UNIVERSITY
OF TECHNOLOGY
SYDNEY

Optimisation Models and Algorithms for Scheduling Transportation

by Yefei Zhang

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Dr Hanyu Gu, A/Prof Dr Yakov Zinder

University of Technology Sydney
Faculty of Science

January 2024

Certificate of Original Authorship

I, Yefei Zhang, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Mathematical and Physical Sciences, Faculty of Science at the University of Technology Sydney.
This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.
This document has not been submitted for qualifications at any other academic institution.
This research is supported by an Australian Government Research Training Program Scholarship

Signature of student:

Production Note:
Signature removed prior to publication.

Date: January 23, 2024

Abstract

The thesis is concerned with optimisation models and algorithms for vehicle routing problems (VRPs). An optimisation framework called the Lagrangian ILS is developed which aims at efficiently solving the industry problems. This optimisation framework adaptively adjusts the weights of the coefficients in the iterated local search metaheuristic using the Lagrangian relaxation technique.

Three VRPs are studied in this thesis. For each problem, a problem-specific optimisation procedure is derived under the Lagrangian ILS framework. The first problem studied in this thesis is the Workforce Scheduling and Routing Problem where the objective is weighted on the total cost of outsourcing and the total cost of travelling. A novel optimisation procedure referred to as the Lagrangian ILS has been tested on a standard benchmark on this topic. The results of the computational experiments demonstrate the superior performance of Lagrangian ILS in comparison with the state-of-the-art algorithm for this problem. The effectiveness of utilising the Lagrangian ILS is particularly noticeable in large instances, even when the Lagrangian ILS uses only half of the permissible number of iterations.

The second problem considered in this thesis is a Simultaneous Pickup and Delivery Problem suggested by an Australian transportation company. The problem has ordered objectives. The primary objective is to maximise the number of served customers and the secondary objective is to minimise the total travel time. The thesis formulates the problem into a mixed integer program and solves the problem with a new optimisation procedure called ILS2O. The performance of the ILS2O is evaluated on three sets of benchmarks. One comprises real-world instances provided by the industry partner and the other two are derived from a standard benchmark for VRPs. The results of the computational experiments have demonstrated that the ILS2O produces solutions with high quality and high consistency within the time frame imposed by the industry partner.

The last problem studied is a Simultaneous Pickup and Delivery Problem with Preloading under Uncertainty (SPDPP) which is also motivated by an Australian transportation company. In this problem, customers are revealed in two stages. The assignment of customers in the first stage is called preloading. Since preloading is determined without knowing customers in the second stage, this problem is a stochastic vehicle routing problem. The thesis formulates the SPDPP as a 2-stage stochastic program and solves it using the Sample Average Approximation (SAA) approach. An optimisation procedure called ILS-SAA is proposed to accommodate the non-anticipativity constraints. The performance of ILS-SAA is tested on instances derived from historical data provided by the industry partner. Results of the computational experiments indicate that ILS-SAA yields favourable solutions within a reasonable time frame.

Thesis supervisors: Dr Hanyu Gu, Associate Professor Dr Yakov Zinder

Acknowledgements

- I would like to express my deepest gratitude to my supervisors, Dr. Hanyu Gu and Associate Professor Dr. Yakov Zinder, for sharing their knowledge and expertise with me. Their patience and guidance have made this work possible.
- Many thanks to the director of the Australian National Courier, Mr James Taylor, for giving me the opportunity to work on challenging real-world problems.
- I would like to thank my family for their constant support throughout my research.
- I would like to dedicate this thesis to my mother, Mianchun Wang, who has always believed in me

List of publications

Hanyu Gu, Yefei Zhang, and Yakov Zinder. An efficient optimisation procedure for the workforce scheduling and routing problem: Lagrangian relaxation and iterated local search. Computers & Operations Research, 144:105829, 2022. https://doi.org/10.1016/j.cor.2022.105829.

Hanyu Gu, Yefei Zhang, and Yakov Zinder. Lagrangian relaxation in iterated local search for the workforce scheduling and routing problem. International Symposium on Experimental Algorithms (SEA), Springer, pages 527 − 540, 2019.

Hanyu Gu, Lucy MacMillan, Yefei Zhang, and Yakov Zinder. Iterated local search with neighbourhood reduction for the pickups and deliveries problem arising in retail industry. In Optimization and Learning: 4th International Conference, OLA 2021, Catania, Italy, June 21-23, 2021, Proceedings, pages 190 − 202. Springer, 2021.

Contents

List of Tables

List of Figures

1 Introduction

The thesis is concerned with optimisation models and algorithms for vehicle routing problems. The research presented in this thesis is conducted in collaboration with a leading Australian transportation company that provides home delivery services to clients in the retail industry.

The aim of the research is to develop an optimisation component of an integrated human-computer managerial system where the developed optimisation software is a tool used by the so-called scheduler (the staff responsible for the allocation of customers to vehicles). In such a system, a scheduler can use the optimisation software either for producing an initial version of the allocation that the scheduler can correct if it is necessary, for improving some already existing version of the allocation or for producing some alternative version of the allocation. This interactive mode imposes the restriction that the solution must be produced in seconds rather than minutes.

The vehicle routing problem (VRP) was introduced in Dantzig and Ramser (1959) and has become one of the most active fields of operations research. Several surveys have appeared for VRP and its variants, for example, Laporte (2009), Braekers et al. (2016b), Konstantakopoulos et al. (2020) and several books or chapters of a book have been devoted to VRP, for example, Golden et al. (2008), Toth and Vigo (2002b) and Toth and Vigo (2014). A wide variety of optimisation procedures has been developed for the VRP and its variations, including but not limited to branch-and-price algorithm Bettinelli et al. (2014); branch-and-cut algorithm Wolfinger and Salazar-González (2021); branch-cut-and-price algorithm Subramanian et al. (2013); iterated local search Ibaraki et al. (2008), Xie et al. (2017), Gu et al. (2019); iterated local search with hybrid neighbourhood search Zhou et al. (2020); variable neighbourhood search Chen et al. (2020); genetic algorithm Algethami et al. (2019); simulated annealing Wang et al. (2015); large neighbourhood search Wolfinger (2021); adaptive large neighbourhood search François et al. (2019); tabu search Cordeau et al. (1997), Cordeau et al. (2001); tree-based search algorithm with large composite neighbourhood Schneider and Löffler (2019); memetic algorithm Nagata et al. (2010); parallel iterated tabu search Cordeau and Maischberger (2012); hybrid adaptive large neighbourhood search with tabu search Pan et al. (2021).

Among all these methods, the optimisation procedures that use local search permitting infeasible solutions are particularly interesting (see, for example, Cordeau et al. (1997), Cordeau et al. (2001), Schneider and Löffler (2019), Ibaraki et al. (2008), Xie et al. (2017), Zhou et al. (2020), Gu et al. (2019), Nagata et al. (2010), Cordeau and Maischberger (2012), Pan et al. (2021)). A local search method typically generates a sequence of feasible solutions, where each subsequent solution has a better value of the objective function than the previous one. Since VRPs have complex constraints such as time windows, vehicle capacities, maximum shift duration, etc., such a sequence of feasible solutions that leads to a desired solution may be difficult to find. The local search permitting infeasible solutions attempts to overcome this difficulty by exploring a larger set of solutions that includes both feasible solutions and infeasible solutions.

The advantages of permitting infeasible solutions are evident in Xie et al. (2017) which proposes an iterated local search (ILS) method for the Workforce Scheduling and Routing Problem (WSRP). By allowing the violation of time window and maximum shift duration constraints, the evaluation of the neighbourhoods can be implemented with the highly efficient concatenation technique Vidal et al. (2013) which significantly reduces the solution time. Additionally, a small number of iterations are needed to find good solutions. It is reported in Xie et al. (2017) that instances with 100 tasks can be solved in at most 40 seconds on average. This superior performance on computational time indicates that ILS permitting infeasible solutions can be a promising approach for the studied industry problem in the thesis.

Most optimisation procedures permitting infeasible solutions in the local search component construct an augmented objective function that comprises the original objective function and penalties for the violation of the constraints. The penalty for each constraint is computed as a measure of the violation multiplied by a certain weight, and weights for the constraints are updated by multiplying them with some constants.

The author of the thesis implemented the ILS in Xie et al. (2017) for the industry problem and observed that the constants used to update the weights for the penalty are critical for the performance of the ILS. Setting these constants with appropriate values requires tedious and time-consuming computational experiments. Tuning the setting for these constants is based on the average performance of the ILS over a benchmark of instances. Since the ILS is a stochastic procedure, the average performance is used over a certain number of independent runs. It has been noticed that the ILS can produce much worse solutions on some runs which indicates the robustness of ILS is not satisfactory. Accordingly, there is no guarantee that the ILS will perform well on new instances for industry applications.

This thesis develops an optimisation framework that amalgamates the iterated local search metaheuristic Lourenço et al. (2019) and Lagrangian relaxation technique Fisher (1981). This framework referred to as the Lagrangian ILS framework views the weights for the violation of constraints as Lagrange multipliers and chooses their initial values as well as dynamically updates them correspondingly. By creating new optimisation algorithms, this framework has been applied to three practical VRPs. The outcome of the computational experiments has demonstrated that the algorithms implemented within this framework produce excellent performance with respect to computational efficiency, solution quality, and consistency. Below is an outline for each of the problems considered in this thesis along with a brief overview of the main results presented.

Workforce Scheduling and Routing Problem

The first problem considered in this thesis is the WSRP which has applications ranging from home health care to manpower allocation Castillo-Salazar et al. (2016), Fikar and Hirsch (2017), Paraskevopoulos et al. (2017). This problem is concerned with the allocation of tasks (requests for service, customers, patients) to the service providers (technicians, nurses). The tasks have different locations, and the service providers need to spend a significant amount of time travelling between these locations. The version of the WSRP studied in this thesis considers several real-world restrictions, including, time windows for the tasks, the maximum shift duration for the service providers, and the compatibility between the service providers and tasks. These restrictions were also reflected in the second problem studied in this thesis (the industry problem). In addition, if a task cannot be allocated to a service provider, this task incurs a penalty which will be referred to as the cost of outsourcing. The objective of the problem is to minimise the total cost of travelling and outsourcing.

The thesis presents a new optimisation procedure for the WSRP that amalgamates the iterated local search and Lagrangian relaxation. This optimisation procedure referred to as the Lagrangian ILS has been tested on a set of benchmark instances from the literature, which is regarded as standard in the publications on this topic. The results of the computational experiments have shown that the Lagrangian ILS outperforms the state-of-the-art algorithm for WSRP described in Xie et al. (2017) both in terms of solution quality and computational time. The advantage of using the Lagrangian ILS becomes more pronounced in large instances, as it outperforms the algorithm in Xie et al. (2017), even when the Lagrangian ILS is limited to use only half of the permissible number of iterations.

Multi-attribute Simultaneous Pickup and Delivery Problem

The second problem considered in this thesis is a Simultaneous Pickup and Delivery Problem that reflects many real-world restrictions and practices. This problem referred to as the Multi-attribute Simultaneous Pickup and Delivery Problem (MASPDP) is suggested by the industry partner. The MASPDP belongs to a class of problems commonly referred to as the last-mile delivery that concerns the final stage of a supply chain – the direct delivery from a depot to customers Lim et al. (2018), Boysen et al. (2021). It is also the most expensive and least efficient part of the supply chain Gevaers et al. (2011). Due to urbanisation and the growth of e-commerce, this problem has become critical in the past decades.

In comparison with the WSRP, the MASPDP considers many additional real-world restrictions and practices, including, open routes, vehicles with different capacities, a roster that specifies the time when a vehicle can load at the depot, and simultaneous pickup and delivery. The objectives for the MASPDP are ordered where the primary objective is to maximise the number of served customers and the secondary objective is to minimise the total travel time. These restrictions and practices make the problem belong to a class of VRPs so-called rich VRP Lahyani et al. (2015), Kramer et al. (2019).

This thesis introduces two new mixed integer programming formulations for the MASPDP and presents a novel optimisation procedure. This optimisation procedure referred to as the ILS2O is a further development of the Lagrangian ILS by taking into account the bi-objective optimisation. In addition, the ILS2O uses a neighbourhood reduction technique that dynamically reduces the search space and leads the search to a more promising solution. The performance of the ILS2O is tested on two sets of benchmarks. One is real-world instances provided by the industry partner and the other one is derived from a standard benchmark for VRPs. The results of the computational experiments have demonstrated that the ILS2O produces robust and high-quality solutions within the time frame imposed by the industry partner.

Simultaneous Pickup and Delivery Problem with Preloading under Uncertainty

Following the advice from the industry partner, the last problem considered in this thesis is a Simultaneous Pickup and Delivery Problem with Preloading under Uncertainty (SPDPP). In this problem, two groups of vehicles are considered. The vehicles in the first group must return to the depot after serving all the allocated customers, load some demands for the next day and return to their designated end locations. Such an operation is referred to as preloading. The preloading is done when only a subset of all customers is known, whereas the assignment of customers to the vehicles of the second group is done when all remaining customers are known.

The reason for preloading is to deal with the limited capacity at the depot. Following a customer's purchase, the depot will receive the customer's demand and wait for a vehicle to fulfil the delivery. If the depot runs out the space when a customer's demand arrives that means the depot cannot store the demand before its delivery commences, then this customer must be outsourced which is expensive. By loading some demands to the vehicles, preloading helps to lighten the heavy burden of the depot. As a result, preloading could reduce the total outsourcing cost. The preloading feature makes the problem unique in comparison with other stochastic VRPs in the literature.

In this thesis, the SPDPP is formulated as a 2-stage stochastic program and is solved by the sample average approximation (SAA) approach. The thesis introduces a new optimisation procedure that is referred to as the ILS-SAA for the SAA approach. To handle the non-anticipativity constraints where the first stage solution in the SAA approach is the same for all the scenarios in the second stage, the ILS-SAA made several new developments. These developments change how the initial feasible solutions are constructed, how local search is performed with a new local search operator, as well as how the perturbation mechanism works. In addition, the ILS-SAA is also extended from the Lagrangian ILS for WSRP. The computational experiments are conducted on a set of instances derived from the historical data provided by the industry partner. The results of computational experiments have demonstrated that the ILS-SAA produces a good solution in a reasonable time.

1.1 Thesis organisation

The remaining part of the thesis is organised as follows.

- Chapter 2 provides an overview of the literature in the related research areas.

- Chapter 3 describes the Lagrangian ILS for the WSRP which is an amalgamation of iterated local search and Lagrangian relaxation. The idea of such amalgamation was initially presented at the Analysis of Experimental Algorithms 2019 conference and included in the refereed conference proceeding as "Hanyu Gu, Yefei Zhang, and Yakov Zinder. Lagrangian relaxation in iterated local search for the workforce scheduling and routing problem. International Symposium on Experimental Algorithms (SEA), Springer, pages 527540, 2019.". The Lagrangian ILS described in Chapter 3 is a new implementation of this idea. The work presented in Chapter 3 was published as "Hanyu Gu, Yefei Zhang, and Yakov Zinder. An efficient optimisation procedure for the workforce scheduling and routing problem: Lagrangian relaxation and iterated local search. Computers & Operations Research, 144:105829, 2022. https://doi.org/10.1016/j.cor.2022.105829." and was awarded the "Science HDR Student Paper of the Month, April – May 2022" in University of Technology Sydney.

- Chapter 4 describes the ILS2O for the MASPDP. The neighbourhood reduction technique utilised in the ILS2O was successfully tested on a version of MASPDP with a single objective, i.e., maximising the number of served customers. The research on MASPDP with a single objective was presented at the International Conference on Optimisation and Learning 2021 and included in the refereed conference proceeding as "Hanyu Gu, Lucy MacMillan, Yefei Zhang, and Yakov Zinder. Iterated local search with neighbourhood reduction for the pickups and deliveries problem arising in retail industry. In Optimization and Learning: 4th International Conference, OLA 2021, Catania, Italy, June 21-23, 2021, Proceedings, pages

190202. Springer, 2021."

- Chapter 5 describes the ILS-SAA for the SPDPP.
- Chapter 6 concludes the thesis with a summary of the contributions of the thesis.

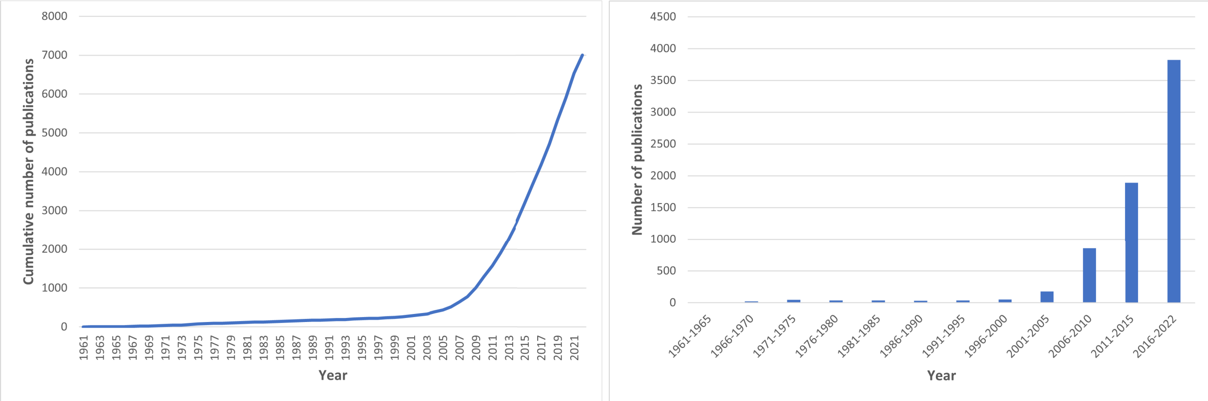
2 Literature review

In this chapter, a literature review is provided focusing on works related to the three problems studied in this thesis, i.e., Workforce Scheduling and Routing (WSRP) in Chapter 3, Multi-attribute Simultaneous Pickup and Delivery Problem (MASPDP) in Chapter 4, and Simultaneous Pickup and Delivery Problem with Preloading under Uncertainty (SPDPP) in Chapter 5. Please note, that this chapter does not attempt to classify and categorise the huge body of existing work on vehicle routing problems. Instead, this chapter focuses on whether there exist publications that study the same problems as the industry problems studied in this thesis and whether there are solution methods that have the potential to solve the industry problems.

The three problems studied in the thesis are considered as the variants of the vehicle routing problem (VRP) introduced by Dantzig and Ramser (1959). The VRP involves finding the optimal routes for a fleet of vehicles to satisfy the demands of customers at different locations. This problem is a generalisation of the travelling salesman problem which is NP-hard in the strong sense Garey and Johnson (1979). Due to the broad applications of VRP in industries such as transportation, logistics, and workforce scheduling, thousands of publications have appeared on this topic. According to data collected on 20th Dec 2022 from Google Scholar, there are 6999 publications that have cited Dantzig and Ramser (1959) (see figure 2.1a). In the past 7 years, Dantzig and Ramser (1959) has been cited more than 3500 times (see figure 2.1b).

2.1 Workforce scheduling and routing problem

One of the important applications of the WSRP is home health care (HHC) where nurses and other caregivers need to visit the HHC recipients. In 2008, there were approximately 12 million people in the US requiring HHC services Chen et al. (2017). In Europe in 2012, between 1% and 5% of the total public health budget was spent on HHC services



(a) Cumulative number of publications that cite Dantzig and Ramser (1959) from 1960 to 2021.

(b) Number of publications that cite Dantzig and Ramser (1959) from 1960 to 2021.

Figure 2.1: Trend on the number of vehicle routing publications.

Fikar and Hirsch (2017). The number of people requiring HHC services is growing every year. According to nla (2007), the population of 60 years old and over, who constitute the largest group of the recipients of HHC services, has been growing in the European Union from 17 percent in 1980 to 22 percent in 2004 and may increase to 32 percent in 2030.

Another important application of the WSRP is technical service where technicians or engineers should visit various locations. One of the examples is British Telecom which needs to allocate its 30,000 field engineers to tasks such as network maintenance, repairs, and installation of appliances Borenstein et al. (2010). The authors of Goel and Meisel (2013) considered the problem of planning the electricity network maintenance operations and tested their algorithm on a set of instances that were derived from data of a German electricity provider. The list of publications in which the optimisation algorithms are tested, using data originating from real-world situations, can be easily extended. For example, Peng et al. (2013) is concerned with the allocation of engineers to rail inspections and Dohn et al. (2009) is concerned with the allocation of workers to tasks in some of Europe's major airports.

Due to its numerous applications, the WSRP has attracted significant attention in the literature Castillo-Salazar et al. (2016), Fikar and Hirsch (2017), Paraskevopoulos et al. (2017). Depending on the application, the problem may contain some additional assumptions and constraints, but all these variations assume that both travel time between locations and service times are essential. The publications on the WSRP often assume that each service provider has certain skills and can be assigned only to the tasks for which these skills are sufficient. Another frequently used assumption in the publications on WSRP is the possibility of outsourcing. Furthermore, the WSRP is not concerned with the capacity of vehicles which is important in problems with a heterogeneous fleet of vehicles such as the problems considered in Koç et al. (2016).

As has been mentioned above, different applications may require additional assumptions and constraints which lead to a number of variations of the WSRP. Thus, the tasks may have different priorities Xu and Chiu (2001); each task may have the associated time window within which the service is to be provided Dohn et al. (2009), Peng et al. (2013), Braekers et al. (2016a), Polnik et al. (2021); the service providers may work in teams Kovacs et al. (2012) or some tasks may require the simultaneous involvement of several service providers Dohn et al. (2009), Polnik et al. (2021); a task may require some specific tools or spare parts which must be carried by the allocated service provider Pillac et al. (2013). The majority of the publications on the WSRP and its variations require a solution for a single time period (normally a day or a shift). In contrast, the problem studied in Guastaroba et al. (2021), is concerned with the planning horizon which is comprised of several such time periods. Other additional assumptions and constraints considered in the literature include lunch breaks Liu et al. (2017) and tasks which are comprised of several stages Pereira et al. (2020). A comprehensive discussion on the main characteristics of the WSRP and on various additional assumptions can be found in the surveys Castillo-Salazar et al. (2016), Fikar and Hirsch (2017), and Paraskevopoulos et al. (2017).

Even particular cases of the WSRP such as the travelling salesman problem and the makespan minimisation scheduling problem for parallel identical machines are NP-hard in the strong sense Garey and Johnson (1979). A wide variety of the optimisation procedures, developed for the WSRP and its variations, includes mixed integer programming with decomposition Laesanklang et al. (2015), Laesanklang et al. (2016); branch-and-price algorithms Dohn et al. (2009), Liu et al. (2017); algorithms based on Lagrangian relaxation Fathollahi-Fard et al. (2018), Gu et al. (2019); iterated local search Xie et al. (2017), Gu et al. (2019); iterated local search with hybrid neighbourhood search Zhou et al. (2020); genetic algorithms Shi et al. (2017), Algethami et al. (2016), Algethami and Landa-Silva (2017), Algethami et al. (2019); large neighbourhood search Goel and Meisel (2013), Braekers et al. (2016a); adaptive large neighbourhood search Kovacs et al.

(2012),Guastaroba et al. (2021); greedy randomised adaptive search Xu and Chiu (2001); ant colony optimisation Pereira et al. (2020); artificial bee colony optimisation Yurtkuran et al. (2018); matheuristic Pillac et al. (2013); and constraint programming Polnik et al. (2021).

The WSRP studied in Chapter 3 follows the problems studied in Kovacs et al. (2012), Xie et al. (2017), and Zhou et al. (2020). The problem is concerned with the allocation of tasks to the service providers. The tasks have different locations, and the service providers need to spend significant time travelling between these locations. The constraints of this problem include the time window for tasks, maximum duration on shift length and compatibility between tasks and service providers. The objective is to minimise the total cost of travelling and outsourcing.

2.2 Multi-attribute Simultaneous Pickup and Delivery Problem

Chapter 4 studies the Multi-attribute Simultaneous Pickup and Delivery Problem (MASPDP) which is an extension of the classic vehicle routing problem with additional attributes. The six decades of extensive research on VRP and its variants have resulted in huge progress in the development of vehicle routing methodology. In recent years, there is an increasing focus on solving complex VRPs that arise in real-life Caceres-Cruz et al. (2014), Lahyani et al. (2015), which is also referred to as rich VRPs (RVRPs). According to the definition provided in Lahyani et al. (2015), an RVRP should have at least nine additional physical characteristics with respect to the classical VRP. A physical characteristic is sometimes called an attribute or a feature in the literature (see for example, Vidal et al. (2013), Vidal et al. (2014)). The MASPDP contains 10 features, which qualifies it as an RVRP. In what follows, several features that have been added to the VRP in the literature are discussed.

- **Simultaneous pickup and delivery**

  The simultaneous pickup and delivery problem (SPDP) is another generalisation of the VRP, which is first introduced by Min (1989). This problem reflects the practical situations where a customer often requires both pickup and delivery. The

SPDP is also a generalisation of the mixed pickup and delivery problem where a customer can request either pickup or delivery only Kamsopa et al. (2021). For more than 30 years, many publications have studied the SPDP and several surveys have appeared in the literature, for example, Parragh et al. (2008a), Parragh et al. (2008b), Berbeglia et al. (2007), Koç et al. (2020), and Bouanane et al. (2022). One of the important applications of the SPDP is reverse logistics where the vehicles deliver products and at the same time, collect end-of-life products (see, for example, Agrawal et al. (2015) and Govindan et al. (2015)). In contrast to SPDP, Goetschalckx and Jacobs-Blecha (1989) and Reil et al. (2018) have studied the VRP with backhaul where the vehicles only consider pickup after the last delivery.

- **Time windows**

A time window designates the earliest time (left end of the time window) and the latest time (right end of the time window) when the service corresponds to a customer can commence. There are several publications that combine the time window constraints with the SPDP. For example, DENG et al. (2009), Gan et al. (2012), Wang and Chen (2012), Wang et al. (2015), Mingyong and Erbao (2010), Liu et al. (2013), Wang et al. (2016), and Shi et al. (2020). There are also surveys available on vehicle routing problem with time windows, for example, Bräysy and Gendreau (2005a), Bräysy and Gendreau (2005b).

In the literature, the time window constraints have been treated in two ways. The time windows can be treated as hard constraints. More specifically, the vehicle arrives at a customer before the left end of the time window resulting in a wait before service can begin. On the other hand, arriving at the customer after the right end of the time window is not allowed. Examples can be found in Mingyong and Erbao (2010), Wang and Chen (2012), Liu et al. (2013), Wang et al. (2015), and Shi et al. (2020). The problems studied in this thesis also treat the time windows as hard constraints.

Another way is to treat time windows as soft constraints. For example, in Fu et al. (2008), DENG et al. (2009), Castro-Gutierrez et al. (2011), Gan et al. (2012), Wang et al. (2016), a waiting time is incurred if the vehicle arrives at a customer before the left end of the time window. If the vehicle arrives at the customer after the right end of the time window, a delaying time is incurred. Then, the total waiting time and total delaying time are minimised in the objective function. This feature is motivated by the situation when slightly violating the time window constraints is not a critical breach of service requirements. In this case, relaxing the time windows may result in lower cost solutions requiring fewer vehicles, shorter travel distances, and less travel time Chiang and Russell (2004), Fu et al. (2008).

- **Restriction on shift length**

  Another feature in the problems studied in Chapter 4 is the restriction on shift length. This is motivated by the regulation imposed on the drivers that they cannot work longer than a certain amount of time per shift. Several publications have also considered this feature, for example, Seixas and Mendes (2013), Alcaraz et al. (2019). In contrast, the restriction on shift length is relaxed in Moon et al. (2012) with overtime labour cost. In addition, some publications are concerned with the break time for drivers or rest areas for vehicles after working long hours during a shift, for example, Ceselli et al. (2009), Coelho et al. (2016), Kamsopa et al. (2021).

- **Open route**

  The VRP with open route is known as the open VRP (OVRP). This variant is first considered by Sariklis and Powell (2000) where a vehicle does not return to the depot after serving the last customer in the route. This variant reflects the situation when the company outsources the service to subcontractors who have owned vehicles by themselves Simeonova et al. (2020). As mentioned in Chapter 1, these subcontractors also have their own depot suitable for temporary storage. This feature commonly appears in the service industry and retail sector Russell et al. (2008), Lahyani et al. (2015). In the literature, this feature has attracted enough attention in the recent two decades. For example, Tarantilis and Kiranoudis (2002), Brandão (2004), Tarantilis et al. (2005), Fu et al. (2005), Letchford et al. (2007), Li et al. (2007), Rieck and Zimmermann (2010), Ceselli et al. (2009), and Li et al. (2012).

- **Weight and volume**

  In the literature, the demand of a customer is often characterised by a single type (either weight or volume), for example, Gajpal and Abad (2010), Halvorsen-Weare and Savelsbergh (2016), Bouzid et al. (2017), Schneider and Löffler (2019). Only a few publications consider both weight and volume. The publications Bortfeldt (2012) and Reil et al. (2018) study vehicle routing problems with 3-dimensional loading constraints where the demand of each customer is a set of rectangular items specified by weight, width, length, and height. The publication Sabar et al. (2020) only considers 2-dimensional loading constraints where the demand is characterised by weight, width and length. For the problems studied in Bortfeldt (2012) Reil et al. (2018), and Sabar et al. (2020), the goal is not only to construct routes for the vehicles but also to determine how vehicles are loaded. In contrast to these publications, the problem studied in Chapter 4 assumes that the drivers know how vehicles can be loaded and the demand (both pickup and delivery) of a customer is characterised by weight and volume computed by width, length, and height.

- **Heterogeneous fleet of vehicles**

  The VRP with a heterogeneous fleet of vehicles reflects the real-world situation that customers are served by a fleet of heterogeneous vehicles Koç et al. (2016). This variant referred to as the Heterogeneous VRP (HVRP) was first introduced in Golden et al. (1984). In the past 30 years, it becomes a very active field of research (Yaman (2006), Baldacci et al. (2008), Li et al. (2012), Seixas and Mendes (2013), Bettinelli et al. (2014), Yao et al. (2016), Simeonova et al. (2018), Bevilaqua et al. (2019), and Sabar et al. (2020)). In Avci and Topaloglu (2016), Nepomuceno et al. (2019), Kamsopa et al. (2021), Keçeci et al. (2021), the SPDS with heterogeneous fleet of vehicles is studied. In publications Bortfeldt (2012), Reil et al. (2018), the vehicles are characterised by their weight capacity, maximum width, maximum length, and maximum height, whereas in Sabar et al. (2020), the vehicles are characterised by their weight capacity, maximum width and maximum length. For the problem considered in Chapter 4, the vehicles are characterised by two types of capacity, i.e. weight and volume.

- **Incompatibility**

  In real-life VRP, incompatibility can often appear between customers and vehicles. Therefore, this feature has also attracted much attention. For example, in Alcaraz et al. (2019) and Ceselli et al. (2009), customers can order different types of goods and not all goods can be transported by a single vehicle. In contrast, in Seixas and Mendes (2013) and Kramer et al. (2019), due to accessibility restrictions at the delivery location, some customers can only be served by specific vehicles. The incompatibility constraints considered in Chapter 4 are similar to the ones considered in Seixas and Mendes (2013) and Kramer et al. (2019).

- **Ordered objectives**

  The problem studied in Chapter 4 has two objectives and they are ordered. The first-order objective is to maximise the number of served customers and the second-order objective is to minimise the total travel time.

  In the literature, the most common objectives considered in vehicle routing publications are the minimisation of the number of vehicles and the minimisation of total travel cost (usually computed from total travel distance or total travel time). Examples can be found in Wang et al. (2015), Sabar et al. (2020), Shi et al. (2020). In Chapter 4, the problem also considers the minimisation of total travel time. Such consideration is motivated by the massive expense caused by real-world transportation activities with respect to both the economy and the environment. According to Toth and Vigo (2002a), the use of computerised procedures for the distribution process planning produces significant savings (generally from 5% to 20%) in global transportation costs. On the other hand, in Europe in 2010, transportation activities were responsible for approximately 20% of greenhouse gas emissions Schneider et al. (2014).

  Another area that is closely related to the problems studied in this thesis is the VRP with profit. This setting reflects the real-world situation when there is an insufficient number of vehicles to fulfil the demand for a set of customers Gansterer et al. (2017). Such a situation forced the company to make a profitable selection of customers. The selected customers will be served by the vehicles available whereas the remaining customers are outsourced to subcontractors. This setting of the VRPs is also been extensively studied in the literature. For example, in Archetti et al. (2009), the profitable tour problem is studied where the objective is maximising the difference between the total collected profit and the cost of the total distance travelled, in Li et al. (2016), the pickup and delivery problem with time windows, profits, and reserved requests where the objective is to maximise the difference between the sum of payments of served requests and the total transportation cost.

Some publications consider more than two objectives. For example, in Wang et al. (2016), the minimisation of five objectives is considered including, the number of vehicles; total travel distance; travel time of the longest route; total waiting time; and total delay time. The publication Hornstra et al. (2020) maximises the profit of revenue after deducting fuel costs, the cost of using a vehicle, driver wage cost, penalty cost and overtime cost. Many other objectives have been considered by the vast literature. For example, the minimisation on outsourcing cost Alcaraz et al. (2019); the maximisation on customers' satisfaction Fan (2011); the minimisation on the handling costs for the pickup items at the rear of the vehicle Hornstra et al. (2020).

- **Other attributes considered in literature**

  Many features have not been considered in MASPDP. For example, a vehicle can re-load at the depot and can have routes for multiple trips Rieck and Zimmermann (2010), Cattaruzza et al. (2018); the demand of a customer can be satisfied by multiple visits Archetti and Speranza (2008), Nagy et al. (2013), Polat (2017); instead of a single depot, the vehicles can depart from different depots Nagy and Salhi (2005), Rahimi-Vahed et al. (2013), Salhi et al. (2014); the customers can require delivery for a variety of products with different temperature requirements Martins et al. (2019); a limit is applied to the number of customers that can be served by a route Kramer et al. (2019). In Chapter 4, the problem is concerned with scheduling for a single period, i.e., usually a day or a shift. In contrast, the publication Kamsopa et al. (2021) studies the SPDP with multiple such periods. In this thesis, some features that have appeared in VRP are discussed. The comprehensive classification and taxonomic survey can be found in Lahyani et al. (2015), Koç et al. (2020), Simeonova et al. (2020), and Bouanane et al. (2022).

In Table 2.1, a summary of some of the recent publications is given in terms of the considered attributes in MASPDP studied in Chapter 4. One of the important attributes of the MASPDP in Chapter 4 is the roster that specifies the time when a vehicle can load at the depot. This feature is motivated by the fact that the depot has limited loading space. To the best of the author's knowledge, no publication has considered this attribute (as shown in Table 2.1).

Table 2.1: VRP with multiple features

	Alcaraz et al. (2019), Kramer et al. (2019), Seixas and Mendes (2013)	Ceselli et al. (2009)	Hornstra et al. (2020)	Kassem and Chen (2013), Wang et al. (2016), Wang et al. (2015)	Chen et al. (2020)	Avci and Topaloglu (2016)	Kamsopa et al. (2021)
Time windows	✓	✓		✓	✓		✓
Open routes		✓			✓		
Simultaneous weight and volume		✓					
Heterogeneous vehicles	✓	✓				✓	✓
Incompatibility	✓	✓					
Roster							
Simultaneous pickup and delivery			✓	✓		✓	
Restriction on shift length	✓	✓					✓
Minimise cost	✓	✓	✓	✓	✓		✓
Maximise profit							✓

2.3 Simultaneous Pickup and Delivery Problem with Preloading under Uncertainty

In real-life applications, some parameters are usually unknown due to the presence of uncertainty Gendreau et al. (2014). These parameters will only be revealed after the decision phase. This significantly increases the difficulty of computing a solution for the problem. In Chapter 5, the problem takes into account preloading with unknown customers' demands that is motivated by the limited storage capacity of the depot.

In the publications Gendreau et al. (1995) and Laporte et al. (2002), the VRP with stochastic demands is studied where the demand is revealed after arriving at the location of a customer. The publication Gendreau et al. (1995) also considered stochastic customers where a customer may no longer require a visit after the route is constructed. The uncertainty may arise from other sources. For example, Zhang et al. (2012) considers stochastic travel time (service time); Sungur et al. (2010) and Lei et al. (2012) consider pure stochastic service time; Bektaş and Laporte (2011) considers stochastic time window; Keskin et al. (2021) considers stochastic waiting time; Zhu and Sheu (2018) considers

SPDP with stochastic demand. The comprehensive discussion on the main characteristics of the stochastic vehicle routing problem can be found in the surveys Ritzinger et al. (2016), Gendreau et al. (2016), Oyola et al. (2018). To the best of the author's knowledge, the SPDPP described in Chapter 5 is a new problem that has never been studied in the literature.

2.4 Solution approaches

Different methods have been explored to solve the VRP and its variants. Some approaches focus on finding the optimal solutions or obtaining good lower bounds such as branch-and-price algorithm Bettinelli et al. (2014); branch-and-cut algorithm Wolfinger and Salazar-González (2021); branch-cut-and-price algorithm Subramanian et al. (2013). There are also approaches that focus on finding approximate solutions including

- heuristics, such as Sweep heuristic Clarke and Wright (1964), local search Focacci et al. (2003);
- metaheuristics, such as iterated local search Lourenço et al. (2019), variable neighbourhood search Hansen et al. (2019); tabu search Gendreau and Potvin (2019);
- and matheuristics Kramer et al. (2015), Sartori and Buriol (2020), Doerner and Schmid (2010), Archetti and Speranza (2014).

In this section, the solution methods used to tackle the VRP and its variants are discussed.

2.4.1 Mixed integer programming formulation

Different mathematical models have been introduced for VRP and its variants. There are models that use variables with three indexes. For example, $x_{i,j}^k$ is a binary variable that indicates whether vehicle k travels between customer i and j. Examples can be found in Mosheiov (1998), Dethloff (2001), Kallehauge et al. (2005), Montané and Galvao (2006), Baldacci et al. (2008), Wang et al. (2015).

Another class of models is known as the two-index model. In this model, a binary variable $x_{i,j}$ indicates whether a vehicle travels between customer i and j. Several publications have used this type of model to investigate the lower bound of the VRPs. For example, Yaman (2006) developed six formulations and valid inequality for the heterogeneous VRP; Baldacci et al. (2009) developed a two-index commodity flows formulation and two new classes of valid inequalities; Subramanian et al. (2011), Subramanian et al. (2013), and Rieck and Zimmermann (2013) presented the two-index vehicle flow models. In Chapter 4, the MASPDP is formulated into two different models, one is a three-index model and the other one is a two-index model.

2.4.2 Lagrangian relaxation

Many combinatorial optimisation problems are complicated by side constraints Fisher (1981). Lagrangian relaxation relaxes a subset of the side constraints which leads to a "relatively easier" problem. This "easy" problem referred to as the Lagrangian problem contains an augmented objective function which is comprised of the objective function of the original problem and a measure of the violation of the relaxed constraints multiplied by a vector of coefficients known as the Lagrange multipliers. The optimal objective value of the Lagrangian problem is a lower bound of the optimal objective value of the original minimisation problem. The Lagrangian relaxation method has been discussed in a number of publications, for example, Geoffrion (1974), Fisher (1981), Lemaréchal (2001), Guignard (2003), Frangioni (2005). This thesis proposes algorithms that are an amalgamation of the Lagrangian relaxation technique and iterated local search metaheuristics. The main goal of the algorithms is to find good feasible solutions that differ from the goal of conventional Lagrangian relaxation approaches i.e., obtaining a good bound.

2.4.3 Iterated Local search permitting infeasible solutions

The iterated local search (ILS) is a metaheuristic that iteratively produces a sequence of solutions. Each solution in this sequence is generated by an embedded optimisation procedure, typically a local search algorithm Lourenço et al. (2019). Let s' be a feasible solution and s^* records the current best feasible solution. The pseudocode below outlines the basic ILS.

ILS

1: $s^* \leftarrow$ Generate initial feasible solution
2: $s' \leftarrow s^*$
3: **while** Stopping criterion is not satisfied **do**
4: $s' \leftarrow$ Local Search(s')
5: $s^* \leftarrow$ Acceptance Criterion(s^*, s')
6: $s' \leftarrow$ Randomly generate new feasible solution from s^*
7: **end while**
8: **return** s^*

The ILS has many applications on VRP and its variants, for example, Ibaraki et al. (2008), Penna et al. (2013), Li et al. (2015), Xie et al. (2017), Gu et al. (2019) Zhou et al. (2020), Öztaş and Tuş (2022), Gu et al. (2022b). A comprehensive discussion on the applications for the ILS can be found in Lourenço et al. (2019).

The optimisation procedure discussed in Chapter 3 is an amalgamation of the iterated local search (ILS) Lourenço et al. (2019) and Lagrangian relaxation Fisher (1981). In what follows, this procedure will be referred to as the Lagrangian ILS. The idea of such amalgamation was first introduced in Gu et al. (2019) and stemmed from the observation that the performance of local search often can be improved by permitting the violation of some constraints and by introducing an augmented objective function which comprises the original objective function and a penalty for the violation of the constraints. This phenomenon was used in various algorithms (see, for example, Cordeau et al. (1997), Cordeau et al. (2001), Schneider and Löffler (2019), Ibaraki et al. (2008), Xie et al. (2017), Zhou et al. (2020), Gu et al. (2019), Nagata et al. (2010), Cordeau and Maischberger (2012), Pan et al. (2021)) and can be anticipated given the nature of local search Gendreau and Potvin (2019). Indeed, local search generates a sequence of solutions where each subsequent solution has a better value of the objective function than the preceding one. Since such a sequence of monotonic (in terms of the objective values) feasible solutions that lead to a desired solution may not exist or be difficult to find, the use of the infeasible solutions may significantly facilitate the construction of a sequence that renders a desired solution.

To the best of the author's knowledge, in all publications in which the constraint violation is permitted, the penalty for each constraint is computed as a measure of the violation multiplied by a certain coefficient (weight) and these weights either remain unchanged during the entire optimisation (see, for example, Nagata et al. (2010)), or are updated by multiplying them by some constants which remain the same during the entire optimisation (see, for example, Cordeau et al. (1997), Cordeau et al. (2001), Schneider and Löffler (2019), Ibaraki et al. (2008), Xie et al. (2017), Zhou et al. (2020), Pan et al. (2021), Cordeau and Maischberger (2012)). In both cases, these constants are determined as a result of tedious computational experiments. In contrast, Gu et al. (2019) introduced a mixed integer linear programming formulation that permits considering the weights as Lagrange multipliers and choosing their initial values as well as dynamically updating them correspondingly. According to Gu et al. (2019), the application of local search to the Lagrangian relaxation of the original problem rather than to the original problem itself is motivated by the observation that this new problem, although remaining difficult from the computational point of view, is more amenable to local search. In other words, the use of the Lagrangian relaxation is dictated not by the complexity consideration but by the suitability for the optimisation method − local search. This distinguishes Gu et al. (2019) from the conventional Lagrangian relaxation approach where the main goal is to obtain a tight bound for the optimal value of the objective function with, if necessary, the subsequent conversion of the obtained infeasible solution into a feasible one Fisher (1981), Fathollahi-Fard et al. (2018).

2.4.4 Multi-objective VRPs

In some real-world applications, the problem can have multiple objectives or even conflicting objectives like the one considered in Chapter 4. The most common method to tackle multi-objective optimisation is weighted sum Coello Coello (1999), for example, Kovacs et al. (2012), Xie et al. (2017). The weighted sum combines multiple objectives into a single function. In this function, a weight is associated with each objective which indicates the preference. The advantage of this method is that it is relatively easy to implement. However, determining appropriate values for the weights involves tedious computational experiments.

Another method to tackle multi-objective optimization problems is known as the lexicographic method Fishburn (1974). This method requires a pre-specified preference regarding the objectives. Then, the method attempts to find a better solution with respect to each objective one at a time. This method is suitable for problems when the preferences on the objectives are easily established. For example, in Shi et al. (2020), a lexicographic-based two-stage algorithm is used to solve the SPDP with time windows. In their problem, the primary objective is to minimise the number of vehicles and the secondary objective is to minimise total travel distance. According to Castro-Gutierrez (2012), this method produces good results when the number of objectives is small, typically 2 or 3 objectives. The comprehensive surveys about the multi-objective VRPs can be found in Jozefowiez et al. (2008), Zajac and Huber (2021).

2.4.5 Sample average approximation approach for 2-stage stochastic program

In the SPDPP studied in Chapter 5, customers are revealed in two stages and some routes are determined without knowing customers in the second stage. In Chapter 5, to capture the stochastic customers, the studied SPDPP is formulated as a two-stage stochastic program Birge and Louveaux (2011). The stochastic programming has been applied to various problems such as the scheduling problem with random processing times Gu et al. (2022a); underground mine scheduling with random duration and economic value for each underground mining activity Nesbitt et al. (2021); spare parts inventory management problem with random deployment situations Johannsmann et al. (2022)

Let Ω be a set of all scenarios, P be the probability of occurrence for scenario $\omega \in \Omega$, and $\mathbb{E}_{\omega\in\Omega}$ be the mathematical expectation with respect to ω. A 2-stage stochastic program can be represented as follows.

$$\max c^T x + \mathbb{E}_{\omega\in\Omega}[Q(x,\omega)] \tag{2.1}$$

subject to:

$$Ax = b \tag{2.2}$$

$$x \geq 0 \tag{2.3}$$

where for a particular realization $\tilde{\omega}$ of ω, $Q(x,\tilde{\omega})$ is defined as

$$Q(x,\tilde{\omega}) = \max q^T(\tilde{\omega})y \tag{2.4}$$

subject to:

$$W(\tilde{\omega})y = h(\tilde{\omega}) - T(\tilde{\omega}) \tag{2.5}$$

$$y \geq 0 \tag{2.6}$$

where x is a set of first-stage decision variables; y is a set of second-stage decision variables; c^T, A, b are input data for the first stage; $q^T(\tilde{\omega})$, $W(\tilde{\omega})$, $h^T(\tilde{\omega})$, $T(\tilde{\omega})$ are input data for the second stage correspond to $\tilde{\omega} \in \Omega$.

Sample average approximation

Since $Q(x, \omega)$ itself requires solving a combinatorial optimisation problem, and $\mathbb{E}_\omega[Q(x, \omega)]$ is difficult to compute, in Chapter 5, the sample average approximation (SAA) approach is used to solve the SPDPP. This approach replaces the 2-stage stochastic program with a deterministic mixed integer program (MIP) and approximates $\mathbb{E}_\omega[Q(x, \omega)]$ with a sample average function. More specifically, let $S = \{\omega^1, \omega^2, ..., \omega^{|S|}\}$ be a sample of ω, then

$$\mathbb{E}_\omega[Q(x, \omega)] \approx \frac{1}{|S|} \sum_{s=1}^{s=|S|} Q(x, \omega^s). \tag{2.7}$$

It has been shown in Kleywegt et al. (2002) that with the increase in the sample size, an optimal solution to the MIP model for the SAA approach provides the exact optimal solution of the 2-stage stochastic program.

2.5 Benchmarks for vehicle routing problems

Since the VRP was introduced by Dantzig and Ramser (1959), many publications have derived their own benchmarks. The most well-known benchmarks are introduced by Solomon (1987) for VRP with time window which is also known as the Solomon benchmarks. Many publications use Solomon benchmarks as the basis of their own benchmark, for example, Russell (1995), Li and Lim (2003), Ma et al. (2012), Kovacs et al. (2012), Zhang et al. (2017), Yang et al. (2017). In the Solomon benchmarks, each customer (depot) has X, and Y coordinates for the location, and the distance matrix is symmetric. In contrast, some publications consider the asymmetric distance matrix where the distance between point A to point B can be different from the distance between point B to point A, for example, Toth and Vigo (1999), Almoustafa et al. (2013).

