, le problème de génération d'offres multi-périodes avec les demandes de ramassage et de livraison, les fenêtre de temps, les demandes sélectives et les demandes réservées, et les profits peuvent être définis sur un graphique dirigé  $G = (V, E)$  où  $V$  est l'ensemble des nœuds comprenant tous les nœuds de ramassage et de livraison et de dépôt du transporteur

et  $E$  l'ensemble des arcs. L'ensemble de nœuds est défini par  $V = \{0, \dots, 2n + 1\}$ , où  $n$  représente le nombre de demandes, 0 et  $2n+1$  représentent le dépôt du transporteur,  $i$  et  $n + i$  représentent le point de collecte et le point de livraison de la demandes  $i$ ,  $i = 1, 2, \dots, n$ . Soit  $W$  l'ensemble des noeuds à exception du noeud dépôt. Comme mentionné précédemment, le problème est le BGP multi-période avec l'ensemble de périodes indiqué par  $H$ . Dans le problème, le transporteur a une flotte finie de véhicules homogènes dont l'indice est donné par  $K = \{1, 2, \dots, VK\}$  où  $VK$  est le nombre maximum de véhicules. La capacité de chaque véhicule est indiquée par  $Q$  et la charge de chaque véhicule ne peut dépasser sa capacité. L'ensemble  $P = \{1, 2, \dots, n\}$  est l'ensemble des nœuds de reprise de toutes les demandes et l'ensemble des nœuds de livraison est donné par  $D = \{n + 1, \dots, 2n\}$ . Dans notre modèle, chaque demandes  $i$  a son noeud de ramassage  $i$  et son noeud de livraison  $i + n$ . La demande du noeud de ramassage de la demandes  $i$  est indiquée par  $d_i$ , tandis que la demande du noeud de livraison de la même demandes est indiquée par  $d_{i+n}$ , avec  $d_{i+n} = -d_i$ . Le noeud de livraison de chaque demande doit être visité après son noeud de ramassage sur le même itinéraire. L'ensemble de toutes les demandes est indiqué par  $R$ , où  $R = (\bigcup_{l \in H} R_r^l) \cup R_s$ .  $R_r^l$  est l'ensemble des demandes réservées qui doivent être traitées dans la période  $l$  et  $R_s$  l'ensemble des demandes sélectionnées. Chaque demandes sélective a une fenêtre de période de service et deux fenêtres de temps: La fenêtre de période de service détermine les périodes auxquelles la demande sélective peut être exécutée et les deux fenêtres de temps déterminent à quels moments de chaque période le noeud de ramassage et le noeud de livraison de la demande peuvent être visités par un véhicule qui sert la demande. Les demandes sélectives et réservées sont associées à deux fenêtres de temps alors que seules les demandes sélectives sont associées à une fenêtre de période de service (la période pendant laquelle chaque demandes réservée doit être servie est prédéfinie). La fenêtre temporelle du noeud de ramassage  $i$  et du noeud de livraison  $i + n$  de la demandes  $i$  sont indiqués respectivement par  $[e_i, l_i]$  et  $[e_{i+n}, l_{i+n}]$ . La fenêtre de période de service pour chaque demandes sélective  $i$  est représentée par  $[E_i, L_i]$ . Chaque demande réservée  $i \in R_r^l$  doit être servie dans sa période prédéfinie  $l$ ,  $l \in H$ . Le temps de déplacement et le coût de transport entre deux nœuds  $i$  et  $j$  sont supposés être les mêmes et sont indiqués par  $t_{ij}$  et  $c_{ij}$ , respectivement. La durée maximale de chaque itinéraire  $i$  est limitée par  $T$ . Le problème de la génération d'offres sur plusieurs périodes peut être formulé sous la forme d'un modèle de programmation linéaire à entiers mixtes. Dans le modèle, les paramètres  $BM_{ij} = l_j - e_i$  sont utilisés pour formuler linéairement les contraintes de fenêtre de temps tandis que  $C_i = Q + d_i$  est utilisé en contraintes de capacité. Les variables de décision du modèle comprennent les variables binaires,  $x_{ijkh}$  et  $y_{ikh}$  et les variables réelles  $U_{ikh}$  et  $CV_{ikh}$  définies comme suit:

$$x_{ijkh} = \begin{cases} 1 & \text{Si et seulement si dans la période } h, \text{ le véhicule } k \text{ visite directement le noeud } j \\ & \text{après le noeud } i. \\ 0 & \text{Dans le cas contraire.} \end{cases}$$

A.3. PROBLÈME DE GÉNÉRATION D'ENCHÈRES SUR PLUSIEURS PÉRIODES AVEC  
DEMANDES DE RAMASSAGE ET DE LIVRAISON, PROFITS ET DEMANDES RÉSERVÉES

---

$$y_{ikh} = \begin{cases} 1 & \text{Si et seulement si la demande } i \text{ est servie par le véhicule } k \text{ dans} \\ & \text{la période } h. \\ 0 & \text{Dans le cas contraire.} \end{cases}$$

$U_{ikh}$  = Temps d'arrivée du véhicule  $K$  au nœud  $i$  dans la période  $h$

$CV_{ikh}$  = Charge de véhicule  $k$  quand il quitte le nœud  $i$  dans la période  $h$

$$\max(\sum_{i \in R} \sum_{h \in H} \sum_{k \in K} p_i y_{ikh} - \sum_{h \in H} \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{ij} x_{ijk})$$

sujet à:

$$\sum_{j \in V; j \neq i} x_{jikh} - \sum_{j \in V; j \neq i} x_{ijk} = 0 \quad \forall i \in W, \forall k \in K, \forall h \in H \quad (\text{A.23})$$

$$\sum_{k \in K} \sum_{j \in P; j \neq 0} \sum_{h \in H} x_{0jkh} \geq 1 \quad (\text{A.24})$$

$$\sum_{k \in K} \sum_{i \in D; i \neq (2n+1)} \sum_{h \in H} x_{i(2n+1)kh} \geq 1 \quad (\text{A.25})$$

$$\sum_{k \in K} y_{ikh} = 1 \quad \forall l \in H, \forall i \in R_r^l \quad (\text{A.26})$$

$$\sum_{h \in [E_i, L_i]} \sum_{k \in K} y_{ikh} \leq 1 \quad \forall i \in R_s \quad (\text{A.27})$$

$$\sum_{j \in V; j \neq i, 0} x_{j(n+i)kh} = y_{ikh} \quad \forall i \in P, k \in K, \forall h \in H \quad (\text{A.28})$$

$$\sum_{j \in V; j \neq i, 2n+1} x_{j(i+n)kh} = y_{ikh} \quad \forall i \in P, k \in K, \forall h \in H \quad (\text{A.29})$$

$$U_{ikh} + t_{i,(n+i)} + s_i \leq U_{(n+i),kh} \quad \forall i \in P, k \in K, \forall h \in H \quad (\text{A.30})$$

$$U_{jkh} \geq U_{ikh} + t_{ij} + s_i - BM_{ij}(1 - x_{ijkh}) \quad \forall i \in V, k \in K, \forall h \in H, \forall j \in V \quad (\text{A.31})$$

$$e_i y_{ikh} \leq U_{ikh} \leq l_i + (1 - y_{ikh}) \sum_{j \in V} BM_{ij} \quad \forall h \in H, \forall i \in V, \forall k \in K \quad (\text{A.32})$$

$$U_{ikh} + t_{ij} + s_i - BM_{ij}(1 - x_{ijkh}) \leq T \quad \forall i \in V, k \in K, \forall h \in H, \forall j \in \{0\} \quad (\text{A.33})$$