The instances used in Chapter 4 are provided by an Australian transportation company, whereas the instances used in Chapter 5 are derived from historical data provided by this company. Although the distance matrix is symmetric, it is different compared with the distance matrix used in the literature. In the instances used in Chapters 4 and 5, customers within a suburb have a constant distance from each other, while the distance between two customers in different suburbs is calculated based on the coordinates of the suburbs. In the Solomon benchmark, the size of the distance matrix depends on the number of customers. As mentioned in Arnold et al. (2019), if each customer has a distinct location, the total number of entries needed to store the distance matrix is N^2, where N is the total number of customers plus depot. In contrast, the size of the distance matrix used in Chapter 4 and 5 depends on the number of existing suburbs. This setting reduces the size of the distance matrix.

3

An Efficient Optimisation Procedure for the Workforce Scheduling and Routing Problem: Lagrangian Relaxation and Iterated Local Search

Abstract

This chapter studies the Workforce Scheduling and Routing Problem where certain service providers complete tasks at different locations. The presented optimisation procedure is an amalgamation of the iterated local search and Lagrangian relaxation. This optimisation procedure has been tested on benchmark problems from the literature and showed superior performance in comparison with a previously published implementation of the iterated local search.

3.1 Introduction

This chapter presents a new optimisation algorithm for the Workforce Scheduling and Routing Problem (WSRP). This problem is concerned with the allocation of tasks (requests for service, customers, patients) to the service providers (technicians, nurses). The tasks have different locations, and the service providers need to spend significant time travelling between these locations. The service providers depart from some service centre (depot) and return to this service centre after the completion of all allocated tasks. The service providers may have different skills, and therefore, each task is given a subset of the set of service providers which can be assigned to this task. If a task cannot be allocated to a service provider, this task incurs a penalty which will be referred to as the cost of outsourcing. The goal is to minimise the total cost of travelling and outsourcing.

Even very particular cases of the WSRP such as the travelling salesman problem and the makespan minimisation scheduling problem for parallel identical machines are

NP-hard in the strong sense Garey and Johnson (1979). A wide variety of the optimisation procedures, developed for the WSRP and its variations, includes mixed integer programming with decomposition Laesanklang et al. (2015), Laesanklang et al. (2016); branch-and-price algorithms Dohn et al. (2009), Liu et al. (2017); algorithms based on Lagrangian relaxation Fathollahi-Fard et al. (2018), Gu et al. (2019); iterated local search Xie et al. (2017), Gu et al. (2019); iterated local search with hybrid neighbourhood search Zhou et al. (2020); genetic algorithms Shi et al. (2017), Algethami et al. (2016), Algethami and Landa-Silva (2017), Algethami et al. (2019); large neighbourhood search Goel and Meisel (2013), Braekers et al. (2016a); adaptive large neighbourhood search Kovacs et al. (2012),Guastaroba et al. (2021); greedy randomised adaptive search Xu and Chiu (2001); ant colony optimisation Pereira et al. (2020); artificial bee colony optimisation Yurtkuran et al. (2018); matheuristic Pillac et al. (2013); and constraint programming Polnik et al. (2021).

The optimisation procedure discussed below is an amalgamation of the iterated local search (ILS) Lourenço et al. (2019) and Lagrangian relaxation Fisher (1981). In what follows, this procedure will be referred to as the Lagrangian ILS. The idea of such amalgamation was first introduced in Gu et al. (2019) and stemmed from the observation that the performance of local search often can be improved by permitting the violation of some constraints and by introducing an augmented objective function which comprises the original objective function and a penalty for the violation of the constraints. This phenomenon was used in various algorithms (see, for example, Cordeau et al. (1997), Cordeau et al. (2001), Schneider and Löffler (2019), Ibaraki et al. (2008), Xie et al. (2017), Zhou et al. (2020), Gu et al. (2019), Nagata et al. (2010), Cordeau and Maischberger (2012), Pan et al. (2021)) and can be anticipated given the nature of local search Gendreau and Potvin (2019). Indeed, local search generates a sequence of solutions where each subsequent solution has a better value of the objective function than the preceding one. Since such a sequence of monotonic (in terms of the objective values) feasible solutions that lead to a desired solution may not exist or be difficult to find, the use of the infeasible solutions may significantly facilitate the construction of a sequence that renders a desired solution.

To the best of the author's knowledge, in all publications in which the constraint violation is permitted, the penalty for each constraint is computed as a measure of the violation multiplied by a certain coefficient (weight) and these weights either remain unchanged during the entire optimisation (see, for example, Nagata et al. (2010)), or are updated by multiplying them by some constants which remain the same during the entire optimisation (see, for example, Cordeau et al. (1997), Cordeau et al. (2001), Schneider and Löffler (2019), Ibaraki et al. (2008), Xie et al. (2017), Zhou et al. (2020), Pan et al. (2021), Cordeau and Maischberger (2012)). In both cases, these constants are determined as a result of tedious computational experiments. In contrast, Gu et al. (2019) introduced a mixed integer linear programming formulation which permits to consider the weights as Lagrange multipliers and to choose their initial values as well as to dynamically update them correspondingly. According to Gu et al. (2019), the application of local search to the Lagrangian relaxation of the original problem rather than to the original problem itself is motivated by the observation that this new problem, although remaining very difficult from the computational point of view, is more amenable to local search. In other words, the use of the Lagrangian relaxation is dictated not by the complexity consideration but by the suitability for the optimisation method − local search. This distinguishes Gu et al. (2019) from the conventional Lagrangian relaxation approach where the main goal is to obtain a tight bound for the optimal value of the objective function with, if necessary, the subsequent conversion of the obtained infeasible solution into a feasible one Fisher (1981), Fathollahi-Fard et al. (2018).

The results presented in Gu et al. (2019) indicate that the development of algorithms based on the idea of the amalgamation of ILS and Lagrangian relaxation is a promising direction of research. Having in common the same idea, these algorithms can be quite different, depending on what local search subroutines are used, how these subroutines are implemented and how they interact, as well as on how and when the Lagrange multipliers (weights in the penalty component of the augmented objective function) are updated. This chapter presents a new algorithm (Lagrangian ILS) that significantly outperforms the original algorithm in Gu et al. (2019) both, in terms of speed as well as in terms of solution quality. Although the Lagrangian ILS and the algorithm in Gu et al. (2019) are based on the same idea of amalgamation of ILS and Lagrangian relaxation, the implementation of this idea in these two algorithms is quite different. The superior performance of the Lagrangian ILS was achieved by the following changes:

(a) The structure of the algorithm in Gu et al. (2019), which was mostly the same as in Xie et al. (2017), has been redesigned

- by changing the events which trigger the update of the weights in the penalty for the violation of constraints;
- by changing the events which trigger the perturbation of the best currently known feasible solution which is used to escape a local minimum.

(b) The use of the neighbourhoods in the local search in Gu et al. (2019), which was the same as in Xie et al. (2017), has been changed

- by replacing the search in the neighbourhood generated by three types of transformations of a current solution by the successive search in the three separate neighbourhoods, each for one type of transformations;
- by implementing a new method of choosing the output for each neighbourhood.

(c) The local search procedure in Gu et al. (2019) has been significantly enhanced by implementing the advanced method of the evaluation of the elements of a neighbourhood.

The improvements, outlined in (a), have completely changed the optimisation process which is now a different sequence of applications of local search and perturbations. Furthermore, since the weights are now updated at different stages of optimisation, the objective function at each application of local search also differs from that in Gu et al. (2019). The improvements, outlined in (b), have completely changed the local search subroutine which now successively explores several neighbourhoods and uses a novel method of choosing the output that takes into account the amalgamation of ILS and Lagrangian relaxation. The improvement, mentioned in (c), has dramatically sped up the evaluation of solutions in a neighbourhood.

More specifically, the algorithm in Gu et al. (2019) updates weights at each iteration of the local search after finishing the exploration of the neighbourhood of the current solution. This is the usual point when the weights are updated in typical implementations of the local search with constraints violation (see, for example, Cordeau et al. (1997), Cordeau et al. (2001), Schneider and Löffler (2019), Xie et al. (2017), Zhou et al. (2020), Cordeau and Maischberger (2012), Pan et al. (2021)). In contrast, in the spirit of Lagrangian relaxation, the Lagrangian ILS updates the weights (viewed in this algorithm as Lagrange multipliers) only when it finds a local minimum which is infeasible for the original problem.

In Gu et al. (2019), the perturbation subroutine is called when either a feasible solution has been found (regardless of the quality of this solution), or the limit on the attempts to find a feasible solution is reached. Observe that in Xie et al. (2017), Zhou et al. (2020), Ibaraki et al. (2008), Cordeau and Maischberger (2012) the perturbation is called when the local minimum has been found regardless of its feasibility. In contrast to all these publications, in the Lagrangian ILS, the perturbation subroutine is called if either a feasible local minimum has been found or the limit on the attempts to find such a feasible local minimum has been reached.

The local search in the Lagrangian ILS is significantly more efficient in comparison with that in Gu et al. (2019) partly because it uses an advanced technique for evaluating solutions in the neighbourhood of a current solution. The utilised technique was originally introduced in Nagata et al. (2010) and Vidal et al. (2013). In the Lagrangian ILS, the ideas of Nagata et al. (2010) and Vidal et al. (2013) are further developed by a new integer linear programming formulation that reflects this technique. Consequently, this leads to changes in how the weights in the penalty component of the augmented objective function are initialised and updated.

Another enhancement of the local search in the Lagrangian ILS in comparison with the local search in Gu et al. (2019) is the method of choosing the output solution for a neighbourhood. In Gu et al. (2019), similar to Xie et al. (2017), the local search uses two types of neighbourhoods. For the first type, the output is a solution with the smallest number of unallocated tasks among all solutions with an improved value of the augmented objective function. For the second type, the output is a solution with the smallest value on the augmented objective function. In contrast to Gu et al. (2019), the Lagrangian ILS uses four types of neighbourhoods and chooses the output for each neighbourhood, by considering both the value of the augmented objective function and the value of the original objective function. This improves the entire optimisation procedure by taking into account the nature of the amalgamation of ILS and Lagrangian relaxation.

The remaining part of the chapter is organised as follows. A description of the considered problem is given in Section 3.2. Section 3.3 presents the proposed optimisation algorithm and its subroutines. Section 3.4 provides the results of computational experiments. Section 3.5 concludes the chapter.

3.2 Problem Statement

Following Kovacs et al. (2012), Xie et al. (2017), it is convenient to describe the considered problem, using directed graph $G(V, A)$ with the set of vertices $V = \{0, 1, ..., n, n+1\}$ and the set of arcs A. In set V, vertices 0 and $n + 1$ represent the depot, and the vertices constituting the set $C = \{1, ..., n\}$ represent the tasks. Vertex 0 is used when a departure from the depot is considered and vertex $n + 1$ is used when the arrival at the depot is considered. The route of each service provider is a set of arcs. If arc (i, j) is on the route of a service provider, the service provider must travel directly from the location associated with vertex i to the location associated with vertex j. Hence, the route of each service provider is a directed path from vertex 0 to vertex $n + 1$. The set of arcs A contains the set $A_0 = \{(0, i)|i \in C \cup \{n + 1\}\}$, the set $A_{n+1} = \{(i, n + 1)|i \in C \cup \{0\}\}$, and the set $A_C = \{(i, j)|i \neq j, i \in C, j \in C\}$. In other words, $A = A_C \cup A_0 \cup A_{n+1}$ and the subgraph $G(C, A_C)$ is complete.

For each $i \in C$, let $d_i > 0$ be the duration of the service required by task i. For the sake of convenience, it is assumed that vertex 0 has $d_0 = 0$. Each vertex in C has an associated time window. For $i \in C$, the associated time window $[e_i, l_i]$ designates the time interval when the service required by task i can commence. In addition, a service provider can leave the depot (vertex 0) and return to the depot (vertex $n+1$) only within the given time window $[e_0, l_{n+1}]$. In what follows, it is assumed that $e_0 = 0$.

Each arc $(i, j) \in A$ has the associated travel time $t_{i,j}$, and for any three arcs (i, j), (i, h) and (h, j), the travel times satisfy

$$t_{i,j} \leq t_{i,h} + t_{h,j}. \tag{3.1}$$

The inclusion of an arc (i, j) in the route of a service provider incurs the cost $c_{i,j}$. Since vertices 0 and $n + 1$ represent the same depot, $c_{0,n+1} = 0$ and $t_{0,n+1} = 0$. Consequently, for each $i \in C$, $t_{0,i} \leq l_i$. If the route of a service provider contains the arc $(0, i)$, then the earliest time that the service provider can commence the service at task i is $\max\{t_{0,i}, e_i\}$. If, for $i \in C$ and $j \in C$, the route of a service provider contains the arc (i, j) and the service provider completes the service at task i at time t, then the earliest time that the service can start at task j is $\max\{t + t_{i,j}, e_j\}$. In other words, even if a service provider can arrive at task i prior to the point in time e_i, the service commences only at e_i. Furthermore, there exists an upper bound D on the length of the time interval between the departure of a service provider from the depot and the return of this service provider to the depot.

Each task needs only one service provider, but not all service providers may be qualified for certain tasks. Let K be the set of service providers. For $k \in K$, $i \in C$, the parameter q_i^k is 1 if service provider k is qualified for task i, whereas this assignment is not allowed if $q_i^k = 0$. If task i is not allocated to any service provider, this task must be outsourced at cost o_i.

The goal is to find the routes for service providers which minimises the total cost, including the traveling and outsourcing costs.

Let

$$
\begin{aligned}
b_0^k &= \text{the time when service provider } k \text{ leaves the depot} \\
b_{n+1}^k &= \text{the time when service provider } k \text{ returns to the depot} \\
b_i^k &= \text{the time when service provider } k \text{ starts task } i \\
y_i &= \begin{cases} 1 & \text{if task } i \text{ is outsourced;} \\ 0 & \text{otherwise} \end{cases} \\
x_{i,j}^k &= \begin{cases} 1 & \text{if } i \text{ and } j \text{ are two consecutive tasks in the route} \\ & \text{of service provider } k \text{, i.e., this route contains } (i,j); \\ 0 & \text{otherwise} \end{cases} \\
z_i^k &= \begin{cases} 1 & \text{if task } i \text{ is served by service provider } k; \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

The considered problem can be formulated as follows:

$$
\min \sum_{k \in K} \sum_{(i,j) \in A} c_{i,j} x_{i,j}^k + \sum_{i \in C} o_i y_i \tag{3.2}
$$

subject to:

$$\sum_{k \in K} z_i^k + y_i = 1, \quad \forall i \in C \tag{3.3}$$

$$z_i^k \leq q_i^k, \quad \forall k \in K, \ \forall i \in C \tag{3.4}$$

$$\sum_{j \in V \setminus \{0\}} x_{0,j}^k = 1, \quad \forall k \in K \tag{3.5}$$

$$\sum_{i \in V \setminus \{n+1\}} x_{i,n+1}^k = 1, \quad \forall k \in K \tag{3.6}$$

$$z_h^k = \sum_{i \in V \setminus \{n+1\}} x_{i,h}^k, \quad \forall k \in K, \ \forall h \in C \tag{3.7}$$

$$z_i^k = \sum_{j \in V \setminus \{0\}} x_{i,j}^k, \quad \forall i \in C, \forall k \in K \tag{3.8}$$

$$b_i^k + (d_i + t_{i,j}) x_{i,j}^k \leq b_j^k + l_i(1 - x_{i,j}^k), \quad \forall k \in K, \ \forall (i,j) \in A \tag{3.9}$$

$$e_i \leq b_i^k, \quad \forall k \in K, \ \forall i \in C \cup \{0\} \tag{3.10}$$

$$b_i^k \leq l_i, \quad \forall k \in K, \ \forall i \in C \cup \{n+1\} \tag{3.11}$$

$$b_{n+1}^k - b_0^k \leq D, \quad \forall k \in K \tag{3.12}$$

$$x_{i,j}^k \in \{0,1\}, \quad \forall k \in K, \ \forall (i,j) \in A \tag{3.13}$$

$$z_i^k \in \{0,1\}, \quad \forall k \in K, \forall i \in C \tag{3.14}$$

$$y_i \in \{0,1\}, \quad \forall i \in C \tag{3.15}$$

The objective function (3.2) minimises the sum of the total travel cost and the total cost for outsourcing. Constraints (3.3) stipulate that a task is either outsourced or served by exactly one service provider. Constraints (3.4) ensure that a service provider can be assigned to a task only if this service provider is qualified for this task. Constraints (3.5) guarantee that a service provider can leave the depot at most once, whereas according to (3.6), a service provider can return to the depot at most once. Observe that, $x_{0,n+1}^k = 1$ means that service provider k does not leave the depot at all. Constraints (3.7) and (3.8) enforce that a service provider is assigned to a task if and only if this task is on the route of this service provider. Constraints (3.9) state that if (i, j) is on the route of service provider k, the service at task j can start only after the completion of the service at task i plus the travel time between task i and task j. Constraints (3.10) and (3.11) require that the service of a task should commence within the time window associated with this task. According to the constraints (3.12), the duration of the shift of a service provider cannot exceed the allowed maximum duration. Observe that (3.9) − (3.11) can eliminate subtours. In what follows, the problem defined by (3.2) − (3.15) will be referred to as the original problem.

3.3 Lagrangian ILS Framework

In any local search procedure, the evaluation of the solutions constituting neighbourhoods, which involves the exchange of arcs and the reallocation of visits, is time-consuming. In particular, in the presence of time windows, it is crucial to have an efficient technique for measuring the violation of the right-end points of the time windows (in the Lagrangian ILS as well as in all publications known to the authors, only the right-end points of the time window can be violated). This issue was addressed in Nagata et al. (2010) and then in Vidal et al. (2013) by the technique based on the idea of time warps. Since its introduction, this technique has been successfully used in a number of vehicle routing algorithms, for example, Schneider et al. (2013), Kramer et al. (2015), Hiermann et al. (2016), Xie et al. (2017), François et al. (2019), Pan et al. (2021).

This technique is also the core of the search component of the Lagrangian ILS. The time warps can be introduced as follows. Consider a route

$$(0, i_1), (i_1, i_2), ..., (i_r, n + 1), \tag{3.16}$$

where i_1, ..., i_r are tasks allocated to the corresponding service provider and listed in the order in which this service provider visits their locations. It is convenient to denote $i_0 = 0$ and $i_{r+1} = n+1$. Let b_{i_0} be the time when the service provider leaves the depot in (3.16). The time warps for this route are the values u_{i_1}, ..., $u_{i_{r+1}}$ that can be computed recursively (together with the auxiliary values B_{i_1}, ..., $B_{i_{r+1}}$) using (3.17) − (3.20) below.

$$B_{i_1} = \max\{e_{i_1},\ b_{i_0} + t_{i_0,i_1}\}, \tag{3.17}$$

$$u_{i_j} = \max\{0, B_{i_j} - l_{i_j}\}, \quad 1 \le j \le r + 1, \tag{3.18}$$

$$B_{i_{j+1}} = \max\{e_{i_{j+1}},\ B_{i_j} - u_{i_j} + d_{i_j} + t_{i_j,i_{j+1}}\}, \quad 1 \le j \le r - 1, \tag{3.19}$$

$$B_{i_{r+1}} = B_{i_r} - u_{i_r} + d_{i_r} + t_{i_r,i_{r+1}}. \tag{3.20}$$

It is easy to see, that by virtue of (3.17), B_{i_1} is the earliest possible time when the service of task i_1 can commence. Furthermore, if all time warps are zero, (3.17) and (3.19) indicate that B_{i_1}, ..., B_{i_r} are the earliest time when tasks in (3.16) can commence and $B_{i_{r+1}}$ is the earliest possible arrival time at the depot. On the other hand, since all time warps are zero, (3.18) implies that B_{i_1}, ..., $B_{i_{r+1}}$ do not violate their respective time windows. Therefore, if all time warps are zero, then the corresponding route is feasible with respect to the time windows. Since B_{i_1} is the earliest possible time when the service of task i_1 can commence, by (3.18), $u_{i_1} > 0$ implies that the route (3.16) violates the time window for i_1. Suppose that, for some $1 < j \leq r+1$, $u_{i_j} > 0$ and $u_{i_g} = 0$ for all $1 \leq g < j$. Then, according to (3.18), B_{i_j} violates the time window for the task i_j, and according to (3.17) and (3.19), B_{i_1}, ..., B_{i_j} are the earliest possible times when the service of the tasks i_1, ...,i_j can commence. Hence, the route (3.16) violates the time window for task i_j. Summarising the above discussion, in order to check whether or not the route (3.16) violates at least one time window, it suffices to check whether or not $\sum_{j=1}^{r+1} u_{i_j}$ is zero. Observe that, for any value $u_{i_j} > 0$, the value of $u_{i_{j+1}}$ is the same. This property significantly facilitates the recalculation of $\sum_{j=1}^{r+1} u_{i_j}$ in the course of local search.

The key idea of the Lagrangian ILS is the amalgamation of the iterated local search metaheuristic and the Lagrangian relaxation method. This approach requires an alternative mixed integer linear programming formulation of the WSRP that includes variables that reflect the violation of the time windows and the limit D. Since the zero time warps indicate that the time windows are not violated, the time warps are an ideal building block for such formulation. The formulation (3.21) – (3.39) below is equivalent to (3.2) – (3.15), but in contrast to (3.2) – (3.15), involves the new variables u_i^k, for all $i \in V \setminus \{0\}$ and $k \in K$, and v_k, for all $k \in K$. If task i is allocated to service provider k, then u_i^k is the time warp associated with this task in the route of this service provider. According to the formulation below, each v^k is not less than the violation by the service provider k of the permissible shift duration.

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{i,j} x_{i,j}^k + \sum_{i \in C} o_i y_i \tag{3.21}$$

subject to:

$$\sum_{k\in K} z_i^k + y_i = 1, \quad \forall i \in C \tag{3.22}$$

$$z_i^k \le q_i^k, \quad \forall k \in K,\ \forall i \in C \tag{3.23}$$

$$\sum_{j\in V\setminus\{0\}} x_{0,j}^k = 1, \quad \forall k \in K \tag{3.24}$$

$$\sum_{i\in V\setminus\{n+1\}} x_{i,n+1}^k = 1, \quad \forall k \in K \tag{3.25}$$

$$z_h^k = \sum_{i\in V\setminus\{n+1\}} x_{i,h}^k, \quad \forall k \in K,\ \forall h \in C \tag{3.26}$$

$$z_i^k = \sum_{j\in V\setminus\{0\}} x_{i,j}^k, \quad \forall i \in C,\ \forall k \in K \tag{3.27}$$

$$b_0^k + (d_0 + t_{0,j})x_{0,j}^k \le b_j^k + l_{n+1}(1 - x_{0,j}^k), \quad \forall k \in K,\ \forall j \in C \cup \{n+1\} \tag{3.28}$$

$$b_i^k - u_i^k + (d_i + t_{i,j})x_{i,j}^k \le b_j^k + l_i(1 - x_{i,j}^k), \quad \forall k \in K,\ \forall (i,j) \in A_C \cup A_{n+1} \tag{3.29}$$

$$e_i \le b_i^k, \quad \forall k \in K,\ \forall i \in C \cup \{0\} \tag{3.30}$$

$$b_i^k - l_i \le u_i^k, \quad \forall k \in K,\ \forall i \in C \cup \{n+1\} \tag{3.31}$$

$$b_{n+1}^k - b_0^k + \sum_{i\in V\setminus\{0\}} u_i^k \le D + v_k, \quad \forall k \in K \tag{3.32}$$

$$\sum_{k\in K} \sum_{i\in V\setminus\{0\}} u_i^k \le 0 \tag{3.33}$$

$$\sum_{k\in K} v_k \le 0 \tag{3.34}$$

$$x_{i,j}^k \in \{0,1\}, \quad \forall k \in K,\ \forall (i,j) \in A \tag{3.35}$$

$$z_i^k \in \{0,1\}, \quad \forall k \in K,\ i \in C \tag{3.36}$$

$$y_i \in \{0,1\}, \quad \forall i \in C \tag{3.37}$$

$$u_i^k \ge 0, \quad \forall k \in K,\ i \in V \setminus \{0\} \tag{3.38}$$

$$v_k \ge 0, \quad \forall k \in K \tag{3.39}$$

The objective function (3.21) and the constraints (3.22) − (3.27), (3.30), (3.35) − (3.37) are the same as in the formulation (3.2) − (3.15). Constraints (3.29) correspond to (3.17), (3.19) and (3.20) in the definition of time warps, whereas constraints (3.31) correspond to (3.18). Constraints (3.33) and (3.38) ensure that time warps u_i^k are zero, whereas the constraints (3.34) and (3.39) enforce that v_k are zero. Consequently, constraints (3.29), (3.31), and (3.32) become constraints (3.9), (3.11), and (3.12) in the formulation (3.2) − (3.15), respectively.

The dualisation of (3.33) and (3.34), using Lagrange multipliers $\alpha \geq 0$ and $\beta \geq 0$, gives the following Lagrangian relaxation of the mixed integer linear program (3.21) − (3.39)

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{i,j} x_{i,j}^k + \sum_{i \in C} o_i y_i + \alpha \sum_{k \in K} \sum_{i \in V \setminus \{0\}} u_i^k + \beta \sum_{k \in K} v_k \tag{3.40}$$

subject to:

$$(3.22) - (3.32), (3.35) - (3.39)$$

In what follows, this Lagrangian relaxation will be referred to as LR problem.

The presented optimisation procedure is comprised of the following main components:

- subroutine INITIAL that constructs a feasible solution for the given instance of the problem (3.2) − (3.15) which is the best currently known feasible solution at the start of the optimisation procedure;
- subroutine START(s) that perturbs the solution s which is the output of the subroutine INITIAL;
- subroutine SEARCH(s) which is a local search procedure for the LR problem that starts by exploring the neighbourhood of s;
- subroutine ADJUST(α, β, s) that updates α and β, when the current local minimum s is infeasible for the original problem;
- subroutine PERTURB(s, h) that perturbs the best currently known feasible solution s, taking into account the number of runs h which has failed to improve s;
- subroutine WEIGHTS(s') that computes the initial values of α and β, using either $s' =$ START(s^*) or $s' =$ PERTURB(s^*, h) where s^* is the best currently known feasible solution.

Let $f(s)$ be the objective function value for a solution s of the mixed integer linear program (3.2) − (3.15). The pseudocode below outlines the Lagrangian ILS.

Lagrangian ILS

1: $s^* \leftarrow$ INITIAL
2: $s' \leftarrow$ START(s^*)
3: $h \leftarrow 1$
4: **while** $h \leq M$ **do**
5:     **if** s' is feasible **and** $f(s') < f(s^*)$ **then**
6:         $s^* \leftarrow s'$
7:     **end if**
8:     $\{\alpha, \beta\} \leftarrow$ WEIGHTS(s')
9:     $s' \leftarrow$ SEARCH(s')
10:     $e \leftarrow 1$
11:     **while** $e \leq E$ **and** s' is infeasible **do**
12:         $\{\alpha, \beta\} \leftarrow$ ADJUST(α, β, s')
13:         $s' \leftarrow$ SEARCH(s')
14:         $e \leftarrow e + 1$
15:     **end while**
16:     **if** s' is feasible **and** $f(s') < f(s^*)$ **then**
17:         $s^* \leftarrow s'$
18:         $h \leftarrow 0$
19:     **end if**
20:     $s' \leftarrow$ PERTURB(s^*, h)
21:     $h \leftarrow h + 1$
22: **end while**
23: **return** s^*

The Lagrangian ILS starts with the subroutine INITIAL (line 1) which generates a feasible solution for the original problem, that is the problem (3.2) – (3.15). Until a better solution for the original problem has been found, the solution generated by INITIAL is considered as the best currently known solution. In the pseudocode above, at all stages of the Lagrangian ILS, the best currently known solution for the original problem is denoted by s^*.

The parameter M (line 4) determines the maximal permissible number of consecutive attempts (WHILE loop lines 4 – 22) to find a feasible solution for the original problem with a better value of the objective function (3.2). Each such attempt starts with a solution which is a perturbation of the best currently known solution s^* for the original problem. For the first attempt (the first iteration of the WHILE loop lines 4 – 22), the perturbed solution is produced by the subroutine START (line 2), whereas, for all subsequent attempts, the perturbed solutions are generated by the subroutine PERTURB (line 20).

Each attempt to find a better solution for the original problem is a sequence of applications of local search. Each such application (a call of the subroutine SEARCH) finds a local minimum for the LR problem. The first call of SEARCH (line 9) is preceded by the call of the subroutine WEIGHTS (line 8) that computes the initial values of Lagrange multipliers (the initial weights specifying the penalty for the violation of constraints (3.33), (3.34)). Each of the subsequent calls of SEARCH during an iteration of the WHILE loop lines $4-22$ is preceded by the adjustment of the Lagrange multipliers (subroutine ADJUST in line 12). The repetition of applications of local search (WHILE loop lines $11-15$) terminates when either the number of applications of local search exceeds the given permissible number E, or the current local minimum for the LR problem is feasible for the original problem. The starting solution of the local search performed by the subroutine SEARCH, called in the WHILE loop lines $11-15$, is the local minimum found as a result of the previous call of the subroutine SEARCH.

3.3.1 Subroutine INITIAL

The subroutine INITIAL is an iterative algorithm that constructs a feasible solution for the original problem, using a list of tasks and a list of service providers. The service providers are listed in non-increasing order of $\sum_{i\in C} q_i^k$, $k \in K$, which is the number of tasks that service provider k is qualified to serve. The tasks are ordered in a non-decreasing order of the angle in their polar coordinates where the pole is the depot and the polar axis is specified by the direction to the location of a randomly chosen task. The idea to use polar coordinates in constructing a feasible solution can be traced back at least to Gillett and Miller (1974). At the beginning of the first iteration, the current list of tasks contains all tasks; the current routes of service providers are empty; and the current value of the objective function is zero. At each iteration, the algorithm scans the list of service providers and attempts to insert the first task from the current list of tasks into the route of each service provider in such a way that this insertion does not violate the route feasibility. If there exist multiple feasible positions, then this task is inserted into a route and a feasible position in this route which gives the smallest increase in the current value of the objective function. If there is no feasible position available, this task is outsourced (since outsourcing is permissible, this does not lead to an infeasible solution) and the current value of the objective function is increased by the cost of outsourcing. In both cases, the current list of tasks is updated by eliminating the first task in this list and the next iteration of the subroutine INITIAL begins. This procedure terminates when the current list of tasks becomes empty, i.e., each task is either allocated to some service provider or outsourced. The rationale behind the choice of this particular method is two-fold: it generates a reasonably good solution and it is similar to the method in Xie et al. (2017) which eliminates the impact of the starting solution in the comparison of the Lagrangian ILS and the algorithm in Xie et al. (2017).

3.3.2 Subroutine START

The input for the subroutine START is the solution constructed by the subroutine INITIAL. If this solution does not have outsourced tasks, then the output of START is the same solution. In the case when the input has outsourced tasks, the output of START is produced by randomly choosing one of the outsourced tasks, and then inserting it into the route of one of the qualified service providers in such a way that this insertion results in the smallest increase of (3.40) with $\alpha = \beta = 1$.

3.3.3 Subroutine WEIGHTS

The input of the subroutine WEIGHTS is an output of either the subroutine START or the subroutine PERTURB. The output of the subroutine WEIGHTS is α and β in (3.40), which are the weights used to calculate the penalty for the violation of constraints. For input solution s, the violation of time windows $u_i^k(s)$, and the violation of permissible shift duration $v_k(s)$ are calculated based on the time warp technique in Vidal et al. (2013). The subroutine WEIGHTS computes the weights in the penalty for the violation of constraints (the values of Lagrange multipliers) as follows:

$$\alpha = \sum_{k \in K} \sum_{i \in V \setminus \{0\}} u_i^k(s) \quad \text{and} \quad \beta = \sum_{k \in K} v_k(s)$$

So, each call of the subroutine WEIGHTS results in Lagrange multipliers (weights) that reflect the violation of constraints by the input solution.

3.3.4 Subroutine SEARCH

The subroutine SEARCH(s) attempts to solve the Lagrangian relaxation problem for fixed values of the Lagrange multipliers (for fixed weights in the augmented objective function), using four local search optimisation procedures, each with one of the four operators N_0, N_1, N_2, N_3. The operators implemented in this subroutine are commonly used in the field of vehicle routing and can be found in many algorithms reported in the literature (see for example, Laporte et al. (2000), Bräysy and Gendreau (2005a), Bräysy and Gendreau (2005b), and Kindervater and Savelsbergh (2018)). Each operator N_i transforms an input solution s, by applying transformations (moves) from the set of transformations associated with this operator, and returns as the result some solution s' (denoted $s' = N_i(s)$) where s' is either the input solution s, or one of the transformations of s.

The set of transformations, associated with N_0, is comprised of all transformations that

- for two routes, interchange a sequence of up to two consecutive visits in one route with a sequence of up to two consecutive visits in another route, including the transformations that only use a sequence from one route and an insertion position in another;
- interchange a sequence of up to two consecutive visits in a route (the tasks in this sequence become outsourced) with at most one outsourced task, including the transformations which either do not use an outsourced task or instead of the sequence of visits use only an insertion position in the route.

The set of transformations associated with operator N_1 is comprised of all transformations that extract one visit from a route and insert it into a different position of the same route. Operator N_2 is similar to N_1, but, instead of one visit, each transformation performed by N_2 extracts a sequence of two consecutive visits and inserts this sequence into a different position of the same route. Each transformation performed by N_3 reverses the order of a sequence of consecutive visits in a route.

Although the transformations associated with operators N_0, N_1, N_2, and N_3 are among the most commonly used in the vehicle routing algorithms, the rule of choosing $N_i(s)$ differs from the rules reported in the literature. This rule is the same for all four operators

N_0, N_1, N_2, N_3 and reflects the nature of the Lagrangian ILS: it aims at improving the value of the augmented objective function $f_{LR}(\cdot)$ but at the same time takes into account the minimisation of the original objective function $f(\cdot)$. For a current solution s, according to this rule, if there is no solution in the neighbourhood of s with the value of the augmented objective function smaller than $f_{LR}(s)$, then $s = N_i(s)$, i.e., the output is the current solution s. If there are solutions in the neighbourhood of s that have the value of the augmented objective function smaller than $f_{LR}(s)$, then the output is one of them, say s', that has the smallest value of $\left\lfloor \frac{f(s')}{\psi} \right\rfloor$ where ψ is a fixed positive integer which is the same for all four operators N_0, N_1, N_2, N_3. If there are several such solutions, the output is one of them with the smallest value of $f_{LR}(s')$. Observe that if $\psi = 1$, the output is the solution with the smallest value of the original objective function among all solutions with the value of the augmented objective function less than $f_{LR}(s)$, whereas if ψ is very large, then the output is a solution with the smallest value of the augmented objective function.

The subroutine SEARCH(s) requires an input solution s and can be outlined as follows:

SEARCH(s)

```
1:  repeat
2:     s̄ ← s
3:     for i ← 0; i ≤ 3; i ← i + 1 do
4:        repeat
5:           s' ← s
6:           s ← N_i(s)
7:        until f_LR(s') = f_LR(s)
8:     end for
9:  until f_LR(s̄) = f_LR(s)
10: return s
```

The subroutine SEARCH(s) with an input solution s is an iterative optimisation procedure (REPEAT loop lines $1 - 9$) where each iteration is comprised of the application of four local search algorithms (FOR loop lines $3 - 8$). The first local search algorithm uses the operator N_0, the second uses the operator N_1, the third uses the operator N_2, and the fourth uses the operator N_3. At the first iteration, the local search with the operator N_0 is applied to the input solution s. When the local search with the operator N_i (REPEAT loop lines $4 - 7$) finds a local minimum with respect to N_i, this local minimum is used as an input to the local search with operator $N_{(i+1)\text{mod}4}$. The subroutine SEARCH terminates when all four local search algorithms fail to further improve the value of the augmented objective function.

3.3.5 Adjustment of multipliers

As has been discussed above, the weights α and β of the penalty for the violation of constraints (3.33), (3.34) are computed prior to each call of the subroutine SEARCH and remain unchanged till the next call of this subroutine, i.e., remain unchanged during each run of SEARCH. Prior to a call of the subroutine SEARCH, α and β are computed either by the subroutine WEIGHTS or by the subroutine ADJUST. If an optimal solution $\hat{s}$ of the Lagrangian relaxation (3.40), (3.22) $-$ (3.32), (3.35) $-$ (3.39) can be found, then according to a commonly used version of the Lagrangian relaxation method Fisher (1981), Guignard (2003), the weights α and β are updated to

$$\alpha + \tau \sum_{k\in K} \sum_{i\in V\setminus\{0\}} u_i^k(\hat{s}) \quad \text{and} \quad \beta + \tau \sum_{k\in K} v_k(\hat{s}) \tag{3.41}$$

where $u_i^k(\hat{s})$, $i \in V \setminus \{0\}$, $k \in K$, and $v_k(\hat{s})$, $k \in K$ are the violations of constraints (3.33), (3.34) caused by $\hat{s}$, and

$$\tau = \frac{\eta\left(f(s^*) - f_{LR}(\hat{s})\right)}{\left(\sum_{k\in K}\sum_{i\in V\setminus\{0\}} u_i^k(\hat{s})\right)^2 + \left(\sum_{k\in K} v_k(\hat{s})\right)^2} \tag{3.42}$$

where η is a positive parameter, $f(\cdot)$ is the original objective function, $f_{LR}(\cdot)$ is the augmented objective function (the objective function for the LR problem), and s^* is the best currently known solution for the original problem. Since the subroutine SEARCH cannot guarantee the optimal solution $\hat{s}$, instead of (3.41), the Lagrangian ILS uses

$$\alpha + \tau \sum_{k\in K} \sum_{i\in V\setminus\{0\}} u_i^k(s) \quad \text{and} \quad \beta + \tau \sum_{k\in K} v_k(s) \tag{3.43}$$

where $u_i^k(s)$, $i \in V \setminus \{0\}$, $k \in K$, and $v_k(s)$, $k \in K$ are the violations of constraints (3.33), (3.34) by the output s of the subroutine SEARCH, and

$$\tau = \frac{\gamma\, f(s^*)}{\left(\sum_{k\in K}\sum_{i\in V\setminus\{0\}} u_i^k(s)\right)^2 + \left(\sum_{k\in K} v_k(s)\right)^2} \tag{3.44}$$

where γ is a positive parameter. The choice of γ and an analysis of the sensitivity of the optimisation procedure to the different values of γ will be discussed in Section 3.4.2. Observe that $f(s^*) - f_{LR}(\hat{s})$ in (3.42) cannot be replaced by $f(s^*) - f_{LR}(s)$, since this may results in a negative τ.

3.3.6 Perturbation

The perturbation mechanism, used in the Lagrangian ILS is the same as in Xie et al. (2017). If there exists at least one outsourced task, the subroutine PERTURB randomly chooses an outsourced task; evaluates all possible insertions of this task into the existing routes; and inserts it in the position that gives the smallest value of (3.40) when $\alpha = \beta = 1$. If there is no outsourced task, this stage of the perturbation is skipped. Then the subroutine PERTURB repetitively applies the exchange operation that, randomly chooses two routes; randomly chooses a sequence of consecutive visits in each of the chosen routes; and interchanges these sequences. The number of applications of the exchange operation depends on the number of consecutive calls of the subroutine SEARCH that result in no improvement of the value of the objective function (3.2) of the original problem, which is the counter h in the pseudocode of the Lagrangian ILS. Starting with one exchange of two sequences, the number of exchanges (iterations) increases by one each time when h increases by a chosen increment.

3.4 Computational experiments

This section presents the results of computational experiments aimed at evaluating the performance of the Lagrangian ILS by comparing it with the performance of CPLEX 12.10, the state-of-the-art iterated local search algorithm Xie et al. (2017), and a modification of the algorithm in Xie et al. (2017) presented in Zhou et al. (2020). The iterated local search in Xie et al. (2017) will be referred to as ILS and its modification presented in Zhou et al. (2020) will be referred to as the iterated local search with hybrid neighbourhood search (ILS-HNS). The algorithm in Zhou et al. (2020) uses two local search subroutines, one of which is the local search procedure described in Xie et al. (2017). Another local search subroutine in Zhou et al. (2020), at each iteration, outsources several already scheduled tasks and then, using the entire set of outsourced tasks, tries to insert some of them into the existing routes. At each cycle of optimisation, one of these two subroutines is chosen randomly and is applied until the limit on the number of failed attempts to improve the value of the objective function is reached. After that, the perturbation is applied and a new cycle starts by randomly choosing one of the two local search subroutines.

All computational experiments were conducted on a computer with Intel Xeon CPU E5-2697 v3 2.60GHz and 4GB RAM. To eliminate the differences that may be caused by different hardware and a different implementation of the algorithm in Xie et al. (2017), the author produced an implementation of the ILS which will be referred to as the Implemented ILS. The Implemented ILS and the Lagrangian ILS were programmed in C++ and compiled with g++, using the optimisation level O3, which is aimed at reducing the running time of the executable file. Moreover, both implementations use the same computer code for the evaluation of each neighbourhood thereby eliminating the differences that may be caused by different programming techniques or compilers.

As far as the comparison with the ILS is concerned, i.e., the comparison with the algorithm in Xie et al. (2017), the computational experiments use the benchmark instances comprised of 25, 50, and 100 tasks. These instances are the same as in Xie et al. (2017) (please also see Kovacs et al. (2012)). The instances with 100 tasks can be downloaded from `https://prolog.univie.ac.at/research/STRSP/` and are based on the Solomon data sets R101, R103, R201, R203, C101, C103, C201, C203, RC101, RC103, RC201, RC203 in Solomon (1987) with additional compatibility restrictions in Cordeau et al. (2010). The instances belong to two categories, "NoTeam Reduced" and "NoTeam Complete". For instances within the category "NoTeam Complete", the number of service providers is sufficiently large such that all tasks can be assigned to a service provider. In contrast to "NoTeam Complete", the number of service providers used in the instances within the category "NoTeam Reduced" is reduced such that it is not possible to assign all tasks Kovacs et al. (2012). The instances with 25 and 50 tasks were generated according to Kovacs et al. (2012) by taking the first 25 and 50 tasks in the instances with 100 tasks and by taking a few service providers in the instances from "NoTeam Complete". The number of service providers in each instance with 25 or 50 tasks is reported in Table A.1 in A.1. CPLEX was able to obtain an optimal solution for instances with 25 and 50 tasks. Its performance on these instances is reported in Table A.1 in A.1 and is summarised in Table 3.1. As far as the instances with 100 tasks are concerned, CPLEX obtained an optimal solution only for a few of them and most of them could not find an optimal solution or even a feasible solution within a time limit of 4 hours and with a memory limit of 4GB. The corresponding results are reported in Tables A.3 and A.5 in A.1.

The ILS in Xie et al. (2017) solves each instance using a multi-start mechanism that runs the iterated local search five times, each time with a new starting solution. The output of one application of this multi-start algorithm is the best solution obtained in these 5 runs. Furthermore, in Xie et al. (2017), the ILS is applied to each instance 5 times. Therefore, the Lagrangian ILS was applied to each instance 25 times, each time with a different starting solution, which was generated by one application of the subroutine INITIAL. These 25 applications were split into 5 groups, each comprised of 5 applications. The output obtained by a group is defined as the best solution obtained by the attempts constituting the group. So, each group is a counterpart of one application of the ILS. Therefore, the total time required by all five attempts constituting a group is a counterpart of the time required by one application of the ILS. For the comparison with the results presented in Xie et al. (2017), only the result of each group and the time required by each group were recorded.

The modification of the ILS, presented in Zhou et al. (2020), is compared in Zhou et al. (2020) with the ILS, using the computational experiments methodology in Xie et al. (2017) and a subset of instances used in Xie et al. (2017). Therefore, the Lagrangian ILS is compared below with the ILS-HNS, using the computational experiments methodology described above. This comparison uses the information provided in Zhou et al. (2020) which reports only the results for instances in Xie et al. (2017) with 100 tasks and does not provide any information on the computational time required by the ILS-HNS on these instances.

Parameter settings are identical for the Implemented ILS and ILS as in Xie et al. (2017). For Lagrangian ILS, the maximum number of exchange operations in the subroutine PERTURB is five, which is the same as ILS; the parameter E is 100; the parameter M is computed according to $\omega(|C| + \lambda|K|)$, where C is the set of all tasks; K is the set of all service providers; ω is a parameter to control M; $\lambda = 10$. The Lagrangian ILS increases the number of exchange operations in perturbation after each $M/5$ sequential iterations that fails to improve the value of the objective function.

The solution quality of the studied algorithms is measured by the percentage relative difference

$$\frac{Reference - Obj}{Reference} \times 100 \tag{3.45}$$

where Obj is the objective value obtained by the corresponding algorithm and $Reference$ is the objective value either presented in Xie et al. (2017) (Table 3.2 and 3.3), or produced by CPLEX (Table 3.1), or obtained by Lagrangian ILS (Table 3.6).

For readers' convenience, the computational results for performance comparisons are shown in overview tables, while the detailed results for each instance are provided in A.1 and A.2. In the overview tables, the instances are grouped according to the geographic distribution (C, R or RC), compatibility restriction (5x4, 6x6, 7x4), and the number of tasks (25, 50, 100). For example, small (25 tasks) instances "C101 5x4", "C201 5x4", and "C203 5x4" form the group named "C 5x4". In the overview tables, each row displays the average results on instances of the same group. The values in the last row of each overview table are the average values over all instances, i.e., they are the same as in Tables in A.1 and A.2. Observe that, since the groups in Table 3.1 may contain different numbers of instances, a value in the last row in this table may not be the average over the values in the corresponding column.

In what follows, Section 3.4.1 compares the performance of the Lagrangian ILS with the performance of CPLEX, the performance of the Implemented ILS, the performance of the ILS reported in Xie et al. (2017), and the performance of the ILS-HNS reported in Zhou et al. (2020). Section 3.4.2 analyses how the performance of the Lagrangian ILS changes with the variation of several parameters.

3.4.1 Comparison of the performance

This subsection reports the results obtained by the Lagrangian ILS, ILS, Implemented ILS, and ILS-HNS for $\omega = 1$. In addition, $\psi = 50$ and $\gamma = 2$ are used for the Lagrangian ILS. The results for the small (25 tasks) and medium (50 tasks) instances are reported in Table 3.1. Tables 3.2 and 3.3 present results for the two categories, "NoTeam Reduced" and "NoTeam Complete" comprising large (100 tasks) instances. To compare the performance of Lagrangian ILS, ILS, Implemented ILS, and ILS-HNS, the comprehensive Wilcoxon tests with Bonferroni correction are conducted and p-values are displayed in

Tables 3.4 and 3.5.

The performance of the Lagrangian ILS, ILS in Xie et al. (2017), Implemented ILS, and CPLEX on the small (25 tasks) and medium (50 tasks) instances is given in Table 3.1. The ILS-HNS is not included in this table since Zhou et al. (2020) does not report the performance on these instances. The first three columns in this table, as well as in Table 3.2 and Table 3.3, are the instance group's name, the number of tasks, and the average number of service providers. For CPLEX, the optimal objective value and computational time are given in the columns Opt^* and sec^*, respectively. Each column $\%^*$ gives the percentage difference (3.45) of the objective value with respect to the objective value obtained by CPLEX. Each column sec_a gives the average computation time required by the respective optimisation procedure. As has been discussed above, the Lagrangian ILS is applied to each instance 25 times and these 25 attempts constitute 5 groups. The column $|Opt|$ shows the number of groups that obtained an optimal solution. For the Lagrangian ILS, the column sec_a is complemented by columns sec_w and sec_b which give the worst and best time required by the Lagrangian ILS. To facilitate the reading, the best values obtained by various algorithms are in bold.

Table 3.1: Comparison between the performance of CPLEX, ILS in Xie et al. (2017), Implemented ILS and Lagrangian ILS on small and medium instances

			CPLEX		**ILS**		**Implemented ILS**		**Lagrangian ILS**				
Instances	$\lvert C\rvert$	$\lvert K\rvert$	Opt^*	sec^*	$\%^*$	sec_a	$\%^*$	sec_a	$\%^*$	$\lvert Opt\rvert$	sec_a	sec_w	sec_b
C5x4	25	2.67	656.87	7.84	-0.15	0.12	-0.03	0.20	**0.00**	5.00	0.20	0.67	**0.00**
R5x4	25	3.00	1643.06	0.08	**0.00**	0.13	-0.08	0.50	**0.00**	5.00	0.30	0.50	**0.00**
RC5x4	25	3.50	663.73	1.87	-0.01	0.22	**0.00**	0.70	-0.23	3.00	0.30	1.00	**0.00**
C6x6	25	2.67	1025.02	1.07	**0.00**	0.04	-0.56	0.27	**0.00**	5.00	0.20	1.00	**0.00**
R6x6	25	3.00	2117.24	0.07	-1.84	0.18	-0.77	0.60	**0.00**	5.00	0.30	1.00	**0.00**
RC6x6	25	3.50	1295.35	3.52	**0.00**	0.19	**0.00**	0.50	**0.00**	5.00	0.30	1.00	**0.00**
C7x4	25	3.00	720.87	59.55	**0.00**	0.06	-0.83	0.25	**0.00**	5.00	0.15	0.75	**0.00**
R7x4	25	2.67	1418.91	38.44	**0.00**	0.07	-0.53	0.20	**0.00**	5.00	0.20	0.67	**0.00**
RC7x4	25	3.50	1318.62	0.27	**0.00**	0.09	-0.10	0.40	**0.00**	5.00	0.20	1.00	**0.00**
C5x4	50	5.00	844.77	0.38	**0.00**	**0.94**	**0.00**	4.30	**0.00**	5.00	1.80	2.50	1.50
R5x4	50	5.00	2807.69	8.19	-0.26	4.49	-0.99	7.20	**0.00**	5.00	**3.60**	5.00	**3.00**
C6x6	50	5.00	1179.39	28.74	**0.00**	**1.31**	**0.00**	5.10	**0.00**	5.00	2.50	3.00	2.00
R6x6	50	5.00	3419.01	52.61	-0.07	**2.31**	-0.10	6.40	**0.00**	5.00	3.90	5.00	3.00
C7x4	50	5.00	1334.38	0.44	-0.42	**0.82**	**0.00**	2.60	**0.00**	5.00	2.40	3.00	2.00
R7x4	50	5.00	3008.52	23.31	-0.06	**1.47**	-0.06	4.60	**0.00**	5.00	2.40	3.00	1.50
Average			1469.98	17.69	-0.16	**0.72**	-0.31	1.97	**-0.01**	4.89	1.10	1.77	0.74

CPLEX can find optimal solutions for all the instance groups within 60 seconds. It is worth pointing out that introducing z_h^k and constraint (3.7) in the MIP model, which is not common in the workforce scheduling and routing literature, can dramatically reduce the solution time for CPLEX. For groups "R 5x4" and "R 6x6" with 25 tasks, and for groups "C 5x4" and "C 7x4" with 50 tasks, CPLEX surprisingly requires less time than the other algorithms on average. In Table 3.1, a zero relative difference indicates that the algorithm can constantly obtain optimal objective value in all 5 runs for the corresponding instance group. It can be seen that the Lagrangian ILS obtains the optimal objective value constantly for 14 out of 15 groups. In contrast, the ILS in Xie et al. (2017) obtains optimal objective value constantly for 8 out of 15 groups, and the Implemented ILS achieves 5 out of 15 groups. The Lagrangian ILS also produces a better average objective value (displayed in the last row of Table 3.1) than the Implemented ILS and ILS.

The Lagrangian ILS described in this chapter is a new version of the algorithm described in Gu et al. (2019). As mentioned in the introduction of this chapter, although both versions use the idea of amalgamation of the ILS and the Lagrangian relaxation, the implementation on the new version is quite different compared with the original version. The original version of the Lagrangian ILS was only tested on instances with 25 and 50 tasks. In many of these instances, the original version of the Lagrangian ILS fails to constantly produce the optimal solutions. In contrast, the new version described in this chapter constantly finds optimal solutions in almost every instance with 25 and 50 tasks. Therefore, the Lagrangian ILS significantly outperforms the original version of this algorithm in terms of solution quality. Furthermore, the computational time required by the new version of the Lagrangian ILS is also less than the time required by the original version of the algorithm.

Tables 3.2 and 3.3 compare the performance of Lagrangian ILS, ILS, Implemented ILS, and ILS-HNS on large (100 tasks) instances from "NoTeam Reduced" and "NoTeam Complete", respectively. In these two tables, the columns Average, Worst, and Best show the average, worst, and best objective values reported in Xie et al. (2017). The columns $\%_a$, $\%_w$, and $\%_b$ give the percentage difference of the average, worst, and best objective values. According to (3.45), a positive (negative) percentage difference indicates that the corresponding algorithm produces better (worse) objective values relative to the value produced by the ILS. The objective values obtained by the Lagrangian ILS are in bold if they are better than the values obtained by ILS, Implemented ILS, and ILS-HNS. The computational times required by the Lagrangian ILS are in bold if they are smaller than the time required by ILS and Implemented ILS. Since the Implemented ILS is just a re-implementation of the ILS in Xie et al. (2017), for the Implemented ILS, Tables 3.2 and 3.3 contain only the average objective values and the average computational times.

The worst and best objective values on each instance obtained by the Implemented ILS can be found in Tables A.3 and A.5 in A.1.

Table 3.2: Comparison between ILS in Xie et al. (2017), Implemented ILS, ILS-HNS in Zhou et al. (2020), Lagrangian ILS on large instances from category "NoTeam Reduced"

		ILS				Implemented ILS		ILS-HNS			Lagrangian ILS			
Inst	$\|K\|$	Average	Worst	Best	sec_a	$\%_a$	sec_a	$\%_a$	$\%_w$	$\%_b$	$\%_a$	$\%_w$	$\%_b$	sec_a
C5x4	6.00	3424.33	3477.38	3344.75	**24.75**	0.26	68.55	-3.37	-2.77	-4.61	**2.14**	**2.68**	**0.53**	28.30
R5x4	8.00	3138.24	3195.59	**3098.84**	38.32	-0.48	115.45	-6.17	-5.09	-6.87	**0.30**	**1.61**	-0.19	**31.55**
RC5x4	8.00	3256.57	3315.27	3201.97	28.58	0.39	103.95	-6.15	-8.31	-4.88	**2.03**	**3.21**	**0.97**	**26.90**
C6x6	6.00	4638.83	4698.82	4599.93	**27.98**	-0.33	67.15	-3.13	-2.95	-2.94	**1.41**	**1.79**	**1.19**	40.10
R6x6	8.50	3583.12	3651.18	3532.18	**43.27**	-0.52	142.25	-11.02	-13.26	-8.52	**0.11**	**0.60**	**0.59**	45.30
RC6x6	8.00	3631.39	3701.79	3562.49	**35.09**	-0.36	113.65	-18.75	-19.80	-18.12	**2.40**	**2.45**	**0.94**	42.70
C7x4	6.50	3112.48	3180.04	3074.77	**15.67**	0.11	45.70	-2.38	-1.87	-2.56	**1.38**	**2.27**	**0.46**	22.25
R7x4	9.50	3090.79	3138.41	3035.33	21.64	0.12	85.80	-3.25	-2.67	-3.78	**0.86**	**1.15**	**0.61**	**20.80**
RC7x4	8.50	3324.75	3368.44	3295.20	**18.33**	-0.51	81.60	-6.35	-7.91	-3.82	**0.49**	**1.02**	**0.35**	20.10
Average		3466.72	3525.21	3416.16	**28.18**	-0.15	91.57	-6.73	-7.18	-6.23	**1.23**	**1.86**	**0.60**	30.89

In Table 2, the Lagrangian ILS produces a better average objective value than that produced by ILS in Xie et al. (2017), Implemented ILS and ILS-HNS in Zhou et al. (2020) for all instance groups within "NoTeam Reduced". For worst and best objective values, the Lagrangian ILS outperforms ILS, Implemented ILS, and ILS-HNS for 8 out of 9 instance groups. Moreover, the Lagrangian ILS requires a noticeably shorter computational time compared with the Implemented ILS. However, a minor difference is observed regarding the computation time between the Lagrangian ILS and ILS. The Implemented ILS is comparable to the ILS in respect of the solution quality but consumes significantly longer computational time.

In Table 3, the Lagrangian ILS obtains the best results for 8 out of 9 instance groups with regard to the average and worst objective values among all the algorithms. For the best objective values, the Lagrangian ILS outperforms all the algorithms in 7 out of 9

Table 3.3: Comparison between ILS in Xie et al. (2017), Implemented ILS, ILS-HNS in Zhou et al. (2020), Lagrangian ILS on large instances from category "NoTeam Complete"

		ILS				Implemented ILS		ILS-HNS			Lagrangian ILS			
Inst	$\|K\|$	Average	Worst	Best	sec_a	$\%_a$	sec_a	$\%_a$	$\%_w$	$\%_b$	$\%_a$	$\%_w$	$\%_b$	sec_a
C5x4	12.50	1087.88	1097.41	1080.58	25.68	-0.13	90.15	**0.71**	**1.24**	**0.19**	0.50	0.70	0.08	**19.40**
R5x4	16.00	1362.17	1370.76	1355.67	38.46	-0.34	318.70	0.13	-0.03	0.10	**0.60**	**0.94**	**0.24**	**37.40**
RC5x4	15.50	1455.74	1470.94	1436.95	40.59	0.41	270.80	0.55	0.51	0.20	**1.58**	**2.17**	**0.61**	**35.05**
C6x6	11.50	854.99	866.82	846.22	46.46	0.42	163.95	0.49	0.98	**0.18**	**0.81**	**1.77**	0.14	**36.80**
R6x6	16.50	1272.40	1280.55	1267.48	63.25	0.10	399.30	-0.54	-1.27	-0.08	**0.63**	**1.09**	**0.45**	**60.15**
RC6x6	16.00	1344.67	1358.40	1333.02	62.48	0.31	329.00	-0.26	-0.53	0.47	**1.48**	**1.86**	**1.10**	**52.80**
C7x4	12.50	1253.19	1269.70	1242.65	19.68	-1.28	83.85	0.44	1.02	0.19	**0.66**	**1.20**	**0.31**	**14.85**
R7x4	19.00	1427.00	1438.08	1420.31	30.16	-1.15	245.60	-0.23	-0.71	0.08	**0.31**	**0.58**	**0.21**	**26.15**
RC7x4	16.00	1550.13	1563.53	1540.31	26.74	0.00	185.20	-0.07	-0.33	0.51	**1.17**	**0.91**	**1.13**	**20.40**
Average		1289.79	1301.80	1280.35	39.28	-0.18	231.84	0.13	0.10	0.20	**0.86**	**1.25**	**0.48**	**33.67**

instance groups. Moreover, the Lagrangian ILS is also faster than the Implemented ILS and ILS. Same as in Table 3.2, the Implemented ILS is on par with ILS in terms of solution quality but is much slower. This difference may be caused by different hardware and/or different computer code. It should be noted that in Xie et al. (2017), the experiments were conducted on a computer with Intel Core i5-3570 3.40GHz which is faster than the computer used for experiments in this chapter. Given that the Lagrangian ILS and Implemented ILS run on the same hardware and share the computer code for the evaluation of each neighbourhood, the superior performance of the Lagrangian ILS on both solution quality and computation time is mainly attributed to the novel design of the subroutine SEARCH resulting in a highly efficient global search capability.

To statistically compare the performance of the Lagrangian ILS, ILS in Xie et al. (2017), Implemented ILS, and ILS-HNS in Zhou et al. (2020), the Wilcoxon test Conover (1999) with Bonferroni correction Demšar (2006) at 95% confidence interval is applied based on the results for each instance (see Tables A.1, A.2, A.3, A.4, A.5 in A.1). In addition to the results obtained by the Lagrangian ILS with $\omega = 1$ (see Tables in A.1), the Wilcoxon test with Bonferroni correction also uses the results obtained by the Lagrangian ILS with $\omega = 0.5$ for large (100 tasks) instances (see Tables A.6 and A.7 in A.2). Please note that $\omega = 0.5$ reduces the number of permissible iterations by half relative to $\omega = 1$. For the two-tailed Wilcoxon test, the null hypothesis is "the solution quality (computational time) of algorithm A is similar to algorithm B", while for the one-tailed Wilcoxon test, the hypothesis is "the solution quality (computational time) of Lagrangian ILS is similar to or worse than algorithm B". The comprehensive Wilcoxon test analysis was conducted using R-studio Kloke and McKean (2015), and the corresponding p-values are reported in Tables 3.4 and 3.5.

In the matter of solution quality, for large instances from both categories "NoTeam Reduced" and "NoTeam Complete", the p-values at 95% confidence interval for both two-tailed and one-tailed Wilcoxon tests on Lagrangian ILS versus ILS in Xie et al. (2017), Lagrangian ILS versus Implemented ILS, and Lagrangian ILS versus ILS-HNS in Zhou et al. (2020) are much smaller than 0.05. These are strong evidence that the Lagrangian ILS produces significantly better solutions than ILS, Implemented ILS, and ILS-HNS. For small and medium instances, the p-value for the two-tailed Wilcoxon test on Lagrangian ILS versus ILS is 0.0550 which indicates that the two algorithms have similar performance.

Table 3.4: Wilcoxon tests at 95% confidence interval between the performance of the Lagrangian ILS, ILS in Xie et al. (2017), Implemented ILS, and ILS-HNS in Zhou et al. (2020)

	p-value			
Algorithm A **Versus** Algorithm B	Average	Worst	Best	sec_a
Small and Medium instances				
Lagrangian ILS ($\omega = 1$) **Versus** ILS (two-tailed)	0.0550	N/A	N/A	0.0000
Lagrangian ILS ($\omega = 1$) **Versus** ILS (one-tailed)	0.0277	N/A	N/A	1.0000
Lagrangian ILS ($\omega = 1$) **Versus** Implemented ILS (two-tailed)	0.0160	N/A	N/A	0.0001
Lagrangian ILS ($\omega = 1$) **Versus** Implemented ILS (one-tailed)	0.0078	N/A	N/A	0.0000
Implemented ILS **Versus** ILS (two-tailed)	0.6550	N/A	N/A	0.0000
Large instances from category "NoTeam Reduced"				
Lagrangian ILS ($\omega = 1$) **Versus** ILS (two-tailed)	0.0000	0.0000	0.0000	0.6900
Lagrangian ILS ($\omega = 1$) **Versus** ILS (one-tailed)	0.0000	0.0000	0.0000	1.0000
Lagrangian ILS ($\omega = 1$) **Versus** Implemented ILS (two-tailed)	0.0057	0.0053	0.0003	0.0000
Lagrangian ILS ($\omega = 1$) **Versus** Implemented ILS (one-tailed)	0.0028	0.0027	0.0002	0.0000
Lagrangian ILS ($\omega = 1$) **Versus** ILS-HNS (two-tailed)	0.0000	0.0000	0.0000	N/A
Lagrangian ILS ($\omega = 1$) **Versus** ILS-HNS (one-tailed)	0.0000	0.0000	0.0000	N/A
Implemented ILS **Versus** ILS (two-tailed)	1.0000	1.0000	0.9384	0.0000
Large instances from category "NoTeam Complete"				
Lagrangian ILS ($\omega = 1$) **Versus** ILS (two-tailed)	0.0000	0.0000	0.0003	0.0016
Lagrangian ILS ($\omega = 1$) **Versus** ILS (one-tailed)	0.0000	0.0000	0.0002	0.0008
Lagrangian ILS ($\omega = 1$) **Versus** Implemented ILS (two-tailed)	0.0000	0.0000	0.0000	0.0000
Lagrangian ILS ($\omega = 1$) **Versus** Implemented ILS (one-tailed)	0.0000	0.0000	0.0000	0.0000
Lagrangian ILS ($\omega = 1$) **Versus** ILS-HNS (two-tailed)	0.0000	0.0004	0.0102	N/A
Lagrangian ILS ($\omega = 1$) **Versus** ILS-HNS (one-tailed)	0.0000	0.0002	0.0051	N/A
Implemented ILS **Versus** ILS (two-tailed)	1.0000	1.0000	1.0000	0.0000

In contrast, the p-values at 95% confidence interval for both two-tailed and one-tailed Wilcoxon tests on Lagrangian ILS versus Implemented ILS are much smaller than 0.05. This evidence strongly supports that the Lagrangian ILS has better performance than Implemented ILS in terms of solution quality. Besides, for the two-tailed Wilcoxon tests on Implemented ILS versus ILS, the p-values are greater than 0.05, which indicate that the Implemented ILS obtains a similar solution quality as ILS.

In terms of the computational time, for small (25 tasks) and medium (50 tasks) instances, the p-value at 95% confidence interval for the two-tailed Wilcoxon test on Lagrangian ILS versus ILS in Xie et al. (2017) is much smaller than 0.05, which are strong evidence that the two algorithms have different performance. On the other hand, the p-value for the one-tailed Wilcoxon test on Lagrangian ILS versus ILS is greater than 0.05. These two p-values indicate that the Lagrangian ILS is slower than the ILS on small (25 tasks) and medium (50 tasks) instances. For large instances (100 tasks) from the category "NoTeam Reduced", the p-values at 95% confidence interval for both, two-tailed and one-tailed Wilcoxon tests on Lagrangian ILS versus ILS are greater than 0.05, which indicate that the time required for both algorithms are about the same. For large instances (100 tasks) from the category "NoTeam Complete", the p-values at 95% confidence interval for both two-tailed and one-tailed Wilcoxon tests on Lagrangian ILS versus ILS are smaller than 0.05, which are strong evidence that the Lagrangian ILS is faster than ILS. The p-values at 95% confidence interval for both two-tailed and one-tailed Wilcoxon tests on the Lagrangian ILS versus Implemented ILS are much smaller than 0.05 in all cases, which are strong evidence that the Lagrangian ILS is faster than the Implemented ILS.

For Lagrangian ILS with $\omega = 0.5$, the Wilcoxon tests show that for large instances from both categories, "NoTeam Reduced" and "NoTeam Complete", the Lagrangian ILS outperforms both ILS and ILS-HNS for average and worst objective values. For the best objective values, the Lagrangian ILS with $\omega = 0.5$ is able to obtain similar results compared with ILS but better results compared with ILS-HNS for instances from category "NoTeam Reduced", whereas for instances from category "NoTeam Complete", the Lagrangian ILS outperforms ILS but produces similar results compared with ILS-HNS. For Lagrangian ILS and Implemented ILS, with regards to the solution quality, the two algorithms have similar performance on instances from the category "NoTeam Reduced", whereas for instances from the category "NoTeam Complete", the Lagrangian ILS outperforms the Implemented ILS. In terms of computational time, the Lagrangian ILS is faster than ILS and Implemented ILS in both categories. The p-values obtained from the Wilcoxon tests are provided in Table 3.5, and the results obtained from the Lagrangian ILS with $\omega = 0.5$ on each instance are provided in A.2 (see Tables A.6 and A.7).

Table 3.5: Wilcoxon tests at 95% confidence interval between the performance of ILS in Xie et al. (2017), Implemented ILS, ILS-HNS in Zhou et al. (2020), and Lagrangian ILS with $\omega = 0.5$.

	p-value			
Algorithm A **Versus** Algorithm B	Average	Worst	Best	sec_a
Large instances from category "NoTeam Reduced"				
Lagrangian ILS ($\omega = 0.5$) **Versus** ILS (two-tailed)	0.0237	0.0008	0.1468	0.0000
Lagrangian ILS ($\omega = 0.5$) **Versus** ILS (one-tailed)	0.0118	0.0004	0.0734	0.0000
Lagrangian ILS ($\omega = 0.5$) **Versus** Implemented ILS (two-tailed)	1.0000	1.0000	0.1712	0.0000
Lagrangian ILS ($\omega = 0.5$) **Versus** Implemented ILS (one-tailed)	0.9095	0.8317	0.0856	0.0000
Lagrangian ILS ($\omega = 0.5$) **Versus** ILS-HNS (two-tailed)	0.0000	0.0000	0.0000	N/A
Lagrangian ILS ($\omega = 0.5$) **Versus** ILS-HNS (one-tailed)	0.0000	0.0000	0.0000	N/A
Large instances from category "NoTeam Complete"				
Lagrangian ILS ($\omega = 0.5$) **Versus** ILS (two-tailed)	0.0000	0.0004	0.0014	0.0000
Lagrangian ILS ($\omega = 0.5$) **Versus** ILS (one-tailed)	0.0000	0.0002	0.0007	0.0000
Lagrangian ILS ($\omega = 0.5$) **Versus** Implemented ILS (two-tailed)	0.0000	0.0001	0.0006	0.0000
Lagrangian ILS ($\omega = 0.5$) **Versus** Implemented ILS (one-tailed)	0.0000	0.0000	0.0003	0.0000
Lagrangian ILS ($\omega = 0.5$) **Versus** ILS-HNS (two-tailed)	0.0008	0.0010	0.2681	N/A
Lagrangian ILS ($\omega = 0.5$) **Versus** ILS-HNS (one-tailed)	0.0004	0.0005	0.1341	N/A

3.4.2 Sensitivity analysis

Our empirical experience indicates that "NoTeam Reduced" is harder to solve than "NoTeam Complete", and the performance is more sensitive to the parameter settings. The reason could be that the weight on unallocated tasks is relatively high in the studied problem, and the objective value will dramatically deteriorate even if only one more task is not allocated. In this section, using the large instances from the category "NoTeam Reduced", the performance of the Lagrangian ILS is analysed with the variation of several parameters, including ω, ψ, and γ.

Table 3.6 presents the analysis on the performance of the Lagrangian ILS with $\omega \in \{0.5, 7, 15\}$ when $\psi = 50$ and $\gamma = 2$. In this table, all the percentage differences are referenced to the corresponding values obtained by the Lagrangian ILS with $\omega = 0.5$. It can be observed that the solution quality improves at the cost of the computational time when ω increases. This is expected since increasing ω can effectively increase the number of permissible iterations for the Lagrangian ILS. Indeed, the Lagrangian ILS with $\omega = 15$ consistently obtains a better solution than the Lagrangian ILS with $\omega = 0.5$ on each instance. Such behaviour is a desired property for choosing the value of ω in practice. In A.2, the detailed results on each instance are provided (see Table A.6). In A.2, we also report the results on the large instances from "NoTeam Complete", which exhibits the same behaviour (see Table A.7).

Table 3.6: Sensitivity analysis on the performance of the Lagrangian ILS with ω when $\psi = 50$ and $\gamma = 2$ for large instances from category NoTeam Reduce

	$w = 0.5$				$w = 7$				$w = 15$			
Instances	Average	Worst	Best	sec_a	$\%_a$	$\%_w$	$\%_b$	sec_a	$\%_a$	$\%_w$	$\%_b$	sec_a
C 5x4	3382.76	3426.39	3342.80	16.30	0.72	0.70	0.35	157.65	1.46	2.19	0.57	356.30
R 5x4	3162.48	3191.25	3123.60	18.00	1.63	1.95	1.57	177.10	1.71	0.90	1.97	323.85
RC 5x4	3220.88	3302.87	3187.24	14.85	2.22	3.72	2.07	140.05	2.18	3.43	2.28	292.85
C 6x6	4582.42	4603.21	4552.87	22.10	0.54	0.56	0.12	221.90	0.62	0.80	0.15	470.65
R 6x6	3607.36	3657.14	3543.74	24.90	2.13	2.56	0.93	243.95	2.09	3.04	0.87	570.30
RC 6x6	3588.10	3653.21	3531.01	23.10	0.92	2.04	0.28	252.40	1.75	2.28	0.65	605.35
C 7x4	3086.91	3120.58	3064.55	12.15	0.99	2.08	0.29	131.30	0.99	2.07	0.28	265.10
R 7x4	3081.14	3112.90	3032.64	11.50	0.79	1.01	0.62	110.35	1.29	0.85	0.56	212.70
RC 7x4	3345.14	3383.03	3318.30	10.45	1.22	1.85	0.73	110.90	1.57	2.53	0.78	231.30
Average	3450.80	3494.51	3410.75	17.04	1.24	1.83	0.77	171.73	1.52	2.01	0.90	369.82

Table 3.7 presents the results obtained from the Lagrangian ILS using a combination of $\psi \in \{5, 50, 150, 400\}$ and $\gamma \in \{0.5, 2, 10, 100\}$ when $\omega = 1$. In this table, the columns Average, Best, and sec_a show the average, best objective value, and average computational time over all instances, respectively. In addition, the best value in each group is in

Table 3.7: Sensitivity analysis on the performance of the Lagrangian ILS with ψ and γ when $\omega = 1$ for large instances from category "NoTeam Reduced"

	Average				**Best**				sec_a			
γ \ ψ	5	50	150	400	5	50	150	400	5	50	150	400
0.5	3449.12	3439.29	3438.41	3449.08	3418.84	3411.08	3404.46	3410.67	61.56	38.59	38.48	32.18
2	3446.21	**3426.64**	3441.74	3449.20	3404.02	**3395.25**	3412.22	3406.17	45.64	30.89	25.87	21.21
10	3452.68	3449.72	3441.43	3459.68	3419.23	3412.92	3407.24	3421.11	51.41	29.29	26.76	22.55
100	3460.67	3450.93	3450.49	3472.80	3425.65	3412.96	3416.30	3424.22	53.59	28.02	24.98	**22.01**

bold, which indicates that the Lagrangian ILS performs the best when $\psi = 50$ and $\gamma = 2$ in terms of average and best objective values. In the subroutine SEARCH (Section 3.3.4), if ψ is large, the neighbourhood operators will favour more on the augmented objective function. In contrast, if ψ is small, then the neighbourhood operators will favour more on the original objective function. Based on the results in Table 3.7, increasing or decreasing ψ relative to $\psi = 50$ can slightly reduce the solution quality, indicating that finding a good balance between the original objective function and the augmented objective function can improve the overall solution quality. On the other hand, it has been observed that increasing ψ can notably reduce computational time. Indeed, the difference between the best and worst times in each row can be as large as 31.58 seconds when $\gamma = 100$. This suggests that favouring the augmented objective function in SEARCH can speed up the algorithm. The parameter γ controls how fast the penalty weights can increase. It can be seen that a very small γ ($\gamma = 0.5$) leads to good solution quality but increases the solution time. When γ is very large ($\gamma = 100$), solution quality deteriorates while solution time improves. The reason is that the penalty weights increase so fast that only feasible solutions can be accepted in the neighbourhood search. In A.2, we provided the detailed results for the analysis on ψ when $\gamma = 2$ (see Table A.8) and the detailed results for the analysis on γ when $\psi = 50$ (see Table A.9).

3.5 Conclusion

This chapter presents a new optimisation procedure for the Workforce Scheduling and Routing Problem. This procedure, referred to as the Lagrangian ILS, is based on the idea of an amalgamation of the iterated local search and Lagrangian relaxation, which was originally introduced in Gu et al. (2019). The computational experiments demonstrated better performance of the Lagrangian ILS in comparison with CPLEX and the state-of-the-art algorithm in Xie et al. (2017) both, in terms of the solution quality and the computational time. The Lagrangian ILS also significantly outperforms the original implementation of the idea of such amalgamation presented in Gu et al. (2019). The computational experiments were conducted on a set of benchmark instances from the literature, which are regarded as standard in the publications on this topic. The superior performance of the Lagrangian ILS is particularly evident on large instances where the Lagrangian ILS outperforms the algorithm in Xie et al. (2017) even when the Lagrangian ILS was allowed to use only a half of the permissible number of iterations.

It is well-known that permission to violate certain constraints can significantly improve the performance of the local search. Given this observation and the outstanding performance of the Lagrangian ILS, the development of algorithms for vehicle routing problems based on an amalgamation of a local search metaheuristic and Lagrangian relaxation can be viewed as a promising direction for future research.

4

Iterated Local Search for the Pickups and Deliveries Problem Arising in Retail Industry with Ordered Objectives

Abstract

This chapter studies a Simultaneous Pickup and Delivery Problem that reflects many real-world restrictions and practices. The objectives of the problem are ordered where the primary objective is to maximise the number of served customers, and the secondary objective is to minimise the total travel time. The problem is formulated as a mixed integer program which is based on three index variables. A novel iterated local search is tested on three sets of instances, one set is provided by the industry partner and the other two sets are derived from benchmark instances in the literature. With a time limit of 1 minute, the results of computational experiments have shown that the proposed algorithm has good performance in terms of solution quality and stability.

4.1 Introduction

This chapter studies a Simultaneous Pickup and Delivery Problem that reflects many real-world restrictions and practices. This problem referred to as the Multi-attribute Simultaneous Pickup and Delivery Problem (MASPDP) considers the following features.

- **Time window** is associated with each customer which specifies the time interval when the service can commence.
- **Open routes** refers to the situation when drivers finish the service of their last customer on the routes and do not return to the depot. This feature is motivated by the use of subcontractors who have their own vehicles and depots suitable for temporary storage.

- **Weight and Volume** are used to characterise the demand corresponding to the customers.

- **Heterogeneous fleet of vehicles** is used. Each vehicle is characterised by its capacity for weight and volume.

- **Incompatibility** is applied between customers and vehicles. Two types of vehicles are considered, i.e., the one-man vehicle and the two-men vehicle. The customers are also classified as either one-man customers or two-men customers. The one-man customer can be served by all vehicles, while two-men customers can only be served by two-men vehicles.

- **Roster** specifies when a vehicle can load at the depot. This feature is motivated by the fact that the depot has limited loading space.

- **Simultaneous pickup and delivery** is considered where customers can request service for both delivery and pickup.

- **Restriction on shift length** is applied to all drivers. For example, drivers can not work longer than 10 hours.

- **Ordered objectives** are considered. The primary objective is to maximise the number of served customers and the secondary objective is to minimise the total travel time.

In spite of the practical importance of these features, as discussed in Chapter 2, to the best of the author's knowledge, no publication studies this problem in its entire complexity.

The restriction on computational time is one minute. This restriction is imposed by the industry partner because the developed optimisation software is a tool interactively used by a scheduler for allocating customers to vehicles. This means that the scheduler can use this software to produce an initial version of the allocation which may require manual adjustments; improve an existing version of the allocation; or produce some alternative version of the allocation. For example, the allocation may contain excessively long routes compared with other routes which is not fair to all drivers. The scheduler may adjust the allocation to make it fair to all drivers or choose an allocation among all the alternative allocations that is fair to all drivers. Therefore, the developed software must respond in seconds rather than in minutes.

The Simultaneous Pickup and Delivery Problem is a generalisation of the vehicle routing problem. Thus, it is NP-hard in the strong sense Garey and Johnson (1979). The majority of the publications on this topic present various heuristics and metaheuristics Parragh et al. (2008a), Parragh et al. (2008b), Koç et al. (2020). Although many publications present heuristics and metaheuristics that are efficient, for example, Vidal et al. (2013), Nagata et al. (2010), Xie et al. (2017), a strict time limit is rarely considered. In this chapter, a new iterated local search optimisation procedure is presented. This iterated local search referred to as the ILS2O achieves a satisfactory performance with the following developments:

- The ILS2O uses the framework that amalgamates the iterated local search and Lagrangian relaxation.
- The ILS2O introduces a method that alternates between the primary objective and the secondary objective during the application of local search.
- The ILS2O uses the neighbourhood reduction technique that dynamically reduces the size of the search space.

The remaining part of this chapter is organised as follows. Section 4.2 presents the mixed integer programming formulations. The ILS2O is described in Section 4.3. Section 4.4 reports the results of computational experiments and Section 4.5 concludes this chapter.

4.2 Problem statement

The considered MASPDP can be stated as follows. Let $G(L, A)$ be a directed graph, where the set of vertices $L = \{0\} \cup C$ and $C = \{1, 2, ..., l\}$, the set of arcs $A = A_D \cup A_C$ and $A_D = \{(0, i)|i \in C\}$, $A_C = \{(i, j)|i \neq j, \forall i, j \in C\}$. Vertex 0 represents the depot and the remaining vertices represent the customers. Each arc $(i, j) \in A$ has an associated travel time $t_{i,j}$.

The delivery to customer $i \in C$ is characterised by its weight w_i^d and volume v_i^d. The pickup from customer $i \in C$ is characterised by its weight w_i^p and volume v_i^p. For customer $i \in C$, the associated time window $[a_i, b_i]$ indicates the earliest and latest time when the corresponding services can start, and let $p_i > 0$ be the service time required to complete the service.

Let T be the set of all vehicles. Each vehicle $i \in T$ is differed by its weight capacity W_i and volume capacity V_i. All vehicles $i \in T$ depart from the same depot and are not required to return to the depot after serving all allocated customers. Due to the loading capacity of the depot, each vehicle $i \in T$ arrives at the depot at the specified starting time r_i with loading time δ_i. Furthermore, there exists an upper bound S_i on the shift time of the drivers in vehicle $i \in T$, which is the length of the time interval between the time when a driver starts loading at the depot and the time when the driver finishes the service of the last allocated customers.

Each customer $i \in C$ can be served only once, but not all vehicles are capable to serve certain customers. In this chapter, two types of vehicles are considered, i.e., the one-man vehicle $T' \subset T$ and the two-men vehicle $T'' \subset T$. The customers are also classified as either one-man customer $C' \subset C$, or two-men customer $C'' \subset C$. The one-man customer can be served by all vehicles, while two-men customer can only be served by two-men vehicles.

While respecting all the constraints on drivers, vehicles, customers and the depot, the primary objective is to maximise the total number of served customers and the secondary objective is to minimise the total travel time.

4.2.1 Three-index model

Two three-index mixed integer programming (MIP) formulations are presented in this section to solve the MASPDP with ordered objectives. These formulations are based on the three-index MIP formulation presented in Gu et al. (2021). The first formulation maximises the number of served customers which is the same as the formulation presented in Gu et al. (2021). The second formulation minimises the total travel time which is modified from the formulation presented in Gu et al. (2021) by changing the objective function and adding a new constraint.

Let x^i_{jk} be a binary variable indicating if customer j is the immediate predecessor of customer k in the route of vehicle i; η^i_j be a binary variable indicating if customer j is allocated to vehicle i; γ^i_j be a binary variable indicating if customer j is the first customer to visit after vehicle i departing from the depot; θ_j^i be a binary variable indicating if customer j is the last customer in the route of vehicle i. Denote the time when the driver in vehicle i starts serving customer k by s_k^i; the weight of the vehicle when leaving customer j by y_j; the volume of the vehicle when leaving customer j by z_j. The considered problem is formulated as follows to maximise the number of served customers:

$$Job = \max \sum_{i \in T} \sum_{j \in C} \eta_j^i \tag{4.1}$$

subject to

$$\sum_{i \in T} \eta_j^i \leq 1, \quad \forall j \in C \tag{4.2}$$

$$\sum_{j \in C} \gamma_j^i \leq 1, \quad \forall i \in T \tag{4.3}$$

$$\gamma_j^i + \sum_{k \in C} x_{k,j}^i = \eta_j^i, \quad \forall i \in T, \forall j \in C \tag{4.4}$$

$$\theta_j^i + \sum_{k \in C} x_{j,k}^i = \eta_j^i, \quad \forall i \in T, \forall j \in C \tag{4.5}$$

$$a_j \leq s_j^i, \quad \forall j \in C, \forall i \in T \tag{4.6}$$

$$s_j^i \leq b_j, \quad \forall j \in C, \forall i \in T \tag{4.7}$$

$$(r_i + \delta_i + t_{0,k})\gamma_k^i \leq s_k^i, \quad \forall i \in T, \forall k \in C \tag{4.8}$$

$$s_j^i + (p_j + t_{j,k})x_{j,k}^i + (a_k - b_j)(1 - x_{j,k}^i) \leq s_k^i, \quad \forall i \in T, \forall (j,k) \in A_C \tag{4.9}$$

$$p_j + s_j^i - r_i - (p_j + b_j - r_i)(1 - \theta_j^i) \leq S_i, \quad \forall j \in C, \forall i \in T \tag{4.10}$$

$$\sum_{k \in C} w_k^d \eta_k^i \leq W_i, \quad \forall i \in T \tag{4.11}$$

$$y_k \leq W_i + (\max_{e \in T} W_e - W_i)(1 - \eta_k^i), \quad \forall i \in T, \forall k \in C \tag{4.12}$$

$$\sum_{j \in C} w_j^d \eta_j^i - w_k^d + w_k^p - (\max_{e \in T} W_e - w_k^d + w_k^p)(1 - \gamma_k^i) \leq y_k, \quad \forall i \in T, \forall k \in C \tag{4.13}$$

$$y_j - w_k^d + w_k^p - (\max_{e \in T} W_e - w_k^d + w_k^p)(1 - x_{j,k}^i) \leq y_k, \quad \forall i \in T, \forall (j,k) \in A_C \tag{4.14}$$

$$\sum_{k \in C} v_k^d \eta_k^i \leq V_i, \quad \forall i \in T \tag{4.15}$$

$$z_k \leq V_i + (\max_{e \in T} V_e - V_i)(1 - \eta_k^i), \quad \forall i \in T, \forall k \in C \tag{4.16}$$

$$\sum_{j \in C} v_j^d \eta_j^i - v_k^d + v_k^p - (\max_{e \in T} V_e - v_k^d + v_k^p)(1 - \gamma_k^i) \leq z_k, \quad \forall i \in T, k \in C \tag{4.17}$$

$$z_j - v_k^d + v_k^p - (\max_{e \in T} V_e - v_k^d + v_k^p)(1 - x_{j,k}^i) \leq z_k, \quad \forall i \in T, \forall (j,k) \in A_C \tag{4.18}$$

$$\sum_{i \in T'} \sum_{k \in C''} \eta_k^i = 0 \tag{4.19}$$

$$x_{j,k}^i \in \{0,1\}, \quad \forall \{j,k\} \in A_C, \forall i \in T \tag{4.20}$$

$$\eta_j^i \in \{0,1\}, \quad \forall i \in T, \forall j \in C \tag{4.21}$$

$$\gamma_j^i \in \{0,1\}, \quad \forall i \in T, \forall j \in C \tag{4.22}$$

$$\theta_j^i \in \{0,1\}, \quad \forall i \in T, \forall j \in C \tag{4.23}$$

The objective function (4.1) maximises the number of served customers. Constraints (4.3) and (4.8) guarantee that a vehicle either stays at the depot or visits exactly one customer. Constraints (4.4) and (4.5) make sure that a vehicle leaves the customer's location except for the last customer. Then, constraints (4.4) and (4.5) together with constraints (4.2) ensure that a customer is visited by at most one vehicle. The arrival times, loading times at the depot, travelling times between vertices, and the time windows are taken into account by (4.8), (4.9) and (4.6)-(4.9) respectively. The shift length, weight capacity, and volume capacity are enforced by (4.10), (4.11)-(4.14), and (4.15)-(4.18) respectively. In addition, (4.6), (4.7), and (4.9) eliminate the subtours by virtue of $p_i > 0$. At last, constraints (4.19) establish the compatibility between customers and vehicles.

Let N be the number of served customers in the solution obtained from solving the model (4.1) $-$ (4.23). Then, the following mixed integer program is considered to minimise the total travel time.

$$Time = \min \sum_{k \in T} \sum_{(i,j) \in A_C} t_{i,j} x_{i,j}^k + \sum_{i \in T} \sum_{j \in C} t_{0,j} \gamma_j^i \tag{4.24}$$

subject to:

$$(4.2) - (4.23)$$

$$\sum_{i \in T} \sum_{j \in C} \eta^i_j \geq N \tag{4.25}$$

The objective function (4.24) minimises the total travel time while the number of the served customers must not below N (constraint (4.25)). In what follows, the problems (4.1)−(4.23) and (4.24)−(4.25) will be referred to as the three-index model.

4.3 ILS for ordered objectives

The ILS2O is another implementation of the amalgamation of the iterated local search and Lagrangian relaxation. Thus, this optimisation procedure also requires an alternative mixed integer linear programming formulation. Since the objectives for the MASPDP are ordered, certain modifications to the mathematical model are required in order to make it compatible with the idea of the amalgamation of the iterated local search and Lagrangian relaxation.

Using the three-index model as an example, the problems (4.1)−(4.23) and (4.24)−(4.25) are modified as the formulation (4.26)−(4.55) below using weighted sum where λ_1 is a non-negative weight for the objective function (4.1) and λ_2 is a non-negative weight for the objective function (4.24). It should be noted that the choice of mathematical models is not important because the performance of the ILS2O does not depend on the number of variables or the number of constraints.

The above modifications are dictated not by the complexity consideration but by the suitability for the optimisation method − local search. Furthermore, to reflect the objectives considered for the MASPDP, the ILS2O uses a method to determine if a solution to the problem (4.26)−(4.55) is better than the current best-known feasible solution. More specifically, a feasible solution is an improving solution if

- this solution serves more customers compared with the current best-known feasible solution or,
- this solution serves the same number of customers as the current best-known feasible

solution with less total travel time.

As in (3.16) − (3.20), let μ_j^i be the time warps for all $i \in T$ and $j \in C$. ψ_i be the violation of working duration for all $i \in T$, ω_i be the maximum violation on weight capacity for all $i \in T$, ν_i be the maximum violation on volume capacity for all $i \in T$. The considered problem is modified as follows.

$$f = \max \quad \lambda_1 \sum_{i \in T} \sum_{j \in C} \eta_j^i - \lambda_2 \left(\sum_{k \in T} \sum_{(i,j) \in A_C} t_{i,j} x_{i,j}^k + \sum_{i \in T} \sum_{j \in C} t_{0,j} \gamma_j^i \right) \tag{4.26}$$

subject to

$$\sum_{i \in T} \eta_j^i \leq 1, \quad \forall j \in C \tag{4.27}$$

$$\sum_{j \in C} \gamma_j^i \leq 1, \quad \forall i \in T \tag{4.28}$$

$$\gamma_j^i + \sum_{k \in C} x_{k,j}^i = \eta_j^i, \quad \forall i \in T, j \in C \tag{4.29}$$

$$\theta_j^i + \sum_{k \in C} x_{j,k}^i = \eta_j^i, \quad \forall i \in T, j \in C \tag{4.30}$$

$$a_j \leq s_j^i, \quad \forall j \in C, i \in T \tag{4.31}$$

$$s_j^i - b_j \leq \mu_j^i, \quad \forall j \in C, i \in T \tag{4.32}$$

$$(r_i + \delta_i + t_{0,k}) \gamma_k^i \leq s_k^i, \quad \forall i \in T, k \in C \tag{4.33}$$

$$s_j^i - \mu_j^i + (p_j + t_{j,k}) x_{j,k}^i + (a_k - b_j)(1 - x_{j,k}^i) \leq s_k^i, \forall i \in T, \forall (j,k) \in A_C \tag{4.34}$$

$$p_j + s_j^i - r_i - (p_j + b_j - r_i)(1 - \theta_j^i) + \sum_{k \in C} \mu_k^i \leq S_i + \psi_i, \forall j \in C, i \in T \tag{4.35}$$

$$\sum_{i \in T} \sum_{j \in C} \mu_j^i \leq 0 \tag{4.36}$$

$$\sum_{i \in T} \psi_i \leq 0 \tag{4.37}$$

$$\sum_{k \in C} w_k^d \eta_k^i \leq W_i + \omega_i, \quad \forall i \in T \tag{4.38}$$

$$y_k \leq W_i + \omega_i + (\max_{e \in T} W_e - W_i)(1 - \eta^i_k), \quad \forall i \in T, k \in C \tag{4.39}$$

$$\sum_{j \in C} w^d_j \eta^i_j - w^d_k + w^p_k - (\max_{e \in T} W_e - w^d_k + w^p_k)(1 - \gamma^i_k) \leq y_k, \quad \forall i \in T, k \in C \tag{4.40}$$

$$y_j - w^d_k + w^p_k - (\max_{e \in T} W_e - w^d_k + w^p_k)(1 - x^i_{j,k}) \leq y_k, \quad \forall i \in T, \forall (j,k) \in A_C \tag{4.41}$$

$$\sum_{i \in T} \omega_i \leq 0 \tag{4.42}$$

$$\sum_{k \in C} v^d_k \eta^i_k \leq V_i + \nu_i, \quad \forall i \in T \tag{4.43}$$

$$z_k \leq V_i + \nu_i + (\max_{e \in T} V_e - V_i)(1 - \eta^i_k), \quad \forall i \in T, k \in C \tag{4.44}$$

$$\sum_{j \in C} v^d_j \eta^i_j - v^d_k + v^p_k - (\max_{e \in T} V_e - v^d_k + v^p_k)(1 - \gamma^i_k) \leq z_k, \forall i \in T, k \in C \tag{4.45}$$

$$z_j - v^d_k + v^p_k - (\max_{e \in T} V_e - v^d_k + v^p_k)(1 - x^i_{j,k}) \leq z_k, \quad \forall i \in T, \forall (j,k) \in A_C \tag{4.46}$$

$$\sum_{i \in T} \nu_i \leq 0 \tag{4.47}$$

$$\sum_{i \in T'} \sum_{k \in C''} \eta^i_k = 0 \tag{4.48}$$

$$x^i_{j,k} \in \{0,1\}, \quad \forall \{j,k\} \in A_C, i \in T \tag{4.49}$$

$$\eta^i_j \in \{0,1\}, \quad \forall i \in T, j \in C \tag{4.50}$$

$$\theta^i_j \in \{0,1\}, \quad i \in T, j \in C \tag{4.51}$$

$$\mu^i_j \geq 0, \quad \forall i \in T, \forall j \in C \tag{4.52}$$

$$\psi_i \geq 0, \quad \forall i \in T \tag{4.53}$$

$$\omega_i \geq 0, \quad \forall j \in C \tag{4.54}$$

$$\nu_i \geq 0, \quad \forall j \in C \tag{4.55}$$

The objective function (4.26) combines the objective functions (4.1) and (4.24) using weights λ_1 and λ_2. The constraints (4.27) − (4.31), (4.33), (4.40), (4.41), (4.45) − (4.51) are the same as the constraints (4.2) − (4.6), (4.8), (4.13), (4.14), (4.17) − (4.23) in the three-index model. The constraints (4.33) and (4.34) correspond to (3.17), (3.19) and (3.20) which define the time warps, whereas the constraints (4.32) correspond to (3.18). The constraints (4.36), (4.37), (4.42), (4.47) guarantee that μ^i_j, ψ_i, ω_i, ν_i are zero.

By dualising constraints (4.36), (4.37), (4.42), and (4.47), using Lagrange multiplier $\alpha > 0$, $\beta > 0$, $\sigma > 0$, $\kappa > 0$, gives the following Lagrangian relaxation of the mixed integer linear program (4.26) − (4.55)

$$\begin{aligned} f_{LR} = \max \ \ & \lambda_1 \sum_{i \in T} \sum_{j \in C} \eta^i_j - \lambda_2 \left(\sum_{k \in T} \sum_{(i,j) \in A_C} t_{i,j} x^k_{i,j} + \sum_{i \in T} \sum_{j \in C} t_{0,j} \gamma^i_j \right) \\ & - \alpha \sum_{i \in T} \sum_{j \in C} \mu^i_j - \beta \sum_{i \in T} \psi_i - \sigma \sum_{i \in T} \omega_i - \kappa \sum_{i \in T} \nu_i \end{aligned} \tag{4.56}$$

subject to:

$$(4.27) - (4.35), (4.38) - (4.41), (4.43) - (4.46), (4.48) - (4.55)$$

In what follows, this Lagrangian relaxation will be referred to as the LR problem.