$$CV_{ikh} + d_i - C_i(1 - x_{ijkh}) \leq CV_{jkh} \quad \forall i \in V, k \in K, \forall h \in H, \forall j \in V \quad (\text{A.34})$$

$$\max\{0, d_i\} \leq CV_{ikm} \leq \min\{Q, Q + d_i\} \quad \forall i \in V, k \in K, \forall h \in H \quad (\text{A.35})$$

$$x_{ijkh} \in \{0, 1\} \quad \forall i, j \in V, \forall k \in K, \forall h \in H \quad (\text{A.36})$$

$$U_{ikh} \geq 0 \quad \forall i \in V, \forall k \in K, \forall h \in H \quad (\text{A.37})$$

$$CV_{ikh} \geq 0 \quad \forall i \in V, \forall k \in K, \forall h \in H \quad (\text{A.38})$$

$$y_{ikh} = 0 \quad \text{pour tout } i \in R_s, \forall k \in K, \text{ pour tout } h \notin (E_i, L_i) \quad \text{and} \quad y_{ikh} = 0 \quad \text{pour tout } i \in R_{r,l}, l \neq h \quad (\text{A.39})$$

La fonction objective représente le bénéfice total du transporteur, qui est égal à la différence entre le total des paiements des demandes de service dans toutes les périodes et le coût total du transport. La contrainte (A.23) garantit que lorsqu'un véhicule arrive à un nœud au cours d'une période, il doit quitter ce nœud au cours de la même période. Les contraintes (A.24) et (A.25) signifient que chaque véhicule quitte son dépôt au cours d'une période doit y retourner au cours de la même période. L'équation (A.26) implique que chaque demande réservée doit être servie dans sa période prédéterminée. L'équation (A.27) indique que chaque demande sélective peut être servie pendant une période comprise dans sa fenêtre de période de service ou ne pas être servie. Les contraintes (A.28) et (A.29) garantissent que si une demande est servie dans une période, son nœud de livraison doit être visité après son nœud de ramassage avec le même véhicule dans la même période. Les équations (A.30)-(A.33) précisent les contraintes de fenêtre de temps sur les nœuds de ramassage et de livraison de chaque demande et la contrainte sur la durée maximale de chaque route. Les contraintes (A.34) et (A.35) assurent des contraintes de capacité des véhicules. Les équations (A.36)-(A.39) décrivent les variables.

#### **A.4 Problème de génération d'enchères sur plusieurs périodes avec prise en compte des délais de livraison**

Dans les entreprises, en plus de minimiser tous les coûts, l'expéditeur est intéressé par la réduction du délai de livraison. La réduction des délais de livraison présente deux avantages. Dans un premier temps, des délais de livraison plus courts rendent agile la gestion de la chaîne d'approvisionnement et la réponse aux commandes est exécutée plus rapidement. Ainsi, elle

améliore la capacité des expéditeurs à être concurrentiels dans l'industrie. Deuxièmement, l'avantage d'un délai de livraison plus court réside dans le choix des transporteurs en fonction des facteurs géographiques. Cela signifie que les expéditeurs distribuent les volumes de fret à un transporteur géographiquement [48]. En fait, en considérant le délai de livraison court, le commissaire-priseur s'implique aussi dans d'autres attributs non seulement économiques. Dans de nombreuses situations d'acquisition, bien que le coût soit un attribut important dans l'évaluation des commandes, les transporteurs se soucient des spécifications de qualité lorsqu'ils estiment les commandes soumises par les fournisseurs [26].

Compte tenu de l'importance du délai de livraison dans BGP, de son application dans les problèmes réels de transport et du fait que cette question n'a pas encore été abordée dans la littérature, nous nous concentrons sur ce sujet dans la présent section. En effet, dans ce dernier, nous avons envisagé l'enchère combinatoire pour la collaboration entre transporteur, chaque transporteur détermine les demandes (échangeables) de soumission (les demandes échangeables à sélectionner) en résolvant un BGP. Après avoir reçu les offres de tous les transporteurs, le commissaire-priseur (une plate-forme d'enchères) détermine les transporteurs gagnants et leurs offres gagnantes. Concernant le commissaire-priseur, il a deux façons d'évaluer chaque offre des transporteurs. L'une consiste à évaluer une offre en fonction de la notation qui est définie comme le coût et le délai de livraison, comme dans l'enchère de notation multi-attributs. L'autre façon d'évaluer une offre est en fonction de ses multiples attributs (dans notre cas, un attribut est le coût, l'autre attribut est le délai de livraison). Par conséquent, chaque transporteur a également deux façons de présenter son offre, l'une consiste à présenter une note (coût et délai de livraison) et l'autre à présenter les deux attributs d'une offre (coût et délai de livraison).

Par conséquent, dans la présent section, nous examinons les deux points de vue sur le problème du transport collaboratif. La première approche est une méthode des objectifs pondérés, c'est-à-dire que nous attribuons un poids à chaque objectif et transformons ensuite deux objectifs en un seul objectif pondéré. L'autre méthode est une méthode bi-objective qui est traitée avec des fonctions objectives séparées sans les transformer en une seule fonction objective. Par conséquent, deux méthodes de traitement avec les deux objectifs supposés conduisent à deux problèmes: le BGP multi-période à objectif unique avec délai de livraison et le BGP multi-période bi-objectif avec délai de livraison.

#### A.4.1 Description et modèle mathématique

Le problème considéré dans ce section est un problème de planification du transport avec les demandes de ramassage et de livraison, les fenêtres de de temps et de période sur un graphique dirigé  $G = (V, E)$ , où  $V$  est l'ensemble des nœuds et  $E$  est l'ensemble des arcs, avec les spécifications suivantes: Chaque demande  $i$  a deux points, un point de ramassage  $i$  et un point de livraison  $n + i$ , où  $n$  représente le nombre de demandes. Aussi, la demande du point de ramassage est égale à la demande du point de livraison. Le nœud de livraison de chaque demande

**Table A.3:** Les paramètres du modèle mathématique présenté.

Symbols	Explication
$V = \{0, \dots, 2n + 1\}$	Ensemble de tous les nœuds et comprend tous les nœuds de ramassage, de livraison et de dépôt de transporteur. Le $n$ signifie le nombre de demandes, 0 et $2n + 1$ indiquent le dépôt du transporteur.
$P = \{1, 2, \dots, n\}$	Ensemble des nœuds de ramassage de toutes les demandes.
$D = \{n + 1, \dots, 2n\}$	Ensemble des noeuds de livraison.
$W$	Ensemble de tous les nœuds de livraison et ramassage.
$H$	Ensemble des périodes de l'horizon de planification.
$K = \{1, 2, \dots, VK\}$	Ensemble fini de véhicules homogènes où $VK$ est le nombre maximum de véhicules.
$Q$	Capacité de chaque véhicule.
$R_l^r$	Demandes réservées et qui doivent être servies dans la période $l \in H$ .
$R_s = (\cup_{l \in H} R_l^s)$	Ensemble de demandes sélectives.
$R = (\cup_{l \in H} R_l^r)$	Ensemble de toutes les demandes.
$R_l^s$	Demande sélective $s$ dans la période $l$ .
$[e_i, l_i]$	Fenêtres de temps du point de ramassage $i$ .
$[e_{i+n}, l_{i+n}]$	Fenêtres de temps du point de livraison du point de ramassage $i$ .
$[E_i, L_i]$	Fenêtre de période de la demande sélective $i$ .
$t_{ij}$	Temps de déplacement entre deux nœuds $i$ et $j$ .
$c_{ij}$	cout de déplacement entre le noeud $i$ et $j$ $i$ and $j$ .
$T$	Durée maximale de chaque route.
$BM_{ij} = l_j - e_i$	Formuler linéairement les contraintes de fenêtre temporelle si $l_j$ est la limite supérieure de la fenêtre de temps pour le nœud $j$ et $e_i$ est la limite inférieure de la fenêtre de temps pour le nœud $i$ .
$CV_i = Q + d_i$	Symbole utilisé pour formuler linéairement les contraintes de capacité du véhicule.
$d_i$	Demande du nœud ramassage de la demande $i$ .
$d_{i+n} = -d_i$	Demande du point de livraison si la demande du point de ramassage est $d_i$ .
$\alpha_i$	Pénalité du délai de livraison pour le client $i$ .