4.3.1 Neighbourhood reduction technique

One of the critical components of ILS2O is the design of proper neighbourhood structures. It has been demonstrated by many publications that permitting infeasible solutions in local search together with the use of an augmented objective function can significantly boost the performance of the meta-heuristics in the field of vehicle routing problem Cordeau et al. (1997), Cordeau et al. (2001), Nagata et al. (2010), Xie et al. (2017). The neighbourhood structures considered in this chapter are defined by the commonly used edge exchange operators. These operators allow the violation of the time window, shift length, weight and volume capacity constraints. In addition, the algorithm presented in this chapter reduces the size of the neighbourhood by only allowing moves that lead to more allocations than the current best-known feasible solution. To be specific, let s be a solution that can be infeasible; $H(s, N)$ be the neighbourhood of s induced by an edge exchange operator N permitting infeasible solutions. The corresponding reduced neighbourhood is defined as

$$\widehat{H_1}(s, N) = \{s' \in H(s, N) | Job(s') > Job(s^*)\}$$

$$\widehat{H_2}(s, N) = \{s' \in H(s, N) | Job(s') = Job(s^*), Time(s') < Time(s^*)\}$$

where s^* is the best-known feasible solution, $\widehat{H_1}(s, N)$ is the neighbourhood for maximising the number of served customers, and $\widehat{H_2}(s, N)$ is the neighbourhood for minimising the total travel time. The reason is that when the proposed ILS2O focuses completely on maximising the number of served customers, searching in a neighbourhood that includes solutions with the same number of served customers as the number of served customers in the current best-known feasible solution does not produce a solution with a higher number of served customers than the current one. Therefore, the neighbourhood reduction technique ignores all solutions with the same or lower number of served customers. This leads to solutions with a higher number of served customers as well as reduces the number of solutions to be considered in the evaluation process.

Please note, in the studied problem, it is permitted to have customers not served. Therefore, feasible solutions can be efficiently generated using simple heuristics (see Section 4.3.3 for more details). It should be noted that the reduced neighbourhood is dynamic since s^* can be updated in the iterative process of ILS2O. Since ILS2O can quickly find good solutions, the size of the reduced neighbourhood becomes significantly smaller after just a few iterations, which leads to faster convergence of the algorithm. Also, the solution process can be more stable because only solutions with more allocations are considered in the local search process.

4.3.2 ILS scheme

Let s^* be the currently best-known feasible solution which is updated through the entire optimisation procedure. The ILS2O is comprised of the following main components.

- The INITIAL procedure constructs a feasible solution of the problem (4.26)−(4.55) which is the current best-known feasible solution at the beginning of the optimisation procedure.

- The VARIABLE_OBJECTIVE_SEARCH(s', s^*) procedure attempts to improve s^* with respect to two different objective functions each at a time by adjusting the weights λ_1 and λ_2.

- The ASSIGN_WEIGHTS(s') procedure computes the initial values of $\alpha, \beta, \sigma, \kappa$ taking into account the constraints violation of the input solution s'.

- The ADJUST_WEIGHTS$(\alpha, \beta, \sigma, \kappa, s')$ procedure updates $\alpha, \beta, \sigma, \kappa$ according to the constraints violation of the input solution s'.

- The SEARCH(s', s^*) procedure constructs a sequence of solutions for the LR problem using different values of $\alpha, \beta, \sigma, \kappa$ computed from either the Assign_weights(s') procedure or the Adjust_weights$(\alpha, \beta, \sigma, \kappa, s')$.

- The STRATEGY(s', s^*, ξ) procedure is a local search procedure that attempts to find a solution that is better than the current best-known solution s^* using a strategy specified by parameter ξ.

- The PERTURB$(h,\ s^*)$ procedure perturbs the current best-known feasible solution s^*, taking into account the number of runs h which has failed to improve s^*.

Let M be the parameter that specifies the maximal permissible number of consecutive attempts to find an improving solution; $Job(\cdot)$ be a function that computes the number of served customers; $Time(\cdot)$ be a function that computes the total travel time. The ILS2O can be outlined by the pseudocode below.

ILS2O

1: $s' \leftarrow$ INITIAL **and** $s^* \leftarrow s'$
2: $h \leftarrow 0$
3: **while** $h \leq M$ **do**
4: $\quad s \leftarrow s^*$
5: $\quad s^* \leftarrow$ VARIABLE_OBJECTIVE_SEARCH(s', s^*)
6: $\quad$ **if** $Job(s^*) > Job(s)$ **then**
7: $\qquad h \leftarrow 0$
8: $\quad$ **else if** $Job(s^*) = Job(s)$ **and** $Time(s^*) < Time(s)$ **then**
9: $\qquad h \leftarrow 0$
10: $\quad$ **end if**
11: $\quad s' \leftarrow$ PERTURB$(h,\ s^*)$
12: $\quad h++$
13: **end while**

The ILS2O starts with a solution to the problem (4.26)−(4.55) which is generated using the INITIAL procedure described in Section 4.3.3. Until a better solution has been found, this solution is the current best-known feasible solution (line 1). The WHILE loop (lines 3−13) repeatedly attempts to find a feasible solution to the problem (4.26)−(4.55). Each such attempt starts with a different solution. For the first attempt (the first iteration of the WHILE loop lines 3 − 13), the starting solution is produced by the INITIAL procedure (line 4.3.3), whereas for all subsequent attempts, the starting solutions are generated by the PERTURB procedure (line 11). The PERTURB procedure is described in Section 4.3.7. It perturbs the best currently known feasible solution s^*, taking into account the number of runs h which has failed to obtain an improving solution.

4.3.3 INITIAL procedure

The INITIAL procedure is a sweep heuristic Gillett and Miller (1974) that constructs a solution for the problem (4.26)−(4.55). First, a list of customers is constructed based on the geographic coordinates of the customers. Then the customers are inserted into a route one by one until no customer can be inserted, in which case a new route is constructed. Since one-man vehicles can only serve one-man customers, whereas two-men vehicles can serve all types of customers, the procedure constructs the routes for one-man vehicles first, then followed by the routes for two-men vehicles. When inserting a customer into the route, the procedure chooses the insertion position that respects all the constraints and gives the smallest increase in travel time. The procedure terminates until either no customers can be inserted into the vehicle's route, or all customers have been allocated.

4.3.4 Local search with variable objectives

Let s' and s^* be the input solutions. The VARIABLE_OBJECTIVE_SEARCH procedure is outlined in the pseudocode below.

VARIABLE_OBJECTIVE_SEARCH

repeat
    $\lambda_1 \leftarrow 1$, $\lambda_2 \leftarrow 0$
    $\{s', s^*\} \leftarrow$SEARCH(s', s^*)
    $s \leftarrow s'$
    $\lambda_1 \leftarrow$ longest travel time of a route among the routes in s^*, $\lambda_2 \leftarrow 1$
    $\{s', s^*\} \leftarrow$SEARCH(s', s^*)
until $f_{LR}(s') = f_{LR}(s)$
return s^*

The VARIABLE_OBJECTIVE_SEARCH applies the SEARCH procedure described in Section 4.3.5 to two different objective functions by alternating the value for the parameters λ_1 and λ_2. As mentioned above, λ_1 is the weight for maximising the number of served customers (for example, the objective function (4.1)) whereas λ_2 is the weight for minimising the total travel time (for example, the objective function (4.24)). The procedure first attempts to find a solution with a higher number of served customers by assigning λ_1 to 1 and λ_2 to 0 (line 2). Then, the procedure attempts to find a solution with a lower total travel time while the objective of maximising the number of served customers remains at a higher priority. This is done by assigning λ_1 to the longest travel time of a route among the routes in s^* and λ_2 to 1 (line 5). This procedure terminates if the SEARCH procedure in line 6 fails to further improve the solution returned by the SEARCH procedures in line 3 with respect to the augmented objective function (4.56). Such an alternation on parameters λ_1 and λ_2 is motivated by the observation that when minimising the travel time, the output solutions may allow more customers to be inserted.

4.3.5 Search strategies

This section describes the SEARCH procedure for the ILS2O. Five different search strategies are described that can be used in the SEARCH procedure. Let $\xi \in \{0, 1, 2, 3, 4\}$ be the parameter that specifies which search strategy is used, the SEARCH procedure is outlined in the pseudocode below.

SEARCH

1: $\{\alpha, \beta, \sigma, \kappa\} \leftarrow$ ASSIGN_WEIGHTS(s')
2: $s \leftarrow s'$
3: $\{s', s^*\} \leftarrow$ STRATEGY(s', s^*, ξ)
4: $e \leftarrow 1$
5: **while** $f_{LR}(s) \neq f_{LR}(s')$ **and** s' is infeasible **and** $e \leq E$ **do**
6: $\quad s \leftarrow s'$
7: $\quad \{\alpha, \beta, \sigma, \kappa\} \leftarrow$ADJUST_WEIGHTS($\alpha, \beta, \sigma, \kappa, s'$)
8: $\quad \{s', s^*\} \leftarrow$ STRATEGY(s', s^*, ξ)
9: $\quad e \leftarrow e + 1$
10: **end while**
11: return $\{s', s^*\}$

Let s' and s^* be the input solutions which are also the output solutions. The SEARCH procedure repeatedly applies the STRATEGY procedure to find a solution with a better value than s' on the augmented objective function (4.56). Each application of the STRATEGY procedure uses a different value for the weights α, β, σ, and κ. For the first application, these weights are assigned by the ASSIGN_WEIGHTS procedure taking into account the constraints violation on s' (line 1). For the subsequent applications, these weights are adjusted using the ADJUST_WEIGHTS procedure (line 7). The SEARCH procedure terminates when either a feasible solution has been found, a local optimal has been found, or the counter e exceeds limit E.

Six neighbourhood operators N_0, N_1, N_2, N_3, $N_{\{0-3\}}$, and $N_{\{1-3\}}$ are considered for the search strategies. These operators are commonly used in the field of vehicle routing and can be found in many algorithms reported in the literature (see for example, Laporte et al. (2000), Bräysy and Gendreau (2005a), Bräysy and Gendreau (2005b), and Kindervater and Savelsbergh (2018)). Each operator N_i transforms an input solution s, by applying transformations (moves) from the set of transformations associated with this operator, and returns as the result some solution s' (denoted $s' = N_i(s)$) where s' is either the input solution s, or one of the transformations of s.

- The Operator N_0
  - interchanges a sequence of up to two consecutive visits in one route with a sequence of up to two consecutive visits in another route, including the transformations that only use a sequence from one route and an insertion position in another;
  - interchanges a sequence of up to two consecutive visits in a route (the customers in this sequence become unserved) with at most one unserved customer, including the transformations which either do not use an unserved customer or instead of the sequence of visits use only an insertion position in the route.
- The Operator N_1 extracts one visit from the route and inserts it into a different position of the same route.
- The Operator N_2 extracts a sequence of two consecutive visits and inserts this sequence into a different position of the same route.
- The Operator N_3 reverses the order of some sequence of consecutive visits in a route.
- The Operator $N_{\{0-3\}}$ comprises all transformations associated with the N_0, N_1, N_2, and N_3.
- The Operator $N_{\{1-3\}}$ comprises all transformations associated with the N_1, N_2, and N_3.

For each of the neighbourhood operators N_0 and $N_{\{0-3\}}$, the output is a solution with the largest number of served customers among all solutions with a better value than the input solution on the augmented objective function (4.56). For each of the neighbourhood operators N_1, N_2, N_3, and $N_{\{1-3\}}$, the output is a solution with the largest value on the augmented objective function (4.56).

In all strategies, if a feasible solution is found that is better than the currently best-known solution s^*, then s^* is updated immediately.

- The first strategy i.e., STRATEGY$(s', s^*, 1)$, picks the best transformation among the transformations associated with N_0.

- The second strategy i.e., STRATEGY$(s', s^*, 2)$, picks the best transformation among all transformations associated with $N_{\{0-3\}}$.

- The third strategy i.e., STRATEGY$(s', s^*, 3)$, uses four local search optimisation procedures, each with one of the four operators N_0, N_1, N_2, N_3. This strategy terminates when a local optimal is found for all four operators.

- The fourth strategy i.e., STRATEGY$(s', s^*, 4)$, applies a local search optimisation procedure with the operator N_0. Then, using the local optimal obtained for the operator N_0, this strategy picks the best transformation among all transformations associated with $N_{\{1-3\}}$.

- The fifth strategy i.e., STRATEGY$(s', s^*, 5)$, iteratively applies the fourth strategy until it fails to obtain a solution with better value on the augmented objective function (4.56) than the current one. This strategy is adapted from the most classical variable neighbourhood search (VNS) described in Hansen et al. (2017), Hansen et al. (2019). The output solution of this strategy is the local optimal for both neighbourhood operators N_0 and $N_{\{1-3\}}$.

STRATEGY$(s', s^*, 1)$

$s' \leftarrow N_0(s')$
if s' is feasible **then**
  if $Job(s^*) < Job(s')$ **then**
    $s^* \leftarrow s'$
  else if $Job(s^*) = Job(s')$ **and** $Time(s^*) > Time(s')$ **then**
    $s^* \leftarrow s'$
  end if
end if
return $\{s', s^*\}$

STRATEGY$(s', s^*, 2)$

$s' \leftarrow N_{\{0-3\}}(s')$
if s' is feasible **then**
  if $Job(s^*) < Job(s')$ **then**
    $s^* \leftarrow s'$
  else if $Job(s^*) = Job(s')$ **and** $Time(s^*) > Time(s')$ **then**
    $s^* \leftarrow s'$
  end if
end if
return $\{s', s^*\}$

STRATEGY$(s', s^*, 3)$

repeat
  $\bar{s} \leftarrow s'$
  for $i \leftarrow 0;\ i < 4;\ i \leftarrow i + 1$ **do**
    repeat
      $s \leftarrow s'$
      $s' \leftarrow N_i(s')$
      if s' is feasible **then**
        if $Job(s^*) < Job(s')$ **then**
          $s^* \leftarrow s'$
        else if $Job(s^*) = Job(s')$ **and** $Time(s^*) > Time(s')$ **then**
          $s^* \leftarrow s'$
        end if
      end if
    until $f_{LR}(s) = f_{LR}(s')$
  end for
until $f_{LR}(\bar{s}) = f_{LR}(s')$
return $\{s', s^*\}$

STRATEGY($s', s^*, 4$)

repeat
  $s \leftarrow s'$
  $s' \leftarrow N_0(s')$
  if s' is feasible **then**
    if $Job(s^*) < Job(s')$ **then**
      $s^* \leftarrow s'$
    else if $Job(s^*) = Job(s')$ **and** $Time(s^*) > Time(s')$ **then**
      $s^* \leftarrow s'$
    end if
  end if
until $f_{LR}(s) = f_{LR}(s')$
$s' \leftarrow N_{\{1-3\}}(s')$
if s' is feasible **then**
  if $Job(s^*) < Job(s')$ **then**
    $s^* \leftarrow s'$
  else if $Job(s^*) = Job(s')$ **and** $Time(s^*) > Time(s')$ **then**
    $s^* \leftarrow s'$
  end if
end if
return $\{s', s^*\}$

STRATEGY($s', s^*, 5$)

repeat
  $s \leftarrow s'$
  $\{s', s^*\} \leftarrow$ STRATEGY($s', s^*, 4$)
until $f_{LR}(s) = f_{LR}(s')$
return $\{s', s^*\}$

4.3.6 Initial value of Lagrange multipliers and their adjustment

The ASSIGN_WEIGHTS procedure assigns initial value for α, β, σ, κ using an input solution s where $\alpha = \sum_{i \in T} \sum_{j \in C} \mu_j^i(s)$; $\beta = \sum_{i \in T} \psi_i(s)$; $\sigma = \sum_{i \in T} \omega_i(s)$; and $\kappa = \sum_{i \in T} \nu_i(s)$. The ADJUST_WEIGHTS procedure updates the value of these weights as follows when an infeasible solution is returned from the STRATEGY procedure.

$$\alpha_{i+1} = \alpha_i + \tau \sum_{i \in T} \sum_{j \in C} \mu_j^i(s) \quad , \quad \beta_{i+1} = \beta_i + \tau \sum_{i \in T} \psi_i(s), \tag{4.57}$$

$$\sigma_{i+1} = \sigma_i + \tau \sum_{i \in T} \omega_i(s), \text{ and } \kappa_{i+1} = \kappa_i + \tau \sum_{i \in T} \nu_i(s) \tag{4.58}$$

where

$$\tau = \frac{\gamma f(s)}{(\sum_{i\in T}\sum_{j\in C}\mu_j^i(s))^2 + (\sum_{i\in T}\psi_i(s))^2 + (\sum_{i\in T}\omega_i(s))^2 + (\sum_{i\in T}\nu_i(s))^2} \tag{4.59}$$

where γ is a positive parameter.

4.3.7 PERTURB procedure

The PERTURB procedure expands the search space by randomly perturbing the current best solution s^*. An unallocated customer is randomly chosen and then inserted into a position among the routes which gives the largest value of (4.56) when $\alpha = \beta = \sigma = \psi = 1$. Then, two randomly selected sequences of consecutive customers are swapped between two randomly selected routes. This random swap will be performed multiple times which depends on the counter h in the pseudocode for the ILS2O. To be specific, the number of swaps starts from one and increases by one each time when counter h in the pseudocode for the ILS2O increases. The current best solution s^* may also be updated in this process.

4.4 Computational experiments

This section presents the results of computational experiments. To evaluate the performance of the proposed ILS2O, its performance is compared with the performance of an iterated local search with neighbourhood reduction (ILS-NR) described in Gu et al. (2021). The problem studied in Gu et al. (2021) contains the same constraints as the problem studied in this chapter. However, the objective of the problem studied in Gu et al. (2021) only maximises the number of served customers. In addition, the performance of the ILS2O is compared with the performance of CPLEX, an iterated local search (ILS) adapted from Xie et al. (2017), and a two-stage algorithm. The ILS considers the weighted sum objective function (4.26) where the value for λ_1 is the longest travel time of a route in a solution generated by 4.3.3 and the value for λ_2 is 1. The two-stage algorithm considers the two objectives one at a time. In the first stage, the ILS is used to find a solution to the problem (4.1)−(4.23). The solution produced by the first stage is used as the starting solution of the second stage. In the second stage, the ILS is used again to find a solution to the problem (4.24)−(4.25). In what follows, this algorithm is referred to as 2Phase.

To evaluate the performance of CPLEX with the three-index model, a time limit of 6 hours and a memory limit of 8GB are given for CPLEX. In addition, the objective functions used in CPLEX for the three-index model are weighted sum because this approach is easy to implement. To be clear, the three-index model with weighted sum is referred to as the weighted three-index model and is presented in B.1. As the ILS, the weighted sum objective function uses the same λ_1 and λ_2. For each instance from the RW benchmark, Solomon benchmark, and Solomon benchmark Ver2, the value of λ_1 is presented in the column titled λ_1 in Tables 4.2 and 4.3 and λ_2 is one for all instances.

The ILS2O, ILS, ILS-NR, and 2Phase were applied 30 times, each time with a different starting solution which is generated by the procedure described in Section 4.3.3. To eliminate the impact of the starting solution, each individual application of these algorithms uses the same starting solution. The parameters settings for the ILS and 2Phase are identical as in Xie et al. (2017) and the parameters settings for ILS-NR are the same as suggested in Gu et al. (2021). For ILS2O, the maximum number of exchange operations in the subroutine PERTURB is five; the parameter E is 100; the parameter M is computed according to $\omega(|C| + \Lambda|T|)$, where C is the set of all customers; T is the set of all vehicles; ω is a parameter to control M; similar to Xie et al. (2017) and Penna et al. (2013) Λ is 10. Similar to the Lagrangian ILS, the ILS2O increases the number of exchange operations in perturbation after each $M/5$ sequential iterations that fail to obtain an improving solution. The positive parameter γ in (4.59) is 2.

All methods were programmed in C++ and compiled with g++, using the optimisation level O3 and all computational experiments were conducted on a computer with Intel Xeon CPU E5-2697 v3 2.60GHz and 8GB RAM. In addition, the version for CPLEX is 12.10. In what follows, Section 4.4.1 discusses the benchmark instances used in the computational experiments. Section 4.4.2 analyses the performance of ILS2O with different search strategies. Section 4.4.3 compares the performance of ILS2O with the performance of CPLEX, ILS, and 2Phase.

4.4.1 Test instances

The computational experiments are conducted on three sets of instances. One set is provided by the industry partner. Since the considered problem has never appeared in the literature, this motivated the author to introduce the second and third sets of instances which can be downloaded from https://www.dropbox.com/scl/fo/k97c6i8vyry4y51lfxhwc/h?rlkey=9x6f582apvgx5c136knl9r24r&dl=0.

The first benchmark is a set of 60 instances with up to 100 customers. The second benchmark is a set of 56 instances that combines the Solomon data sets Solomon (1987) and data provided by the industry partner with a roster specifying the vehicles' starting time; and loading time for drivers at the depot. Furthermore, each delivery or pickup has the associated weight and volume and each vehicle has a capacity on weight and volume. The third benchmark is a set of 51 instances which also combines the Solomon data with the data provided by the industry partner. In addition to the features considered in the second benchmark, the third benchmark also considers the compatibility between customers and vehicles. For all three benchmarks, the travel times are rounded to an integer value. In what follows, the first benchmark will be referred to as the RW benchmark, whereas the second benchmark will be referred to as the Solomon benchmark and the third benchmark will be referred to as the Solomon benchmark Ver2.

Each instance in the RW benchmark contains a MASPDP that occurred on a particular day. These instances are categorised into six types, "M", "'R", "T", "A", "B", and "C". Each type represents the situation of a particular depot. For each instance, the depot and the customers have a suburb number that represents the location. Each suburb number has its longitude and altitude which are used to construct the distance matrix. For the distance between customers (or distance between depot and customers) with the same suburb number, it is assumed to be 1 mile. Furthermore, it is also assumed that all vehicles require 2 minutes to travel 1 mile. In addition, all drivers cannot work longer than 10 hours.

The instances in the Solomon benchmark and Solomon benchmark Ver2 combine the Solomon data sets with 100 customers Solomon (1987) and the data provided by the industry partner. These instances were generated in the following way. First, the data provided by the industry partner is pre-processed and obtains four lists L_1, L_2, L_3, and L_4. Each element in list L_1 (L_2) contains the weight and volume of the delivery and pickup required by a two-men customer (one-man customer) whereas each element in list L_3 (L_4) contains the arrival time at the depot; depart time from the depot (arrival time at the depot plus loading time); and the capacity on weight and volume of a two-men vehicle (one-man vehicle). For each instance of the Solomon benchmark, a customer from the corresponding instance of the Solomon data sets was randomly paired with an element in either L_1 or L_2 and the data regarding a vehicle was randomly selected from either L_3 or L_4. Besides, the Solomon benchmark takes into account the information on instances from the RW benchmark i.e., the ratios between the number of two-men customers and the total number of customers, as well as the ratios between the number of two-men vehicles and the total number of vehicles. A ratio with regards to the two-men customers is randomly selected and is used to determine the number of customers which is paired with L_1. Moreover, a ratio with regards to the two-men vehicle is also randomly selected and is used to determine the number of two-men vehicles which is selected from L_3. The rest of the customers will be paired with L_2 and the rest of the vehicles will be selected from L_4. Since the Solomon data sets have different lengths on time horizons and different densities on time windows, the starting time and loading time for drivers at the depot are adjusted to make them suitable for the time window and time horizon of a particular instance. At last, the number of vehicles in each instance was also adjusted by preliminary tests, so that, the number of vehicles is not sufficient to allocate all customers.

4.4.2 Analysis on search strategies

Since the ILS2O alternates between two objectives (see Section 4.3.4), the five search strategies described in Section 4.3.5 are selected twice, one for each objective. These choices have resulted in 25 different combinations. The results obtained from ILS2O for these 25 combinations are presented in Table 4.1.

In Table 4.1, the columns Average and sec_a show the average objective values and the average computational time. In each row, different strategies are tested for maximising the number of served customers while the choice for minimising the total travel time is fixed, and vice versa for each column. The best objective values are underlined in this table. They are obtained by Strategy 3 for both objectives. 57.10 is the average number of served customers whereas 683.94 is the average total travel time. All other entries in column Average are the relative percentage differences using the best objective values as the reference. For the first percentage, if it is positive then the average number of served customs is less than 57.10 whereas for the second percentage, if it is negative then the average total travel time is larger than 683.94. In terms of computation time, using Strategy 3 for both objectives requires 22.89 seconds which is efficient. Since the performance of Strategy 3 is promising for both objectives, this setting is used for the ILS2O in the following computational experiments.

Table 4.1: Analysis on the performance of ILS2O with different search strategies on instances from Solomon benchmark

	S1		**S2**		**S3**		**S4**		**S5**	
	Average	sec_a	Average	sec_a	Average	sec_a	Average	sec_a	Average	sec_a
S1	3.36%| − 53.3%	1.54	2.99%| − 53.42%	1.61	0.33%| − 54.16%	9.52	0.40%| − 54.73%	9.28	0.33%| − 54.34%	9.19
S2	2.98%| − 52.14%	1.83	2.68%| − 52.27%	1.85	0.26%| − 53.53%	10.36	0.30%| − 54.25%	9.91	0.28%| − 53.72%	10.10
S3	0.21%| − 0.42%	13.92	0.21%| − 1.87%	13.79	57.10|683.94	22.89	0.00%| − 0.04%	22.54	0.00| − 0.02%	22.77
S4	0.25%| − 3.97%	15.14	0.21%| − 5.71%	15.01	0.02%| − 4.50%	23.45	0.02%| − 4.37%	22.86	0.02%| − 4.45%	23.04
S5	0.26%| − 0.90%	13.85	0.23%| − 2.20	13.51	0.02%| − 0.78%	22.43	0.02%| − 0.52%	21.66	0.02%| − 0.43%	21.80

4.4.3 Comparison of the performance

This section reports the results obtained by the ILS2O, ILS, 2Phase, ILS-NR and CPLEX. The results obtained from CPLEX on instances from the RW benchmark, Solomon benchmark, and Solomon benchmark Ver2 are presented in Tables 4.2, 4.3, and 4.4. The average results obtained from the ILS2O, 2Phase and ILS on instances from the RW benchmark, the Solomon benchmark and the Solomon benchmark Ver2 are presented in Tables 4.5, 4.8, and 4.12 whereas the best and worst cases obtained by these algorithms are reported in Tables 4.6, 4.9, and 4.12. In addition, the average results, best cases and worst cases obtained by the ILS-NR are reported in Tables 4.7, 4.10, and 4.13.

In Tables 4.2, 4.3, and 4.4, the first column presents the instances' name, and the columns $|C|$ and $|T|$ present the number of customers and the number of vehicles. Each column O1 presents the number of served customers in the solution obtained by CPLEX; each column O2 presents the total travel time incurred in the solution obtained by CPLEX; each column Gap(%) presents the optimality gap; each column Time(s) presents the computational time required by CPLEX.

With a time limit of 6 hours and a memory limit of 8 GB, CPLEX cannot find the optimal solution for most of the instances from all three benchmarks. CPLEX cannot even find optimal solutions for many instances with customers less than 50, whereas CPLEX can find optimal solutions in the instances for the Workforce Scheduling and Routing Problem (WSRP) with 25 and 50 tasks. This indicates that the MASPDP studied in this chapter is computationally more challenging than the instances for WSRP studied in Chapter 3. The WSRP studied in Chapter 3 contains the time window constraints, shift duration constraints, and compatibility constraints. The MASPDP studied in this chapter contains many additional constraints, including a heterogeneous fleet of vehicles; weight and volume of the demands; maximum shift length on the drivers; open routes; and a roster specifying the order of vehicle loading at the depot. Considering these additional features simultaneously makes the problem more challenging than the instances for the WSRP. Furthermore, for instances "M20171009" and "M20171010", CPLEX with the weighted three-index model can obtain the optimal solutions. For instances "c101", "c201", and "r101" from the Solomon benchmark, CPLEX with the weighted three-index model has produced the optimal solutions. For instances from the Solomon benchmark Ver2, CPLEX cannot obtain the optimal solution for almost every instance except the instance "r101".

In Tables 4.5 − 4.13, the groups ILS2O, ILS, 2Phase, and ILS-NR report the results obtained by ILS2O, ILS, 2Phase, and ILS-NR, respectively. In Tables, 4.5, 4.7, 4.8, 4.10, 4.11, 4.13, the columns Average and sec_a report the average objective value and average computational time. The column StdV presents the standard deviation over 30 runs. In column Average (StdV), the first number is the average value (standard deviation) with respect to the number of served customers and the second number is the average value (standard deviation) with respect to the total travel time. For the readers' convenience, the best values obtained by these algorithms are underlined.

Furthermore, the best case and the worst case obtained by ILS2O, ILS, 2Phase, and ILS-NR are reported. For 30 runs, the best case represents the solution that gives the best number of served customers with the smallest total travel time whereas the worst case represents the solution that gives the worst number of served customers with the largest total travel time. In Tables 4.6, 4.9, 4.12, the number of served customers and the total travel time for the best solution and the worst solution obtained by the ILS2O, ILS, and 2Phase are reported. The column bestj (bestd) reports the number of served customers (total travel time) of the best solution whereas the column worstj (worstd) reports the number of served customers (total travel time) of the worst solution. The best solution and worst solution obtained by the ILS-NR are also reported which can be found in Tables 4.7, 4.10, and 4.13. In these tables, the group Best (Worst) contains the objective values of the best solution (worst solution) where column O1 reports the

Table 4.2: Comparison of performance between weighted three-index model on instances from RW benchmark

				Weighted three-index			
	$\|C\|$	$\|T\|$	λ_1	O1	O2	Gap(%)	Time(s)
M20170723	30	3	314	28	608	0.00	7349.72
M20170724	26	2	334	22	468	0.00	5603.80
M20170725	14	2	350	14	312	0.00	0.95
M20171008	28	2	302	26	592	5.22	36000.00
M20171009	22	2	314	21	438	0.00	10178.07
M20171010	22	2	410	17	518	0.00	30096.77
M20171016	34	2	338	26	476	19.93	36000.00
M20171017	24	2	332	22	632	4.22	36000.00
M20171021	34	2	332	26	356	18.76	36000.00
M20171024	17	2	288	17	440	0.00	3.25
M20171030	37	2	404	30	580	16.04	36000.00
M20171222	72	7	316	56	1852	37.44	36000.00
M20171223	70	5	454	64	1182	11.24	36000.00
M20171224	70	5	348	55	1134	29.94	36000.00
M20171225	70	5	352	55	1124	21.08	36000.00
R20170723	47	5	266	47	580	0.84	36000.00
R20170724	65	3	290	53	628	4.71	36000.00
R20170725	43	4	220	42	726	3.76	4651.49
R20171008	88	6	256	74	1130	21.34	36000.00
R20171009	63	4	268	55	884	6.48	36000.00
R20171010	44	5	268	44	580	0.62	36000.00
R20171016	72	5	294	66	1250	10.22	36000.00
R20171017	37	4	402	36	734	2.56	36000.00
R20171021	60	5	204	55	856	10.15	8187.09
R20171024	53	6	334	53	668	0.67	36000.00
R20171030	71	7	344	70	1380	4.54	36000.00
R20171212	52	4	342	50	992	5.72	36000.00
R20171219	52	4	268	50	880	5.19	36000.00
R20171222	62	4	256	55	902	13.95	36000.00
R20171223	70	5	320	67	1030	5.37	36000.00
R20171224	70	5	296	60	1160	7.10	36000.00
R20171225	70	5	304	68	1174	5.43	36000.00
T20170723	64	5	228	64	570	1.54	36000.00
T20170724	70	5	194	69	684	2.29	9525.44
T20170725	57	4	210	55	630	5.72	36000.00
T20171008	65	8	308	65	826	1.43	36000.00
T20171009	43	7	332	43	564	0.30	36000.00
T20171010	46	5	380	46	508	0.41	36000.00
T20171016	63	7	312	63	776	1.79	36000.00
T20171017	56	4	467	52	670	5.39	36000.00
T20171021	76	4	206	60	608	7.74	36000.00
T20171024	62	4	272	54	860	8.13	36000.00
T20171030	36	5	244	36	302	0.30	36000.00
T20171212	63	7	238	63	1078	3.59	36000.00
T20171219	54	5	318	54	734	1.49	36000.00
T20171222	91	7	236	79	972	16.20	36000.00
T20171223	70	5	296	66	990	8.43	36000.00
T20171224	70	5	262	63	942	12.26	9447.68
T20171225	70	5	398	67	1008	5.41	36000.00
T20171226	70	5	376	66	904	6.54	36000.00
A20171016	100	4	418	61	1074	15.06	36000.00
A20171222	100	7	458	81	1612	17.06	36000.00
B20171008	100	6	344	75	1302	13.34	36000.00
B20171016	100	5	322	77	1146	8.63	7275.28
B20171030	100	7	352	84	1406	11.85	36000.00
B20171222	100	4	346	63	940	31.19	36000.00
C20170724	100	5	332	90	918	6.55	36000.00
C20171016	100	7	258	94	1274	8.38	36000.00
C20171021	100	4	268	73	950	13.75	36000.00
C20171222	100	7	380	91	1478	12.02	36000.00

Table 4.3: Comparison of performance between weighted three-index model on instances from Solomon benchmark

	$\|C\|$	$\|T\|$	λ_1	Weighted three-index			
				O1	O2	Gap(%)	Time(s)
c101	100	5	133	46	226	0.00	585.53
c102	100	2	146	20	149	147.54	4903.90
c103	100	6	140	61	526	56.32	36000.00
c104	100	3	158	28	165	133.77	5841.00
c105	100	5	153	46	365	44.76	36000.00
c106	100	5	197	46	289	81.24	19827.53
c107	100	5	176	48	424	67.38	36000.00
c108	100	6	159	52	570	88.16	36000.00
c109	100	5	116	49	437	90.63	36000.00
c201	100	3	188	66	701	0.00	612.97
c202	100	3	253	64	852	3.85	5040.40
c203	100	5	146	81	1277	9.76	36000.00
c204	100	6	225	95	1273	6.21	36000.00
c205	100	3	139	65	536	1.96	36000.00
c206	100	3	262	77	881	7.72	36000.00
c207	100	3	285	71	824	4.05	36000.00
c208	100	4	211	82	798	3.23	36000.00
r101	100	3	105	23	183	0.00	13.66
r102	100	6	99	47	473	112.58	36000.00
r103	100	5	111	45	335	93.35	36000.00
r104	100	4	105	38	262	131.72	36000.00
r105	100	5	94	41	383	72.62	36000.00
r106	100	3	96	28	193	139.40	36000.00
r107	100	4	108	36	293	107.20	36000.00
r108	100	3	106	29	184	141.60	36000.00
r109	100	3	120	29	196	115.16	36000.00
r110	100	3	96	29	178	149.29	36000.00
r111	100	3	105	31	175	166.98	36000.00
r112	100	3	96	30	188	131.89	36000.00
r201	100	3	250	51	816	15.67	7702.84
r202	100	5	362	94	1683	6.53	36000.00
r203	100	6	222	89	1603	10.64	36000.00
r204	100	5	287	83	857	4.09	36000.00
r205	100	3	199	63	886	9.35	36000.00
r206	100	5	272	92	1020	3.97	36000.00
r207	100	3	267	63	755	4.75	36000.00
r208	100	3	246	75	726	2.80	36000.00
r209	100	4	251	76	1060	5.92	36000.00
r210	100	8	250	70	1242	47.26	36000.00
r211	100	4	158	73	1039	10.25	36000.00
rc101	100	3	129	27	248	67.90	36000.00
rc102	100	2	131	18	148	249.54	36000.00
rc103	100	3	124	23	267	178.61	6406.04
rc104	100	3	102	28	250	168.18	36000.00
rc105	100	5	137	42	354	88.11	36000.00
rc106	100	3	123	26	204	149.51	36000.00
rc107	100	3	117	27	216	130.23	36000.00
rc108	100	6	113	40	418	154.63	36000.00
rc201	100	3	342	58	1020	18.30	36000.00
rc202	100	3	337	50	956	18.95	36000.00
rc203	100	3	217	58	1132	22.23	36000.00
rc204	100	3	253	57	1064	29.43	2381.00
rc205	100	3	229	56	973	12.11	36000.00
rc206	100	3	330	65	880	11.20	36000.00
rc207	100	6	420	79	1551	8.43	36000.00
rc208	100	7	255	72	1221	44.42	36000.00

Table 4.4: Comparison of performance between weighted three-index model on instances from Solomon Ver2 benchmark

				Weighted three-index			
	$\|C\|$	$\|T\|$	λ_1	O1	O2	Gap(%)	Time(s)
c102	100	2	165	20	208	80.81	36000
c103	100	6	192	47	610	28.61	36000
c104	100	3	150	28	191	93.75	36000
c105	100	5	196	42	494	34.97	36000
c106	100	5	191	42	413	70.89	36000
c107	100	5	297	43	517	45.32	36000
c108	100	6	164	45	405	78.03	36000
c109	100	5	300	40	538	77.77	36000
c201	100	3	545	57	1273	1.30	36000
c202	100	3	462	56	1145	3.62	36000
c203	100	5	581	73	875	2.25	36000
c204	100	6	386	95	1252	3.23	36000
c205	100	3	728	60	1075	0.68	36000
c206	100	3	540	70	1133	6.05	36000
c207	100	3	630	67	1073	3.32	36000
c208	100	4	516	80	1320	3.62	5404
r101	100	3	119	22	200	0.00	2
r102	100	6	115	47	445	105.97	36000
r103	100	5	105	34	319	115.49	11098
r104	100	4	109	36	305	143.98	36000
r105	100	5	104	26	264	48.12	36000
r106	100	3	104	27	248	121.85	36000
r107	100	4	116	33	347	71.63	36000
r108	100	3	88	28	235	130.39	36000
r109	100	3	121	22	243	101.02	36000
r110	100	3	102	25	203	165.87	10407
r111	100	3	113	27	236	85.16	36000
r112	100	3	91	26	219	155.28	36000
r201	100	3	217	46	718	8.69	36000
r202	100	5	520	66	869	8.35	36000
r203	100	6	458	84	1371	9.58	36000
r204	100	5	373	74	872	3.60	36000
r205	100	3	281	45	680	4.81	36000
r206	100	5	387	77	870	2.13	36000
r207	100	3	435	58	739	4.59	36000
r209	100	4	445	70	1101	2.98	36000
r210	100	8	334	85	1293	3.43	4651
r211	100	4	326	69	999	9.16	13490
rc101	100	3	128	24	264	17.19	36000
rc102	100	2	117	16	148	187.21	36000
rc103	100	3	136	22	248	182.99	36000
rc105	100	5	125	30	3750	66.79	36000
rc106	100	3	121	25	251	118.64	36000
rc107	100	3	128	25	259	89.22	36000
rc108	100	6	114	38	442	135.02	3977
rc203	100	3	372	51	969	20.19	36000
rc204	100	3	399	52	683	2.48	36000
rc205	100	3	582	53	1097	9.09	36000
rc206	100	3	509	45	881	14.54	3007
rc207	100	6	519	68	1602	10.96	2747
rc208	100	7	504	57	1042	56.08	3129

number of served customers and column O2 reports the total travel time.

For 60 instances from the RW benchmark, the ILS2O obtains solutions that serve more customers than the solutions obtained by the ILS in 22 instances. For the remaining instances, the ILS2O obtains solutions that serve the same number of customers but lower total travel time than the solutions obtained by the ILS in 13 instances. The ILS either finds better solutions or the same solutions in 25 instances. Comparing the solutions obtained by the 2Phase algorithm, the ILS2O finds solutions with a higher number of served customers in 27 instances. In 16 instances, the ILS2O finds solutions with the same number of served customers but lower total travel time. In the remaining 17 instances, the 2Phase algorithm finds better solutions than the solutions obtained by the ILS2O. Moreover, for all instances, the ILS2O outperforms the ILS-NR in terms of solution quality.

Out of the 56 instances from the Solomon benchmark, the solutions obtained by the ILS2O serve more customers than the solutions obtained by the ILS in 36 instances. For 9 instances out of 56 instances, the solutions obtained by the ILS2O serve the same number of customers with a lower total travel time compared with the solutions obtained by the ILS. For the remaining instances, the ILS finds better solutions. Comparing the solutions obtained by the 2Phase algorithm, the ILS2O finds solutions with a higher number of served customers in 41 instances. In 8 instances, the ILS2O finds solutions with the same number of served customers with a lower total travel time. For the remaining 7 instances, the 2Phase algorithm finds better solutions than the solutions obtained by the ILS2O. Similar to the results for the RW benchmark, the solutions obtained from ILS2O are better than the solutions obtained by the ILS-NR for all instances.

For 51 instances from the Solomon benchmark Ver2, there are 19 instances that the ILS2O obtains solutions with a higher number of served customers than the solutions obtained by the ILS. For the remaining instances, the ILS2O obtains solutions that serve the same number of customers with a lower total travel time than the solutions obtained by the ILS in 12 instances. The ILS finds better solutions in the remaining 20 instances. Comparing the solutions obtained by the 2Phase algorithm, the ILS2O outperforms the 2Phase algorithm in 40 instances in terms of the number of served customers. In 7 instances, the ILS2O finds solutions with the same number of served customers with a lower total travel time. In the remaining 4 instances, the 2Phase algorithm finds better solutions than the solutions obtained by the ILS2O. Similar to the results for the RW benchmark and Solomon benchmark, the ILS2O again outperforms the ILS-NR in all instances from the Solomon benchmark Ver2.

In addition, this section also investigates the consistency of the ILS2O, ILS, 2Phase, and ILS-NR. For instances from all three sets of benchmarks, the ILS2O is more stable than the ILS, 2Phase, and ILS-NR in terms of both the number of served customers and the total travel time. For 50 out of 60 instances from the RW benchmark, the standard deviation obtained from ILS2O for the number of served customers is 0. This means that the ILS2O constantly finds solutions with the same number of served customers. For instances from the Solomon benchmark and Solomon benchmark Ver2, larger variances have been observed. However, it can be observed that the ILS2O is still more stable in comparison with the ILS, 2Phase, and ILS-NR in terms of both objectives.

In terms of computational time, the ILS requires more time in comparison with the ILS2O, 2Phase, and ILS-NR. The author believes that the local search struggles to find the local optimal when the objective function is a weighted sum. Although the ILS2O alternates between the two objectives, the computational time is still competitive with the computational time required by the ILS, 2Phase, and ILS-NR after observing its promising performance on the solution quality and consistency.

The effectiveness of the neighbourhood reduction technique has been verified in Gu et al. (2021) with the ILS-NR whereas the effectiveness of the Lagrangian ILS framework has been verified in Gu et al. (2022b). With the comparisons of performance on the solutions obtained by the ILS2O and ILS-NR, the ILS2O outperforms ILS-NR in all cases. Since the ILS2O is an algorithm under the Lagrangian ILS framework with the neighbourhood reduction technique, these comparisons indicate that both the Lagrangian ILS framework and the neighbourhood reduction technique are effective in solving the MASPDP studied in this chapter.

4.5 Conclusion

This chapter considers a practical vehicle routing problem with simultaneous pickups and deliveries and ordered objectives. The problem is formulated into a three-index mathematical formulation. To tackle the problem, this chapter described an iterated local

Table 4.5: Comparison of performance between ILS2O , ILS and 2Phase ILS on instances from RW benchmark

			ILS2O			**ILS**			**2Phase**		
	$\|C\|$	$\|T\|$	Average	StdV	sec_a	Average	StdV	sec_a	Average	StdV	sec_a
M20170723	30	3	28.00|610.93	0.00|2.08	0.53	26.93|837.20	0.64|71.18	0.37	28.00|719.87	0.00|135.20	0.60
M20170724	26	2	22.00|470.27	0.00|8.15	0.33	21.80|493.80	1.10|24.11	0.33	22.00|487.67	0.00|37.08	0.47
M20170725	14	2	14.00|312.00	0.00|0.00	0.00	14.00|320.87	0.00|6.86	0.17	14.00|321.80	0.00|6.53	0.13
M20171008	28	2	26.00|564.60	0.00|6.95	0.53	25.83|576.07	0.38|22.42	0.43	25.13|544.60	0.51|53.02	0.50
M20171009	22	2	21.00|438.00	0.00|0.00	0.03	19.40|555.00	0.77|42.87	0.13	21.00|438.67	0.00|1.52	0.17
M20171010	22	2	17.00|520.40	0.00|4.88	0.10	16.80|559.40	0.76|40.48	0.17	17.00|533.47	0.00|10.41	0.20
M20171016	34	2	27.00|535.67	0.00|10.09	0.97	26.97|567.13	0.18|31.12	1.47	26.43|516.67	0.50|56.87	0.80
M20171017	24	2	22.00|632.53	0.00|1.38	0.30	22.00|632.40	0.00|1.22	0.27	21.77|616.80	0.43|29.02	0.40
M20171021	34	2	27.93|475.47	0.25|31.55	0.83	27.77|472.87	0.43|34.44	1.20	27.13|405.73	0.35|36.38	0.90
M20171024	17	2	17.00|440.00	0.00|0.00	0.03	17.00|440.00	0.00|0.00	0.03	17.00|440.27	0.00|1.01	0.03
M20171030	37	2	30.00|542.00	0.00|0.00	0.93	30.00|558.67	0.00|12.70	2.00	29.63|537.27	0.56|39.14	0.97
M20171222	72	7	71.00|1468.53	0.00|44.79	25.90	71.00|1505.73	0.00|60.06	26.03	70.97|1465.87	0.18|68.62	28.60
M20171223	70	5	69.33|1289.27	0.48|91.60	17.17	68.70|1229.27	0.53|75.03	27.57	68.07|1076.80	0.45|129.30	22.40
M20171224	70	5	59.00|1087.20	0.00|22.70	17.50	59.00|1125.40	0.00|47.45	46.70	58.90|1479.40	0.31|145.34	12.23
M20171225	70	5	60.00|1202.80	0.00|53.66	21.87	59.80|1259.67	0.41|81.84	24.80	59.03|1367.87	0.61|244.44	13.13
R20170723	47	5	47.00|559.47	0.00|7.84	1.77	47.00|562.13	0.00|8.10	2.70	47.00|562.40	0.00|6.13	2.07
R20170724	65	3	53.97|660.20	0.18|23.05	11.97	54.00|659.27	0.00|22.78	13.23	53.03|569.07	0.32|36.63	10.97
R20170725	43	4	42.00|511.67	0.00|13.07	5.10	42.00|501.73	0.00|18.65	3.23	42.00|501.13	0.00|20.42	4.13
R20171008	88	6	86.00|921.20	0.00|25.76	45.30	86.00|924.13	0.00|32.30	47.67	86.00|906.67	0.00|34.41	55.50
R20171009	63	4	57.00|848.73	0.00|22.63	16.40	57.00|872.27	0.00|45.84	15.77	56.73|821.07	0.45|87.06	18.07
R20171010	44	5	44.00|574.13	0.00|3.52	1.27	44.00|573.27	0.00|2.49	2.47	44.00|573.80	0.00|3.29	2.00
R20171016	72	5	70.00|1022.47	0.00|31.01	16.70	70.00|974.40	0.00|35.47	26.07	70.00|985.60	0.00|43.86	30.17
R20171017	37	4	37.00|1065.00	0.00|31.82	1.17	36.67|951.60	0.48|145.43	2.13	36.87|1016.93	0.35|114.83	1.83
R20171021	60	5	58.00|694.07	0.00|25.34	15.07	58.00|688.80	0.00|33.94	14.50	58.00|673.00	0.00|37.62	15.53
R20171024	53	6	53.00|636.67	0.00|10.57	2.77	53.00|635.60	0.00|15.84	5.50	53.00|633.87	0.00|7.39	5.67
R20171030	71	7	71.00|1059.27	0.00|39.51	10.70	71.00|1054.27	0.00|43.28	23.70	71.00|1067.73	0.00|45.37	23.50
R20171212	52	4	52.00|912.67	0.00|10.51	3.30	52.00|915.93	0.00|14.92	6.90	52.00|918.53	0.00|16.51	6.17
R20171219	52	4	51.00|714.73	0.00|17.09	5.37	51.00|707.07	0.00|21.44	7.03	51.00|704.13	0.00|19.38	9.03
R20171222	62	4	59.00|816.40	0.00|35.00	15.10	58.97|840.13	0.18|45.75	17.17	58.13|742.40	0.35|50.78	13.90
R20171223	70	5	69.00|885.67	0.00|25.83	25.83	69.00|914.07	0.00|49.38	20.70	68.80|860.73	0.41|57.75	25.57
R20171224	70	5	62.00|1069.67	0.00|37.52	21.13	62.00|1045.33	0.00|41.29	21.80	62.00|1065.67	0.00|56.84	16.37
R20171225	70	5	70.00|901.47	0.00|19.75	7.43	70.00|905.87	0.00|22.56	13.43	70.00|900.33	0.00|27.73	9.63
T20170723	64	5	64.00|431.07	0.00|6.72	5.13	64.00|430.33	0.00|6.75	8.83	64.00|432.27	0.00|5.87	5.40
T20170724	70	5	69.00|539.73	0.00|12.99	25.20	69.00|527.67	0.00|13.78	16.13	69.00|516.73	0.00|16.37	20.47
T20170725	57	4	57.00|547.80	0.00|13.90	5.20	57.00|545.87	0.00|13.46	7.77	57.00|550.07	0.00|16.68	6.90
T20171008	65	8	65.00|629.33	0.00|7.15	4.83	65.00|628.67	0.00|8.02	13.53	65.00|628.87	0.00|8.64	13.40
T20171009	43	7	43.00|567.47	0.00|4.33	1.53	43.00| 569.47	0.00|6.93	4.17	43.00|569.60	0.00|7.30	3.90
T20171010	46	5	46.00|481.73	0.00|5.17	2.00	46.00|481.60	0.00|5.79	2.20	46.00|483.93	0.00|5.52	2.10
T20171016	63	7	63.00|504.67	0.00|5.16	5.10	63.00|502.53	0.00|6.19	7.87	63.00|503.60	0.00|6.88	6.10
T20171017	56	4	54.00|763.47	0.00|4.55	13.00	53.93|763.00	0.25|44.74	15.60	53.37|661.13	0.49|83.29	11.97
T20171021	76	4	63.00|751.20	0.00|26.68	29.57	62.60|661.67	0.50|111.03	33.57	62.53|653.20	0.51|109.17	28.93
T20171024	62	4	57.00|899.53	0.00|31.87	12.83	57.00|885.60	0.00|28.40	18.33	56.43|836.73	0.50|71.09	15.50
T20171030	36	5	36.00|302.87	0.00|2.27	0.77	36.00|304.33	0.00|2.88	0.87	36.00|303.73	0.00|2.66	0.87
T20171212	63	7	63.00|660.87	0.00|4.02	4.30	63.00|660.93	0.00|3.23	5.67	63.00|658.67	0.00|3.94	5.47
T20171219	54	5	54.00|560.27	0.00|5.75	3.37	54.00|559.27	0.00|8.53	5.43	54.00|561.60	0.00|7.71	4.43
T20171222	91	7	89.00|963.60	0.00|39.56	57.90	89.00|931.53	0.00|37.91	54.07	89.00|918.40	0.00|37.64	56.77
T20171223	70	5	70.00|877.87	0.00|19.93	8.03	70.00|874.27	0.00|26.01	16.17	70.00|879.00	0.00|27.36	14.10
T20171224	70	5	69.00|1171.40	0.00|44.76	20.53	68.63|1022.27	0.49|113.05	32.13	68.07|893.13	0.25|58.92	29.53
T20171225	70	5	69.13|879.93	0.35|129.11	28.00	69.03|847.13	0.18|70.97	27.30	69.00|834.27	0.00|21.78	32.13
T20171226	70	5	68.00|761.93	0.00|18.14	32.17	68.00|759.20	0.00|20.45	23.30	68.00|751.60	0.00|22.32	27.27
A20171016	100	4	63.77|1093.33	0.43|61.71	26.00	63.57|1122.87	0.57|66.91	31.97	62.53|1035.80	0.68|82.56	19.40
A20171222	100	7	87.17|1578.67	0.53|82.85	57.53	86.67|1576.07	0.55|67.73	58.23	85.23|1607.67	0.57|253.91	37.30
B20171008	100	6	81.17|1086.67	0.38|73.64	55.43	81.10|1083.33	0.31|58.33	45.70	80.37|982.07	0.49|80.34	49.03
B20171016	100	5	81.00|1187.93	0.00|37.65	36.23	80.83|1221.47	0.38|47.67	43.70	79.73|1118.67	0.58|85.19	37.70
B20171030	100	7	89.13|1347.33	0.35|82.79	59.90	88.80|1347.27	0.41|75.69	57.50	88.07|1264.93	0.45|127.74	53.97
B20171222	100	4	71.57|981.60	0.50|47.94	39.90	71.40|1009.47	0.50|42.50	48.57	69.20|829.20	0.71|60.86	33.07
C20170724	100	5	94.00|806.13	0.00|20.30	46.53	93.83|769.53	0.38|71.53	56.37	93.07|637.47	0.25|63.15	58.97
C20171016	100	7	98.00|847.40	0.00|30.63	42.03	98.00|779.20	0.00|31.38	59.73	98.00|806.40	0.00|35.23	60.00
C20171021	100	4	77.77|784.33	0.43|44.40	56.57	77.80|787.73	0.41|54.33	53.57	76.70|682.07	0.75|89.56	45.07
C20171222	100	7	99.00|1330.67	0.00|42.31	59.87	99.00|1266.67	0.00|44.90	59.87	99.00|1289.93	0.00|51.36	59.67
Average			57.25|796.37	0.06|26.07	17.21	57.13|796.31	0.18|37.34	19.73	56.90|772.44	0.20|51.72	17.85

Table 4.6: Comparison on the best and worst performance between ILS2O, ILS and 2Phase ILS on instances from RW benchmark

			ILS2O				**ILS**				**2Phase**			
	$\|C\|$	$\|T\|$	bestj	bestd	worstj	worstd	bestj	bestd	worstj	worstd	bestj	bestd	worstj	worstd
M20170723	30	3	28	608	28	616	28	628	26	914	28	610	28	994
M20170724	26	2	22	468	22	512	22	468	16	568	22	468	22	666
M20170725	14	2	14	312	14	312	14	312	14	326	14	312	14	326
M20171008	28	2	26	560	26	576	26	560	25	548	26	560	24	448
M20171009	22	2	21	438	21	438	21	438	18	602	21	438	21	442
M20171010	22	2	17	518	17	530	17	518	14	738	17	518	17	548
M20171016	34	2	27	528	27	562	27	536	26	448	27	528	26	596
M20171017	24	2	22	632	22	636	22	632	22	636	22	632	21	580
M20171021	34	2	28	456	27	384	28	456	27	448	28	456	27	426
M20171024	17	2	17	440	17	440	17	440	17	440	17	440	17	444
M20171030	37	2	30	542	30	542	30	542	30	580	30	542	28	420
M20171222	72	7	71	1350	71	1534	71	1400	71	1608	71	1356	70	1306
M20171223	70	5	70	1306	69	1366	70	1396	68	1282	69	1168	67	944
M20171224	70	5	59	1038	59	1126	59	1048	59	1246	59	1152	58	1522
M20171225	70	5	60	1074	60	1338	60	1194	59	1220	60	1182	58	1620
R20170723	47	5	47	548	47	574	47	548	47	576	47	548	47	578
R20170724	65	3	54	604	53	612	54	608	54	714	54	626	52	454
R20170725	43	4	42	482	42	538	42	478	42	550	42	476	42	548
R20171008	88	6	86	864	86	990	86	872	86	986	86	848	86	978
R20171009	63	4	57	806	57	916	57	792	57	976	57	772	56	790
R20171010	44	5	44	572	44	584	44	572	44	580	44	572	44	582
R20171016	72	5	70	972	70	1104	70	916	70	1078	70	916	70	1112
R20171017	37	4	37	1036	37	1154	37	1036	36	774	37	1036	36	746
R20171021	60	5	58	644	58	758	58	640	58	760	58	590	58	766
R20171024	53	6	53	628	53	678	53	628	53	702	53	628	53	654
R20171030	71	7	71	986	71	1156	71	990	71	1178	71	976	71	1164
R20171212	52	4	52	896	52	944	52	894	52	960	52	894	52	956
R20171219	52	4	51	686	51	762	51	672	51	770	51	676	51	760
R20171222	62	4	59	758	59	914	59	778	58	772	59	804	58	804
R20171223	70	5	69	840	69	930	69	810	69	1032	69	774	68	786
R20171224	70	5	62	980	62	1170	62	966	62	1124	62	976	62	1176
R20171225	70	5	70	870	70	942	70	868	70	954	70	862	70	946
T20170723	64	5	64	418	64	442	64	420	64	450	64	420	64	446
T20170724	70	5	69	520	69	576	69	506	69	552	69	496	69	560
T20170725	57	4	57	524	57	590	57	526	57	574	57	526	57	606
T20171008	65	8	65	620	65	644	65	620	65	650	65	620	65	648
T20171009	43	7	43	564	43	578	43	564	43	588	43	564	43	588
T20171010	46	5	46	476	46	500	46	476	46	502	46	476	46	500
T20171016	63	7	63	492	63	514	63	492	63	514	63	492	63	518
T20171017	56	4	54	760	54	780	54	760	53	632	54	762	53	628
T20171021	76	4	63	704	63	802	63	668	62	562	63	698	62	606
T20171024	62	4	57	846	57	966	57	822	57	948	57	848	56	824
T20171030	36	5	36	302	36	310	36	302	36	310	36	302	36	310
T20171212	63	7	63	652	63	670	63	656	63	672	63	652	63	666
T20171219	54	5	54	552	54	576	54	554	54	586	54	554	54	578
T20171222	91	7	89	884	89	1040	89	856	89	1002	89	850	89	1012
T20171223	70	5	70	836	70	922	70	834	70	934	70	828	70	940
T20171224	70	5	69	1086	69	1262	69	1016	68	962	69	1016	68	930
T20171225	70	5	70	1196	69	856	70	1210	69	888	69	794	69	888
T20171226	70	5	68	722	68	798	68	704	68	790	68	704	68	800
A20171016	100	4	64	1000	63	1072	64	998	62	1036	63	958	61	1086
A20171222	100	7	88	1612	86	1526	88	1700	86	1638	86	1468	84	1476
B20171008	100	6	82	1184	81	1098	82	1200	81	1136	81	1000	80	1066
B20171016	100	5	81	1110	81	1276	81	1144	80	1224	81	1272	79	1088
B20171030	100	7	90	1426	89	1370	89	1258	88	1372	89	1366	87	1100
B20171222	100	4	72	960	71	1018	72	1002	71	1052	70	828	68	798
C20170724	100	5	94	742	94	842	94	732	93	678	94	842	93	680
C20171016	100	7	98	778	98	910	98	714	98	856	98	734	98	876
C20171021	100	4	78	772	77	744	78	748	77	740	78	748	74	486
C20171222	100	7	99	1246	99	1408	99	1172	99	1356	99	1210	99	1462
Average			57.33	773.77	57.15	828.80	57.32	771.50	56.72	821.57	57.18	756.07	56.53	787.47

Table 4.7: The performance ILS-NR on instances from RW benchmark

			Average		Std V		Time(s)	Best		Worst	
	$\lvert C\rvert$	$\lvert T\rvert$	O1	O2	O1	O2		O1	O2	O1	O2
M20170723	30	3	28.00	812.00	0.00	0.00	0.13	28	812	28	812
M20170724	26	2	21.93	645.33	0.25	40.80	0.10	22	558	21	680
M20170725	14	2	14.00	446.00	0.00	0.00	0.00	14	446	14	446
M20171008	28	2	24.63	617.93	0.56	27.93	0.17	26	620	24	678
M20171009	22	2	21.00	635.40	0.00	43.40	0.03	21	562	21	718
M20171010	22	2	17.00	681.27	0.00	52.72	0.07	17	582	17	798
M20171016	34	2	26.10	629.40	0.31	35.81	0.30	27	604	26	686
M20171017	24	2	21.30	662.73	0.53	20.28	0.10	22	640	20	708
M20171021	34	2	26.87	541.33	0.43	27.96	0.33	28	572	26	582
M20171024	17	2	16.90	559.73	0.31	67.96	0.00	17	464	16	444
M20171030	37	2	28.90	628.87	0.40	14.61	0.40	30	622	28	648
M20171222	72	7	69.43	2233.33	0.50	88.68	3.47	70	2114	69	2392
M20171223	70	5	65.80	1538.73	0.61	59.98	3.47	67	1494	65	1616
M20171224	70	5	57.47	1526.67	0.63	66.25	3.20	59	1504	56	1534
M20171225	70	5	57.50	1572.80	0.57	66.84	3.40	59	1458	57	1672
R20170723	47	5	47.00	878.00	0.00	0.00	0.00	47	878	47	878
R20170724	65	3	52.13	776.07	0.43	24.94	2.60	53	756	51	802
R20170725	43	4	42.00	1048.33	0.00	79.79	0.50	42	850	42	1220
R20171008	88	6	85.60	1636.13	0.50	57.48	7.40	86	1486	85	1746
R20171009	63	4	55.27	1174.93	0.45	43.31	2.37	56	1140	55	1258
R20171010	44	5	44.00	916.00	0.00	0.00	0.00	44	916	44	916
R20171016	72	5	68.67	1431.67	0.48	60.04	3.50	69	1300	68	1540
R20171017	37	4	35.93	1256.73	0.25	97.35	0.37	36	1066	35	1386
R20171021	60	5	58.00	1343.67	0.00	70.72	1.80	58	1164	58	1466
R20171024	53	6	53.00	1096.00	0.00	0.00	0.00	53	1096	53	1096
R20171030	71	7	70.67	1962.00	0.48	123.73	1.43	71	1856	70	2098
R20171212	52	4	51.43	1216.13	0.77	30.25	0.67	52	1156	50	1244
R20171219	52	4	50.47	1142.20	0.51	38.65	1.20	51	1096	50	1196
R20171222	62	4	57.03	1168.33	0.41	43.16	2.47	58	1172	56	1196
R20171223	70	5	67.73	1367.60	0.45	47.04	3.23	68	1272	67	1414
R20171224	70	5	60.70	1508.60	0.53	52.43	3.27	62	1478	60	1608
R20171225	70	5	69.77	1482.80	0.43	48.05	0.93	70	1408	69	1514
T20170723	64	5	64.00	618.00	0.00	0.00	0.00	64	618	64	618
T20170724	70	5	69.00	1157.07	0.00	88.72	2.63	69	970	69	1316
T20170725	57	4	56.77	958.67	0.43	46.72	0.57	57	884	56	1024
T20171008	65	8	65.00	1118.00	0.00	0.00	0.00	65	1118	65	1118
T20171009	43	7	43.00	1184.00	0.00	0.00	0.00	43	1184	43	1184
T20171010	46	5	46.00	830.00	0.00	0.00	0.00	46	830	46	830
T20171016	63	7	63.00	836.00	0.00	0.00	0.00	63	836	63	836
T20171017	56	4	52.53	1049.53	0.51	51.46	1.43	53	948	52	1124
T20171021	76	4	61.93	1003.93	0.25	63.19	4.23	62	862	61	1036
T20171024	62	4	55.33	1144.20	0.48	55.88	2.30	56	1088	55	1250
T20171030	36	5	36.00	668.00	0.00	0.00	0.00	36	668	36	668
T20171212	63	7	63.00	1072.00	0.00	0.00	0.00	63	1072	63	1072
T20171219	54	5	54.00	1242.00	0.00	0.00	0.00	54	1242	54	1242
T20171222	91	7	88.73	1948.47	0.45	95.09	7.47	89	1804	88	2030
T20171223	70	5	69.93	1383.13	0.25	46.20	0.67	70	1320	69	1354
T20171224	70	5	67.10	1395.53	0.31	52.72	3.77	68	1380	67	1492
T20171225	70	5	68.53	1324.27	0.51	63.53	3.23	69	1200	68	1444
T20171226	70	5	67.97	1319.93	0.18	67.73	3.07	68	1188	67	1230
A20171016	100	4	61.73	1275.60	0.64	35.37	4.90	63	1236	60	1260
A20171222	100	7	82.70	2137.33	0.60	92.71	9.20	84	2108	82	2274
B20171008	100	6	79.60	1561.60	0.50	60.04	8.70	80	1462	79	1684
B20171016	100	5	78.93	1520.73	0.69	38.36	8.03	80	1506	78	1534
B20171030	100	7	86.93	2027.47	0.64	92.12	9.40	88	1876	86	2184
B20171222	100	4	67.97	1134.33	0.76	33.19	7.90	70	1102	67	1172
C20170724	100	5	92.83	1165.40	0.38	61.26	8.77	93	1068	92	1176
C20171016	100	7	97.97	1640.60	0.18	92.91	6.97	98	1470	97	1684
C20171021	100	4	75.40	1021.00	0.50	41.18	9.67	76	962	75	1100
C20171222	100	7	97.87	2059.20	0.43	63.27	10.20	99	2032	97	2204
Average			56.33	1182.21	0.33	44.53	2.67	56.82	1119.60	55.78	1235.53

Table 4.8: Comparison of performance between ILS2O, ILS and 2Phase ILS on instances from Solomon benchmark

			ILS2O			ILS			2Phase		
	$\vert C \vert$	$\vert T \vert$	Average	StdV	sec_a	Average	StdV	sec_a	Average	StdV	sec_a
c101	100	5	46.00|227.23	0.00|4.33	15.60	46.00|248.23	0.00|21.66	34.10	46.00|272.83	0.00|44.54	10.40
c102	100	2	22.00|86.20	0.00|5.17	4.63	21.93|92.70	0.25|9.17	10.73	21.63|105.37	0.49|21.23	3.17
c103	100	6	63.70|419.93	0.47|22.04	29.33	63.47|402.30	0.51|22.62	51.27	63.43|402.23	0.50|31.71	23.67
c104	100	3	29.00|118.87	0.00|9.23	9.37	29.00|123.33	0.00|13.85	13.07	29.00|132.13	0.00|20.17	5.10
c105	100	5	46.00|247.40	0.00|23.43	15.17	46.00|240.07	0.00|18.60	37.90	46.00|289.57	0.00|57.84	15.23
c106	100	5	46.97|244.57	0.18|41.80	17.83	46.60|264.73	0.50|30.27	39.37	46.90|265.07	0.31|55.07	12.67
c107	100	5	49.00|305.47	0.00|19.07	18.63	48.97|315.13	0.18|13.13	39.43	48.57|339.73	0.50|40.22	12.23
c108	100	6	56.77|390.93	0.43|32.80	23.13	56.60|363.80	0.50|33.22	34.73	56.00|384.87	0.45|89.11	18.83
c109	100	5	51.00|272.97	0.00|31.57	21.30	51.00|282.70	0.00|24.82	46.50	51.00|293.60	0.00|37.26	14.70
c201	100	3	66.00|1062.90	0.25|143.47	40.73	65.47|888.83	0.57|127.83	27.37	65.83|1613.30	0.38|409.37	20.77
c202	100	3	64.00|708.13	0.00|52.84	30.47	63.97|669.57	0.18|67.68	30.73	64.00|864.10	0.00|340.67	24.70
c203	100	5	82.00|683.73	0.00|45.51	51.93	82.00|687.77	0.00|43.64	55.40	82.00|720.20	0.00|55.67	46.93
c204	100	6	97.00|899.23	0.00|46.01	59.07	96.77|811.50	0.43|77.76	59.87	97.00|914.53	0.00|49.62	59.93
c205	100	3	65.00|704.30	0.00|66.97	30.40	64.90|594.80	0.31|59.71	28.30	65.00|671.70	0.00|82.60	27.67
c206	100	3	80.03|1186.97	0.18|108.39	50.67	79.87|992.00	0.35|70.30	48.80	79.40|984.63	0.56|132.33	49.73
c207	100	3	72.00|991.03	0.00|59.81	34.10	71.77|725.47	0.43|93.62	37.13	71.90|831.50	0.31|98.37	39.90
c208	100	4	82.00|772.43	0.00|52.43	52.47	82.00|626.13	0.00|29.95	52.40	82.00|650.60	0.00|42.76	52.73
r101	100	3	22.97|187.47	0.18|4.27	10.70	22.97|188.97	0.18|5.42	23.37	22.07|228.90	0.37|21.34	3.03
r102	100	6	55.80|436.90	0.41|19.09	22.70	54.93|438.43	0.58|19.50	56.90	53.77|460.77	0.94|50.22	15.00
r103	100	5	49.83|321.50	0.46|10.03	17.27	49.53|328.27	0.51|12.67	47.43	48.87|317.07	0.73|15.69	12.67
r104	100	4	45.77|253.10	0.43|9.27	15.87	45.23|255.17	0.43|9.60	47.70	44.57|244.83	0.73|12.17	9.47
r105	100	5	43.87|344.43	0.35|15.40	16.17	43.87|346.37	0.35|18.15	44.83	42.57|363.87	0.77|35.20	10.63
r106	100	3	32.63|187.80	0.49|5.27	11.17	32.27|190.27	0.45|7.78	22.87	31.77|180.40	0.57|7.91	5.77
r107	100	4	42.50|251.53	0.57|7.11	15.00	42.17|258.30	0.38|9.08	34.50	40.97|239.30	0.67|12.57	9.13
r108	100	3	35.00|171.93	0.00|7.20	10.23	34.90|173.17	0.31|6.29	28.30	34.07|170.57	0.58|8.44	5.37
r109	100	3	31.33|191.33	0.48|5.97	8.90	31.37|193.47	0.49|6.86	22.03	30.23|190.30	1.10|16.01	5.03
r110	100	3	31.70|177.70	0.47|10.57	8.43	31.60|176.93	0.50|7.33	34.43	31.00|178.67	0.91|12.67	4.83
r111	100	3	34.03|179.53	0.41|7.10	10.40	33.93|180.93	0.25|5.90	31.57	33.03|175.90	1.00|9.03	5.83
r112	100	3	35.03|170.53	0.32|7.90	9.40	34.57|170.07	0.50|7.66	25.50	33.53|170.37	0.82|13.43	5.47
r201	100	3	54.23|832.20	0.43|58.73	45.23	54.00|788.50	0.00|35.97	30.63	53.57|874.00	0.50|118.09	19.50
r202	100	5	95.97|1113.00	0.18|43.26	54.70	95.37|1039.63	0.49|87.16	58.80	95.23|1016.20	0.43|99.56	57.37
r203	100	6	92.97|1304.53	0.18|128.27	38.30	92.27|958.43	0.45|116.63	59.90	92.53|1092.70	0.51|173.22	58.80
r204	100	5	84.60|891.87	0.50|164.75	51.33	84.50|790.27	0.51|112.45	57.43	84.03|691.80	0.18|59.24	54.53
r205	100	3	66.00|854.50	0.00|32.30	26.77	66.00|819.77	0.00|37.42	39.33	65.60|820.00	0.50|60.83	32.33
r206	100	5	93.00|924.10	0.00|24.71	58.23	93.00|883.93	0.00|18.41	59.77	93.00|890.57	0.00|38.75	57.30
r207	100	3	64.00|667.50	0.00|17.81	40.83	64.00|622.97	0.00|16.91	43.80	64.00|652.83	0.00|30.55	36.60
r208	100	3	75.00|589.40	0.00|12.23	46.83	75.00|583.10	0.00|15.73	53.30	75.00|585.77	0.00|18.03	37.63
r209	100	4	78.00|1000.53	0.00|31.81	50.97	77.70|896.20	0.47|73.87	56.83	77.03|810.70	0.18|45.23	47.37
r210	100	8	98.00|845.73	0.00|21.12	60.00	98.00|832.63	0.00|19.08	55.73	98.00|823.43	0.00|25.25	58.53
r211	100	4	76.00|739.47	0.00|28.71	48.30	76.00|704.53	0.00|23.18	56.10	76.00|713.30	0.00|35.30	50.63
rc101	100	3	26.93|247.70	0.25|4.11	13.27	26.53|252.30	0.73|13.74	24.30	25.87|264.63	0.78|17.41	3.63
rc102	100	2	20.77|159.33	0.50|3.01	5.33	20.07|154.63	0.37|9.38	19.70	18.67|163.43	1.06|13.35	1.77
rc103	100	3	28.77|232.10	0.43|11.16	9.87	28.37|224.03	0.49|17.03	24.67	26.67|217.43	0.66|21.15	4.13
rc104	100	3	34.63|229.00	0.67|9.47	9.93	33.90|213.97	0.40|12.35	27.90	32.67|220.20	0.96|19.03	5.00
rc105	100	5	46.03|400.13	0.41|16.42	15.77	45.50|402.17	0.51|12.80	43.87	43.77|426.50	1.19|39.91	8.90
rc106	100	3	29.87|229.90	0.35|4.47	6.60	29.70|230.27	0.53|6.27	27.67	28.00|244.10	1.02|21.05	3.97
rc107	100	3	30.60|233.00	0.50|13.38	8.80	30.57|237.03	0.50|15.59	26.47	29.17|232.83	0.95|16.19	4.13
rc108	100	6	59.83|407.93	1.18|11.26	18.87	58.43|407.93	0.82|15.17	55.20	56.03|422.07	1.69|31.22	12.70
rc201	100	3	60.73|1188.07	0.45|62.83	37.93	60.53|1180.57	0.57|69.33	48.33	58.77|1133.43	0.77|151.55	24.37
rc202	100	3	53.00|782.93	0.00|22.69	45.53	53.00|808.53	0.00|35.30	38.03	52.70|811.90	0.47|85.58	20.47
rc203	100	3	62.00|775.77	0.00|25.70	54.67	62.00|786.43	0.00|32.20	51.70	61.13|747.33	0.35|69.45	37.03
rc204	100	3	70.00|804.17	0.00|33.89	27.40	69.93|765.00	0.25|65.75	41.10	69.77|763.70	0.43|88.45	41.87
rc205	100	3	59.00|1041.73	0.00|28.14	34.07	59.00|1022.47	0.00|46.31	34.00	58.43|976.40	0.50|88.17	26.70
rc206	100	3	68.80|1010.43	0.41|52.11	43.33	68.47|977.73	0.51|71.10	51.03	67.23|892.80	0.63|70.92	31.07
rc207	100	6	82.00|1049.20	0.00|31.64	56.17	82.00|971.83	0.00|43.02	60.00	81.97|980.50	0.18|52.97	56.37
rc208	100	7	99.00|804.83	0.00|28.29	52.80	99.00|819.60	0.00|33.38	58.10	99.00|828.57	0.00|47.25	58.83
Average			56.94|563.48	0.22|32.08	28.98	56.72|528.64	0.30|34.46	40.54	56.18|552.75	0.48|60.02	24.68

Table 4.9: Comparison on the best and worst performance between ILS2O, ILS and 2Phase ILS on instances from Solomon benchmark

			ILS2O				**ILS**				**2Phase**			
	$\|C\|$	$\|T\|$	bestj	bestd	worstj	worstd	bestj	bestd	worstj	worstd	bestj	bestd	worstj	worstd
c101	100	5	46	226	46	250	46	227	46	310	46	226	46	407
c102	100	2	22	84	22	99	22	84	21	117	22	84	21	177
c103	100	6	64	395	63	437	64	378	63	468	64	397	63	432
c104	100	3	29	106	29	138	29	106	29	151	29	106	29	186
c105	100	5	46	225	46	310	46	225	46	301	46	225	46	454
c106	100	5	47	219	46	260	47	223	46	305	47	219	46	323
c107	100	5	49	285	49	362	49	296	48	319	49	289	48	484
c108	100	6	57	325	56	428	57	327	56	414	57	322	55	649
c109	100	5	51	233	51	341	51	234	51	347	51	233	51	383
c201	100	3	67	1073	66	1217	67	1101	65	1036	66	1076	65	1724
c202	100	3	64	604	64	802	64	558	63	528	64	570	64	1959
c203	100	5	82	585	82	782	82	582	82	773	82	607	82	818
c204	100	6	97	816	97	993	97	784	96	696	97	813	97	1026
c205	100	3	65	557	65	817	65	526	64	542	65	523	65	845
c206	100	3	81	1687	80	1267	80	909	79	901	80	1008	78	847
c207	100	3	72	869	72	1127	72	652	71	652	72	679	71	699
c208	100	4	82	639	82	875	82	569	82	679	82	568	82	752
r101	100	3	23	183	22	176	23	183	22	170	23	227	21	222
r102	100	6	56	416	55	453	56	444	54	441	56	437	52	546
r103	100	5	51	334	49	324	50	317	49	338	50	307	47	296
r104	100	4	46	241	45	254	46	252	45	286	46	248	43	258
r105	100	5	44	328	43	351	44	328	43	354	44	342	41	414
r106	100	3	33	187	32	189	33	192	32	212	33	187	31	192
r107	100	4	43	244	41	242	43	252	42	277	42	235	40	254
r108	100	3	35	161	35	184	35	161	34	175	35	165	33	197
r109	100	3	32	189	31	204	32	189	31	209	32	198	28	239
r110	100	3	32	165	31	185	32	170	31	185	32	167	29	194
r111	100	3	35	186	33	192	34	166	33	179	34	170	30	195
r112	100	3	36	180	34	168	35	161	34	184	35	165	31	218
r201	100	3	55	878	54	862	54	715	54	862	54	793	53	883
r202	100	5	96	1059	95	966	96	1077	95	1121	96	1090	95	1051
r203	100	6	93	1165	92	862	93	1068	92	967	93	1161	92	984
r204	100	5	85	955	84	723	85	828	84	733	85	954	84	775
r205	100	3	66	799	66	936	66	758	66	913	66	793	65	824
r206	100	5	93	844	93	962	93	854	93	916	93	843	93	999
r207	100	3	64	611	64	698	64	588	64	667	64	587	64	721
r208	100	3	75	565	75	610	75	555	75	618	75	552	75	639
r209	100	4	78	931	78	1057	78	857	77	818	78	921	77	875
r210	100	8	98	802	98	894	98	786	98	871	98	778	98	881
r211	100	4	76	682	76	785	76	669	76	771	76	650	76	790
rc101	100	3	27	243	26	244	27	248	24	300	27	252	24	309
rc102	100	2	21	160	19	160	21	160	19	140	21	160	17	195
rc103	100	3	29	226	28	247	29	235	28	233	28	220	25	219
rc104	100	3	35	219	33	230	35	249	33	231	34	206	31	217
rc105	100	5	47	411	45	409	46	391	45	424	46	427	41	412
rc106	100	3	30	228	29	237	30	228	28	231	30	232	26	300
rc107	100	3	31	227	30	252	31	228	30	232	31	247	27	218
rc108	100	6	61	407	58	430	60	416	57	404	59	425	51	420
rc201	100	3	61	1135	60	1201	61	1114	59	1100	60	1094	57	905
rc202	100	3	53	738	53	823	53	755	53	891	53	756	52	826
rc203	100	3	62	728	62	826	62	722	62	854	62	801	61	787
rc204	100	3	70	741	70	870	70	693	69	625	70	683	69	700
rc205	100	3	59	949	59	1095	59	956	59	1133	59	967	58	1044
rc206	100	3	69	938	68	981	69	949	68	1053	68	893	66	837
rc207	100	6	82	981	82	1104	82	867	82	1065	82	877	81	864
rc208	100	7	99	747	99	859	99	747	99	887	99	723	99	951
Average			57.18	537.70	56.48	584.82	57.05	505.52	56.20	546.59	56.93	515.68	55.21	607.43

Table 4.10: The performance ILS-NR on instances from Solomon benchmark

			Average		Std V		Time(s)	Best		Worst	
	$\|C\|$	$\|T\|$	O1	O2	O1	O2		O1	O2	O1	O2
c101	100	5	46.00	409.67	0.00	40.49	5.53	46	336	46	473
c102	100	2	21.40	156.97	0.50	42.39	1.73	22	101	21	213
c103	100	6	62.80	590.97	0.55	31.20	11.20	64	520	62	641
c104	100	3	29.00	250.17	0.00	29.83	2.23	29	169	29	299
c105	100	5	45.87	429.43	0.35	47.33	6.07	46	362	45	543
c106	100	5	45.93	480.57	0.37	47.10	6.37	47	487	45	614
c107	100	5	48.20	484.33	0.41	52.46	6.50	49	364	48	567
c108	100	6	55.70	600.67	0.47	38.71	7.70	56	531	55	682
c109	100	5	50.80	462.03	0.41	29.08	7.03	51	415	50	538
c201	100	3	65.80	1917.20	0.41	176.53	10.47	66	1571	65	2207
c202	100	3	64.00	1932.37	0.00	133.11	8.93	64	1668	64	2208
c203	100	5	82.00	2703.93	0.00	192.28	9.53	82	2292	82	3042
c204	100	6	97.00	3087.17	0.00	249.11	6.80	97	2524	97	3593
c205	100	3	65.00	1898.33	0.00	185.67	8.80	65	1490	65	2240
c206	100	3	79.97	1808.23	0.18	109.14	16.73	80	1520	79	1769
c207	100	3	72.00	1975.90	0.00	133.46	9.10	72	1753	72	2277
c208	100	4	82.00	2508.27	0.00	174.61	9.57	82	2022	82	2827
r101	100	3	22.07	234.43	0.45	18.34	1.87	23	209	21	258
r102	100	6	53.23	504.97	0.86	14.61	5.63	55	479	52	523
r103	100	5	48.17	372.10	0.70	10.87	4.93	49	347	46	389
r104	100	4	44.00	294.23	0.83	12.65	4.20	45	278	42	317
r105	100	5	42.60	405.00	0.67	11.52	4.47	44	386	41	419
r106	100	3	31.40	215.07	0.62	9.38	2.67	32	204	30	232
r107	100	4	40.60	296.17	0.93	10.05	4.37	42	277	38	315
r108	100	3	33.37	211.00	0.89	8.69	2.40	35	193	31	226
r109	100	3	30.13	216.57	0.90	11.59	2.20	32	198	28	239
r110	100	3	30.63	206.77	1.00	12.20	2.17	32	184	28	229
r111	100	3	32.67	208.17	1.21	10.14	2.40	34	193	30	228
r112	100	3	33.30	201.00	0.99	10.35	2.53	35	180	31	223
r201	100	3	53.67	1278.63	0.48	85.60	7.80	54	1107	53	1355
r202	100	5	95.53	2184.87	0.51	168.29	14.73	96	1924	95	2412
r203	100	6	92.73	2439.13	0.45	133.31	14.37	93	2268	92	2486
r204	100	5	84.50	2231.17	0.51	187.00	7.93	85	2068	84	2416
r205	100	3	66.00	1411.27	0.00	58.73	9.47	66	1226	66	1490
r206	100	5	93.00	2273.40	0.00	102.38	11.13	93	2053	93	2440
r207	100	3	64.00	1502.47	0.00	67.29	6.93	64	1366	64	1623
r208	100	3	75.00	1564.43	0.00	63.44	6.63	75	1355	75	1679
r209	100	4	77.80	1807.63	0.41	113.91	12.80	78	1656	77	1700
r210	100	8	98.00	1875.83	0.00	285.64	6.87	98	1271	98	2332
r211	100	4	76.00	1747.50	0.00	135.26	8.10	76	1524	76	2018
rc101	100	3	25.67	275.33	0.66	12.17	1.93	27	255	24	284
rc102	100	2	18.57	182.47	1.04	12.47	0.87	21	161	17	200
rc103	100	3	26.67	271.13	0.71	14.82	1.63	28	265	25	294
rc104	100	3	33.00	264.80	1.08	12.84	2.17	35	243	31	297
rc105	100	5	43.47	465.63	1.41	20.09	4.23	46	455	41	507
rc106	100	3	27.57	270.03	0.90	16.87	1.80	29	248	26	303
rc107	100	3	29.37	264.27	0.96	14.97	1.83	31	251	27	301
rc108	100	6	55.17	478.17	2.25	22.79	5.17	59	442	50	518
rc201	100	3	59.00	1426.93	0.74	49.62	9.50	60	1339	58	1477
rc202	100	3	53.00	1383.07	0.00	68.37	6.60	53	1241	53	1490
rc203	100	3	61.23	1390.70	0.43	43.52	9.07	62	1375	61	1441
rc204	100	3	70.00	1457.83	0.00	41.60	7.37	70	1344	70	1533
rc205	100	3	58.40	1492.63	0.50	75.83	7.73	59	1314	58	1630
rc206	100	3	67.77	1359.70	0.77	59.11	11.30	69	1327	67	1468
rc207	100	6	82.00	2433.47	0.00	114.96	12.80	82	2149	82	2669
rc208	100	7	99.00	2410.10	0.00	267.27	6.73	99	1892	99	2947
Average			56.10	1092.93	0.49	73.23	6.64	56.86	953.07	55.13	1207.88

Table 4.11: Comparison of performance between ILS2O, ILS and 2Phase ILS on instances from Solomon Ver2 benchmark

			ILS2O			ILS			2Phase		
	$\|C\|$	$\|T\|$	Average	StdV	sec_a	Average	StdV	sec_a	Average	StdV	sec_a
c102	100	2	20.00|204.67	0.00|2.04	6.93	20.00|205.57	0.00|2.94	10.27	19.50| 223.37	0.51|24.69	3.20
c103	100	6	49.00|510.60	0.00|8.59	24.37	49.00|512.97	0.00|9.61	54.67	49.00| 514.43	0.00|10.31	22.77
c104	100	3	29.00|159.23	0.00|4.75	11.70	29.00|170.33	0.00|9.76	15.03	29.00| 161.73	0.00|14.96	7.50
c105	100	5	43.00|480.03	0.00|12.94	23.40	42.87|488.83	0.35|23.64	40.07	42.53| 564.07	0.51|84.23	13.40
c106	100	5	44.03|437.83	0.18|41.38	24.90	44.00|444.53	0.00|21.54	51.33	43.83| 483.50	0.38|94.45	18.63
c107	100	5	44.00|536.77	0.00|15.66	22.37	44.00|554.33	0.00|16.50	44.77	44.00| 619.70	0.00|90.20	16.87
c108	100	6	47.77|443.63	0.43|22.23	28.03	47.80|443.77	0.41|27.94	38.17	47.13| 471.67	0.43|55.77	23.73
c109	100	5	42.00|486.07	0.00|11.10	18.50	42.00|501.43	0.00|27.90	54.87	41.80| 505.60	0.41|52.29	16.43
c201	100	3	57.07|1360.93	0.25|86.80	51.27	57.37|1426.40	0.49|135.83	45.67	56.80| 1556.07	0.48|225.48	25.33
c202	100	3	56.93|1232.80	0.25|67.42	42.17	56.60|1200.63	0.50|119.71	47.37	56.10| 1333.33	0.31|356.68	23.47
c203	100	5	74.00|777.37	0.00|28.61	56.23	74.00|820.53	0.00|40.69	60.00	74.00| 849.57	0.00|50.86	51.57
c204	100	6	96.00|1086.30	0.00|49.32	59.73	96.00|1023.90	0.00|58.52	60.00	96.00| 1067.57	0.00|66.23	60.00
c205	100	3	60.00|1229.90	0.00|45.44	38.73	60.00|1164.83	0.00|39.66	37.73	60.00| 1185.83	0.00|51.96	41.33
c206	100	3	71.83|1415.53	0.38|117.48	57.93	72.00|1405.03	0.00|22.42	57.67	70.90| 1227.73	0.31|60.49	51.70
c207	100	3	68.00|1128.00	0.00|44.37	48.50	68.00|1062.83	0.00|35.87	47.47	67.90| 1115.17	0.40|79.59	49.03
c208	100	4	81.00|1195.97	0.00|65.29	58.43	81.00|1039.60	0.00|41.71	59.60	80.37| 945.07	0.49|146.60	59.70
r101	100	3	21.90|199.77	0.31|1.01	10.63	21.30|187.63	0.47|9.59	31.87	21.07| 231.63	0.45|23.68	4.50
r102	100	6	54.63|452.10	0.49|12.21	37.00	53.50|451.37	0.68|18.08	60.00	51.97| 462.97	0.76|45.93	22.80
r103	100	5	41.97|376.23	0.18|13.67	24.43	42.03|375.73	0.18|14.30	50.03	40.83| 398.30	0.83|38.53	11.57
r104	100	4	44.07|265.27	0.58|10.48	24.73	43.73|265.33	0.45|6.39	55.13	43.23| 260.67	0.77|12.23	16.47
r105	100	5	26.00|249.77	0.00|2.16	10.20	26.00|254.13	0.00|4.07	23.67	25.97| 309.03	0.18|42.80	5.23
r106	100	3	29.87|195.90	0.35|4.94	14.67	29.80|197.73	0.41|5.98	30.37	29.40| 205.40	0.50|15.68	8.47
r107	100	4	35.00|284.50	0.00|2.54	17.83	35.00|290.07	0.00|5.97	31.63	34.20| 299.77	0.89|23.46	9.73
r108	100	3	31.97|188.10	0.18|4.84	16.73	32.00|189.90	0.00|4.56	34.37	31.43| 194.97	0.57|12.60	8.67
r109	100	3	24.00|196.67	0.00|6.48	5.77	24.00|197.83	0.00|9.20	23.87	23.50| 236.63	0.73|22.26	3.73
r110	100	3	29.67|191.30	0.48|4.36	23.07	29.33|199.50	0.48|8.86	34.17	28.93| 196.33	0.74|13.52	8.97
r111	100	3	28.77|218.77	0.43|9.89	7.70	28.93|220.47	0.25|5.35	22.50	27.73| 216.23	0.83|16.97	6.20
r112	100	3	33.00|180.23	0.64|9.16	15.60	32.80|184.70	0.61|11.88	35.23	31.90| 184.87	0.61|15.13	8.23
r201	100	3	47.00|828.00	0.00|31.90	46.23	46.57|781.63	0.50|79.40	39.07	46.17| 789.37	0.38|127.64	22.00
r202	100	5	70.00|1023.80	0.00|25.46	58.03	69.73|1012.37	0.45|74.39	60.00	69.17| 926.13	0.38|81.88	52.20
r203	100	6	89.00|1037.13	0.00|27.68	60.00	89.00|1053.00	0.00|42.59	60.00	89.00| 1130.20	0.00|48.43	60.00
r204	100	5	75.00|806.50	0.00|15.29	54.33	74.97|782.00	0.18|31.35	55.47	74.77| 1065.53	0.43|458.39	58.23
r205	100	3	45.77|735.40	0.43|36.99	14.23	45.93|744.87	0.25|19.88	21.13	44.63| 673.07	0.49|95.36	17.17
r206	100	5	78.00|970.13	0.00|20.99	55.23	77.93|955.10	0.25|44.15	59.37	78.00| 990.03	0.00|44.53	60.00
r207	100	3	60.00|851.03	0.00|11.34	35.53	60.00|847.70	0.00|22.11	52.13	59.90| 845.80	0.31|38.92	48.20
r209	100	4	71.00|910.83	0.00|25.53	40.00	71.00|876.90	0.00|23.53	60.00	70.90| 886.60	0.31|35.42	54.47
r210	100	8	86.00|1162.67	0.00|34.33	57.27	86.00|1025.17	0.00|21.97	59.97	86.00| 1045.73	0.00|29.04	59.87
r211	100	4	73.00|760.07	0.00|14.79	48.63	73.00|725.87	0.00|21.97	57.20	73.00| 723.20	0.00|29.15	51.80
rc101	100	3	23.90|269.10	0.31|4.04	11.83	23.90|268.07	0.55|9.48	28.60	23.17| 280.60	0.46|12.98	5.23
rc102	100	2	17.93|172.73	0.25|1.60	4.77	16.90|165.23	0.71|11.67	20.23	16.27| 176.93	1.08|19.39	4.57
rc103	100	3	25.47|243.90	0.51|17.60	17.17	25.30|236.27	0.47|17.89	34.80	24.60| 253.27	0.72|22.49	6.23
rc105	100	5	29.97|387.47	0.18|7.77	12.67	29.83|393.03	0.38|5.30	26.17	29.13| 429.47	0.82|43.26	6.93
rc106	100	3	26.03|243.97	0.18|6.20	9.73	26.23|247.87	0.50|12.02	30.93	25.00| 248.97	0.87|20.10	5.93
rc107	100	3	26.87|256.50	0.35|11.05	11.33	26.90|258.23	0.31|8.86	29.07	26.47| 258.03	0.51|19.92	6.70
rc108	100	6	51.87|470.30	0.51|20.93	30.43	51.60|458.53	0.62|20.43	55.97	50.07| 470.20	1.08|37.27	18.83
rc203	100	3	54.00|849.53	0.00|25.40	58.03	54.00|862.63	0.00|25.01	57.73	53.60| 860.40	0.56|68.16	44.80
rc204	100	3	52.93|745.73	0.25|24.15	23.23	52.90|761.57	0.31|34.77	34.43	51.43| 629.50	0.73|55.31	15.87
rc205	100	3	55.00|1128.83	0.00|25.12	45.33	55.00|1124.67	0.00|30.56	41.03	54.53| 1080.30	0.51|95.17	39.27
rc206	100	3	48.50|910.60	0.51|59.22	25.43	48.47|915.50	0.51|69.19	42.07	46.80| 841.87	0.55|109.95	16.27
rc207	100	6	73.00|1156.57	0.00|32.42	57.47	73.00|1137.37	0.00|34.98	59.20	72.57| 1122.90	0.86|91.97	55.37
rc208	100	7	87.00|924.00	0.00|25.00	54.13	87.00|940.90	0.00|25.51	60.00	87.00| 932.53	0.00|30.05	60.00
Average			50.03|637.82	0.17|23.72	31.62	49.95|628.52	0.22|27.95	43.18	49.44| 640.82	0.45|65.93	26.85

Table 4.12: Comparison on the best and worst performance between ILS2O, ILS and 2Phase ILS on instances from Solomon Ver2 benchmark