**Table A.4:** Les variables de décision du modèle considéré.

Symbols	Explication
$x_{ijkh} = 1$	Si et seulement si le véhicule $k$ visite directement le nœud $j$ après le nœud $i$ dans la période $h$ .
$y_{ikh} = 1$	Si et seulement si la demande $i$ est servie par le véhicule $k$ dans la période $h$ .
$U_{ikh}$	Heure d'arrivée du véhicule $k$ au nœud $i$ dans la période $h$ .
$CV_{ikh}$	Capacité du véhicule $k$ quand il quitte le nœud $i$ dans la période $h$ .
$y'_{ikh}$	Variable auxiliaire utilisant dans la linéarisation d'un modèle.

doit être visité après son nœud de ramassage sur le même itinéraire. Le nombre de clients, leur localisation, le type de véhicule et leur taille sont prédéfinis. Il y a d'autres hypothèses

A.4. PROBLÈME DE GÉNÉRATION D'ENCHÈRES SUR PLUSIEURS PÉRIODES AVEC  
PRISE EN COMPTE DES DÉLAIS DE LIVRAISON

---

à ce problème comme le cas où la quantité de ramassage/livraison est inférieure à la capacité du véhicule et tous les véhicules sont homogènes. Le coût de la satisfaction de la demande des clients dépend de la distance totale parcourue par les véhicules. Chaque demande sélective a une fenêtre de période de service et deux fenêtres de temps. La fenêtre de période de service définit les périodes pendant lesquelles chaque demande sélective peut être traitée et les deux fenêtres de temps déterminent à quels moments de chaque période le nœud de ramassage et le nœud de livraison de la demande peuvent être visités par un véhicule qui dessert cette demande, par exemple les fenêtres matin et après-midi séparées par une pause déjeuner. Les demandes sélectives et réservées sont associées à deux fenêtres de temps, alors que seules les demandes sélectives sont associées à une fenêtre de période de service (la période pendant laquelle chaque demande réservée doit être servie est prédéfinie). Si une demande sélective est servie, la période la plus appropriée pour la servir devrait être déterminée.

Le problème de la génération d'offres sur plusieurs périodes peut être formulé sous la forme d'un modèle de programmation linéaire à entiers mixtes. Le but du problème est de répondre à toutes les demandes réservées dans leur période prédéfinie et de déterminer quelles demandes sélectives doivent être traitées pour augmenter le bénéfice total. Les paramètres du problème sont énumérés dans le Tableau A.3, tandis que les variables de décision utilisées sont présentées dans le Tableau A.4.

$\max (\sum_{i \in R} \sum_{h \in H} \sum_{k \in K} p_i y_{ikh} - \sum_{h \in H} \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{ij} x_{ijk} - \sum_{j \in R_s} \alpha_j (\sum_{h \in [E_i, L_i]} \sum_{k \in K} y_{jkh}))$   
sujet à:

$$\sum_{j \in V; j \neq i} x_{jik} - \sum_{j \in V; j \neq i} x_{ijk} = 0 \quad \forall i \in W, \forall k \in K, \forall h \in H \quad (\text{A.40})$$

$$\sum_{k \in K} \sum_{j \in P; j \neq 0} \sum_{h \in H} x_{0jkh} \geq 1 \quad (\text{A.41})$$

$$\sum_{k \in K} \sum_{i \in D; i \neq (2n+1)} \sum_{h \in H} x_{i(2n+1)kh} \geq 1 \quad (\text{A.42})$$

$$\sum_{k \in K} y_{ikh} = 1 \quad \forall h \in H, \forall i \in R_l^r \quad (\text{A.43})$$

$$\sum_{h \in [E_i, L_i]} \sum_{k \in K} y_{ikh} \leq 1 \quad \forall i \in R_s \quad (\text{A.44})$$

$$\sum_{j \in V; j \neq i, 0} x_{j(n+i)kh} = y_{ikh} \quad \forall i \in P, k \in K, \forall h \in H \quad (\text{A.45})$$

$$\sum_{j \in V, j \neq i, 2n+1} x_{j(i+n)kh} = y_{ikh} \quad \forall i \in P, k \in K, \forall h \in H \quad (\text{A.46})$$

$$U_{ikh} + t_{i,(n+i)} + s_i \leq U_{(n+i),kh} \quad \forall i \in P, k \in K, \forall h \in H \quad (\text{A.47})$$

$$U_{jkh} \geq U_{ikh} + t_{ij} + s_i - BM_{ij}(1 - x_{ijkh}) \quad \forall i \in V, k \in K, \forall h \in H, \forall j \in V \quad (\text{A.48})$$

$$e_i y_{ikh} \leq U_{ikh} \leq l_i + (1 - y_{ikh}) \sum_{j \in V} BM_{ij} \quad \forall h \in H, \forall i \in V, \forall k \in K \quad (\text{A.49})$$

$$U_{ikh} + t_{ij} + s_i - BM_{ij}(1 - x_{ijkh}) \leq T \quad \forall i \in V, k \in K, \forall h \in H, \forall j \in \{0\} \quad (\text{A.50})$$

$$CV_{ikh} + d_i - Q(1 - x_{ijkh}) \leq CV_{jkh} \quad \forall i \in V, k \in K, \forall h \in H, \forall j \in V \quad (\text{A.51})$$

$$\max\{0, d_i\} \leq CV_{ikm} \leq \min\{Q, d_i\} \quad \forall i \in V, k \in K, \forall h \in H \quad (\text{A.52})$$

$$x_{ijkh} \in \{0, 1\} \quad \forall i, j \in V, \forall k \in K, \forall h \in H \quad (\text{A.53})$$

$$U_{ikh} \geq 0 \quad \forall i \in V, \forall k \in K, \forall h \in H \quad (\text{A.54})$$

$$CV_{ikh} \geq 0 \quad \forall i \in V, \forall k \in K, \forall h \in H \quad (\text{A.55})$$

$$y_{ikh} = 0 \quad \text{pour tout } i \in R_s, \forall k \in K, \text{ pour tout } h \notin (E_i, L_i) \quad \text{and} \quad y_{ikh} = 0 \quad \text{pour tout } i \in R_{rl, l \neq h} \quad (\text{A.56})$$

$$y'_{jkh} \geq h \times y_{jkh} - E_j \quad \forall j \in V, \forall k \in K, \forall h \in H \quad (\text{A.57})$$

$$y'_{jkh} \geq 0 \quad \forall j \in V, \forall k \in K, \forall h \in H \quad (\text{A.58})$$