			ILS2O				**ILS**				**2Phase**			
	$\|C\|$	$\|T\|$	bestj	bestd	worstj	worstd	bestj	bestd	worstj	worstd	bestj	bestd	worstj	worstd
c102	100	2	20	204	20	214	20	204	20	218	20	204	19	290
c103	100	6	49	503	49	529	49	505	49	539	49	503	49	537
c104	100	3	29	149	29	176	29	153	29	198	29	149	29	212
c105	100	5	43	462	43	513	43	473	42	491	43	468	42	779
c106	100	5	45	595	44	473	44	414	44	496	44	374	43	665
c107	100	5	44	516	44	579	44	516	44	585	44	547	44	824
c108	100	6	48	434	47	488	48	434	47	460	48	434	46	443
c109	100	5	42	475	42	513	42	475	42	590	42	475	41	724
c201	100	3	58	1580	57	1491	58	1573	57	1383	58	1634	56	1941
c202	100	3	57	1155	56	1055	57	1177	56	1123	57	1831	56	1996
c203	100	5	74	722	74	835	74	747	74	898	74	758	74	1006
c204	100	6	96	993	96	1182	96	910	96	1107	96	944	96	1201
c205	100	3	60	1112	60	1326	60	1061	60	1226	60	1075	60	1291
c206	100	3	72	1383	71	1299	72	1370	72	1467	71	1183	70	1105
c207	100	3	68	990	68	1206	68	980	68	1131	68	1049	66	861
c208	100	4	81	1071	81	1288	81	946	81	1122	81	1041	80	887
r101	100	3	22	200	21	197	22	200	21	193	22	237	20	209
r102	100	6	55	438	54	485	55	456	52	464	53	406	50	394
r103	100	5	42	361	41	350	43	416	42	411	42	371	38	464
r104	100	4	45	263	43	284	44	257	43	263	44	250	42	304
r105	100	5	26	248	26	253	26	248	26	259	26	259	25	346
r106	100	3	30	191	29	206	30	191	29	208	30	191	29	246
r107	100	4	35	279	35	290	35	284	35	309	35	287	32	332
r108	100	3	32	179	31	196	32	182	32	203	32	183	30	222
r109	100	3	24	188	24	204	24	188	24	218	24	215	22	281
r110	100	3	30	188	29	201	30	200	29	213	30	189	27	224
r111	100	3	29	207	28	230	29	204	28	224	29	216	26	209
r112	100	3	34	191	32	193	34	191	32	214	33	179	31	230
r201	100	3	47	773	47	891	47	795	46	730	47	983	46	990
r202	100	5	70	966	70	1072	70	985	69	963	70	980	69	940
r203	100	6	89	991	89	1092	89	990	89	1147	89	1036	89	1214
r204	100	5	75	771	75	833	75	730	74	729	75	794	74	731
r205	100	3	46	735	45	697	46	735	45	695	45	657	44	646
r206	100	5	78	926	78	1003	78	900	77	875	78	890	78	1094
r207	100	3	60	825	60	867	60	809	60	931	60	803	59	764
r209	100	4	71	861	71	968	71	825	71	917	71	831	70	836
r210	100	8	86	1089	86	1253	86	980	86	1087	86	976	86	1096
r211	100	4	73	710	73	782	73	656	73	760	73	669	73	783
rc101	100	3	24	264	23	276	24	264	21	230	24	268	22	269
rc102	100	2	18	173	17	167	18	173	16	149	18	173	14	217
rc103	100	3	26	249	25	263	26	250	25	235	26	269	23	247
rc105	100	5	30	382	29	415	30	384	29	408	30	383	28	521
rc106	100	3	27	266	26	247	27	266	25	253	27	266	24	282
rc107	100	3	27	250	26	237	27	250	26	244	27	250	26	313
rc108	100	6	53	486	51	522	53	486	51	467	52	477	48	488
rc203	100	3	54	799	54	890	54	811	54	899	54	841	52	676
rc204	100	3	53	732	52	701	53	732	52	700	53	757	50	584
rc205	100	3	55	1091	55	1188	55	1085	55	1210	55	1121	54	1044
rc206	100	3	49	948	48	947	49	954	48	867	48	894	46	808
rc207	100	6	73	1087	73	1211	73	1074	73	1187	73	1078	69	828
rc208	100	7	87	883	87	1013	87	889	87	991	87	874	87	1008
Average			50.22	618.31	49.69	662.57	50.20	607.41	49.53	644.84	50.04	625.92	48.51	678.47

Table 4.13: The performance ILS-NR on instances from Solomon benchmark Ver2

			Average		**Std V**		**Time(s)**	**Best**		**Worst**	
	$\|C\|$	$\|T\|$	O1	O2	O1	O2		O1	O2	O1	O2
c102	100	2	19.40	277.57	0.50	44.37	2.00	20	210	19	362
c103	100	6	49.00	955.70	0.00	114.99	5.30	49	777	49	1173
c104	100	3	29.00	269.43	0.00	18.48	3.90	29	232	29	300
c105	100	5	42.43	655.10	0.50	61.88	8.97	43	567	42	764
c106	100	5	43.63	632.03	0.49	56.59	11.20	44	559	43	768
c107	100	5	44.00	744.60	0.00	32.73	7.33	44	676	44	803
c108	100	6	47.07	737.47	0.37	55.64	8.73	48	672	46	837
c109	100	5	41.83	756.97	0.38	59.77	5.63	42	609	41	879
c201	100	3	56.80	1697.23	0.41	105.94	12.80	57	1497	56	1804
c202	100	3	56.10	1870.87	0.31	152.20	9.60	57	1699	56	2151
c203	100	5	74.00	2328.30	0.00	183.76	12.23	74	1972	74	2667
c204	100	6	96.00	2939.37	0.00	275.76	13.23	96	2380	96	3582
c205	100	3	60.00	2025.77	0.00	151.48	9.83	60	1783	60	2360
c206	100	3	71.53	1928.77	0.51	126.52	20.53	72	1638	71	2109
c207	100	3	68.00	2093.13	0.00	133.22	14.13	68	1815	68	2325
c208	100	4	80.80	2551.50	0.41	174.09	19.57	81	2263	80	2974
r101	100	3	21.03	245.00	0.49	11.62	3.27	22	234	20	253
r102	100	6	52.10	515.60	0.84	16.35	11.43	54	492	51	552
r103	100	5	41.03	421.73	0.96	14.79	5.90	42	397	38	455
r104	100	4	42.27	310.77	1.20	14.16	7.23	45	287	40	322
r105	100	5	25.80	336.67	0.48	27.23	3.13	26	284	24	396
r106	100	3	29.17	235.23	0.59	11.93	4.03	30	202	28	255
r107	100	4	34.03	336.07	0.81	16.39	4.67	35	301	32	361
r108	100	3	31.10	223.73	0.80	11.49	4.33	32	205	30	250
r109	100	3	23.50	253.10	0.68	10.77	2.03	24	233	22	268
r110	100	3	28.87	224.13	0.63	9.49	3.83	30	216	27	244
r111	100	3	27.53	248.17	0.57	8.67	3.03	28	219	26	261
r112	100	3	31.70	212.27	0.65	10.06	4.27	33	206	31	229
r201	100	3	46.60	1056.90	0.50	58.64	9.13	47	963	46	1126
r202	100	5	69.63	1622.07	0.49	108.08	13.63	70	1542	69	1768
r203	100	6	89.00	2318.73	0.00	119.23	18.40	89	2036	89	2610
r204	100	5	74.77	1842.30	0.43	140.45	11.90	75	1601	74	2003
r205	100	3	44.70	1040.80	0.53	98.92	3.60	46	1161	44	1190
r206	100	5	78.00	1870.53	0.00	103.78	10.50	78	1687	78	2073
r207	100	3	60.00	1484.53	0.00	66.92	8.97	60	1349	60	1606
r209	100	4	71.00	1693.87	0.00	85.78	13.03	71	1523	71	1865
r210	100	8	86.00	2253.60	0.00	132.70	10.77	86	1978	86	2503
r211	100	4	73.00	1689.70	0.00	102.20	13.63	73	1474	73	1906
rc101	100	3	23.07	290.00	0.83	11.47	3.03	24	275	21	312
rc102	100	2	16.50	195.03	1.07	11.26	1.67	18	182	14	216
rc103	100	3	24.77	289.33	0.68	10.97	2.80	26	295	23	321
rc105	100	5	28.97	479.50	0.85	30.20	3.17	30	431	27	470
rc106	100	3	24.70	280.00	1.06	14.41	2.70	27	270	23	301
rc107	100	3	26.40	291.27	0.56	10.66	3.07	27	280	25	280
rc108	100	6	49.83	529.20	1.26	19.52	7.67	51	495	46	528
rc203	100	3	53.97	1415.57	0.18	66.45	9.80	54	1296	53	1463
rc204	100	3	52.10	1413.40	0.40	99.28	6.33	53	1432	51	1403
rc205	100	3	54.83	1535.90	0.38	74.26	10.50	55	1347	54	1611
rc206	100	3	47.10	1263.20	0.80	58.94	4.60	48	1206	46	1316
rc207	100	6	73.00	2042.53	0.00	129.64	16.93	73	1785	73	2341
rc208	100	7	87.00	2389.90	0.00	157.99	11.63	87	2003	87	2703
			49.46	1084.59	0.44	71.02	8.23	50.06	965.41	48.55	1208.22

search that alternatives between the two objectives. The computational experiments were carried out on three sets of benchmarks. One is provided by the industry partner and the other two are derived from the standard Solomon benchmark for vehicle routing problems. The results of the computational experiments demonstrate the good performance of the proposed algorithm in terms of computational time, solution quality and stability.

5 Simultaneous Pickup and Delivery Problem with Preloading under Uncertainty

Abstract

This chapter studies a Simultaneous Pickup and Delivery Problem with Preloading under Uncertainty. Two groups of vehicles are considered. The assignment of customers to the vehicles of the first group (preloading) is done when only a subset of all customers is known, whereas the assignment of customers to the vehicles of the second group is done when all remaining customers are known. This problem is formulated as a two-stage stochastic program and solved by the sample average approximation approach. An optimisation algorithm under the Lagrangian ILS framework is described for the sample average approximation approach. The results of the computational experiment on a set of instances derived from historical data provided by the industry partner have shown the proposed algorithm has good performance in terms of computational time and solution quality.

5.1 Introduction

The problem studied in this chapter is suggested by the industry partner who provides the next-day delivery service in the retail sector, i.e., a customer's demand is fulfilled on the following day Yaman et al. (2012), Wollenburg et al. (2018). In this problem, some vehicles must return to the depot after serving all the allocated customers and load some demands that need to be fulfilled on the next day. This practice is called preloading. These vehicles return to the designated locations (depots owned by the drivers) after the preloading and begin their routes for the next day directly from the designated locations. The purpose of preloading is to deal with the limited capacity at the depot. When a customer makes a purchase, the depot will receive the customer's demand and wait for a vehicle to fulfil the delivery. If the depot runs out the space when a customer's demand arrives that means the depot cannot store the demand before its delivery commences, then this customer must be outsourced which is expensive. The preloading loosens the heavy burden at the depot by moving certain demands to the vehicles. Thereby, reducing the total outsourcing cost.

Although preloading can be beneficial by accommodating more customers, there are certain limitations. For example, if the total customer's demands are less than the capacity of the depot, then instead of preloading, it is better to make the decisions for the allocation of customers to vehicles when all customer's demands are revealed. The decisions for the allocation of customers for preloading are also complex. One of the reasons is due to the uncertainty of customers' demands after the preloading. Another reason is because of the drivers' preferences. After serving the last customer, the driver may end up at a location far away from the depot and is not willing to come back to the depot. Therefore, it is difficult to determine the drivers who are suitable to perform the preloading. In addition, the finishing times may vary for drivers, for example, a driver may finish early and require a wait for preloading to commence. Therefore, it is also difficult to determine the appropriate starting time for preloading.

In this chapter, the preloading problem is simplified to investigate the potential to apply the stochastic programming approach. The simplified version assumes that the selection of the vehicles for preloading is not part of the model. In addition, the time for vehicles loading at the depot as well as the time for vehicles preloading at the depot are specified by a roster. In the latter case, the roster ensures a sufficient number of vehicles available when preloading commences. The problem also considers features studied in Chapter 4, including simultaneous pickup and delivery; a heterogeneous fleet of vehicles; time windows for customers; compatibility between customers and vehicles; open routes; restriction on shift length; vehicles' capacity (for weight and volume). This simplified version is called Simultaneous Pickup and Delivery Problem with Preloading under Uncertainty (SPDPP) and is modelled as a 2-stage stochastic program.

The decision stages are illustrated in Figure 5.1. In Stage 1, some vehicles are required to do preloading at the depot at a certain time of the day. As mentioned above, it is assumed that the selection of the vehicles for preloading is not part of the model. Decisions

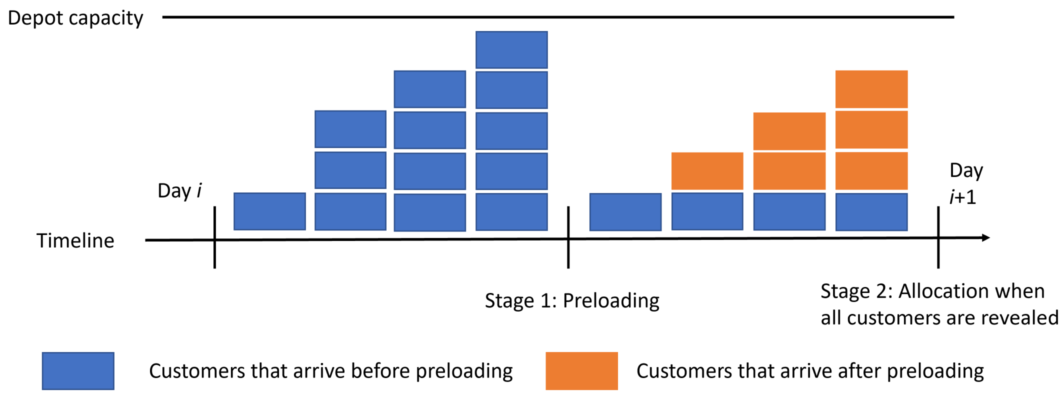


Figure 5.1: Illustration of the decision stages in the SPDPP

need to be made on the allocation of customers at the depot to the preloading vehicles although not all customers are known. In Stage 2, when all customers are revealed, the vehicles that are not involved in preloading need to come to the depot and load the delivery items for the remaining customers (typically at the beginning of the next day). The goal for the problem is to maximise the expected total number of allocated customers.

The SPDPP is formulated as a 2-stage stochastic program and is tackled by the sample average approximation (SAA) approach. The SAA approach generates a set of scenarios and converts the 2-stage stochastic model into a deterministic mixed integer programming (MIP) model. The SAA solution converges asymptotically to the optimal solution of the 2-stage model when the sample size approaches infinity Kleywegt et al. (2002). Since the second stage of the SPDPP is a vehicle routing problem, the size of the MIP model increases quadratically with the number of scenarios.

In this chapter, the optimisation procedure developed for the SAA approach is another implementation of the iterated local search under the Lagrangian ILS framework and is referred to as the ILS-SAA. Since the allocation for the first stage in the SAA approach must be the same for all the scenarios in the second stage which is called the non-anticipativity constraints, the ILS-SAA made several modifications compared with the Lagrangian ILS presented in Chapter 3. These modifications include how the initial feasible solutions are constructed, how local search is performed with a new local search operator, as well as how the perturbation mechanism works.

(a) The procedure to construct the initial feasible solution is 2-stage which is different compared with the corresponding procedure in the Lagrangian ILS in Chapter 3. It first constructs the first stage solution for the SAA approach which is the route for preloading vehicles. Then, taking the unallocated customers from the first-stage solution together with the customers who appeared after the preloading in each scenario, the second-stage solution is constructed which comprises the routes for other vehicles in each scenario.

(b) The perturbation mechanism is also different in comparison with the perturbation mechanism in the Lagrangian ILS. In the ILS-SAA, the perturbation changes the first-stage solution first which involves an unallocated preloading customer with respect to the first-stage solution. If this customer is allocated in some scenarios of the second-stage solution, then, in order to respect the non-anticipativity constraints, this customer is removed from the second-stage solution. After the perturbation of the first-stage solution. The mechanism changes the routes for each scenario in the second-stage solution.

(c) Since the neighbourhood operators considered in the Lagrangian ILS in Chapter 3 do not handle the non-anticipativity constraints, the local search in the ILS-SAA considers a new operator that transforms the input solution taking into account both the first-stage and second-stage solution simultaneously and at the same time respecting the non-anticipativity constraints. This operator allows the ILS-SAA to explore suitable solutions for the SAA approach.

In addition, the number of Lagrange multipliers used in the ILS-SAA is larger than the number of Lagrange multipliers used in the Lagrangian ILS in Chapter 3. In Lagrangian ILS, a single Lagrange multiplier is used for the total violation of certain constraints, for example, β in (3.40) is used for the total violation on maximum duration among all routes. In ILS-SAA, a single Lagrange multiplier for one type of constraint for both first-stage and second-stage solutions is inadequate. Therefore, the ILS-SAA introduces Lagrange multipliers for both the first-stage and second-stage solutions. For the second-stage solution, the ILS-SAA introduces Lagrange multipliers for each scenario in the second stage. That means a Lagrange multiplier is the penalty for the total violation of certain constraints on all routes in either the first-stage solution or a scenario in the second stage.

To evaluate the performance of the ILS-SAA, a set of instances derived from historical data provided by the industrial partner. The results of the computational experiments obtained from solving the MIP model of the SAA approach with a single scenario, 5 scenarios and 10 scenarios have demonstrated that the proposed ILS-SAA has good performance in terms of computational time and solution quality, especially for the instances with 10 scenarios.

The remaining part of the chapter is organised as follows. Section 5.2 presents the 2-stage stochastic programming formulation for the SPDPP and the formulation for the sample average approximation approach. Section 5.3 describes the iterated local search designed for this approach. Then, the results of the computational experiments are reported in Section 5.4. At last, Section 5.5 concludes the chapter.

5.2 Problem statement

The SPDPP is formulated below as a 2-stage stochastic program. Two sets of vehicles are considered, T_1 and T_2, and two sets of customers are considered, C_1 and C_2. The first set of vehicles T_1 can only serve customers constituting the set C_1. The Preloading involves the assignment of customers in C_1 to the vehicles in T_1 and the construction of routes for these vehicles where these decisions are made without any knowledge of C_2. Furthermore, each vehicle in T_1 has an associated depot (owned by the driver) and the corresponding route is constructed under the assumption that this vehicle can depart from its depot at any time. When customers are allocated to vehicles in T_2 and the corresponding routes are constructed under the condition that all vehicles depart from the depot (owned by the industry partner) and are loaded according to the roster. The assignment of customers to the vehicles in T_2 is made using all available information: the subset of C_1 comprised of all customers who are not allocated to the vehicles in T_1 as well as all customers in C_2.

Let $G = \{L, A\}$ be a directed graph where the set of vertices $L = \{0\} \cup C_1 \cup C_2 \cup D$, $C_1 = \{1, 2, ..., l_1\}$, $C_2 = \{l_1{+}1, l_1{+}2, ..., l_1{+}l_2\}$, and $D = \{l_1{+}l_2{+}1, l_1{+}l_2{+}2, ..., l_1{+}l_2{+}d\}$. The set of arcs $A = A_0 \cup A_C \cup A_D$ where $A_0 = \{(0, i) | i \in C_1 \cup C_2\}$, $A_C = \{(i, j) | i \neq j, \forall i, j \in C_1 \cup C_2\}$, $A_D = \{(i, j) | \forall i \in D, j \in C_1\}$. Vertex 0 represents the depot; vertices in C_1 represent the customers who can be assigned to vehicles in T_1; vertices in C_2 represent the customers who occur after the construction of routes for vehicles in T_1; vertices in D represent the depots owned by the subcontractors. Each arc $(i, j) \in A$ has an associated travel time $t_{i,j}$.

For each customer $i \in C_2$, $\tilde{\zeta}_i$ is a random variable which indicates whether customer i occurs or not (0:no, 1:yes). Let $\zeta^\omega = \{\zeta_i^\omega | i \in C_2\}$ be a particular realisation of the random vector $\tilde{\zeta} = \{\tilde{\zeta}_i | i \in C_2\}$. The delivery of customer $i \in C_1 \cup C_2$ is characterised by its weight w_i^d and volume v_i^d. The pickup of customer $i \in C_1 \cup C_2$ is also characterised by its weight w_i^p and volume v_i^p. Furthermore, for customer $i \in C_1 \cup C_2$, the associated time window $[a_i, b_i]$ indicates the earliest and latest time when the subcontractor can start the corresponding services, and let $p_i > 0$ be the service time required for the subcontractor to complete the service.

Each vehicle $i \in T_1 \cup T_2$ is characterised by its weight capacity W_i and volume capacity V_i. Each vehicle $i \in T_1$ departs from its own home location, whereas all vehicles $i \in T_2$ depart from the same depot. A vehicle is not required to return to the depot after serving its allocated customers. The subcontractor in vehicle $i \in T_1 \cup T_2$ finishes the shift after serving the last allocated customer. Due to the loading capacity of the depot, each vehicle $i \in T_2$ arrives at the depot at the specified starting time r_i with loading time δ_i. Furthermore, there exists a maximal duration Ψ_i on the shift time of the subcontractor in vehicle $i \in T_1 \cup T_2$, which is the length of the time interval between the time when the subcontractor starts loading at the depot and the time when subcontractor finishes the service of the last allocated customer.

Each customer $i \in C_1 \cup C_2$ can be allocated only once, but not all vehicles are capable to serve certain customers. Two types of vehicles are considered, i.e., $T' \subset T_1 \cup T_2$ and $T'' \subset T_1 \cup T_2$. The customers are also classified into two types $C_1' \subset C_1$ ($C_2' \subset C_2$) and $C_1'' \subset C_1$ ($C_2'' \subset C_2$). The vehicles in $T'' \cap T_1$ ($T'' \cap T_2$) can serve all customers in C_1 (C_2) whereas the vehicles in $T' \cap T_1$ ($T' \cap T_2$) can only serve customers in C_1' (C_2').

Let $\mathbb{E}_\omega(\cdot)$ denotes the mathematical expectation operator taken with respect to $\tilde{\zeta}$. The objective is to maximise the total expected number of allocated customer services while respecting all the constraints on subcontractors, vehicles, customers, depot, and non-anticipatity.

$$\max \sum_{i \in T_1} \sum_{j \in C_1} \eta_j^i + \mathbb{E}_\omega[Q(\rho, \omega)] \tag{5.1}$$

Table 5.1: Symbols for 2-stage formulation

Variables	
x^i_{jk}	If customer $j \in C_1$ is the immediate predecessor of customer $k \in C_1$ in the route of vehicle $i \in T_1$ (0:no, 1:yes).
$\widehat{x}^i_{jk}$	If customer $j \in C_1 \cup C_2$ is the immediate predecessor of customer $k \in C_1 \cup C_2$ in the route of vehicle $i \in T_2$ (0:no, 1:yes).
ρ_i	If customer $i \in C_1$ is not allocated to any vehicles in T_1 (0:no, 1:yes).
η^i_j	If customer $j \in C_1$ is allocated to vehicle $i \in T_1$ (0:no, 1:yes).
$\widehat{\eta}^i_j$	If customer $j \in C_1 \cup C_2$ is allocated to vehicle $i \in T_2$ (0:no, 1:yes).
γ^i_j	If customer $j \in C_1$ is the first customer to visit after vehicle $i \in T_1$ departing from the depot (0:no, 1:yes).
$\widehat{\gamma}^i_j$	If customer $j \in C_1 \cup C_2$ is the first customer to visit after vehicle $i \in T_2$ departing from the depot (0:no, 1:yes).
θ^i_j	If customer $j \in C_1$ is the last customer in the route of vehicle $i \in T_1$ (0:no, 1:yes).
$\widehat{\theta}^i_j$	If customer $j \in C_1 \cup C_2$ is the last customer in the route of vehicle $i \in T_2$ (0:no, 1:yes).
ψ^i_j	The time when subcontractor in vehicle $i \in T_1$ starts serving customer $j \in C_1$.
$\widehat{\psi}^i_j$	The time when subcontractor in vehicle $i \in T_2$ starts serving customer $j \in C_1 \cup C_2$.
y_j	The weight of the vehicle when leaving customer $j \in C_1 \cup C_2$.
z_j	the volume of the vehicle when leaving customer $j \in C_1 \cup C_2$.

subject to:

$$\sum_{i \in T_1} \eta^i_j + \rho_j = 1, \quad \forall j \in C_1 \tag{5.2}$$

$$\sum_{j \in C_1} \gamma^i_j \leq 1, \quad \forall i \in T_1 \tag{5.3}$$

$$\gamma^i_j + \sum_{k \in C_1} x^i_{k,j} = \eta^i_j, \quad \forall i \in T_1, \forall j \in C_1 \tag{5.4}$$

$$\theta^i_j + \sum_{k \in C_1} x^i_{j,k} = \eta^i_j, \quad \forall i \in T_1, \forall j \in C_1 \tag{5.5}$$

$$a_j \leq \psi^i_j, \quad \forall i \in T_1, \forall j \in C_1 \tag{5.6}$$

$$\psi^i_j \leq b_j, \quad \forall i \in T_1, \forall j \in C_1 \tag{5.7}$$

$$(r_i + t_{i,k})\gamma^i_k \leq \psi^i_k, \quad \forall i \in T_1, \forall k \in C_1 \tag{5.8}$$

$$\psi^i_j + (p_j + t_{j,k})x^i_{j,k} + (a_k - b_j)(1 - x^i_{j,k}) \leq \psi^i_k, \quad \forall i \in T_1, \forall j \in C_1, \forall k \in C_1, k \neq j \tag{5.9}$$

$$p_j + \psi_j^i - r_i - (p_j + b_j - r_i)(1-\theta_j^i) \le \Psi_i, \quad i \in T_1, \forall j \in C_1 \tag{5.10}$$

$$\sum_{k \in C_1} w_k^d \eta_k^i \le W_i, \quad \forall i \in T_1 \tag{5.11}$$

$$y_k \le W_i + (\max_{e \in T_1} W_e - W_i)(1-\eta_k^i), \quad \forall i \in T_1, k \in C_1 \tag{5.12}$$

$$\sum_{j \in C_1} w_j^d \eta_j^i - w_k^d + w_k^p - (\max_{e \in T_1} W_e - w_k^d + w_k^p)(1-\gamma_k^i) \le y_k, \quad \forall i \in T_1, k \in C_1 \tag{5.13}$$

$$y_j - w_k^d + w_k^p - (\max_{e \in T_1} W_e - w_k^d + w_k^p)(1 - x_{j,k}^i) \le y_k, \quad \forall i \in T_1, \forall j \in C_1, \forall k \in C_1, k \ne j \tag{5.14}$$

$$\sum_{k \in C_1} v_k^d \eta_k^i \le V_i, \quad \forall i \in T_1 \tag{5.15}$$

$$z_k \le V_i + (\max_{e \in T_1} V_e - V_i)(1-\eta_k^i), \quad \forall i \in T_1, k \in C_1 \tag{5.16}$$

$$\sum_{j \in C_1} v_j^d \eta_j^i - v_k^d + v_k^p - (\max_{e \in T_1} V_e - v_k^d + v_k^p)(1-\gamma_k^i) \le z_k, \quad \forall i \in T_1, k \in C_1 \tag{5.17}$$

$$z_j - v_k^d + v_k^p - (\max_{e \in T_1} V_e - v_k^d + v_k^p)(1 - x_{j,k}^i) \le z_k, \quad \forall i \in T_1, \forall j \in C_1, \forall k \in C_1, k \ne j \tag{5.18}$$

$$\sum_{i \in T' \cap T_1} \sum_{k \in C_1''} \eta_k^i = 0 \tag{5.19}$$

$$x_{j,k}^i \in \{0,1\}, \quad \forall j \in C_1, \forall k \in C_1, k \ne j, \forall i \in T_1 \tag{5.20}$$

$$\eta_j^i \in \{0,1\}, \quad \forall i \in T_1, \forall j \in C_1 \tag{5.21}$$

$$\gamma_j^i \in \{0,1\}, \quad \forall i \in T_1, \forall j \in C_1 \tag{5.22}$$

$$\theta_j^i \in \{0,1\}, \quad \forall i \in T_1, \forall j \in C_1 \tag{5.23}$$

$$\rho_j \in \{0,1\}, \quad \forall j \in C_1 \tag{5.24}$$

where for a particular scenario ω, $Q(\rho, \omega)$ is defined as

$$Q(\rho, \omega) = \max \sum_{i \in T_2} \sum_{j \in C_1 \cup C_2} \widehat{\eta}_j^i \tag{5.25}$$

subject to:

$$ \tag{5.26} $$

$$\sum_{i \in T_2} \widehat{\eta}_j^i \leq 1, \quad \forall j \in C_2 \tag{5.27}$$

$$\sum_{i \in T_2} \widehat{\eta}_j^i \leq \rho_j, \quad \forall j \in C_1 \tag{5.28}$$

$$\sum_{i \in T_2} \widehat{\eta}_j^i \leq \zeta_j^\omega, \quad \forall j \in C_2 \tag{5.29}$$

$$\sum_{j \in C_1 \cup C_2} \widehat{\gamma}_j^i \leq 1, \quad \forall i \in T_2 \tag{5.30}$$

$$\widehat{\gamma}_j^i + \sum_{k \in C_1 \cup C_2} \widehat{x}_{kj}^i = \widehat{\eta}_j^i, \quad \forall i \in T_2, j \in C_1 \cup C_2 \tag{5.31}$$

$$\widehat{\theta}_j^i + \sum_{k \in C_1 \cup C_2} \widehat{x}_{jk}^i = \widehat{\eta}_j^i, \quad \forall i \in T_2, j \in C_1 \cup C_2 \tag{5.32}$$

$$a_j \leq \widehat{\psi}_j^i, \quad \forall j \in C_1 \cup C_2, i \in T_2 \tag{5.33}$$

$$\widehat{\psi}_j^i \leq b_j, \quad \forall j \in C_1 \cup C_2, i \in T_2 \tag{5.34}$$

$$(r_i + \delta_i + t_{0,k})\widehat{\gamma}_k^i \leq \widehat{\psi}_k^i, \quad \forall i \in T_2, k \in C_1 \cup C_2 \tag{5.35}$$

$$\begin{aligned} &\widehat{\psi}_j^i + (p_j + t_{jk})\widehat{x}_{jk}^i + (a_k - b_j)(1 - \widehat{x}_{jk}^i) \leq \widehat{\psi}_k^i, \\ &\qquad \forall i \in T_2, j \in C_1 \cup C_2, k \in C_1 \cup C_2, k \neq j \end{aligned} \tag{5.36}$$

$$p_j + \widehat{\psi}_j^i - r_i - (p_j + b_j - r_i)(1 - \widehat{\theta}_j^i) \leq \widehat{\Psi}_i, \quad \forall j \in C_1 \cup C_2, i \in T_2 \tag{5.37}$$

$$\sum_{k \in C_1 \cup C_2} w_k^d \widehat{\eta}_k^i \leq W_i, \quad \forall i \in T_2 \tag{5.38}$$

$$y_k \leq W_i + (\max_{e \in T_2} W_e - W_i)(1 - \widehat{\eta}_k^i), \quad \forall i \in T_2, k \in C_1 \cup C_2 \tag{5.39}$$

$$\begin{aligned} &\sum_{j \in C_1 \cup C_2} w_j^d \widehat{\eta}_j^i - w_k^d + w_k^p - (\max_{e \in T_2} W_e - w_k^d + w_k^p)(1 - \widehat{\gamma}_k^i) \leq y_k, \\ &\qquad \forall i \in T_2, k \in C_1 \cup C_2 \end{aligned} \tag{5.40}$$

$$\begin{aligned} &y_j - w_k^d + w_k^p - (\max_{e \in T_2} W_e - w_k^d + w_k^p)(1 - \widehat{x}_{jk}^i) \leq y_k, \\ &\qquad \forall i \in T_2, \forall j \in C_1 \cup C_2, k \in C_1 \cup C_2, k \neq j \end{aligned} \tag{5.41}$$

$$\sum_{k \in C_1 \cup C_2} v_k^d \widehat{\eta}_k^i \leq V_i, \quad \forall i \in T_2 \tag{5.42}$$

$$z_k \leq V_i + (\max_{e \in T_2} V_e - V_i)(1 - \widehat{\eta}_k^i), \quad \forall i \in T_2, k \in C_1 \cup C_2 \tag{5.43}$$

$$\sum_{j \in C_1 \cup C_2} v_j^d \eta_j^i - v_k^d + v_k^p - (\max_{e \in T_2} V_e - v_k^d + v_k^p)(1 - \widehat{\gamma}_k^i) \leq z_k, \quad \forall i \in T_2, k \in C_1 \cup C_2 \tag{5.44}$$

$$z_j - v_k^d + v_k^p - (\max_{e \in T_2} V_e - v_k^d + v_k^p)(1 - \widehat{x}_{jk}^i) \leq z_k, \quad \forall i \in T_2, \forall j \in C_1 \cup C_2, k \in C_1 \cup C_2, k \neq j \tag{5.45}$$

$$\sum_{i \in T' \cap T_2} \sum_{k \in C_1'' \cup C_2''} \widehat{\eta}_k^i = 0 \tag{5.46}$$

$$\widehat{x}_{jk}^i \in \{0, 1\}, \quad \forall j \in C_1 \cup C_2, \forall k \in C_1 \cup C_2, k \neq j, \forall i \in T_2 \tag{5.47}$$

$$\widehat{\eta}_j^i \in \{0, 1\}, \quad \forall i \in T_2, \forall j \in C_1 \cup C_2 \tag{5.48}$$

$$\widehat{\gamma}_j^i \in \{0, 1\}, \quad \forall i \in T_2, \forall j \in C_1 \cup C_2 \tag{5.49}$$

$$\widehat{\theta}_j^i \in \{0, 1\}, \quad \forall i \in T_2, \forall j \in C_1 \cup C_2 \tag{5.50}$$

The objective function (5.1) maximises the expected total number of allocated customers. Constraints (5.2) ensure a customer in C_1 is allocated to at most one vehicle in T_1. Constraints (5.3) and (5.8) guarantee that a vehicle either stays at its depot or visits exactly one customer. Constraints (5.4) and (5.5) ensure that each customer must have an immediate successor from the same route except for the last customer. The time when a shift can commence, the travelling time between locations, and the time windows are stipulated by (5.8), (5.9) and (5.6) − (5.9) respectively. The shift length, weight capacity, volume capacity, and compatibility between customers in C_1 and vehicles in T_1 are enforced by (5.10), (5.11)-(5.14), (5.15)-(5.18), and (5.19) respectively.

By virtue of constraints (5.28), the objective function (5.25) for $Q(\rho, \omega)$ maximises the total number of allocated customers including customers from C_2 and customers from C_1 who are not allocated in preloading. The constraints (5.27) and (5.29) ensure a vehicle can only serve customers that occurred in scenario ω at most once. Constraints (5.35) guarantee that a vehicle either stays at the depot or spends sufficient time to load and then visits the first customer in its route. The constraints (5.30) − (5.34), (5.37) − (5.45) have the same purposes as the constraints (5.3) − (5.7), (5.10) − (5.18). The differences between these constraints are that C_1 is replaced by $C_1 \cup C_2$ and T_1 is replaced by T_2.

Furthermore, constraints (5.6), 5.7, and (5.9) eliminate subtours by virtue of $p_i > 0$. Similarly, (5.33), (5.34), and (5.36) eliminate subtours by virtue of $p_i > 0$. It should be noted that the second stage of the problem is exactly the same as the problem (4.1)−(4.23) considered in Chapter 4.

It should be noted that the 2-stage model can be solved independently at each stage if all customers at stage 1 can be allocated by preloading.

5.2.1 Sample average approximation formulation

Let $S = \{\zeta^1, \zeta^2, ..., \zeta^{|S|}\}$ be a sample of $\tilde{\zeta}$. The 2-stage stochastic programming model described above can be approximated by the sample average approximation Kleywegt et al. (2002) using the following deterministic mixed integer linear programming model. In what follows, this model will be referred to as the SAA model.

$$\max \sum_{i\in T_1}\sum_{j\in C_1} \eta^i_j + \frac{1}{|S|}\sum_{s\in S}\sum_{i\in T_2}\sum_{j\in C_1\cup C_2} \widehat{\eta}^i_{js} \tag{5.51}$$

subject to:

$$(5.2) - (5.24)$$

$$\sum_{i\in T_2} \widehat{\eta}^i_{js} \le \zeta^s_j, \quad \forall j\in C_2, \forall s\in S \tag{5.52}$$

$$\sum_{i\in T_2} \widehat{\eta}^i_{js} \le 1, \quad \forall j\in C_2, \forall s\in S \tag{5.53}$$

$$\sum_{i\in T_2} \widehat{\eta}^i_{js} \le \rho_j, \quad \forall j\in C_1, \forall s\in S \tag{5.54}$$

$$\sum_{j\in C_1\cup C_2} \widehat{\gamma}^i_{js} \le 1, \quad \forall i\in T_2, \forall s\in S \tag{5.55}$$

$$\widehat{\gamma}^i_{js} + \sum_{k\in C_1\cup C_2} \widehat{x}^i_{kjs} = \widehat{\eta}^i_{js}, \quad \forall i\in T_2, \forall j\in C_1\cup C_2, \forall s\in S \tag{5.56}$$

$$\widehat{\theta}^i_{js} + \sum_{k\in C_1\cup C_2} \widehat{x}^i_{jks} = \widehat{\eta}^i_{js}, \quad \forall i\in T_2, \forall j\in C_1\cup C_2, \forall s\in S \tag{5.57}$$

$$a_j \le \widehat{\psi}^i_{js} \le b_j, \quad \forall j\in C_1\cup C_2, \forall i\in T_2, \forall s\in S \tag{5.58}$$

$$(r_i + \delta_i + t_{0k})\widehat{\gamma}^i_{ks} \leq \widehat{\psi}^i_{ks}, \quad \forall i \in T_2, \forall k \in C_1 \cup C_2, \forall s \in S \tag{5.59}$$

$$\widehat{\psi}^i_{js} + (p_j + t_{jk})\widehat{x}^i_{jks} + (a_k - b_j)(1 - \widehat{x}^i_{jks}) \leq \widehat{\psi}^i_{ks}, \quad \forall i \in T_2, \forall j \in C_1 \cup C_2, \forall k \in C_1 \cup C_2, k \neq j, \forall s \in S \tag{5.60}$$

$$p_j + \widehat{\psi}^i_{js} - r_i - (p_j + b_j - r_i)(1 - \widehat{\theta}^i_{js}) \leq \Psi_i, \quad \forall j \in C_1 \cup C_2, \forall i \in T_2, \forall s \in S \tag{5.61}$$

$$\sum_{k \in C_1 \cup C_2} w^d_k \widehat{\eta}^i_{ks} \leq W_i, \quad \forall i \in T_2, \forall s \in S \tag{5.62}$$

$$y_{ks} \leq W_i + (\max_{e \in T_2} W_e - W_i)(1 - \widehat{\eta}^i_{ks}), \quad \forall i \in T_2, \forall k \in C_1 \cup C_2, \forall s \in S \tag{5.63}$$

$$\sum_{j \in C_1 \cup C_2} w^d_j \widehat{\eta}^i_{js} - w^d_k + w^p_k - (\max_{e \in T_2} W_e - w^d_k + w^p_k)(1 - \widehat{\gamma}^i_{ks}) \leq y_{ks}, \quad \forall i \in T_2, \forall k \in C_1 \cup C_2, \forall s \in S \tag{5.64}$$

$$y_{js} - w^d_k + w^p_k - (\max_{e \in T_2} W_e - w^d_k + w^p_k)(1 - \widehat{x}^i_{jks}) \leq y_{ks}, \quad \forall i \in T_2, \forall j \in C_1 \cup C_2, \forall k \in C_1 \cup C_2, k \neq j, \forall s \in S \tag{5.65}$$

$$\sum_{k \in C_1 \cup C_2} v^d_k \widehat{\eta}^i_{ks} \leq V_i, \quad \forall i \in T_2, \forall s \in S \tag{5.66}$$

$$z_{ks} \leq V_i + (\max_{e \in T_2} V_e - V_i)(1 - \widehat{\eta}^i_{ks}), \quad \forall i \in T_2, \forall k \in C_1 \cup C_2, \forall s \in S \tag{5.67}$$

$$\sum_{j \in C_1 \cup C_2} v^d_j \widehat{\eta}^i_{js} - v^d_k + v^p_k - (\max_{e \in T_2} V_e - v^d_k + v^p_k)(1 - \widehat{\gamma}^i_{ks}) \leq z_{ks}, \quad \forall i \in T_2, \forall k \in C_1 \cup C_2, \forall s \in S \tag{5.68}$$

$$z_{js} - v^d_k + v^p_k - (\max_{e \in T_2} V_e - v^d_k + v^p_k)(1 - \widehat{x}^i_{jks}) \leq z_{ks}, \quad \forall i \in T_2, \forall j \in C_1 \cup C_2, k \in C_1 \cup C_2, k \neq j, \forall s \in S \tag{5.69}$$

$$\sum_{i \in T' \cup T_2} \sum_{k \in C''_1 \cup C''_2} \widehat{\eta}^i_{ks} = 0, \quad \forall s \in S \tag{5.70}$$

$$\widehat{x}^i_{jks} \in \{0, 1\}, \quad \forall j \in C_1 \cup C_2, \forall k \in C_1 \cup C_2, k \neq j, \forall i \in T_2, \forall s \in S \tag{5.71}$$

$$\widehat{\eta}^i_{js} \in \{0, 1\}, \quad \forall i \in T_2, \forall j \in C_1 \cup C_2, \forall s \in S \tag{5.72}$$

$$\widehat{\gamma}^i_{js} \in \{0, 1\}, \quad \forall i \in T_2, \forall j \in C_1 \cup C_2, \forall s \in S \tag{5.73}$$

$$\widehat{\theta}^i_{js} \in \{0, 1\}, \quad \forall i \in T_2, \forall j \in C_1 \cup C_2, \forall s \in S \tag{5.74}$$

5.3 Iterated local search for sample average approximation

This section describes the iterated local search designed for the SAA model of the considered SPDPP. In what follows, the iterated local search is referred to as ILS-SAA. Similar to the iterated local search described in Chapter 4, the ILS-SAA is another implementation of the Lagrangian ILS described in Chapter 3. Therefore, the ILS-SAA also requires an alternative mixed integer linear programming formulation. The formulation (5.75) - (5.106) below is equivalent to (5.51) − (5.74), but in contrast to (5.51) − (5.74), involves the following new variables.

Table 5.2: Symbols for ILS-SAA

Variables	
μ^i_j	The time warps associated with customer $j \in C_1$ in the route of vehicle $i \in T_1$ defined in (3.16) − (3.20).
τ_i	The violation by the vehicle $i \in T_1$ of the permissible shift duration.
ϕ_i	The maximum violation on weight capacity for vehicle $i \in T_1$.
φ_i	The maximum violation on volume capacity for the vehicle $i \in T_1$.
$\widehat{\mu}^i_{js}$	The time warps (defined in (3.16) − (3.20)) associated with customer $j \in C_1 \cup C_2$ in the route of vehicle $i \in T_2$ for scenario $s \in S$.
$\widehat{\tau}_{is}$	The violation by the vehicle $i \in T_2$ of the permissible shift duration for scenario $s \in S$.
$\widehat{\phi}_{is}$	The maximum violation on weight capacity for the vehicle $i \in T_2$ in scenario $s \in S$.
$\widehat{\varphi}_{is}$	The maximum violation on volume capacity for vehicle $i \in T_2$ in scenario $s \in S$.

$$\max \sum_{i \in T_1} \sum_{j \in C_2} \eta^i_j + \frac{1}{|S|} \sum_{s \in S} \sum_{i \in T_2} \sum_{j \in C_1 \cup C_2} \widehat{\eta}^i_{js} \tag{5.75}$$

subject to:

$$(5.2) - (5.6), (5.8), (5.13), (5.14), (5.17) - (5.24),$$

$$(5.52) - (5.57), (5.59), (5.64), (5.65), (5.68) - (5.74),$$

$$\psi^i_j - b_j \leq \mu^i_j, \quad \forall i \in T_1, \forall j \in C_1 \tag{5.76}$$

$$\psi_j^i - \mu_j^i + (p_j + t_{j,k})x_{j,k}^i + (a_k - b_j)(1 - x_{j,k}^i) \leq \psi_k^i, \quad \forall i \in T_1, \forall j \in C_1, \forall k \in C_1, k \neq j \tag{5.77}$$

$$p_j + \psi_j^i - r_i - (p_j + b_j - r_i)(1 - \theta_j^i) + \sum_{k \in C_1} \mu_k^i \leq \Psi_i + \tau_i, \quad i \in T_1, \forall j \in C_1 \tag{5.78}$$

$$\sum_{i \in T_1} \sum_{j \in C_1} \mu_j^i \leq 0 \tag{5.79}$$

$$\sum_{i \in T_1} \tau_i \leq 0 \tag{5.80}$$

$$\sum_{k \in C_1} w_k^d \eta_k^i \leq W_i + \phi_i, \quad \forall i \in T_1 \tag{5.81}$$

$$y_k \leq W_i + \phi_i + (\max_{e \in T_1} W_e - W_i)(1 - \eta_k^i), \quad \forall i \in T_1, k \in C_1 \tag{5.82}$$

$$\sum_{i \in T_1} \phi_i \leq 0 \tag{5.83}$$

$$\sum_{k \in C_1} v_k^d \eta_k^i \leq V_i + \varphi_i, \quad \forall i \in T_1 \tag{5.84}$$

$$z_k \leq V_i + \varphi_i + (\max_{e \in T_1} V_e - V_i)(1 - \eta_k^i), \quad \forall i \in T_1, k \in C_1 \tag{5.85}$$

$$\sum_{i \in T_1} \varphi_i \leq 0 \tag{5.86}$$

$$a_j \leq \widehat{\psi}_{js}^i, \quad \forall j \in C_1 \cup C_2, \forall i \in T_2, \forall s \in S \tag{5.87}$$

$$\widehat{\psi}_{js}^i - b_j \leq \widehat{\mu}_{js}^i, \quad \forall j \in C_1 \cup C_2, \forall i \in T_2, \forall s \in S \tag{5.88}$$

$$\widehat{\psi}_{js}^i - \widehat{\mu}_{js}^i + (p_j + t_{jk})\widehat{x}_{jks}^i + (a_k - b_j)(1 - \widehat{x}_{jks}^i) \leq \widehat{\psi}_{ks}^i, \quad \forall i \in T_2, \forall j \in C_1 \cup C_2, \forall k \in C_1 \cup C_2, k \neq j, \forall s \in S \tag{5.89}$$

$$p_j + \widehat{\psi}_{js}^i - r_i - (p_j + b_j - r_i)(1 - \widehat{\theta}_{js}^i) \leq \Psi_i + \widehat{\tau}_{is}, \forall j \in C_1 \cup C_2, \forall i \in T_2, \forall s \in S \tag{5.90}$$

$$\sum_{i \in T_2} \sum_{j \in C_1 \cup C_2} \widehat{\mu}_{js}^i \leq 0, \forall s \in S \tag{5.91}$$

$$\sum_{i \in T_2} \widehat{\tau}_{is} \leq 0, \forall s \in S \tag{5.92}$$

$$\sum_{k \in C_1 \cup C_2} w_k^d \widehat{\eta}_{ks}^i \leq W_i + \widehat{\phi}_{is}, \quad \forall i \in T_2, \forall s \in S \tag{5.93}$$

$$y_{ks} \leq W_i + \widehat{\phi}_{is} + (\max_{e \in T_2} W_e - W_i)(1 - \widehat{\eta}_{ks}^i), \quad \forall i \in T_2, \forall k \in C_1 \cup C_2, \forall s \in S \tag{5.94}$$

$$\sum_{i \in T_2} \widehat{\phi}_{is} \leq 0, \forall s \in S \tag{5.95}$$

$$\sum_{k \in C_1 \cup C_2} v_k^d \widehat{\eta}_{ks}^i \leq V_i + \widehat{\varphi}_i, \quad \forall i \in T_2, \forall s \in S \tag{5.96}$$

$$z_{ks} \leq V_i + \widehat{\varphi}_i + (\max_{e \in T_2} V_e - V_i)(1 - \widehat{\eta}_{ks}^i), \quad \forall i \in T_2, \forall k \in C_1 \cup C_2, \forall s \in S \tag{5.97}$$

$$\sum_{i \in T_2} \widehat{\varphi}_{is} \leq 0, \forall s \in S \tag{5.98}$$

$$\mu_j^i \geq 0, \forall i \in T_1, j \in C_1 \tag{5.99}$$

$$\tau_i \geq 0, \forall i \in T_1 \tag{5.100}$$

$$\phi_i \geq 0, \forall i \in T_1 \tag{5.101}$$

$$\varphi_i \geq 0, \forall i \in T_1 \tag{5.102}$$

$$\widehat{\mu}_{js}^i \geq 0, \forall i \in T_2, j \in C_1 \cup C_2, s \in S \tag{5.103}$$

$$\widehat{\tau}_{is} \geq 0, \forall i \in T_2, s \in S \tag{5.104}$$

$$\widehat{\phi}_{is} \geq 0, \forall i \in T_2, s \in S \tag{5.105}$$

$$\widehat{\varphi}_{is} \geq 0, \forall i \in T_2, s \in S \tag{5.106}$$

The objective function (5.75) is the same as in the objective function (5.51) in the SAA model. Constraints (5.77), (5.89) correspond to (3.17), (3.19) and (3.20) in the definition of time warps, whereas constraints (5.76) and (5.88) correspond to (3.18). The constraints (5.79), (5.80), (5.83), (5.86), (5.91), (5.92), (5.95), (5.98) guarantee that μ_j^i, τ_i, ϕ_i, φ_i, $\widehat{\mu}_{js}^i$, $\widehat{\tau}_{is}$, $\widehat{\phi}_{is}$, $\widehat{\varphi}_{is}$ are zero.

The dualisation of (5.79), (5.80), (5.83), (5.86), (5.91), (5.92), (5.95), (5.98), using Lagrange multipliers $\alpha > 0$, $\beta > 0$, $\sigma > 0$, $\kappa > 0$, $\alpha_s > 0$, $\beta_s > 0$, $\sigma_s > 0$, $\kappa_s > 0$, for all $s \in S$, gives the following Lagrangian relaxation.

$$\begin{aligned}
\max \sum_{i \in T_1} \sum_{j \in C_1} \eta_j^i - \alpha \sum_{i \in T_1} \sum_{j \in C_1} \mu_j^i - \beta \sum_{i \in T_1} \tau_i - \sigma \sum_{i \in T_1} \phi_i - \kappa \sum_{i \in T_1} \varphi_i \\
+ \frac{1}{|S|} \sum_{s \in S} \sum_{i \in T_2} \sum_{j \in C_1 \cup C_2} \widehat{\eta}_{js}^i \\
- \sum_{s \in S} \alpha_s \sum_{i \in T_2} \sum_{j \in C_1 \cup C_2} \widehat{\mu}_{js}^i - \sum_{s \in S} \beta_s \sum_{i \in T_2} \widehat{\tau}_{is} \\
- \sum_{s \in S} \sigma_s \sum_{i \in T_2} \widehat{\phi}_{is} - \sum_{s \in S} \kappa_s \sum_{i \in T_2} \widehat{\varphi}_{is}
\end{aligned} \tag{5.107}$$

subject to:

$$(5.2) - (5.6), (5.76) - (5.78), (5.81), (5.82),$$

$$(5.13), (5.14), (5.84), (5.85), (5.17) - (5.24),$$

$$(5.52) - (5.57), (5.87) - (5.90), (5.93), (5.94),$$

$$(5.64), (5.65), (5.96), (5.97), (5.68) - (5.74), (5.99) - (5.106)$$

As mentioned above, since the ILS-SAA is adapted from the Lagrangian ILS described in Chapter 3, the pseudocode below looks similar to the Lagrangian ILS. However, the main components of the ILS-SAA have been completely redesigned.

ILS-SAA

$\pi^* \leftarrow$ INITIAL-SAA
$h \leftarrow 1$
while $h \leq M$ **do**
  if π' is feasible **and** $f(\pi') < f(\pi^*)$ **then**
    $\pi^* \leftarrow \pi'$
  end if
  $\varkappa \leftarrow$ WEIGHTS-SAA(π')
  $\pi' \leftarrow$ SEARCH-SAA(π')
  $e \leftarrow 1$
  while $e \leq E$ **and** π' is infeasible **do**
    $\varkappa \leftarrow$ ADJUST-SAA$(\varkappa, \pi')$
    $\pi' \leftarrow$ SEARCH-SAA(π')
    $e \leftarrow e + 1$
  end while
  if π' is feasible **and** $f(\pi') < f(\pi^*)$ **then**
    $\pi^* \leftarrow \pi'$
    $h \leftarrow 0$
  end if
  $\pi' \leftarrow$ PERTURB-SAA(π^*, h)
  $h \leftarrow h + 1$
end while
return π^*

- The subroutine INITIAL-SAA constructs the initial feasible solution for the considered problem, whereas the subroutine PERTURB-SAA generates a random solution using the current best feasible solution. Both subroutines must respect the non-anticipativity constraints.

- Let $\varkappa = (\alpha, \beta, \sigma, \kappa, \alpha_1, \alpha_2, ..., \alpha_{|S|}, \beta_1, \beta_2, ..., \beta_{|S|}, \sigma_1, \sigma_2, ..., \sigma_{|S|}, \kappa_1, \kappa_2, ..., \kappa_{|S|})$ comprises all Lagrange multipliers. The subroutine WEIGHTS-SAA computes the initial values for all the Lagrange multipliers, whereas the subroutine ADJUST-SAA updates these values by taking into account the violation of constraints of the input solution.

- Although the scheme for the subroutine SEARCH-SAA is the same as the subroutine SEARCH in the Lagrangian ILS in Chapter 3, the neighbourhood operators used in the subroutine SEARCH-SAA have been redesigned.

5.3.1 Initial solutions for ILS-SAA

The subroutine INITIAL-SAA constructs routes using a sweep heuristic Gillett and Miller (1974). For the first stage solution, a list of customers in C_1 is constructed based on the geographic coordinates of the customers. Then these customers are inserted into a route corresponding to a vehicle in T_1 one by one until no customer can be inserted, in which case a new route is constructed. Since vehicles in $T' \cap T_1$ can only serve customers in C_1', whereas vehicles in $T'' \cap T_1$ can serve all types of customers in C_1, the sweep heuristic constructs the routes for vehicles in $T' \cap T_1$ first, then followed by the routes for vehicles in $T'' \cap T_1$. When inserting a customer into the route, the heuristic chooses the insertion position that respects all the constraints with the smallest increase in travel time. The sweep heuristic terminates until either no customers in C_1 can be inserted into the routes of the vehicle in T_1. Using the same heuristic, the routes for the second stage solution are constructed by generating a list of customers in C_2 and a subset of C_1 comprises customers who are not allocated in the first stage solution.

5.3.2 Subroutine WEIGHTS-SAA

The input of the subroutine WEIGHTS-SAA is an output of either the subroutine START or the subroutine PERTURB-SAA. The output of the subroutine WEIGHTS-SAA is α, β, σ, κ, and α_s, β_s, σ_s, and κ_s for all $s \in S$ in (5.107), which are the weights used to calculate the penalty for the violation of constraints. For input solution π, the violation of time windows $u_j^i(\pi)$ and $u_{js}^i(\pi)$ for all $s \in S$; the violation of permissible shift duration $\tau_i(\pi)$ and $\tau_{is}(\pi)$ for all $s \in S$; the violation of vehicle's weight capacity $\phi_i(\pi)$ and $\phi_{is}(\pi)$; the violation of vehicle's volume capacity $\varphi_i(\pi)$ and $\varphi_{is}(\pi)$ for all $s \in S$ are calculated based on the technique in Vidal et al. (2013). The subroutine WEIGHTS-SAA computes the weights in the penalty for the violation of constraints (the values of Lagrange multipliers) as follows:

$$\alpha = \sum_{i \in T_1} \sum_{j \in C_1} u_j^i(\pi) \ ; \ \beta = \sum_{i \in T_1} \tau_i(\pi) \ ; \ \sigma = \sum_{i \in T_1} \phi_i(\pi) \ ; \ \kappa = \sum_{i \in T_1} \varphi_i(\pi) \ ;$$

$$\alpha_s = \sum_{i \in T_2} \sum_{j \in C_1 \cup C_2} u_{js}^i(\pi) \ , \forall s \in S \ ; \quad \beta_s = \sum_{i \in T_2} \tau_{is}(\pi) \ , \forall s \in S \ ;$$

$$\sigma_s = \sum_{i \in T_2} \phi_{is}(\pi) \ , \forall s \in S \ ; \quad \kappa_s = \sum_{i \in T_2} \varphi_{is}(\pi) \ , \forall s \in S \ .$$

So, each call of the subroutine WEIGHTS-SAA results in Lagrange multipliers (weights) that reflect the violation of constraints by the input solution.

5.3.3 Subroutine SEARCH-SAA

The local search attempts to solve the Lagrangian relaxation problem for fixed values of the Lagrange multipliers (for fixed weights in the augmented objective function), using six local search optimisation procedures, each with one of the six operators N_0, N_1, N_2, N_3, N_4, N_5. Each operator N_i transforms an input solution π, by applying transformations (moves) from the set of transformations associated with this operator, and returns as the result some solution π' (denoted $\pi' = N_i(\pi)$) where π' is either the input solution π, or one of the transformations of π.

N_0 Interchange a sequence of up to two consecutive visits in two different routes of the first-stage solution.

N_1 For a sequence of up to two consecutive visits in a route of a vehicle in T_1 and at most one customer $j \in C_1$ who is not allocated to any vehicles in T_1. Interchange their allocations if customer j is allocated in a scenario of the second-stage solution, otherwise, the customers in the sequence become unallocated. To make it easier to understand, Figure 5.2 shows a solution with 2 scenarios, whereas Figures 5.3 and 5.4 show two examples of the transformation associated with this operator. In Figures 5.3 and 5.4, the circle nodes represent customers in C_1, whereas the square nodes represent customers in C_2

N_2 – For each scenario of the second stage, interchange a sequence of up to two

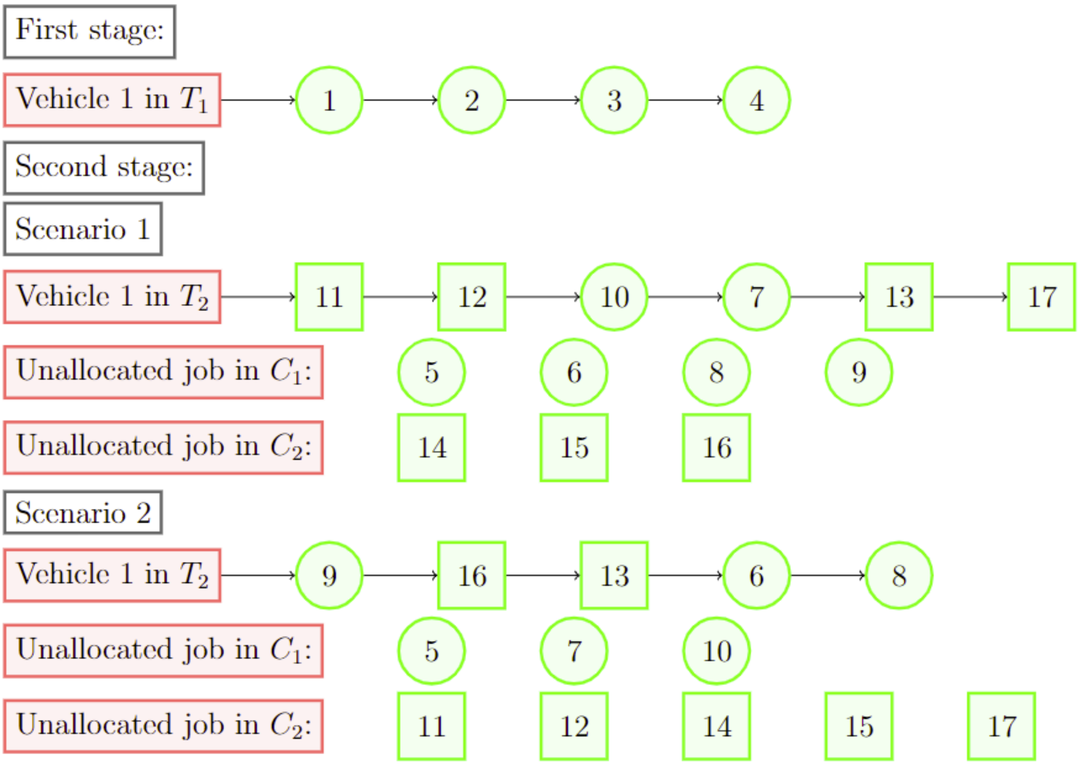


Figure 5.2: Example of a solution with 2 scenarios

consecutive visits in the route of a vehicle in T_2 with a sequence of up to two consecutive visits in the route of another vehicle in T_2.

– For each scenario of the second stage, interchange a sequence of up to two consecutive visits in a route of a vehicle in T_2 (the customers in this sequence become unallocated) with at most one unallocated customer in C_2.

The set of transformations associated with operator N_3 is comprised of all transformations that extract one visit from a route and insert it into a different position of the same route. Operator N_4 is similar to N_3, but, instead of one visit, each transformation performed by N_4 extracts a sequence of two consecutive visits and inserts this sequence into a different position of the same route. Each transformation performed by N_5 reverses the order of a sequence of consecutive visits in a route.

5.3.4 Subroutine ADJUST-SAA

The weights α, β, σ, κ, α_s, β_s, σ_s, and κ_s of the penalty for the violation of constraints (5.79), (5.80), (5.83), (5.86), (5.91), (5.92), (5.95), and (5.98) are computed prior to each call of the subroutine SEARCH-SAA and remain unchanged till the next call of this subroutine. Prior to a call of the subroutine SEARCH-SAA, these penalties are computed either by the subroutine WEIGHTS-SAA, or by the subroutine ADJUST-SAA. If an optimal solution $\hat{\pi}$ of the Lagrangian relaxation problem can be found, then according to a commonly used version of the Lagrangian relaxation method Fisher (1981), Guignard (2003), the weights α, β, σ, κ, α_s, β_s, σ_s, and κ_s are updated to

$$\alpha + \lambda \sum_{i \in T_1} \sum_{j \in C} u_j^i(\hat{\pi}) \tag{5.108}$$

$$\beta + \lambda \sum_{i \in T_1} \tau_i(\hat{\pi}) \tag{5.109}$$

$$\sigma + \lambda \sum_{i \in T_1} \phi_i(\hat{\pi}) \tag{5.110}$$

$$\kappa + \lambda \sum_{i \in T_1} \varphi_i(\hat{\pi}) \tag{5.111}$$

$$\alpha_s + \lambda \sum_{i \in T_2} \sum_{j \in C_1 \cup C_2} u_{js}^i(\hat{\pi}), \forall s \in S \tag{5.112}$$

$$\beta_s + \lambda \sum_{i \in T_2} \tau_{is}(\hat{\pi}), \forall s \in S \tag{5.113}$$

$$\sigma_s + \lambda \sum_{i \in T_2} \phi_{is}(\hat{\pi}), \forall s \in S \tag{5.114}$$

$$\kappa_s + \lambda \sum_{i \in T_2} \varphi_{is}(\hat{\pi}), \forall s \in S \tag{5.115}$$

where $\mu_j^i(\hat{\pi}), i \in T_1, j \in C_1$; $\tau_i(\hat{\pi}), i \in T_1$; $\phi_i(\hat{\pi}), i \in T_1$; $\varphi_i(\hat{\pi}), i \in T_1$; $\mu_{js}^i(\hat{\pi}), i \in T_2, j \in C_1 \cup C_2, s \in S$; $\tau_{is}(\hat{\pi}), i \in T_2, s \in S$; $\phi_{is}(\hat{\pi}), i \in T_2, s \in S$; and $\varphi_{is}(\hat{\pi}), i \in T_2, s \in S$ are the violations of constraints (5.79), (5.80), (5.83), (5.86), (5.91), (5.92), (5.95), and (5.98) caused by solution $\hat{\pi}$, and

$$\lambda = \frac{\Lambda \left(f(\pi^*) - f_{LR}(\hat{\pi}) \right)}{A} \tag{5.116}$$

where Λ is a positive parameter, $f(\cdot)$ is the original objective function, $f_{LR}(\cdot)$ is the augmented objective function (the objective function for the LR problem), and π^* is the

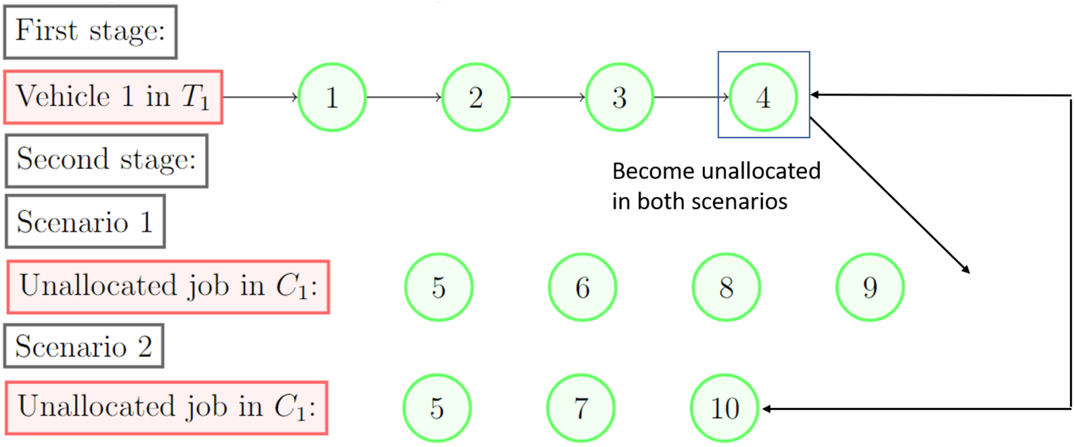


Figure 5.3: First example of a transformation associated with the operator N_1

best currently known solution for the original problem, and

$$A = \Big(\sum_{i \in T_1} \sum_{j \in C_1} \mu_j^i(\hat{\pi})\Big)^2 + \Big(\sum_{i \in T_1} \tau_i(\hat{\pi})\Big)^2 + \Big(\sum_{i \in T_1} \phi_i(\hat{\pi})\Big)^2 + \Big(\sum_{i \in T_1} \varphi_i(\hat{\pi})\Big)^2$$
$$+ \sum_{s \in S} \Big(\Big(\sum_{i \in T_2} \sum_{j \in C_1 \cup C_2} \mu_{js}^i(\hat{\pi})\Big)^2 + \Big(\sum_{i \in T_2} \tau_{is}(\hat{\pi})\Big)^2 + \Big(\sum_{i \in T_2} \phi_{is}(\hat{\pi})\Big)^2 + \Big(\sum_{i \in T_2} \varphi_{is}(\hat{\pi})\Big)^2\Big). \tag{5.117}$$

Since the subroutine SEARCH-SAA cannot guarantee the optimal solution $\hat{\pi}$, (5.116) can result in a negative value. Therefore, instead of (5.108) − (5.115), the ILS-SAA uses

$$\alpha + \lambda \sum_{i \in T_1} \sum_{j \in C} u_j^i(\pi) \tag{5.118}$$

$$\beta + \lambda \sum_{i \in T_1} \tau_i(\pi) \tag{5.119}$$

$$\sigma + \lambda \sum_{i \in T_1} \phi_i(\pi) \tag{5.120}$$

$$\kappa + \lambda \sum_{i \in T_1} \varphi_i(\pi) \tag{5.121}$$

$$\alpha_s + \lambda \sum_{i \in T_2} \sum_{j \in C_1 \cup C_2} u_{js}^i(\pi), \forall s \in S \tag{5.122}$$

$$\beta_s + \lambda \sum_{i \in T_2} \tau_{is}(\pi), \forall s \in S \tag{5.123}$$

$$\sigma_s + \lambda \sum_{i \in T_2} \phi_{is}(\pi), \forall s \in S \tag{5.124}$$

$$\kappa_s + \lambda \sum_{i \in T_2} \varphi_{is}(\pi), \forall s \in S \tag{5.125}$$

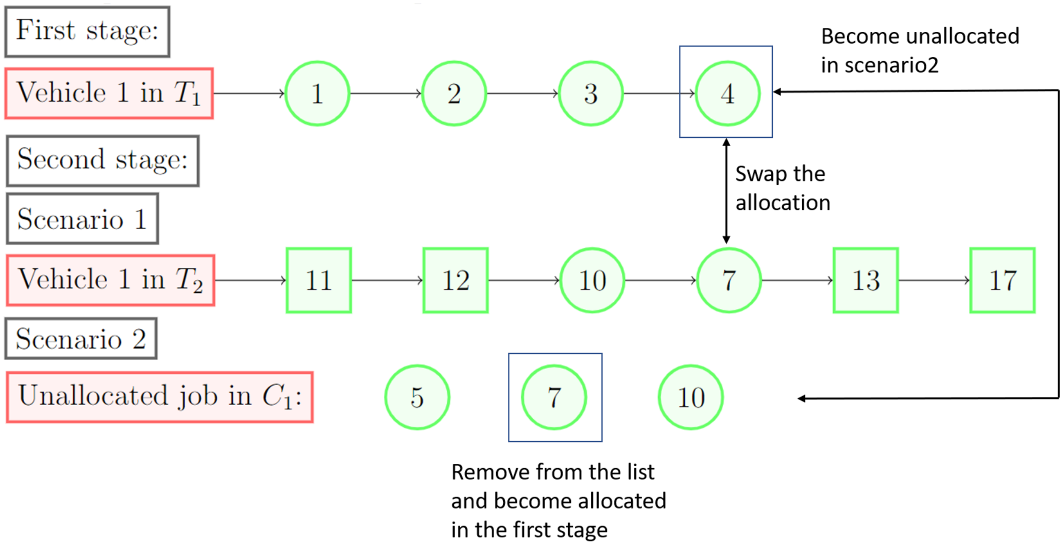


Figure 5.4: Second example of a transformation associated with the operator N_1

where $\mu^i_j(\pi), i \in T_1, j \in C$; $\tau_i(\pi), i \in T_1$; $\phi_i(\pi), i \in T_1$; $\varphi_i(\pi), i \in T_1$; $\mu^i_{js}(\pi), i \in T_2, j \in C_1 \cup C_2, s \in S$; $\tau_{is}(\pi), i \in T_2, s \in S$; $\phi_{is}(\pi), i \in T_2, s \in S$; and $\varphi_{is}(\pi), i \in T_2, s \in S$ are the violations of constraints (5.79), (5.80), (5.83), (5.86), (5.91), (5.92), (5.95), and (5.98) caused by the output π of the subroutine SEARCH-SAA, and

$$\lambda = \frac{\chi f(\pi^*)}{A} \tag{5.126}$$

where χ is a positive parameter and

$$A = \Big(\sum_{i\in T_1}\sum_{j\in C_1}\mu^i_j(\pi)\Big)^2 + \Big(\sum_{i\in T_1}\tau_i(\pi)\Big)^2 + \Big(\sum_{i\in T_1}\phi_i(\pi)\Big)^2 + \Big(\sum_{i\in T_1}\varphi_i(\pi)\Big)^2$$
$$+ \sum_{s\in S}\bigg(\Big(\sum_{i\in T_2}\sum_{j\in C_1\cup C_2}\mu^i_{js}(\pi)\Big)^2 + \Big(\sum_{i\in T_2}\tau_{is}(\pi)\Big)^2 + \Big(\sum_{i\in T_2}\phi_{is}(\pi)\Big)^2 + \Big(\sum_{i\in T_2}\varphi_{is}(\pi)\Big)^2\bigg). \tag{5.127}$$

5.3.5 Perturbation

The PERTURB-SAA procedure expands the search space by randomly perturbing the current best solution s^*. If there are customers in C_1 who are not allocated to a vehicle in T_1, the first-stage solution is modified by randomly choosing one of these customers, and then inserting the customer into the route of a vehicle in T_1 in such a way that this insertion results in the largest increase of (5.107) with $\alpha = 1$, $\beta = 1$, $\sigma = 1$, $\kappa = 1$,

$\alpha_s = 1$, $\beta_s = 1$, $\sigma_s = 1$, $\kappa_s = 1$ for all $s \in S$. If this customer has been allocated in the second-stage solution, this customer is removed from the second stage. For each scenario of the second stage, if there are customers in C_2 who are not allocated to any vehicle in T_2, then one of these customers is randomly selected and inserted into the route of a vehicle in T_2 in such a way that this insertion results in the largest increase of (5.107) with $\alpha = 1$, $\beta = 1$, $\sigma = 1$, $\kappa = 1$, $\alpha_s = 1$, $\beta_s = 1$, $\sigma_s = 1$, $\kappa_s = 1$ for all $s \in S$.

Then, for the first-stage solution and the solution for each scenario of the second stage, the subroutine PERTURB-SAA selects two random sequences of consecutive customers in two random routes; and swaps their position in these two routes. This random swap will be performed multiple times which depends on the counter h in the pseudocode for the ILS-SAA. To be specific, the number of swaps starts from one and increases by one each time when counter h in ILS-SAA increases. The current best solution s^* may also be updated in this process.

5.3.6 ILS-SAA for multiple scenarios

The problem size of the SAA model increases when the number of scenarios considered in the model increases. Therefore, the SAA model with more scenarios is computationally more difficult to solve compared with the SAA model with a single scenario. To solve the SAA model with multiple scenarios, the ILS-SAA is enhanced by the multi-start framework Martí (2003) and will be referred to as the multi-start ILS-SAA (MSILS-SAA). The pseudocode below outlines the MSILS-SAA.

Let $\Re = \{1, 2, \ldots, \varsigma\}$ be the set of scenarios in an SAA model with $|\Re|$ scenarios. The MSILS-SAA applies the ILS-SAA $|\Re|$ times. Each application begins with a different feasible solution. Between lines 6 and 12, an initial feasible solution is constructed. As shown in figure 5.5, at each application, an SAA model with a single scenario is built (line 6). Then ILS-SAA is applied to this model. The first-stage solution of the output will be used as the first-stage solution for the SAA model with $|\Re|$ scenarios. Between lines 7 and 12, the second-stage solution is constructed scenario by scenario. First, $|\Re|$ vehicle routing problems studied in Gu et al. (2021) are built. Each problem uses customers in a scenario and unallocated customers in the first-stage solution (line 9). Then, the ILS presented in Gu et al. (2021) is applied to each problem which results in a complete second-stage solution for the SAA model with $|\Re|$ scenarios. Using the feasible solution

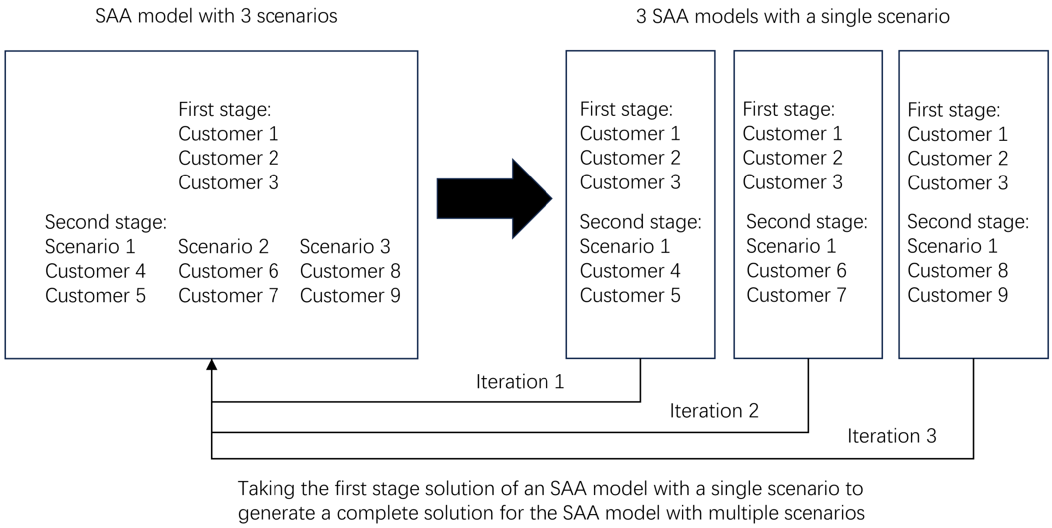


Figure 5.5: Constructing initial feasible solution of MSILS-SAA

as the starting solution, the MSILS-SAA applies the ILS-SAA to solve the SAA models with $|\Re|$ scenarios (line 13). The MSILS-SAA terminates when it runs out of the starting solutions and returns the best feasible solution found so far.

MSILS-SAA

1: $m \leftarrow 1$
2: $\pi \leftarrow$ Empty solution
3: $f(\pi^*) \leftarrow -\infty$
4: **while** $m \in \Re$ **do**
5: Build an SAA model with a single scenario using scenario m
6: $\pi \leftarrow$Apply ILS-SAA to this model and construct the first stage solution
7: $e \leftarrow 1$
8: **while** $e \in \Re$ **do**
9: Build a model studied in Gu et al. (2021) using customers in scenario e and unallocated customers in the first-stage solution
10: $\pi \leftarrow$ Apply ILS in Gu et al. (2021) to this model and construct part of the second stage solution
11: $e \leftarrow e + 1$
12: **end while**
13: $\pi \leftarrow$ ILS-SAA(π)
14: **if** π is feasible **and** $f(\pi) > f(\pi^*)$ **then**
15: $\pi^* \leftarrow \pi$
16: **end if**
17: $m \leftarrow m + 1$
18: **end while**
19: **return** π^*

5.4 Computational experiments

This section presents the results of computational experiments. Since there is no method exists in the literature that solves the same SAA model in this chapter, to evaluate the performance of the proposed ILS-SAA in solving the SAA model, its performance is compared with the performance of CPLEX and an algorithm referred to as the Greedy algorithm. The Greedy algorithm has two phases. In the first phase, CPLEX is used to find the first-stage solution that serves the largest number of customers in C_1 ignoring what customers will appear in C_2. Then, by taking into account the customers that are not allocated in the first stage together with customers who appeared in C_2 from each scenario, the second phase of the Greedy algorithm repeatedly applies the iterated local search described in Gu et al. (2021) to find the solution for each scenario in the second stage that serves the largest number of customers.

Since the neighbourhood reduction technique described in Gu et al. (2021) demonstrates good performance for maximising the number of allocated customers in the deterministic version of the SPDPP, a version of the ILS-SAA using the neighbourhood reduction technique has been implemented. To distinguish between the two versions, the ILS-SAA with neighbourhood reduction technique will be referred to as the Reduced ILS-SAA and the ILS-SAA without neighbourhood reduction technique will be referred to as the NoReduced ILS-SAA.

For both Reduced ILS-SAA and NoReduced ILS-SAA, the maximum number of exchange operations in the subroutine PERTURB-SAA is five, which is the same as the Lagrangian ILS; the parameter E is 100; the parameter M is computed according to $\Theta \times (|C_1| + |C_2| + 10(|T_1| + |T_2|))$, where $C_1 \cup C_2$ is the set of all customers; $T_1 \cup T_2$ is the set of all vehicles; Θ is a parameter to control M. Similar to the Lagrangian ILS, the ILS-SAA increases the number of exchange operations in perturbation after each $M/5$ sequential iterations that fail to obtain an improving solution.

To investigate the performance of the ILS-SAA, CPLEX, and the Greedy algorithm for SAA models with different scenarios, these algorithms were applied to SAA models with 1 scenario, 5 scenarios, and 10 scenarios. The computational experiments did not test the algorithms on SAA models with a larger number of scenarios. As shown in the results below, the solutions obtained from CPLEX become worse when the number of scenarios in the SAA model increases from 1 to 10. The reason is that when the number of scenarios in the SAA model increases the problem size also increases. CPLEX is not able to find good solutions for SAA models with a large number of scenarios within the given time limit and memory limit.

In addition to the comparisons of the objective values obtained from the ILS-SAA, CPLEX, and the Greedy algorithm for solving the SAA models. To evaluate the solution quality of the first-stage solutions obtained from these algorithms, this section also conducted a stochastic analysis. For a first-stage solution produced by either the ILS-SAA, CPLEX or the Greedy algorithm, the stochastic analysis constructs a set of VRPSPDs studied in Gu et al. (2021) using a set of random samples. Each sample is a set of customers in C_2. The VRPSPD studied in Gu et al. (2021) is constructed by combining the customers in a sample with the unallocated customers of C_1 in a particular first-stage solution. For each VRPSPD, the iterated local search described in Gu et al. (2021) is applied. Then, an expected total number of allocated customers is computed for a particular first-stage solution using the numbers of allocated customers obtained from solving the VRPSPDs constructed by this first-stage solution. Please note that the stochastic analysis uses the same set of samples for the first-stage solutions obtained by the ILS-SAA, CPLEX and the Greedy algorithm.

All computational experiments are conducted on a computer with Intel Xeon CPU E5-2697 v3 2.60GHz and 8GB RAM. All algorithms were programmed in C++ and compiled with g++, using the optimisation level O3. The version for CPLEX used for all tests is 12.10. The time limit is 6 hours and the memory limit is 8GB RAM.

In what follows, Section 5.4.1 discusses the benchmark instances used for the computational experiments. Section 5.4.2 analyses how the performance of the ILS-SAA changes with the variation of parameters χ and Θ. In section 5.4.3, the performance of the ILS-SAA is compared with the performance of the Greedy algorithm and CPLEX solving the SAA model with a single scenario. Then, in section 5.4.4, the performance of the MSILS-SAA is compared with the performance of the Greedy algorithm and CPLEX when the SAA model has 5 scenarios and 10 scenarios. Section 5.4.5 presents the results obtained from the stochastic analyses on the first-stage solutions obtained from the Greedy algorithm; CPLEX for SAA models with 1, 5, and 10 scenarios; ILS-SAA for SAA model with 1 scenario; and MSILS-SAA for SAA models with 5 and 10 scenarios.

5.4.1 Test instances

The instances used for the computational experiments were derived from the historical data from September 2021 to April 2022 provided by the industry partner. The data include each customer's demand, location, time window, and service time. In addition, driver rosters and the capacity of the vehicles were also provided. 43 instances were derived which can be classified into seven groups. Each group corresponds to the customers who appeared on a particular day of the week. For example, the customers in the instance "FR1" were randomly selected from a list of customers who appeared on Friday in the historical data. The maximum shift duration for all drivers is 10 hours. Furthermore, for each instance, 94 random samples of customers in C_2 were generated using the same historical data. These random samples were used for the stochastic analysis.

5.4.2 Sensitivity analysis

In this subsection, the performance of the Reduced ILS-SAA and NoReduced ILS-SAA are analysed with the variation of Θ and χ. Table 5.3 (Table 5.4) presents the results obtained from the Reduced ILS-SAA (NoReduced ILS-SAA) using a combination of $\chi \in \{0.5, 2, 5, 10, 50, 100, 1000\}$ when $\Theta = 1$. Table 5.5 (Table 5.6) presents the results obtained from the Reduced ILS-SAA (NoReduced ILS-SAA) using a combination of $\Theta \in \{0.1, 1, 5, 10, 20, 30, 50\}$ when $\chi = 0.5$. In these tables, the group EJ reports the expected number of allocated customers and the group Time(s) reports the computational time.

In Tables 5.3 and 5.4, it can be observed that for both Reduced ILS-SAA and NoReduced ILS-SAA, the algorithms perform the best when $\chi = 0.5$ with respect to the average of the expected number of allocated customers. To facilitate the reading, these values in Tables 5.3 and 5.4 are in bold. The parameter χ controls how fast the Lagrange multipliers can increase (see (5.126)). It can be seen that a small χ ($\chi = 0.5$) leads to good solution quality and short computational time, whereas when χ is large ($\chi = 1000$), the solution quality deteriorates and requires more time. In the following computational experiments, $\chi = 0.5$ is used for both Reduce ILS-SAA and NoReduced ILS-SAA.

Tables 5.5 and 5.6 have shown that the solution quality improves when Θ increases at the cost of computational time. This observation is expected since Θ can increase

Table 5.3: The performance of the Reduced ILS-SAA with χ when $\Theta = 1$

	EJ							Time(s)						
Inst	0.5	2	5	10	50	100	1000	0.5	2	5	10	50	100	1000
FR1	27.00	26.00	26.00	28.00	28.00	26.00	28.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
FR2	38.00	39.00	35.00	38.00	38.00	38.00	38.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00
FR3	35.00	35.00	34.00	34.00	36.00	34.00	35.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
FR4	54.00	52.00	55.00	54.00	54.00	53.00	52.00	1.00	1.00	2.00	1.00	3.00	1.00	1.00
FR5	62.00	63.00	62.00	63.00	62.00	63.00	62.00	1.00	4.00	2.00	2.00	1.00	3.00	2.00
FR6	73.00	80.00	76.00	77.00	76.00	77.00	78.00	1.00	10.00	4.00	3.00	4.00	7.00	5.00
FR7	60.00	58.00	61.00	61.00	61.00	58.00	59.00	3.00	2.00	2.00	3.00	3.00	2.00	2.00
FR8	88.00	89.00	88.00	90.00	91.00	89.00	89.00	5.00	5.00	5.00	6.00	6.00	4.00	4.00
FR9	69.00	68.00	69.00	68.00	69.00	69.00	68.00	3.00	3.00	4.00	3.00	2.00	4.00	2.00
MO1	13.00	13.00	13.00	13.00	13.00	13.00	13.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
MO2	36.00	35.00	36.00	33.00	35.00	35.00	34.00	0.00	0.00	1.00	0.00	1.00	0.00	0.00
MO3	44.00	45.00	45.00	45.00	45.00	47.00	45.00	1.00	0.00	0.00	1.00	0.00	2.00	0.00
MO5	64.00	64.00	65.00	65.00	65.00	64.00	64.00	1.00	2.00	2.00	2.00	2.00	1.00	2.00
MO8	88.00	88.00	86.00	89.00	87.00	89.00	89.00	5.00	4.00	0.00	5.00	8.00	4.00	8.00
MO9	72.00	72.00	71.00	72.00	72.00	72.00	73.00	2.00	2.00	2.00	3.00	3.00	2.00	3.00
SA1	25.00	25.00	24.00	25.00	25.00	25.00	24.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SA2	25.00	23.00	25.00	24.00	23.00	24.00	24.00	0.00	1.00	1.00	1.00	0.00	0.00	0.00
SA3	43.00	43.00	43.00	43.00	42.00	42.00	42.00	1.00	0.00	0.00	0.00	1.00	1.00	0.00
SA4	54.00	52.00	52.00	49.00	50.00	49.00	49.00	1.00	1.00	1.00	1.00	1.00	1.00	2.00
SA5	63.00	64.00	63.00	62.00	62.00	62.00	64.00	2.00	4.00	3.00	2.00	1.00	1.00	3.00
SA6	69.00	69.00	71.00	71.00	68.00	69.00	67.00	2.00	2.00	3.00	2.00	3.00	3.00	2.00
SA7	83.00	82.00	83.00	82.00	81.00	82.00	83.00	6.00	3.00	4.00	3.00	3.00	4.00	4.00
SA8	83.00	84.00	79.00	79.00	80.00	79.00	82.00	4.00	6.00	4.00	5.00	5.00	3.00	7.00
SA9	70.00	70.00	70.00	70.00	70.00	70.00	70.00	2.00	2.00	2.00	2.00	3.00	3.00	2.00
SU1	25.00	23.00	24.00	24.00	25.00	25.00	25.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SU2	22.00	21.00	21.00	21.00	22.00	21.00	21.00	0.00	1.00	0.00	0.00	0.00	0.00	1.00
SU5	56.00	56.00	56.00	56.00	55.00	56.00	56.00	1.00	2.00	2.00	1.00	1.00	1.00	1.00
TH2	35.00	37.00	36.00	36.00	34.00	36.00	36.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
TH4	53.00	53.00	54.00	53.00	54.00	53.00	54.00	1.00	1.00	1.00	1.00	2.00	2.00	2.00
TH9	67.00	67.00	67.00	67.00	66.00	67.00	67.00	2.00	3.00	2.00	2.00	2.00	2.00	2.00
TU1	26.00	26.00	26.00	26.00	26.00	26.00	26.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
TU2	33.00	33.00	32.00	31.00	30.00	32.00	32.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
TU3	42.00	42.00	42.00	41.00	42.00	42.00	41.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00
TU8	81.00	81.00	81.00	81.00	80.00	80.00	79.00	4.00	4.00	3.00	3.00	4.00	5.00	3.00
TU9	67.00	68.00	67.00	68.00	68.00	68.00	67.00	2.00	2.00	2.00	2.00	3.00	4.00	1.00
WE1	15.00	13.00	13.00	13.00	15.00	14.00	14.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
WE2	35.00	35.00	35.00	35.00	33.00	33.00	33.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
WE3	43.00	43.00	42.00	42.00	42.00	43.00	43.00	1.00	0.00	0.00	0.00	0.00	1.00	1.00
WE4	55.00	55.00	53.00	55.00	55.00	55.00	54.00	1.00	1.00	0.00	1.00	1.00	1.00	1.00
WE5	55.00	56.00	54.00	53.00	53.00	55.00	56.00	2.00	3.00	2.00	2.00	2.00	3.00	2.00
WE6	84.00	83.00	83.00	81.00	82.00	83.00	82.00	3.00	3.00	5.00	2.00	3.00	3.00	2.00
WE7	75.00	74.00	74.00	74.00	73.00	74.00	73.00	2.00	4.00	2.00	2.00	3.00	3.00	3.00
WE9	72.00	71.00	72.00	71.00	69.00	71.00	72.00	4.00	3.00	3.00	3.00	2.00	4.00	4.00
Avg	**53.00**	52.93	52.65	52.63	52.49	52.63	52.63	1.53	1.86	1.53	1.58	1.72	1.77	1.74

the number of iterations for both Reduced ILS-SAA and NoReduced ILS-SAA. In addition, the Reduced ILS with $\Theta = 30$ consistently obtains a better solution compared with the Reduced ILS with $\Theta = 0.1$, whereas the NoReduced ILS with $\Theta = 5$ consistently obtains a better solution compared with the NoReduced ILS with $\Theta = 0.1$. This observation suggests that the NoReduced ILS converges faster compared with the Reduced ILS. Overall, the computational times required for both the Reduced ILS-SAA and NoReduced ILS-SAA are acceptable even with $\Theta = 50$. Therefore, in the following computational experiments, $\Theta = 50$ is used for both algorithms.