La fonction objectif correspond au bénéfice total du transporteur, c'est-à-dire à la différence entre les paiement totaux des demandes de service dans toutes les périodes, le coût total du transport et le délai de livraison. Le score considéré, dans la fonction d'objectif unique, est la pénalité du délai de livraison. La contrainte (A.40) garantit que lorsqu'un véhicule arrive à un nœud dans une période, il doit partir dans la même période. Les contraintes (A.41) et (A.42) signifient que chaque véhicule quitte son dépôt dans une période doit y retourner dans la même période. L'équation (A.43) implique que chaque demande réservée doit être servie dans sa période



prédéterminée. L'équation (A.44) indique que chaque demande sélective peut être traitée au cours d'une période de sa fenêtre de service ou ne pas l'être. Les contraintes (A.45) et (A.46) garantissent que si une demande est servie dans une période, son nœud de livraison doit être visité après son nœud de ramassage avec le même véhicule dans la même période. Les équations (A.47)-(A.50) précisent les contraintes de fenêtre de temps sur les nœuds de ramassage et de livraison de chaque demande et la contrainte sur la durée maximale de chaque route. Les contraintes (A.51) et (A.52) contrôlent la capacité du véhicule. Les équations (A.53)-(5.56) définissent toutes les variables. Pour linéariser la dernière composante de la fonction objectif, deux contraintes (5.57) et (5.58) sont utilisées.

Lorsque les expéditeurs se préoccupent surtout du temps de transit (délai de livraison), un temps de transit plus court permet d'offrir un meilleur service de transport. Il est possible de considérer un problème bi-objectif pour maximiser, d'une part, les profits et minimiser d'autre part le temps de transit et pour évaluer l'effet de la minimisation du délai de livraison sur le profit maximal. Cela signifie que l'une des fonctions objectives est la maximisation du profit et l'autre est la minimisation du délai de livraison. Par conséquent, dans le modèle précédent, le délai de livraison sera considéré comme une deuxième fonction objectif.

Première fonction objectif:

$$\max (\sum_{i \in R} \sum_{h \in H} \sum_{k \in K} p_i y_{ikh} - \sum_{h \in H} \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{ij} x_{ijkm})$$

Deuxième fonction objectif:

$$\min \sum_{j \in R_s} \alpha_j (\sum_{h \in [E_i, L_i]} \sum_{k \in K} y_{jkh})$$

## A.5 Conclusion

Au cours des dernières années, la logistique collaborative et surtout le transport collaboratif entre transporteurs sont apparus comme une technique bénéfique pour les petites entreprises et aux transporteurs de fret de taille moyenne afin d'améliorer leur rentabilité en diminuant les trajets de retour à vide par le repositionnement des véhicules et en améliorant leur taux de remplissage. Dans cette thèse, nous nous avons discuté de la planification concertée du transport (CTP) découlant de la collaboration des transporteurs, en particulier de la collaboration des transporteurs dans le transport de chargement partiel (LTL).

Deux sujets clés sont souvent discutés pour la collaboration des transporteurs dans le transport de chargement partiel: le problème de la réaffectation des demandes et le problème de la post-collaboration. Ces problèmes influencent grandement les bénéfices du transporteur. Dans cette thèse, nous nous sommes concentrés sur le problème de la réaffectation des demandes dans le cadre d'une collaboration centralisée entre plusieurs transporteurs et nous avons proposé une approche de collaboration entre transporteurs fondée sur les enchères pour le PTP avec

profits. Dans ce domaine, trois principaux sous-problèmes sont pris en compte: CTP centralisé multi-opérateurs avec demandes de ramassage et de livraison avec fenêtres de temps et demandes réservées, le problème de génération périodique d'enchères et (BGP) avec délai de livraison dans la collaboration entre opérateurs via un échange combinatoire.

Au début, nous avons étudié le contexte de la recherche dans le Chapitre 1 et examiné l'état d'avancement de la recherche sur la planification concertée des transports dans le Chapitre 2. Puis, dans le Chapitre 3, nous nous sommes concentrés sur le CTP centralisé multi-transporteur avec ramassage et livraison avec fenêtres de temps et les demandes réservées (PDPTWR) qui est nouveau défi dans la thématique des tournées de véhicules.

Le PDPTWR centralisé multi-transporteur implique des demandes échangeables, qui peuvent être échangées entre les transporteurs au sein d'une coalition, tandis que chaque transporteur conserve les demandes réservées pour lui-même. Après la présentation d'un modèle de programmation mathématique, trois algorithmes métaheuristiques ont été développés comme méthodes de solution: une recherche adaptative de grand voisinage (ALNS), un algorithme génétique hybride et recuit simulé (GASA) et une recherche tabu améliorée (ITS). Les résultats expérimentaux numériques montrent que nos algorithmes donnent des résultats prometteurs par rapport aux solveurs commerciaux CPLEX. De plus, l'algorithme ALNS surpasse les deux autres algorithmes non seulement pour trouver les meilleures solutions mais aussi parce que son temps de calcul est meilleur que les autres.

Dans le Chapitre 4, un problème de génération d'offres sur plusieurs périodes (BGP) pour un transporteur a été envisagé. Dans ce problème, il y a deux types de demandes différentes, les demandes réservées du transporteur et les demandes sélectives. Chaque demande a un point de ramassage et de livraison avec deux fenêtres de temps. Une fenêtre de période spécifie l'intervalle dans lequel la demande doit être exécutée, alors qu'une fenêtre temporelle indique l'intervalle horaire affectée au point de ramassage et au point de livraison. De plus, chaque demande sélective est associée à un profit qui est le coût de service de la demande fournie par un expéditeur. Une approche hybride combinant un algorithme génétique et un recuit simulé (GASA) a été proposée pour résoudre ce problème. A des fins de comparaison, nous avons utilisé l'algorithme d'optimisation fondé sur la biogéographie (BBO) qui est un algorithme basé sur la population dont les spécifications sont proches à l'GA. Les résultats numériques ont démontré qu'en utilisant CPLEX, les algorithmes proposés peuvent trouver une bonne solution réalisable dans un temps de calcul raisonnable pour des cas importants.

Dans les entreprises, en plus de minimiser tous les coûts, un expéditeur est intéressé par la réduction des délais de livraison. La réduction des délais de livraison présente deux avantages. Dans un premier temps, des délais de livraison plus courts permettent la gestion agile de la chaîne d'approvisionnement et la réponse aux commandes est exécutée plus rapidement. Ainsi, il améliore la capacité des expéditeurs à soutenir la concurrence dans l'industrie. Le deuxième avantage d'un délai de livraison plus court est le choix des transporteurs en fonction des facteurs

géographiques. Cela signifie que les expéditeurs distribuent la marchandise à un transporteur suivant le plan géographique. Finalement, en tenant compte du délai de livraison court, le commissaire-priseur s'implique aussi dans d'autres attributs qui ne sont pas seulement des coûts. Dans de nombreuses situations d'acquisition, bien que le coût soit un facteur important pour évaluer les commandes, les transporteurs se soucient des spécifications de qualité lors de l'estimation de l'offre des commandes passées par les fournisseurs. De ce fait, dans le Chapitre 5, nous avons abordé dans le Chapitre 5 la question du délai de livraison dans le cadre de l'approvisionnement en services de transport, qui est effectué par les expéditeurs au moyen d'enchères pour sélectionner le service de transport auprès des transporteurs à l'étape de la planification. De nombreux expéditeurs utilisent ceci dans leurs tâches de transport quotidiennes dans le monde réel; le service de transport est meilleur lorsque le délai de livraison est plus court. Par conséquent, nous avons envisagé deux façons d'inclure le délai de livraison dans le problème. Premièrement, un problème bi-objectif a été envisagé, où le premier objectif est la maximisation de tous les profits tandis que l'autre est la minimisation du délai de livraison. Après avoir présenté le modèle mathématique, nous l'avons résolu par NSGAI et NPGA et les avons comparés par des métriques bien connues. Deuxièmement, nous avons supposé que le délai d'exécution du problème était inclus comme attribut dans la fonction objectif en utilisant certaines méthodes de prise de décision multicritères (MCDM). De ce fait, GASA et ITS ont été utilisés pour résoudre le modèle. Les résultats indiquent que les deux peuvent obtenir une solution réalisable dans des cas de large taille, au contraire, CPLEX n'a pas pu le faire.

En résumé, les principales contributions de cette thèse comprennent trois nouveaux concepts apparus dans la planification de la collaboration entre transporteurs, la présentation de modèles mathématiques pour eux et le développement de méthodes de solution efficaces: 1) le PDPTWR centralisé à transporteurs multiples, un nouveau problème de tournées des véhicules à plusieurs dépôts pour attribuer les demandes d'échange à un transporteur approprié. 2) le PDPTWR multi-période via BGP: un nouveau problème de routage multi-période des véhicules pour identifier les demandes rentables sur une période. 3) le PDPTWR multi-période via BGP avec minimisation de délais de livraison. Ce dernier point est un nouveau problème de génération d'offres est apparu dans la collaboration des transporteurs LTL afin de réduire les délais de livraison et d'augmenter les profits.

## A.6 Perspectives

Bien que nous ayons suggéré un plan complet pour résoudre le problème de réattribution des demandes dans le cadre de la collaboration entre transporteurs de lots brisés, il reste encore d'autres études à réaliser pour élaborer les modèles et les solutions aux problèmes et les rendre plus appropriés dans les environnements réalistes de collaboration entre transporteurs.

Premièrement, concernant le (Problème de ramassage et de livraison avec plusieurs trans-

porteurs, fenêtres de temps, demandes réservées et demandes échangeables) centralisé à transporteurs multiples, nous avons supposé que toutes les demandes sont déterministes alors que dans un environnement réaliste, les demandes peuvent être stochastiques. Ainsi, dans l'étude future, nous pouvons nous concentrer sur les demandes stochastiques, en particulier les demandes échangeables peuvent être stochastiques.

Deuxièmement, pour le cas de plusieurs périodes sur problème de génération d'enchères avec ramassage et de livraison problème de fenêtres de temps, de profits et de demandes réservées et les délais de livraison, il sera intéressant de traiter les demandes sélectives classifiées pour lesquelles chaque période: si une demande sélective dans chaque catégorie ne peut être servie, il ne sera pas possible de servir toutes les demandes sélectives dans cette même catégorie. Dans un environnement de collaboration réaliste, il est parfois raisonnable d'ignorer un groupe de demandes pour réduire les coûts.

Troisièmement, nous pouvons considérer l'effet de la réduction des émissions de CO<sub>2</sub> à notre modèle dans le troisième problème. Bien qu'il soit important de réduire le délai de livraison pour satisfaire le client, l'émission de CO<sub>2</sub> et l'augmentation des gaz à effet de serre dans l'environnement est une contrainte critique ces dernières années, car elle est devenue un problème majeur dans le monde et la plupart des pays développés prennent en compte des plans pour réduire les effets des émissions de CO<sub>2</sub> dans l'atmosphère. Par conséquent, notre modèle pourrait inclure comme objectifs la réduction des émissions de CO<sub>2</sub> en plus du coût et du profit.

Enfin, les approches d'allocation des bénéfices n'ont pas été abordées dans cette thèse alors que ce sujet est nécessaire à la mise en œuvre de la logistiques collaborative. Dans la littérature, le problème de la réattribution des demandes et celui de l'affectation des bénéfices sont souvent manipulés indépendamment. Il serait préférable que les deux problèmes puissent être examinés ensemble afin d'obtenir de meilleurs résultats qu'auparavant.

## BIBLIOGRAPHY

- [1] Deniz Aksen, Onur Kaya, F Sibel Salman, and Özge Tüncel.  
An adaptive large neighborhood search algorithm for a selective and periodic inventory routing problem.  
*European Journal of Operational Research*, 239(2):413–426, 2014.
- [2] Omar Al Jadaan, Lakishmi Rajamani, and CR Rao.  
Non-dominated ranked genetic algorithm for solving multi-objective optimization problems: Nrga.  
*Journal of Theoretical & Applied Information Technology*, 4(1), 2008.
- [3] Omar Al Jadaan, CR Rao, and Lakishmi Rajamani.  
Parametric study to enhance genetic algorithm performance, using ranked based roulette wheel selection method.  
In *International conference on multidisciplinary information sciences and technology (In-SciT2006)*, volume 2, pages 274–278, 2006.
- [4] Claudia Archetti, Elena Fernández, and Diana L. Huerta-Muñoz.  
The Flexible Periodic Vehicle Routing Problem.  
*Computers and Operations Research*, 85:58–70, 2017.
- [5] Claudia Archetti, Ola Jabali, and M. Grazia Speranza.  
Multi-period Vehicle Routing Problem with Due dates.  
*Computers and Operations Research*, 61:122–134, 2015.
- [6] Heechul Bae and Ilkyeong Moon.  
Multi-depot vehicle routing problem with time windows considering delivery and installation vehicles.  
*Applied Mathematical Modelling*, 40(13-14):6536–6549, 2016.
- [7] Raúl Baños, Julio Ortega, Consolación Gil, Antonio L Márquez, and Francisco De Toro.  
A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows.  
*Computers & Industrial Engineering*, 65(2):286–296, 2013.
- [8] Mark Barratt.

## BIBLIOGRAPHY

---

- Understanding the meaning of collaboration in the supply chain.  
*Supply Chain Management: an international journal*, 9(1):30–42, 2004.
- [9] Haoxun Chen Bo Dai.  
Mathematical model and solution approach for collaborative logistics in less than truckload (LTL) transportation.  
*2009 International Conference on Computers & Industrial Engineering*, pages 779–784, 2009.
- [10] Sylvain Boussier, Dominique Feillet, and Michel Gendreau.  
An exact algorithm for team orienteering problems.  
*4or*, 5(3):211–230, 2007.
- [11] Ann M Brewer, Kenneth J Button, and David A Hensher.  
*Handbook of logistics and supply-chain management*.  
Emerald Group Publishing Limited, 2008.
- [12] Tobias Buer.  
An exact and two heuristic strategies for truthful bidding in combinatorial transport auctions.  
*arXiv preprint arXiv:1406.1928*, 2014.
- [13] Steven E. Butt and Tom M. Cavalier.  
A heuristic for the multiple tour maximum collection problem.  
*Computers and Operations Research*, 21(1):101–111, 1994.
- [14] Roberto Cantu-Funes, M. Angélica Salazar-Aguilar, and Vincent Boyer.  
Multi-Depot Periodic Vehicle Routing Problem with Due Dates and Time Windows.  
*Journal of the Operational Research Society*, 69(2):1–12, 2017.
- [15] Fabian Cevallos and Fang Zhao.  
Minimizing transfer times in public transit network with genetic algorithm.  
*Transportation Research Record: Journal of the Transportation Research Board*, (1971):74–79, 2006.
- [16] I-Ming Chao, Bruce L. Golden, and Edward A. Wasil.  
The team orienteering problem.  
*European Journal of Operational Research*, 88(3):464–474, 1996.
- [17] Haoxun Chen.  
Combinatorial clock-proxy exchange for carrier collaboration in less than truck load transportation.  
*Transportation Research Part E: Logistics and Transportation Review*, 91:152–172, 2016.

- 
- [18] Giorgio Chiandussi, Marco Codegone, Simone Ferrero, and Federico Erminio Varesio. Comparison of multi-objective optimization methodologies for engineering applications. *Computers & Mathematics with Applications*, 63(5):912–942, 2012.
- [19] S Coene, A Arnout, and F C R Spieksma. On a periodic vehicle routing problem. *Journal of the Operational Research Society*, 61(12):1719–1728, 2010.
- [20] Yann Collette and Patrick Siarry. *Multiobjective optimization: principles and case studies*. Springer Science & Business Media, 2013.
- [21] Jean-François Cordeau, Gilbert Laporte, and Anne Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational research society*, 52(8):928–936, 2001.
- [22] Frans Cruijssen, Martine Cools, and Wout Dullaert. Horizontal cooperation in logistics: Opportunities and impediments. *Transportation Research Part E: Logistics and Transportation Review*, 43(2):129–142, 2007.
- [23] Bo Dai and Haoxun Chen. A multi-agent and auction-based framework and approach for carrier collaboration. *Logistics Research*, 3(2-3):101–120, 2011.
- [24] Bo Dai, Haoxun Chen, and Genke Yang. Price-setting based combinatorial auction approach for carrier collaboration with pickup and delivery requests. *Operational Research*, 14(3):361–386, 2014.
- [25] Sophie D’Amours and Mikael Rönnqvist. Issues in collaborative logistics. In *Energy, natural resources and environmental economics*, pages 395–409. Springer, 2010.
- [26] Krishnendu Ghosh Dastidar. Scoring auctions: A brief survey. *Studies in Microeconomics*, 2(1):35–48, 2014.
- [27] Kalyanmoy Deb and Deb Kalyanmoy. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 2001.
- [28] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

## BIBLIOGRAPHY

---

- [29] Yuquan Du, Qiang Meng, and Shuaian Wang.  
Mathematically calculating the transit time of cargo through a liner shipping network with various trans-shipment policies.  
*Maritime Policy & Management*, 44(2):248–270, 2017.
- [30] Ozlem Ergun, Gultekin Kuyzu, and Martin Savelsbergh.  
The lane covering problem.  
*Manuscript*, 2003.
- [31] Ozlem Ergun, Gultekin Kuyzu, and Martin Savelsbergh.  
Reducing truckload transportation costs through collaboration.  
*Transportation science*, 41(2):206–221, 2007.
- [32] Özlem Ergun, Gültekin Kuyzu, and Martin Savelsbergh.  
Shipper collaboration.  
*Computers & Operations Research*, 34(6):1551–1560, 2007.
- [33] Terry L Esper and Lisa R Williams.  
The value of collaborative transportation management (ctm): its relationship to cpfr and information technology.  
*Transportation journal*, pages 55–65, 2003.
- [34] Dominique Feillet, Pierre Dejax, and Michel Gendreau.  
Traveling Salesman Problems with Profits.  
*Transportation Science*, 39(2):188–205, 2005.
- [35] Margaretha Gansterer, Murat Küçüktepe, and Richard F. Hartl.  
The multi-vehicle profitable pickup and delivery problem.  
*OR Spectrum*, 39(1):303–319, 2017.
- [36] Abel Garcia-Najera and John A Bullinaria.  
An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows.  
*Computers & Operations Research*, 38(1):287–300, 2011.
- [37] M Gendreau, G Laporte, and F Semet.  
A tabu search heuristic for the undirected selective traveling salesman problem.  
*European Journal of Operational Research*, 106(2):539–545, 1998.
- [38] Hadi Ghaderi, Wout Dullaert, and Walther Ploos Van Amstel.  
Reducing lead-times and lead-time variance in cooperative distribution networks.  
*International Journal of Shipping and Transport Logistics*, 8(1):51–65, 2016.



- [39] Fred Glover and Manuel Laguna.  
Tabu search.  
In *Handbook of combinatorial optimization*, pages 2093–2229. Springer, 1998.
- [40] Bruce L Golden, Subramanian Raghavan, and Edward A Wasil.  
*The vehicle routing problem: latest advances and new challenges*, volume 43.  
Springer Science & Business Media, 2008.
- [41] Mario Guajardo and Mikael Rönnqvist.  
A review on cost allocation methods in collaborative transportation.  
*International transactions in operational research*, 23(3):371–392, 2016.
- [42] Vahid Hajipour, Reza Zanjirani Farahani, and Parviz Fattahi.  
Bi-objective vibration damping optimization for congested location–pricing problem.  
*Computers & Operations Research*, 70:87–100, 2016.
- [43] Stephanus Daniel Handoko and Hoong Chuin Lau.  
Enabling carrier collaboration via order sharing double auction: a singapore urban logistics perspective.  
*Transportation Research Procedia*, 12:777–786, 2016.
- [44] Stephanus Daniel Handoko, Duc Thien Nguyen, and Hoong Chuin Lau.  
An auction mechanism for the last-mile deliveries via urban consolidation centre.  
In *Automation Science and Engineering (CASE), 2014 IEEE International Conference on*, pages 607–612. IEEE, 2014.
- [45] Salvador Hernández and Srinivas Peeta.  
A carrier collaboration problem for less-than-truckload carriers: characteristics and carrier collaboration model.  
*Transportmetrica A: Transport Science*, 10(4):327–349, 2014.
- [46] Salvador Hernández, Srinivas Peeta, and George Kalafatas.  
A less-than-truckload carrier collaboration planning problem under dynamic capacities.  
*Transportation Research Part E: Logistics and Transportation Review*, 47(6):933–946, 2011.
- [47] John Henry Holland.  
*Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*.  
MIT press, 1992.
- [48] Qian Hu, Zhenzhen Zhang, and Andrew Lim.  
Transportation service procurement problem with transit time.  
*Transportation Research Part B: Methodological*, 86:19–36, 2016.

- [49] C-L Hwang and Abu Syed Md Masud.  
*Multiple objective decision making—methods and applications: a state-of-the-art survey*, volume 164.  
Springer Science & Business Media, 2012.
- [50] OMAR AL JADAAN, LAKSHMI RAJAMANI, and CR Rao.  
Non-dominated ranked genetic algorithm for solving constrained multi-objective optimization problems.  
*Journal of Theoretical & Applied Information Technology*, 5(5), 2009.
- [51] Divya Jayakumar Nair, Hanna Grzybowska, David Rey, and Vinayak Dixit.  
Food Rescue and Delivery.  
*Transportation Research Record: Journal of the Transportation Research Board*, 2548:81–89, 2016.
- [52] Hongmei Jia, Yang Li, Bo Dong, and Hongying Ya.  
An improved tabu search approach to vehicle routing problem.  
*Procedia-Social and Behavioral Sciences*, 96:1208–1217, 2013.
- [53] Voratas Kachitvichyanukul, Pandhapon Sombuntham, and Siwaporn Kunnapapdeelert.  
Two solution representations for solving multi-depot vehicle routing problem with multiple pickup and delivery requests via pso.  
*Computers & Industrial Engineering*, 89:125–136, 2015.
- [54] Shyam Prasad Kodali, Rajesh Kudikala, and Deb Kalyanmoy.  
Multi-objective optimization of surface grinding process using nsga ii.  
In *Emerging Trends in Engineering and Technology, 2008. ICETET'08. First International Conference on*, pages 763–767. IEEE, 2008.
- [55] Herbert Kopfer and Marta Anna Krajewska.  
Approaches for modelling and solving the integrated transportation and forwarding problem.  
*Produktions-und Logistikmanagement, Springer, Berlin Heidelberg New York*, pages 439–458, 2007.
- [56] Herbert Kopfer and Xin Wang.  
Combining vehicle routing with forwarding: extension of the vehicle routing problem by different types of sub-contraction.  
*Journal of Korean Institute of Industrial Engineers*, 35(1):1–14, 2009.
- [57] Marta Anna Krajewska and Herbert Kopfer.  
Transportation planning in freight forwarding companies. Tabu search algorithm for the integrated operational transportation planning problem.

- 
- European Journal of Operational Research*, 197(2):741–751, 2009.
- [58] Vijay Krishna.  
*Auction theory*.  
Academic press, 2009.
- [59] Roy H Kwon, Chi-Guhn Lee, and Zhong Ma.  
An integrated combinatorial auction mechanism for truckload transportation procurement.  
*University of Toronto*, <http://www.mie.utoronto.ca/labs/ilr/MultiRound.pdf>, Working Paper, 2005.
- [60] Nacima Labadie, Renata Mansini, Jan Melechovský, and Roberto Wolfler Calvo.  
The Team Orienteering Problem with Time Windows: An LP-based granular variable neighborhood search.  
*European Journal of Operational Research*, 220(1):15–27, 2012.
- [61] Gilbert Laporte and Silvano Martello.  
The selective travelling salesman problem.  
*Discrete Applied Mathematics*, 26(2-3):193–207, 1990.
- [62] Hoong Chuin Lau, Shih Fen Cheng, Thin Yin Leong, Jong Han Park, and Zhengyi Zhao.  
Multi-period combinatorial auction mechanism for distributed resource allocation and scheduling.  
In *Intelligent Agent Technology, 2007. IAT'07. IEEE / WIC / ACM International Conference on*, pages 407–411. IEEE, 2007.
- [63] Chi-Guhn Lee, Roy H Kwon, and Zhong Ma.  
A carrier's optimal bid generation problem in combinatorial auctions for transportation procurement.  
*Transportation Research Part E: Logistics and Transportation Review*, 43(2):173–191, 2007.
- [64] Jian Li and Wenhua Lu.  
Full truckload vehicle routing problem with profits.  
*Journal of Traffic and Transportation Engineering (English Edition)*, 1(2):146–152, 2014.
- [65] Yuan Li, Haoxun Chen, and Christian Prins.  
Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests.  
*European Journal of Operational Research*, 252(1):27–38, 2016.
- [66] Ran Liu, Zhibin Jiang, Richard YK Fung, Feng Chen, and Xiao Liu.  
Two-phase heuristic algorithms for full truckloads multi-depot capacitated vehicle routing problem in carrier collaboration.

- Computers & Operations Research*, 37(5):950–959, 2010.
- [67] S Afshin Mansouri.  
A multi-objective genetic algorithm for mixed-model sequencing on jit assembly lines.  
*European Journal of Operational Research*, 167(3):696–716, 2005.
- [68] Evangelos Markakis and Amin Saberi.  
On the core of the multicommodity flow game.  
*Decision support systems*, 39(1):3–10, 2005.
- [69] Robert Mason, Chandra Lalwani, and Roger Boughton.  
Combining vertical and horizontal collaboration for transport optimisation.  
*Supply Chain Management: An International Journal*, 12(3):187–199, 2007.
- [70] Hadi Moradi, M Zandieh, and Iraj Mahdavi.  
Non-dominated ranked genetic algorithm for a multi-objective mixed-model assembly line sequencing problem.  
*International Journal of Production Research*, 49(12):3479–3499, 2011.
- [71] Seyed Mohsen Mousavi, Seyed Taghi Akhavan Niaki, Esmaeil Mehdizadeh, and Mohammad Reza Tavarroth.  
The capacitated multi-facility location–allocation problem with probabilistic customer location and demand: two hybrid meta-heuristic algorithms.  
*International Journal of Systems Science*, 44(10):1897–1912, 2013.
- [72] Seyed Mohsen Mousavi, Javad Sadeghi, Seyed Taghi Akhavan Niaki, and Madjid Tavana.  
A bi-objective inventory optimization model under inflation and discount using tuned pareto-based algorithms: Nsga-ii, nrga, and mopso.  
*Applied Soft Computing*, 43:57–72, 2016.
- [73] Amir Abbas Najafi, Seyed Taghi Akhavan Niaki, and Moslem Shahsavar.  
A parameter-tuned genetic algorithm for the resource investment problem with discounted cash flows and generalized precedence relations.  
*Computers & Operations Research*, 36(11):2994–3001, 2009.
- [74] Viet Hung Nguyen and Thi Thu Thuy Nguyen.  
Approximating the asymmetric profitable tour.  
*Electronic Notes in Discrete Mathematics*, 36(C):907–914, 2010.
- [75] Daichi Niizuma, Keiichiro Yasuda, and Atsushi Ishigame.  
Multipoint tabu search based on proximate optimality principle—application of parts concept.  
*IEEJ Transactions on Electrical and Electronic Engineering*, 2(6):635–642, 2007.

- [76] Okan Orsan Ozener, Ozlem Ergun, and Martin Savelsbergh.  
Collaboration for truckload carriers.  
In *INFORMS Annual Conference, San Diego, CA, 2009*.
- [77] Okan Örsan Özener, Özlem Ergun, and Martin Savelsbergh.  
Lane-exchange mechanisms for truckload carrier collaboration.  
*Transportation Science*, 45(1):1–17, 2011.
- [78] Seyed Hamid Reza Pasandideh, Seyed Taghi Akhavan Niaki, and Kobra Asadi.  
Bi-objective optimization of a multi-product multi-period three-echelon supply chain problem under uncertain environments: Nsga-ii and nrga.  
*Information Sciences*, 292:57–74, 2015.
- [79] Seyed Hamid Reza Pasandideh, Seyed Taghi Akhavan Niaki, and Vahid Hajipour.  
A multi-objective facility location model with batch arrivals: two parameter-tuned metaheuristic algorithms.  
*Journal of Intelligent Manufacturing*, 24(2):331–348, 2013.
- [80] GS Peace.  
Taguchi methods, a hands on approach, 1993.
- [81] David Pisinger and Stefan Ropke.  
A general heuristic for vehicle routing problems.  
*Computers & operations research*, 34(8):2403–2435, 2007.
- [82] Seyed Habib A. Rahmati and M. Zandieh.  
A new biogeography-based optimization (BBO) algorithm for the flexible job shop scheduling problem.  
*International Journal of Advanced Manufacturing Technology*, 58(9-12):1115–1129, 2012.
- [83] Seyed Habib A Rahmati, M Zandieh, and M Yazdani.  
Developing two multi-objective evolutionary algorithms for the multi-objective flexible job shop scheduling problem.  
*The International Journal of Advanced Manufacturing Technology*, 64(5-8):915–932, 2013.
- [84] Stefan Ropke and David Pisinger.  
An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows.  
*Transportation Science*, 40(4):455–472, 2006.
- [85] Tuomas Sandholm.  
An algorithm for optimal winner determination in combinatorial auctions.  
1999.

- [86] Christina Schmoltzi and Carl Marcus Wallenburg.  
Horizontal cooperations between logistics service providers: motives, structure, performance.  
*International Journal of Physical Distribution & Logistics Management*, 41(6):552–575, 2011.
- [87] N Semioshkina and G Voigt.  
An overview on taguchi method.  
*Journal of Radiation Research*, 47(2):11–18, 2006.
- [88] Paul Shaw.  
Using constraint programming and local search methods to solve vehicle routing problems.  
In *International conference on principles and practice of constraint programming*, pages 417–431. Springer, 1998.
- [89] Yoav Shoham and Kevin Leyton-Brown.  
*Multiagent systems: Algorithmic, game-theoretic, and logical foundations*.  
Cambridge University Press, 2008.
- [90] Dan Simon.  
Biogeography-based optimization.  
*IEEE Transactions on Evolutionary Computation*, 12(6):702–713, 2008.
- [91] Pawel Sitek and Jarosław Wikarek.  
Capacitated vehicle routing problem with pick-up and alternative delivery (cvrppad): model and implementation using hybrid approach.  
*Annals of Operations Research*, pages 1–21, 2017.
- [92] Jiongjiong Song and Amelia Regan.  
Combinatorial auctions for transportation service procurement: The carrier perspective.  
*Transportation Research Record: Journal of the Transportation Research Board*, (1833):40–46, 2003.
- [93] Jiongjiong Song and Amelia Regan.  
Approximation algorithms for the bid construction problem in combinatorial auctions for the procurement of freight transportation contracts.  
*Transportation Research Part B: Methodological*, 39(10):914–933, 2005.
- [94] R Tavakkoli-Moghaddam, A Makui, and Z Mazloomi.  
A new integrated mathematical model for a bi-objective multi-depot location-routing problem solved by a multi-objective scatter search algorithm.  
*Journal of Manufacturing Systems*, 29(2-3):111–119, 2010.

- [95] Fabien Tricoire, Martin Romauch, Karl F. Doerner, and Richard F. Hartl.  
Heuristics for the multi-period orienteering problem with multiple time windows.  
*Computers and Operations Research*, 37(2):351–367, 2010.
- [96] Resit Unal and Edwin B Dean.  
Taguchi approach to design optimization for quality and cost: an overview.  
1990.
- [97] Lotte Verdonck, AN Caris, Katrien Ramaekers, and Gerrit K Janssens.  
Collaborative logistics from the perspective of road transportation companies.  
*Transport Reviews*, 33(6):700–719, 2013.
- [98] Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins.  
Heuristics for multi-attribute vehicle routing problems: A survey and synthesis.  
*European Journal of Operational Research*, 231(1):1–21, 2013.
- [99] Chen Wang, Hoong Chuin Lau, and Yun Fong Lim.  
A rolling horizon auction mechanism and virtual pricing of shipping capacity for urban consolidation centers.  
In *International Conference on Computational Logistics*, pages 422–436. Springer, 2015.
- [100] Xin Wang and Herbert Kopfer.  
Increasing efficiency of freight carriers through collaborative transport planning: Chances and challenges.  
In *Dynamics and Sustainability in International Logistics and Supply Chain Management- Proceedings of the 6th German-Russian Logistics and SCM Workshop (DR-LOG 2011)*.  
Cuvillier Verlag, Göttingen, pages 41–50, 2011.
- [101] Xin Wang and Herbert Kopfer.  
Collaborative transportation planning of less-than-truckload freight.  
*OR spectrum*, 36(2):357–380, 2014.
- [102] Xin Wang, Herbert Kopfer, and Michel Gendreau.  
Operational transportation planning of freight forwarding companies in horizontal coalitions.  
*European Journal of Operational Research*, 237(3):1133–1141, 2014.
- [103] Xiubin Wang and Mu Xia.  
Combinatorial bid generation problem for transportation service procurement.  
*Transportation Research Record: Journal of the Transportation Research Board*, (1923):189–198, 2005.

## BIBLIOGRAPHY

---

- [104] Jiefeng Xu, Steve Y Chiu, and Fred Glover.  
Using tabu search to solve the steiner tree-star problem in telecommunications network design.  
*Telecommunication Systems*, 6(1):117–125, 1996.
- [105] Yusliza Yusoff, Mohd Salihin Ngadiman, and Azlan Mohd Zain.  
Overview of nsga-ii for optimizing machining process parameters.  
*Procedia Engineering*, 15:3978–3983, 2011.
- [106] Mengdi Zhang, Saurabh Pratap, George Q Huang, and Zhiheng Zhao.  
Optimal collaborative transportation service trading in b2b e-commerce logistics.  
*International Journal of Production Research*, 55(18):5485–5501, 2017.
- [107] Eckart Zitzler.  
*Evolutionary algorithms for multiobjective optimization: Methods and applications*, volume 63.  
Citeseer, 1999.



# Elham JELODARI MAMAGHANI

## Doctorat : Optimisation et Sûreté des Systèmes

### Année 2019

#### Etudes sur le transport collaboratif : modèles et métaheuristiques

Dans le domaine de la logistique collaborative, plusieurs transporteurs forment une alliance pour améliorer leurs opérations de transport et leur rentabilité en échangeant leurs demandes de transport. Dans cette thèse, nous concentrons sur la collaboration des transporteurs dans le transport de chargement partiel (LTL). Plus précisément, trois sous-problèmes de la planification collaborative entre transporteurs sont pris en compte : un problème de planification centralisée de multi-transporteurs avec ramassage et livraison, fenêtres de temps, demandes échangeables et demandes réservées, un problème de génération d'enchères à plusieurs périodes, un problème de ramassage et de livraison, fenêtres de temps, profits, demandes réservées et problème de génération d'enchères à plusieurs périodes avec prise en compte à la fois du coût et du délais de livraison.

Ces sous-problèmes jouent un rôle essentiel dans la planification collaborative de transport entre transporteurs, mais dans la littérature, aucune étude profonde n'a été effectuée sur eux. Nous avons présenté de nouveaux modèles de programmation mathématique pour ces problèmes et développé des heuristiques efficaces pour obtenir des solutions proches de leurs optimums dans un temps de calcul raisonnable. Ces heuristiques proposées sont plus efficaces que le solveur commercial, CPLEX, non seulement en termes de qualité de solution mais aussi en termes de temps de calcul.

**Mots clés :** transports routiers – recherche opérationnelle – métaheuristiques – transports urbains – transport durable – livraison de marchandises.

#### Studies on Collaborative Transportation: Models and Metaheuristics

In collaborative logistics, multiple carriers form an alliance to improve their transportation operations and profitability by exchanging their transportation requests. In this thesis, we focus on the carrier collaboration in Less-than-truckload (LTL) transportation. More precisely three sub-problems of collaborative planning are considered. Centralized multi-carrier problem with pickup and delivery, time windows, exchangeable requests and reserved requests, multi-period Bid Generation Problem with pickup and delivery problem, time windows, profits, reserved requests and multi period Bid Generation Problem with consideration of both cost and delivery lead time.

These sub-problems play a vital role in collaborative transportation planning among carriers, but in the literature, there is no in-depth study on them. We have presented new mathematical programming models for these problems and developed efficient heuristics to obtain solutions close to their optimums in a reasonable computation time. The suggested heuristics are more efficient than commercial solver, CPLEX, not only in terms of solution quality but also in terms of computation time.

**Keywords:** transportation, automotive – operations research – metaheuristics – urban transportation – transportation with time limitation – delivery of goods.

Thèse réalisée en partenariat entre :



Ecole Doctorale "Sciences pour l'Ingénieur"