Table 5.4: The performance of the NoReduced ILS-SAA with χ when $\Theta = 1$

	EJ							**Time(s)**						
Inst	0.5	2	5	10	50	100	1000	0.5	2	5	10	50	100	1000
FR1	28.00	28.00	26.00	26.00	26.00	26.00	27.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
FR2	39.00	38.00	38.00	39.00	38.00	39.00	38.00	1.00	1.00	1.00	1.00	0.00	1.00	1.00
FR3	36.00	35.00	36.00	35.00	36.00	34.00	35.00	0.00	0.00	0.00	1.00	1.00	0.00	0.00
FR4	53.00	55.00	55.00	55.00	53.00	54.00	53.00	2.00	2.00	4.00	2.00	1.00	2.00	2.00
FR5	62.00	62.00	60.00	62.00	65.00	61.00	62.00	2.00	2.00	2.00	2.00	5.00	2.00	2.00
FR6	77.00	79.00	78.00	78.00	79.00	79.00	75.00	5.00	10.00	6.00	7.00	6.00	8.00	5.00
FR7	63.00	63.00	60.00	62.00	63.00	64.00	63.00	8.00	6.00	4.00	11.00	9.00	10.00	7.00
FR8	89.00	88.00	90.00	89.00	89.00	89.00	90.00	6.00	8.00	11.00	7.00	6.00	7.00	10.00
FR9	69.00	68.00	68.00	68.00	69.00	69.00	69.00	3.00	4.00	4.00	4.00	4.00	4.00	4.00
MO1	13.00	13.00	13.00	13.00	13.00	13.00	13.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MO2	36.00	35.00	36.00	36.00	34.00	34.00	34.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
MO3	45.00	45.00	46.00	44.00	45.00	46.00	45.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
MO5	64.00	64.00	64.00	64.00	64.00	64.00	64.00	1.00	1.00	2.00	1.00	2.00	2.00	2.00
MO8	89.00	88.00	90.00	89.00	90.00	89.00	88.00	6.00	4.00	5.00	7.00	7.00	10.00	8.00
MO9	72.00	72.00	72.00	72.00	72.00	72.00	74.00	3.00	3.00	3.00	3.00	3.00	3.00	4.00
SA1	25.00	25.00	25.00	25.00	25.00	25.00	25.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
SA2	23.00	25.00	23.00	22.00	22.00	25.00	19.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
SA3	43.00	43.00	42.00	43.00	43.00	43.00	42.00	1.00	1.00	1.00	0.00	1.00	0.00	1.00
SA4	53.00	52.00	53.00	53.00	53.00	53.00	52.00	2.00	2.00	3.00	1.00	2.00	2.00	2.00
SA5	64.00	62.00	65.00	64.00	63.00	64.00	64.00	4.00	4.00	7.00	4.00	4.00	5.00	4.00
SA6	69.00	69.00	70.00	71.00	68.00	69.00	70.00	2.00	2.00	2.00	3.00	2.00	3.00	3.00
SA7	83.00	82.00	83.00	82.00	83.00	82.00	82.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00
SA8	83.00	83.00	83.00	81.00	77.00	83.00	78.00	5.00	5.00	6.00	5.00	4.00	9.00	5.00
SA9	70.00	70.00	70.00	70.00	70.00	70.00	70.00	2.00	3.00	2.00	3.00	3.00	3.00	2.00
SU1	22.00	22.00	22.00	23.00	23.00	23.00	23.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
SU2	21.00	21.00	21.00	21.00	22.00	21.00	21.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
SU5	56.00	56.00	57.00	57.00	56.00	57.00	56.00	1.00	1.00	2.00	1.00	1.00	1.00	1.00
TH2	36.00	37.00	34.00	35.00	36.00	35.00	35.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00
TH4	53.00	53.00	53.00	54.00	53.00	53.00	54.00	2.00	2.00	1.00	2.00	1.00	1.00	1.00
TH9	67.00	67.00	67.00	67.00	66.00	67.00	66.00	2.00	2.00	3.00	2.00	3.00	4.00	3.00
TU1	26.00	26.00	26.00	26.00	26.00	26.00	26.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
TU2	33.00	32.00	31.00	31.00	30.00	31.00	31.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00
TU3	41.00	42.00	42.00	42.00	38.00	41.00	42.00	0.00	1.00	0.00	0.00	0.00	1.00	1.00
TU8	81.00	81.00	81.00	81.00	81.00	79.00	81.00	4.00	5.00	5.00	4.00	5.00	4.00	4.00
TU9	67.00	68.00	68.00	68.00	68.00	67.00	68.00	3.00	2.00	2.00	4.00	3.00	3.00	3.00
WE1	16.00	15.00	16.00	15.00	15.00	14.00	14.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
WE2	36.00	35.00	35.00	35.00	32.00	31.00	34.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
WE3	43.00	43.00	43.00	43.00	42.00	43.00	42.00	1.00	0.00	1.00	0.00	0.00	1.00	1.00
WE4	55.00	55.00	55.00	54.00	55.00	55.00	55.00	1.00	2.00	1.00	2.00	1.00	2.00	1.00
WE5	55.00	55.00	54.00	54.00	55.00	55.00	55.00	2.00	2.00	3.00	2.00	3.00	2.00	2.00
WE6	85.00	84.00	83.00	84.00	84.00	82.00	84.00	5.00	5.00	6.00	5.00	5.00	5.00	9.00
WE7	75.00	73.00	75.00	74.00	74.00	73.00	73.00	3.00	3.00	4.00	3.00	4.00	4.00	5.00
WE9	72.00	71.00	72.00	71.00	70.00	71.00	72.00	6.00	5.00	15.00	6.00	8.00	9.00	7.00
Avg	**53.21**	53.02	53.05	52.98	52.70	52.81	52.65	2.09	2.21	2.63	2.35	2.40	2.70	2.53

Table 5.5: The performance of the Reduced ILS with Θ when $\chi = 0.5$

	EJ							Time(s)						
Inst	0.1	1	5	10	20	30	50	0.1	1	5	10	20	30	50
FR1	26.00	27.00	28.00	28.00	28.00	28.00	28.00	0.00	0.00	0.00	1.00	3.00	2.00	6.00
FR2	38.00	38.00	40.00	40.00	40.00	40.00	40.00	0.00	0.00	4.00	7.00	6.00	12.00	23.00
FR3	34.00	35.00	35.00	35.00	35.00	36.00	36.00	1.00	1.00	2.00	3.00	6.00	9.00	17.00
FR4	55.00	54.00	56.00	56.00	56.00	56.00	56.00	0.00	1.00	7.00	15.00	25.00	45.00	59.00
FR5	62.00	62.00	62.00	64.00	65.00	64.00	65.00	0.00	1.00	7.00	19.00	36.00	64.00	108.00
FR6	73.00	73.00	73.00	73.00	73.00	73.00	73.00	0.00	1.00	2.00	4.00	7.00	11.00	19.00
FR7	60.00	60.00	63.00	61.00	63.00	63.00	64.00	1.00	3.00	17.00	23.00	55.00	59.00	115.00
FR8	87.00	88.00	89.00	91.00	91.00	90.00	90.00	1.00	5.00	24.00	70.00	89.00	132.00	234.00
FR9	66.00	69.00	69.00	69.00	69.00	70.00	69.00	0.00	3.00	13.00	23.00	45.00	79.00	173.00
MO1	13.00	13.00	13.00	13.00	13.00	13.00	13.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00
MO2	36.00	36.00	36.00	36.00	36.00	36.00	36.00	0.00	0.00	1.00	1.00	3.00	3.00	7.00
MO3	45.00	44.00	46.00	46.00	47.00	46.00	46.00	0.00	1.00	5.00	11.00	23.00	21.00	38.00
MO5	65.00	64.00	65.00	66.00	66.00	66.00	66.00	0.00	1.00	7.00	16.00	52.00	45.00	82.00
MO8	88.00	88.00	89.00	89.00	90.00	90.00	90.00	1.00	5.00	24.00	48.00	81.00	124.00	270.00
MO9	72.00	72.00	72.00	73.00	73.00	75.00	74.00	0.00	2.00	9.00	19.00	39.00	67.00	116.00
SA1	24.00	25.00	25.00	24.00	25.00	25.00	24.00	0.00	0.00	0.00	0.00	1.00	2.00	4.00
SA2	24.00	25.00	25.00	25.00	25.00	25.00	26.00	0.00	0.00	1.00	1.00	3.00	4.00	13.00
SA3	43.00	43.00	43.00	43.00	43.00	43.00	43.00	0.00	1.00	2.00	5.00	8.00	12.00	25.00
SA4	48.00	54.00	54.00	56.00	55.00	53.00	53.00	0.00	1.00	6.00	15.00	17.00	42.00	52.00
SA5	63.00	63.00	62.00	62.00	62.00	65.00	65.00	0.00	2.00	8.00	14.00	28.00	145.00	239.00
SA6	70.00	69.00	71.00	71.00	71.00	71.00	72.00	0.00	2.00	11.00	20.00	45.00	64.00	133.00
SA7	82.00	83.00	83.00	83.00	83.00	83.00	83.00	1.00	6.00	18.00	39.00	66.00	96.00	196.00
SA8	83.00	83.00	84.00	85.00	87.00	86.00	85.00	0.00	4.00	21.00	43.00	137.00	121.00	238.00
SA9	70.00	70.00	70.00	71.00	70.00	70.00	70.00	1.00	2.00	11.00	35.00	44.00	63.00	119.00
SU1	22.00	25.00	25.00	25.00	23.00	23.00	25.00	0.00	0.00	0.00	1.00	1.00	1.00	4.00
SU2	21.00	22.00	21.00	21.00	22.00	21.00	21.00	0.00	0.00	0.00	1.00	2.00	2.00	3.00
SU5	55.00	56.00	56.00	56.00	57.00	57.00	57.00	0.00	1.00	6.00	10.00	20.00	24.00	56.00
TH2	36.00	35.00	36.00	37.00	37.00	37.00	37.00	0.00	0.00	1.00	1.00	3.00	5.00	10.00
TH4	53.00	53.00	54.00	54.00	54.00	54.00	54.00	1.00	1.00	6.00	11.00	18.00	24.00	62.00
TH9	67.00	67.00	67.00	67.00	67.00	67.00	67.00	0.00	2.00	14.00	24.00	45.00	70.00	128.00
TU1	26.00	26.00	26.00	26.00	26.00	26.00	26.00	0.00	0.00	0.00	0.00	1.00	1.00	3.00
TU2	33.00	33.00	33.00	33.00	33.00	33.00	33.00	0.00	0.00	1.00	1.00	2.00	3.00	6.00
TU3	39.00	42.00	42.00	42.00	42.00	42.00	42.00	0.00	1.00	2.00	4.00	7.00	10.00	18.00
TU8	81.00	81.00	82.00	82.00	82.00	82.00	82.00	1.00	4.00	19.00	36.00	66.00	105.00	168.00
TU9	66.00	67.00	70.00	68.00	69.00	68.00	69.00	0.00	2.00	36.00	21.00	40.00	62.00	110.00
WE1	13.00	15.00	14.00	13.00	15.00	15.00	15.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
WE2	35.00	35.00	35.00	35.00	36.00	36.00	36.00	0.00	0.00	1.00	1.00	3.00	8.00	7.00
WE3	43.00	43.00	43.00	43.00	43.00	43.00	43.00	0.00	1.00	2.00	5.00	10.00	14.00	26.00
WE4	55.00	55.00	56.00	56.00	56.00	56.00	56.00	0.00	1.00	6.00	10.00	20.00	29.00	58.00
WE5	54.00	55.00	56.00	55.00	56.00	56.00	56.00	1.00	2.00	8.00	15.00	32.00	42.00	107.00
WE6	84.00	84.00	85.00	85.00	85.00	85.00	85.00	0.00	3.00	17.00	26.00	59.00	98.00	181.00
WE7	75.00	75.00	76.00	76.00	76.00	77.00	76.00	0.00	2.00	14.00	26.00	51.00	106.00	132.00
WE9	70.00	72.00	71.00	72.00	72.00	72.00	72.00	1.00	4.00	11.00	22.00	44.00	68.00	131.00
Avg	52.44	53.00	53.51	53.63	53.88	53.88	53.93	0.23	1.53	8.00	15.05	28.91	44.05	81.37

Table 5.6: The performance of the NoReduced ILS-SAA with Θ when $\chi = 0.5$

	EJ							**Time(s)**						
Inst	0.1	1	5	10	20	30	50	0.1	1	5	10	20	30	50
FR1	26.00	28.00	27.00	27.00	28.00	28.00	28.00	0.00	0.00	1.00	1.00	4.00	6.00	7.00
FR2	38.00	39.00	39.00	40.00	40.00	40.00	40.00	0.00	1.00	3.00	7.00	16.00	23.00	34.00
FR3	35.00	36.00	36.00	36.00	36.00	36.00	36.00	0.00	0.00	2.00	4.00	6.00	10.00	16.00
FR4	52.00	53.00	55.00	56.00	57.00	56.00	57.00	0.00	2.00	10.00	17.00	35.00	59.00	104.00
FR5	61.00	62.00	62.00	62.00	62.00	62.00	65.00	1.00	2.00	9.00	18.00	36.00	62.00	136.00
FR6	76.00	77.00	79.00	79.00	79.00	78.00	80.00	0.00	5.00	32.00	48.00	114.00	141.00	283.00
FR7	56.00	63.00	63.00	64.00	64.00	64.00	64.00	1.00	8.00	27.00	69.00	118.00	217.00	306.00
FR8	87.00	89.00	91.00	91.00	90.00	93.00	92.00	1.00	6.00	34.00	71.00	111.00	205.00	359.00
FR9	69.00	69.00	69.00	69.00	69.00	69.00	69.00	0.00	3.00	16.00	29.00	55.00	97.00	154.00
MO1	13.00	13.00	13.00	13.00	13.00	13.00	13.00	0.00	0.00	0.00	1.00	0.00	1.00	1.00
MO2	36.00	36.00	36.00	36.00	36.00	36.00	36.00	0.00	0.00	1.00	1.00	2.00	3.00	5.00
MO3	45.00	45.00	45.00	46.00	46.00	46.00	46.00	0.00	1.00	4.00	14.00	20.00	32.00	54.00
MO5	64.00	64.00	66.00	66.00	65.00	66.00	66.00	1.00	1.00	8.00	18.00	30.00	51.00	84.00
MO8	90.00	89.00	90.00	89.00	90.00	90.00	90.00	0.00	6.00	31.00	62.00	97.00	194.00	250.00
MO9	72.00	72.00	72.00	74.00	74.00	75.00	74.00	1.00	3.00	15.00	29.00	57.00	102.00	138.00
SA1	24.00	25.00	25.00	26.00	25.00	25.00	26.00	0.00	0.00	0.00	2.00	3.00	5.00	7.00
SA2	25.00	23.00	25.00	25.00	25.00	25.00	26.00	0.00	0.00	1.00	1.00	2.00	2.00	5.00
SA3	42.00	43.00	43.00	43.00	43.00	43.00	43.00	0.00	1.00	2.00	5.00	9.00	14.00	24.00
SA4	51.00	53.00	54.00	55.00	55.00	56.00	56.00	0.00	2.00	9.00	18.00	45.00	69.00	105.00
SA5	62.00	64.00	64.00	64.00	64.00	64.00	65.00	1.00	4.00	19.00	27.00	64.00	108.00	292.00
SA6	69.00	69.00	71.00	72.00	71.00	72.00	72.00	0.00	2.00	12.00	25.00	46.00	82.00	133.00
SA7	82.00	83.00	83.00	83.00	83.00	83.00	83.00	1.00	5.00	21.00	42.00	78.00	132.00	211.00
SA8	83.00	83.00	83.00	85.00	85.00	86.00	85.00	0.00	5.00	22.00	54.00	98.00	181.00	281.00
SA9	70.00	70.00	70.00	70.00	70.00	70.00	70.00	1.00	2.00	11.00	22.00	44.00	69.00	118.00
SU1	22.00	22.00	22.00	22.00	22.00	22.00	22.00	0.00	0.00	0.00	0.00	1.00	1.00	2.00
SU2	21.00	21.00	21.00	22.00	22.00	21.00	22.00	0.00	0.00	0.00	1.00	1.00	2.00	3.00
SU5	56.00	56.00	57.00	57.00	57.00	56.00	57.00	0.00	1.00	8.00	19.00	20.00	32.00	56.00
TH2	35.00	36.00	37.00	37.00	37.00	37.00	37.00	0.00	0.00	0.00	2.00	3.00	4.00	6.00
TH4	52.00	53.00	54.00	54.00	54.00	54.00	54.00	0.00	2.00	7.00	17.00	27.00	42.00	66.00
TH9	67.00	67.00	67.00	67.00	67.00	67.00	67.00	0.00	2.00	10.00	21.00	43.00	74.00	116.00
TU1	26.00	26.00	26.00	26.00	26.00	26.00	26.00	0.00	0.00	1.00	1.00	1.00	3.00	4.00
TU2	33.00	33.00	33.00	33.00	33.00	33.00	33.00	0.00	1.00	0.00	1.00	2.00	3.00	5.00
TU3	38.00	41.00	42.00	42.00	42.00	42.00	42.00	0.00	0.00	2.00	3.00	6.00	11.00	17.00
TU8	81.00	81.00	82.00	82.00	82.00	82.00	82.00	1.00	4.00	23.00	45.00	97.00	142.00	248.00
TU9	68.00	67.00	68.00	69.00	69.00	69.00	69.00	0.00	3.00	11.00	25.00	47.00	90.00	160.00
WE1	13.00	16.00	13.00	15.00	13.00	15.00	15.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00
WE2	35.00	36.00	36.00	36.00	35.00	35.00	36.00	0.00	0.00	1.00	2.00	2.00	3.00	5.00
WE3	43.00	43.00	43.00	43.00	43.00	43.00	43.00	0.00	1.00	2.00	4.00	9.00	14.00	23.00
WE4	55.00	55.00	55.00	55.00	56.00	56.00	56.00	0.00	1.00	6.00	12.00	25.00	45.00	58.00
WE5	55.00	55.00	55.00	55.00	55.00	55.00	56.00	1.00	2.00	9.00	20.00	38.00	71.00	119.00
WE6	84.00	85.00	85.00	85.00	85.00	85.00	85.00	0.00	5.00	21.00	36.00	73.00	131.00	230.00
WE7	75.00	75.00	76.00	76.00	76.00	76.00	76.00	1.00	3.00	16.00	31.00	63.00	100.00	164.00
WE9	70.00	72.00	72.00	72.00	72.00	72.00	72.00	0.00	6.00	36.00	67.00	148.00	217.00	332.00
Avg	52.49	53.21	53.60	53.93	53.86	54.00	54.23	0.26	2.09	10.30	20.74	39.44	66.30	109.81

5.4.3 Performance comparison between CPLEX, Greedy algorithm, and ILS-SAA with single scenario

This section presents the results of solving the SAA model with a single scenario. The performance of the Greedy algorithm, CPLEX, Reduced ILS-SAA, and NoReduced ILS-SAA is compared in Tables 5.7 and 5.8. The first four columns in Table 5.7 show the instance's name, number of customers in both C_1 and C_2, number of vehicles in T_1, and number of vehicles in T_2. The groups Greedy, CPLEX, Reduced ILS-SAA, and NoReduced ILS-SAA report the results obtained from the Greedy algorithm, CPLEX using the SAA model, Reduced ILS-SAA, and NoReduced ILS-SAA, respectively. Each column 1st reports the number of customers that were allocated to vehicles in T_1 and each column Gap(%) reports the optimality gap. In addition, the third column under the group Greedy reports the time required by the first phase, whereas the last column reports the time required by the second phase. Please note that the objective values in columns EJ are in bold if the algorithm obtained the best values compared with other algorithms.

Although the Greedy algorithm uses CPLEX for the first phase, without considering the uncertainty, the problem size for the first phase is relatively small. Therefore, the first phase can still produce good solutions. In addition, the superior performance of the algorithm used in the second phase has been proven in Gu et al. (2021). As shown in Table 5.7, the Greedy algorithm outperforms CPLEX for 39 out of 43 instances with respect to the objective value within the same amount of computational time. Comparing the performance between the Greedy algorithm, Reduced ILS-SAA, and NoReduced ILS-SAA. The Reduced ILS-SAA outperforms the Greedy algorithm for 17 out of 43 instances and produces the same objective values for 19 out of 43 instances. The NoReduced ILS-SAA outperforms the Greedy algorithm for 20 out of 43 instances and produces the same objective values for 18 out of 43 instances. In Table 5.8, with a total of 36 instances, the objective values obtained by the NoReduecd ILS-SAA are better than the values obtained by the Reduced ILS-SAA on 7 instances and are the same on 35 instances. This draws the conclusion that the NoReduced ILS-SAA has better performance than the Reduced ILS-SAA.

The difference between the NoReduced ILS and Reduced ILS-SAA is that the Reduced ILS applies the neighbourhood reduction technique described in Chapter 4. The idea of the neighbourhood reduction technique in Chapter 4 is to find solutions that have better value on the objective function compared with the current best-known feasible solution. The technique ignores those solutions that have objective values worse than the current best-known feasible solution thereby increasing the ability to find good solutions as well as reducing the computational time. In Chapter 4, the effectiveness of the neighbourhood reduction technique has been demonstrated by extensive computational experiments. This contradiction may caused by the difference in the objective functions of the problem studied in this chapter and the problem studied in Chapter 4. The objective of the studied problem in Chapter 4 is to maximise the total number of served customers. The objective value is recorded as an integer number which means a solution is better if it serves at least one more customer than the current best-known feasible solution. In contrast, the objective of the studied problem in this chapter is to maximise the expected total number of served customers and the objective value is recorded as a real number. This difference makes the neighbourhood reduction technique unsuitable for the Reduced ILS-SAA. The reason could be that for the objective function in Chapter 4, many solutions may have the same objective value. Hence, the algorithm in Chapter 4 has many candidate solutions to find feasible solutions with 1 or more number of the served customers than the current best-known feasible solution. It is not the case for the objective function of the problem in this Chapter because each solution of the problem may have its unique objective value.

In terms of computational time, the Reduced ILS-SAA and NoReduced ILS-SAA are incomparably better compared with the Greedy algorithm and CPLEX. With the neighbourhood reduction technique, the Reduced ILS-SAA does require a shorter computational time compared with the NoReduced ILS-SAA. But taking into account the excellent solution quality produced by the NoReduced ILS-SAA, the computational time for the NoReduced ILS-SAA is acceptable.

5.4.4 Performance comparison between CPLEX, Greedy algorithm, and ILS-SAA with multiple scenarios

This section presents the results of solving the SAA model with 5 scenarios and 10 scenarios. Since the results in Section 5.4.3 have shown that the NoReduced ILS-SAA outperforms the Reduced ILS-SAA, this section only reports the results obtained from the NoReduced ILS-SAA.

Table 5.7: Comparison between the performance of Greedy algorithm, CPLEX, Reduced ILS-SAA, NoReduced ILS-SAA with one scenario (part1)

				Greedy					CPLEX			
Inst	$\|C\|$	$\|T_1\|$	$\|T_2\|$	1st	Gap(%)	Time(s)	EJ	Time(s)	1st	EJ	Gap(%)	Time(s)
FR1	15	1	2	9.00	0.00	107.00	27.00	0.00	6.00	26.00	15.38	18346.00
FR2	25	2	2	10.00	150.00	21600.00	**41.00**	1.00	8.00	33.00	45.45	21600.00
FR3	25	2	3	23.00	8.70	21600.00	**36.00**	0.00	20.00	34.00	23.53	21600.00
FR4	35	2	3	21.00	66.67	21600.00	55.00	2.00	14.00	42.00	54.76	21600.00
FR5	35	3	4	29.00	20.69	21600.00	64.00	2.00	22.00	57.00	22.81	21600.00
FR6	45	3	4	29.00	50.00	21600.00	**80.00**	5.00	23.00	69.00	30.43	21600.00
FR7	45	4	3	14.00	221.43	21600.00	62.00	7.00	11.00	49.00	79.59	21600.00
FR8	50	5	4	44.00	13.64	21600.00	90.00	7.00	33.00	77.00	29.87	21600.00
FR9	40	4	4	29.00	37.93	21600.00	**69.00**	2.00	26.00	63.00	25.40	21600.00
MO1	15	1	1	7.00	0.00	6.00	13.00	0.00	6.00	13.00	100.00	21600.00
MO2	25	2	1	20.00	25.00	15308.00	34.00	0.00	17.00	30.00	63.33	13078.00
MO3	25	2	3	19.00	31.58	21600.00	**47.00**	1.00	17.00	45.00	8.89	21600.00
MO5	35	3	3	30.00	16.67	21600.00	64.00	1.00	24.00	61.00	13.11	21600.00
MO8	50	4	4	42.00	19.05	21600.00	88.00	5.00	42.00	82.00	18.29	21600.00
MO9	40	4	4	34.00	14.71	21600.00	72.00	2.00	21.00	62.00	29.03	21600.00
SA1	15	1	2	4.00	0.00	5.00	**26.00**	0.00	4.00	25.00	4.00	21600.00
SA2	25	2	1	14.00	71.43	11587.00	**26.00**	0.00	13.00	22.00	90.91	21600.00
SA3	25	2	3	22.00	13.64	20531.00	**43.00**	0.00	15.00	41.00	12.20	21600.00
SA4	35	2	3	18.00	88.89	9342.00	54.00	3.00	13.00	45.00	55.56	21600.00
SA5	35	3	4	19.00	84.21	21600.00	**66.00**	5.00	13.00	59.00	18.64	21600.00
SA6	45	3	3	38.00	18.42	18169.00	68.00	2.00	30.00	58.00	43.10	21600.00
SA7	45	4	4	38.00	18.42	21600.00	**83.00**	3.00	26.00	72.00	20.83	21600.00
SA8	50	4	3	38.00	31.58	21600.00	80.00	11.00	33.00	64.00	54.69	21600.00
SA9	40	4	4	34.00	17.65	15642.00	**70.00**	2.00	30.00	67.00	19.40	21600.00
SU1	15	1	2	14.00	7.14	21600.00	**25.00**	0.00	9.00	**25.00**	0.00	14012.00
SU2	25	2	1	16.00	56.25	21600.00	21.00	0.00	15.00	20.00	100.00	21600.00
SU5	35	2	3	32.00	9.38	21600.00	**57.00**	1.00	28.00	54.00	27.66	21600.00
TH2	25	2	1	20.00	20.00	21600.00	35.00	0.00	20.00	34.00	29.41	6097.00
TH4	35	2	3	24.00	20.83	21600.00	**54.00**	2.00	18.00	50.00	30.00	21600.00
TH9	40	4	4	34.00	11.76	21600.00	**67.00**	1.00	27.00	62.00	29.03	21600.00
TU1	15	1	2	11.00	0.00	78.00	26.00	0.00	7.00	26.00	3.85	21600.00
TU2	25	2	1	20.00	5.00	21600.00	32.00	0.00	19.00	31.00	29.03	21600.00
TU3	25	2	2	22.00	9.09	10507.00	**42.00**	1.00	18.00	41.00	9.76	13813.00
TU8	50	5	4	43.00	9.30	21600.00	80.00	7.00	32.00	67.00	49.25	21600.00
TU9	40	4	4	33.00	21.21	21600.00	**69.00**	4.00	22.00	63.00	25.40	21600.00
WE1	15	1	1	9.00	0.00	4353.00	15.00	0.00	8.00	**16.00**	62.50	21600.00
WE2	25	2	1	22.00	13.64	21600.00	35.00	0.00	17.00	29.00	55.17	21600.00
WE3	25	2	3	19.00	26.32	21600.00	**43.00**	1.00	13.00	40.00	15.00	21600.00
WE4	35	3	3	29.00	17.24	21600.00	55.00	1.00	25.00	48.00	41.67	21600.00
WE5	35	3	4	29.00	20.69	21600.00	**57.00**	3.00	18.00	45.00	55.56	21600.00
WE6	45	4	4	38.00	18.42	6823.00	83.00	4.00	29.00	70.00	28.57	21600.00
WE7	45	4	3	40.00	12.50	5111.00	75.00	2.00	39.00	69.00	27.54	21600.00
WE9	40	4	4	16.00	150.00	21600.00	**72.00**	7.00	9.00	54.00	46.30	21600.00
Avg				24.56	33.70	17301.60	53.51	2.21	19.53	47.44	35.93	20608.05

Table 5.8: Comparison between the performance of Greedy algorithm, CPLEX, ILS-SAA with one scenario (part2)

				Reduced ILS-SAA			NoReduced ILS-SAA		
Inst	$\|C\|$	$\|T_1\|$	$\|T_2\|$	1st	EJ	Time(s)	1st	EJ	Time(s)
FR1	15	1	2	6.00	**28.00**	6.00	6.00	**28.00**	7.00
FR2	25	2	2	10.00	40.00	23.00	10.00	40.00	34.00
FR3	25	2	3	19.00	**36.00**	17.00	21.00	**36.00**	16.00
FR4	35	2	3	24.00	56.00	59.00	27.00	**57.00**	104.00
FR5	35	3	4	26.00	**65.00**	108.00	28.00	**65.00**	136.00
FR6	45	3	4	31.00	73.00	19.00	30.00	**80.00**	283.00
FR7	45	4	3	17.00	**64.00**	115.00	18.00	**64.00**	306.00
FR8	50	5	4	42.00	90.00	234.00	47.00	**92.00**	359.00
FR9	40	4	4	29.00	**69.00**	173.00	35.00	**69.00**	154.00
MO1	15	1	1	7.00	13.00	2.00	7.00	13.00	1.00
MO2	25	2	1	22.00	**36.00**	7.00	22.00	**36.00**	5.00
MO3	25	2	3	13.00	46.00	38.00	18.00	46.00	54.00
MO5	35	3	3	25.00	**66.00**	82.00	30.00	**66.00**	84.00
MO8	50	4	4	48.00	**90.00**	270.00	49.00	**90.00**	250.00
MO9	40	4	4	33.00	**74.00**	116.00	36.00	**74.00**	138.00
SA1	15	1	2	3.00	24.00	4.00	4.00	**26.00**	7.00
SA2	25	2	1	14.00	**26.00**	13.00	14.00	**26.00**	5.00
SA3	25	2	3	16.00	**43.00**	25.00	19.00	**43.00**	24.00
SA4	35	2	3	16.00	53.00	52.00	20.00	**56.00**	105.00
SA5	35	3	4	15.00	65.00	239.00	16.00	65.00	292.00
SA6	45	3	3	39.00	**72.00**	133.00	42.00	**72.00**	133.00
SA7	45	4	4	38.00	**83.00**	196.00	38.00	**83.00**	211.00
SA8	50	4	3	45.00	**85.00**	238.00	45.00	**85.00**	281.00
SA9	40	4	4	32.00	**70.00**	119.00	36.00	**70.00**	118.00
SU1	15	1	2	11.00	**25.00**	4.00	14.00	22.00	2.00
SU2	25	2	1	16.00	21.00	3.00	16.00	**22.00**	3.00
SU5	35	2	3	29.00	**57.00**	56.00	31.00	**57.00**	56.00
TH2	25	2	1	22.00	**37.00**	10.00	22.00	**37.00**	6.00
TH4	35	2	3	23.00	**54.00**	62.00	23.00	**54.00**	66.00
TH9	40	4	4	35.00	**67.00**	128.00	35.00	**67.00**	116.00
TU1	15	1	2	10.00	26.00	3.00	9.00	26.00	4.00
TU2	25	2	1	20.00	**33.00**	6.00	20.00	**33.00**	5.00
TU3	25	2	2	17.00	**42.00**	18.00	19.00	**42.00**	17.00
TU8	50	5	4	45.00	**82.00**	168.00	44.00	**82.00**	248.00
TU9	40	4	4	34.00	**69.00**	110.00	32.00	**69.00**	160.00
WE1	15	1	1	8.00	15.00	1.00	8.00	15.00	1.00
WE2	25	2	1	22.00	**36.00**	7.00	23.00	**36.00**	5.00
WE3	25	2	3	17.00	**43.00**	26.00	17.00	**43.00**	23.00
WE4	35	3	3	30.00	**56.00**	58.00	30.00	**56.00**	58.00
WE5	35	3	4	23.00	56.00	107.00	29.00	56.00	119.00
WE6	45	4	4	42.00	**85.00**	181.00	43.00	**85.00**	230.00
WE7	45	4	3	41.00	**76.00**	132.00	40.00	**76.00**	164.00
WE9	40	4	4	17.00	**72.00**	131.00	17.00	**72.00**	332.00
Avg				24.00	53.93	81.37	25.35	**54.23**	109.81

Table 5.9 compares the performance of the Greedy algorithm, CPLEX, and NoReduced MSILS-SAA for the SAA model with 5 scenarios. Table 5.10 compares the performance of these methods for the SAA with 10 scenarios. Since the NoReduced MSILS-SAA are comprised of a number of applications of the NoReduced ILS-SAA, in these tables, the best objective value obtained by these applications is reported. The column titled "Best 1st" reports the number of customers allocated to vehicles in T_1, whereas the column titled "Best EJ" reports the expected total number of allocated customers.

With respect to the solution quality obtained for the SAA model with 5 scenarios, the NoReduced MSILS-SAA outperforms both the Greedy algorithm and CPLEX for 34 out of 43 instances. For the SAA model with 10 scenarios, the NoReduced MSILS-SAA outperforms both the Greedy algorithm and CPLEX for 37 of 43 instances. There is one instance that CPLEX cannot even obtain a feasible solution. In terms of computational time, the NoReduced MSILS-SAA is better than both the Greedy algorithm and CPLEX. Both tables show that the stochastic programming approach can be a useful tool for the problem of preloading under uncertainty.

5.4.5 Stochastic analysis on the first-stage solutions

This subsection presents the results for the stochastic analysis on first-stage solutions obtained from CPLEX for solving SAA models with 1 scenario, 5 scenarios, and 10 scenarios; the Greedy algorithm; the NoReduced ILS-SAA for solving the SAA model with 1 scenario; and the NoReduced MSILS-SAA for solving the SAA models with 5 scenarios and 10 scenarios. The groups CPLESSAA1, CPLEXSAA5, and CPLEXSAA10 report the results obtained from the first-stage solution produced by CPLEX from solving SAA models with 1 scenario, 5 scenarios, and 10 scenarios. The group NoReduced ILS-SAA1 presents the result obtained from the first-stage solutions produced by NoReduced ILS-SAA for solving the SAA model with 1 scenario whereas the groups NoReduced MSILS-SAA5 and NoReduced MSILS-SAA10 present the results obtained from the first-stage solutions produced by NoReduced MSILS-SAA for solving the SAA models with 5 scenarios and 10 scenarios. The columns $Time^*(s)$ report the total time used to evaluate the first-stage solution.

In Table 5.11, the expected total number of allocated customers for the first-stage solutions obtained from CPLEX becomes worse when the number of scenarios increases

Table 5.9: Comparison between the performance of Greedy algorithm, CPLEX, NoReduced MSILS-SAA with 5 scenarios

				Greedy			**CPLEX**				**NoReduced MSILS-SAA**		
Inst	$\|C\|$	$\|T_1\|$	$\|T_2\|$	1st	EJ	Time(s)	1st	EJ	Gap(%)	Time(s)	Best 1st	Best EJ	Time(s)
FR1	15	1	2	9.00	26.20	107.00	2.00	24.00	20.83	21600.00	9.00	**26.40**	2.00
FR2	25	2	2	10.00	39.60	21604.00	7.00	29.60	62.84	21600.00	10.00	**39.80**	14.00
FR3	25	2	3	23.00	37.60	21601.00	9.00	27.80	64.03	21600.00	22.00	**37.80**	9.00
FR4	35	2	3	21.00	55.20	21616.00	8.00	24.20	184.30	21600.00	25.00	**57.60**	46.00
FR5	35	3	4	29.00	**63.60**	21611.00	14.00	49.00	42.04	21600.00	26.00	63.20	43.00
FR6	45	3	4	29.00	76.60	21630.00	2.00	17.40	417.24	21600.00	33.00	**78.60**	113.00
FR7	45	4	3	14.00	65.60	21652.00	5.00	30.20	192.72	21600.00	19.00	**68.20**	119.00
FR8	50	5	4	44.00	89.60	21624.00	13.00	31.20	216.67	21600.00	48.00	**92.00**	130.00
FR9	40	4	4	29.00	65.40	21609.00	19.00	46.80	64.10	21600.00	35.00	**66.20**	57.00
MO1	15	1	1	7.00	12.00	6.00	7.00	11.80	126.27	21600.00	6.00	**12.40**	1.00
MO2	25	2	1	20.00	31.20	15309.00	13.00	22.60	112.39	21600.00	22.00	**33.60**	0.00
MO3	25	2	3	19.00	44.60	21604.00	3.00	25.80	86.05	21600.00	19.00	**44.80**	19.00
MO5	35	3	3	30.00	61.40	21604.00	16.00	52.20	33.24	21600.00	28.00	**62.80**	35.00
MO8	50	4	4	42.00	89.20	21630.00	29.00	73.80	31.98	21600.00	46.00	**90.40**	116.00
MO9	40	4	4	34.00	71.80	21613.00	12.00	58.40	36.99	21600.00	34.00	**74.00**	57.00
SA1	15	1	2	4.00	25.60	6.00	2.00	24.00	15.00	21600.00	4.00	**26.20**	3.00
SA2	25	2	1	14.00	**26.40**	11588.00	11.00	21.00	129.05	21600.00	14.00	25.80	2.00
SA3	25	2	3	22.00	44.80	20532.00	10.00	41.00	18.54	21225.00	20.00	**45.00**	12.00
SA4	35	2	3	18.00	57.60	9361.00	11.00	40.60	68.47	21600.00	19.00	**58.80**	49.00
SA5	35	3	4	19.00	**65.20**	21618.00	4.00	46.80	49.57	21600.00	19.00	**65.20**	105.00
SA6	45	3	3	38.00	69.60	18184.00	30.00	55.20	63.04	21600.00	43.00	**75.00**	59.00
SA7	45	4	4	38.00	83.40	21620.00	16.00	55.40	60.29	21600.00	40.00	**84.00**	105.00
SA8	50	4	3	38.00	76.80	21633.00	32.00	63.00	49.22	21600.00	48.00	**82.60**	88.00
SA9	40	4	4	34.00	70.60	15653.00	14.00	53.80	48.33	21600.00	35.00	**72.20**	63.00
SU1	15	1	2	14.00	**26.80**	21600.00	6.00	23.00	22.61	19440.00	14.00	23.00	2.00
SU2	25	2	1	16.00	22.60	21600.00	14.00	19.80	112.12	13999.00	17.00	**23.80**	0.00
SU5	35	2	3	32.00	60.00	21608.00	26.00	51.20	34.37	21600.00	33.00	**60.60**	32.00
TH2	25	2	1	20.00	36.00	21600.00	13.00	23.20	93.10	21600.00	22.00	**38.40**	0.00
TH4	35	2	3	24.00	52.60	21607.00	16.00	44.60	44.39	21600.00	22.00	**53.80**	35.00
TH9	40	4	4	34.00	65.60	21612.00	17.00	33.80	136.69	21600.00	35.00	**66.00**	58.00
TU1	15	1	2	11.00	25.40	79.00	2.00	23.40	17.09	3790.00	9.00	**25.60**	2.00
TU2	25	2	1	20.00	34.60	21600.00	16.00	28.20	50.35	21600.00	20.00	**35.20**	1.00
TU3	25	2	2	22.00	**42.80**	10508.00	6.00	32.00	47.5	21600.00	22.00	**42.80**	6.00
TU8	50	5	4	43.00	84.60	21616.00	22.00	61.40	62.87	21600.00	44.00	**86.40**	98.00
TU9	40	4	4	33.00	70.00	21611.00	13.00	38.40	107.29	21600.00	32.00	**71.80**	64.00
WE1	15	1	1	9.00	**16.20**	4353.00	5.00	12.60	123.63	21600.00	8.00	**16.20**	0.00
WE2	25	2	1	22.00	**35.20**	21600.00	16.00	28.80	62.50	21600.00	22.00	**35.20**	2.00
WE3	25	2	3	19.00	43.40	21603.00	3.00	36.60	36.07	21600.00	17.00	**44.20**	10.00
WE4	35	3	3	29.00	56.20	21606.00	13.00	42.60	63.38	21600.00	30.00	**58.00**	27.00
WE5	35	3	4	29.00	**59.20**	21609.00	13.00	47.60	44.96	21600.00	28.00	59.00	46.00
WE6	45	4	4	38.00	81.00	6840.00	9.00	56.40	59.22	21600.00	43.00	**83.00**	80.00
WE7	45	4	3	40.00	72.20	5122.00	34.00	63.40	37.22	21600.00	41.00	**73.80**	49.00
WE9	40	4	4	16.00	**72.20**	21637.00	6.00	52.00	52.69	21600.00	17.00	**72.20**	125.00
Avg				24.56	53.63	17312.23	12.53	38.25	0.79	20950.09	25.58	**54.69**	43.81

Table 5.10: Comparison between the performance of Greedy algorithm, CPLEX, NoReduced MSILS-SAA with 10 scenarios

				Greedy			**CPLEX**				**NoReduced MSILS-SAA**		
Inst	$\|C\|$	$\|T_1\|$	$\|T_2\|$	1st	EJ	Time(s)	1st	EJ	Gap(%)	Time(s)	1st	EJ	Time(s)
FR1	15	1	2	9.00	26.00	108.00	2.00	23.10	23.38	21600.00	9.00	**26.40**	7.00
FR2	25	2	2	10.00	41.30	21608.00	4.00	30.40	58.53	21600.00	10.00	**41.60**	58.00
FR3	25	2	3	23.00	37.20	21603.00	2.00	14.10	245.88	21600.00	22.00	**37.30**	29.00
FR4	35	2	3	21.00	56.30	21632.00	5.00	17.00	307.06	21600.00	27.00	**58.50**	174.00
FR5	35	3	4	29.00	63.80	21620.00	5.00	14.30	386.01	21600.00	26.00	**64.40**	166.00
FR6	45	3	4	29.00	76.40	21655.00	6.00	33.30	170.27	21600.00	33.00	**78.50**	443.00
FR7	45	4	3	14.00	65.30	21708.00	5.00	20.50	326.83	21600.00	19.00	**68.20**	433.00
FR8	50	5	4	44.00	91.10	21649.00	10.00	16.60	498.19	21600.00	48.00	**92.90**	533.00
FR9	40	4	4	29.00	65.50	21618.00	23.00	55.20	38.41	21600.00	35.00	**66.40**	253.00
MO1	15	1	1	7.00	12.40	6.00	6.00	11.40	135.86	21600.00	7.00	**12.60**	1.00
MO2	25	2	1	20.00	30.50	15309.00	11.00	19.80	142.42	21600.00	22.00	**32.70**	4.00
MO3	25	2	3	19.00	**44.40**	21608.00	2.00	22.80	114.04	21600.00	18.00	44.30	74.00
MO5	35	3	3	30.00	61.20	21609.00	15.00	41.90	65.87	21600.00	31.00	**63.10**	133.00
MO8	50	4	4	42.00	89.80	21655.00	8.00	31.80	206.92	21600.00	49.00	**91.10**	464.00
MO9	40	4	4	34.00	72.60	21623.00	5.00	37.00	115.95	21600.00	34.00	**74.50**	241.00
SA1	15	1	2	4.00	26.20	7.00	1.00	21.80	29.36	21600.00	4.00	**26.80**	14.00
SA2	25	2	1	14.00	**25.60**	11588.00	10.00	17.90	165.36	21600.00	14.00	25.30	5.00
SA3	25	2	3	22.00	44.50	20534.00	5.00	37.00	32.43	16556.00	21.00	**44.70**	50.00
SA4	35	2	3	18.00	56.90	9373.00	7.00	22.80	202.19	21600.00	21.00	**58.50**	195.00
SA5	35	3	4	19.00	65.60	21639.00	0.00	31.70	120.82	21600.00	19.00	**65.80**	442.00
SA6	45	3	3	38.00	69.80	18198.00	18.00	33.30	168.47	21600.00	44.00	**75.30**	239.00
SA7	45	4	4	38.00	83.50	21646.00	7.00	25.00	257.60	21600.00	40.00	**83.90**	435.00
SA8	50	4	3	38.00	77.00	21666.00	6.00	13.10	624.63	21600.00	47.00	**83.40**	372.00
SA9	40	4	4	34.00	72.60	15669.00	3.00	44.40	79.05	21600.00	35.00	**73.40**	272.00
SU1	15	1	2	14.00	**27.40**	21600.00	2.00	18.30	55.74	21600.00	14.00	23.20	9.00
SU2	25	2	1	16.00	23.20	21601.00	10.00	15.60	174.79	21600.00	17.00	**24.10**	3.00
SU5	35	2	3	32.00	60.70	21615.00	0	0	N/A	21600.00	33.00	**61.30**	125.00
TH2	25	2	1	20.00	35.70	21600.00	19.00	30.90	45.31	21600.00	22.00	**38.10**	5.00
TH4	35	2	3	24.00	52.60	21614.00	13.00	30.50	113.10	21600.00	22.00	**53.60**	132.00
TH9	40	4	4	34.00	65.10	21624.00	16.00	28.00	185.71	21600.00	36.00	**65.70**	235.00
TU1	15	1	2	11.00	24.90	79.00	1.00	22.40	19.20	21600.00	9.00	**25.10**	14.00
TU2	25	2	1	20.00	34.20	21601.00	8.00	29.30	47.51	21600.00	20.00	**34.90**	5.00
TU3	25	2	2	22.00	**42.90**	10509.00	8.00	32.00	49.38	21600.00	22.00	**42.90**	32.00
TU8	50	5	4	43.00	84.00	21631.00	10.00	26.30	279.47	21600.00	45.00	**85.80**	395.00
TU9	40	4	4	33.00	70.20	21626.00	7.00	19.10	316.62	21600.00	35.00	**71.60**	256.00
WE1	15	1	1	9.00	16.20	4353.00	5.00	13.10	113.74	21600.00	9.00	**17.00**	1.00
WE2	25	2	1	22.00	35.20	21601.00	16.00	27.50	72.21	21600.00	23.00	**36.00**	7.00
WE3	25	2	3	19.00	43.40	21607.00	2.00	35.80	37.71	21600.00	20.00	**44.20**	45.00
WE4	35	3	3	29.00	56.40	21613.00	5.00	9.10	664.84	21600.00	30.00	**58.10**	114.00
WE5	35	3	4	29.00	**61.40**	21616.00	10.00	29.30	135.84	21600.00	27.00	**61.40**	231.00
WE6	45	4	4	38.00	81.50	6861.00	5.00	24.20	270.66	21600.00	40.00	**83.50**	406.00
WE7	45	4	3	40.00	73.00	5133.00	24.00	44.50	93.48	21600.00	41.00	**74.50**	222.00
WE9	40	4	4	16.00	**73.50**	21677.00	4.00	45.30	75.50	21600.00	18.00	**73.50**	549.00
Avg				24.56	53.88	17323.07	7.70	25.99	173.01	21482.70	26.00	**54.98**	181.93

in the SAA model. This is because as the number of scenarios increases the problem size also increases. CPLEX is not able to construct good first-stage solutions when the problem size becomes large. Furthermore, since CPLEX couldn't produce first-stage solutions with a desired number of allocated customers, this becomes a computational burden for the iterated local search in Gu et al. (2021) to compute the second-stage solutions. This explains why the values in column $Time^*(s)$ are larger compared with the corresponding values obtained from the Greedy algorithm, NoReduced ILS-SAA, and NoReduced MSILS-SAA.

The total expected number of allocated customers of the first-stage solutions obtained from ILS-SAA becomes better when the ILS-SAA solves an SAA model with a larger number of scenarios. For 31 out of 43 instances, even the first-stage solutions obtained from NoReduced ILS-SAA for solving the SAA model with 1 scenario are better than the first-stage solutions obtained from the Greedy algorithm in terms of the total expected number of allocated customers. For the first-stage solutions obtained from the Greedy algorithm, they are obtained by allocating as many as possible for the customer in C_1 ignoring what will happen for customers in C_2. From the stochastic analysis, it has been observed that this strategy may not always be the ideal strategy. For example, for 7 out of 43 instances, the MSILS-SAA10 obtained the first-stage solution with the same number of allocated customers in C_1 as the first-stage solution obtained by the Greedy algorithm and produced a higher expected total number of allocated customers. This is an indication that the choices of which customers to allocate in preloading are important and do have a big impact when the customers in C_2 are revealed. Therefore, it is worth trying the ILS-SAA for an SAA model.

5.5 Conclusion

This chapter studies Simultaneous Pickup and Delivery with Preloading under Uncertainty. This problem is formulated as a two-stage stochastic program and solved by a sample average approximation approach. An Iterated local search, extended from the Lagrangian ILS, is designed for the sample average approximation approaches, Named ILS-SAA. This algorithm is tested on benchmark instances derived from real historical data. The results of computational experiments have shown that ILS-SAA outperforms

Table 5.11: Stochastic analysis on first stage solutions obtained from CPLEX, the Greedy algorithm, NoReduced ILS-SAA, and NoReduced MSILS-SAA (part1)

				CPLEXSAA1			CPLEXSAA5			CPLEXSAA10		
Inst	$\|C\|$	$\|T_1\|$	$\|T_2\|$	1st	EJ	$Time^*(s)$	1st	EJ	$Time^*(s)$	1st	EJ	$Time^*(s)$
FR1	15	1	2	6	25.49	8.00	2	25.54	15.00	2	25.60	14.00
FR2	25	2	2	8	38.48	54.00	7	38.83	58.00	4	37.04	73.00
FR3	25	2	3	20	36.97	21.00	9	31.14	77.00	2	25.53	145.00
FR4	35	2	3	14	53.69	219.00	8	50.99	281.00	5	49.79	337.00
FR5	35	3	4	22	61.68	130.00	14	60.40	233.00	5	60.65	382.00
FR6	45	3	4	23	75.52	362.00	2	68.10	1009.00	6	70.38	876.00
FR7	45	4	3	11	62.44	461.00	5	58.78	575.00	5	58.19	566.00
FR8	50	5	4	33	87.03	417.00	13	80.20	1011.00	10	76.39	1137.00
FR9	40	4	4	26	64.70	128.00	19	63.29	187.00	23	63.56	145.00
MO1	15	1	1	6	12.23	1.00	7	**12.64**	2.00	6	11.81	2.00
MO2	25	2	1	17	28.22	3.00	13	25.40	4.00	11	23.65	4.00
MO3	25	2	3	17	45.06	31.00	3	42.01	103.00	2	42.43	110.00
MO5	35	3	3	24	61.61	93.00	16	58.28	172.00	15	57.27	172.00
MO8	50	4	4	42	91.87	230.00	29	89.33	491.00	8	81.50	1417.00
MO9	40	4	4	21	70.18	245.00	12	70.05	409.00	5	68.09	552.00
SA1	15	1	2	4	26.85	12.00	2	26.17	18.00	1	25.22	19.00
SA2	25	2	1	13	25.13	3.00	11	23.40	4.00	10	23.01	3.00
SA3	25	2	3	15	44.57	36.00	10	44.17	58.00	5	43.14	94.00
SA4	35	2	3	13	55.17	280.00	11	55.21	286.00	7	52.71	328.00
SA5	35	3	4	13	65.40	283.00	4	63.06	486.00	0	61.87	598.00
SA6	45	3	3	30	69.11	253.00	30	66.37	237.00	18	60.35	452.00
SA7	45	4	4	26	80.24	449.00	16	76.49	805.00	7	73.15	1165.00
SA8	50	4	3	33	74.44	421.00	32	75.17	417.00	6	58.54	894.00
SA9	40	4	4	30	73.34	183.00	14	71.13	455.00	3	67.35	802.00
SU1	15	1	2	9	18.36	4.00	6	17.62	5.00	2	13.97	11.00
SU2	25	2	1	15	22.61	4.00	14	21.55	4.00	10	18.50	5.00
SU5	35	2	3	28	60.61	73.00	26	59.77	84.00	0	47.90	398.00
TH2	25	2	1	20	35.82	4.00	13	31.29	6.00	19	34.41	2.00
TH4	35	2	3	18	**54.05**	100.00	16	53.21	114.00	13	51.43	137.00
TH9	40	4	4	27	63.55	225.00	17	59.76	426.00	16	57.28	456.00
TU1	15	1	2	7	24.49	6.00	2	24.49	14.00	1	24.36	22.00
TU2	25	2	1	19	34.45	3.00	16	32.83	4.00	8	27.34	6.00
TU3	25	2	2	18	42.10	22.00	6	36.68	62.00	8	37.45	57.00
TU8	50	5	4	32	80.72	418.00	22	76.78	693.00	10	68.43	930.00
TU9	40	4	4	22	69.93	273.00	13	69.47	465.00	7	66.49	622.00
WE1	15	1	1	8	16.33	0.00	5	13.63	1.00	5	13.68	1.00
WE2	25	2	1	17	31.06	5.00	16	31.17	4.00	16	30.53	4.00
WE3	25	2	3	13	42.51	58.00	3	43.01	134.00	2	42.46	147.00
WE4	35	3	3	25	55.59	106.00	13	51.53	259.00	5	47.31	397.00
WE5	35	3	4	18	58.19	173.00	13	57.15	285.00	10	57.16	332.00
WE6	45	4	4	29	78.67	358.00	9	72.36	850.00	5	71.73	1106.00
WE7	45	4	3	39	72.47	96.00	34	70.07	130.00	24	65.52	218.00
WE9	40	4	4	9	70.97	536.00	6	69.52	601.00	4	68.66	660.00
				19.53	52.60	157.84	12.53	50.42	268.23	7.70	47.95	367.40

Table 5.12: Stochastic analysis on first stage solutions obtained from CPLEX, the Greedy algorithm, NoReduced ILS-SAA, and NoReduced MSILS-SAA (part2)

	Greedy			**NoReduced ILS-SAA1**			**NoReduced MSILS-SAA5**			**NoReduced MSILS-SAA10**		
Inst	1st	EJ	$Time^*(s)$	1st	EJ	$Time^*(s)$	1st	EJ	$Time^*(s)$	1st	EJ	$Time^*(s)$
FR1	9	25.66	5.00	6	**26.44**	9.00	9	25.64	5.00	9	26.00	5.00
FR2	10	41.00	49.00	10	41.33	48.00	10	**41.46**	44.00	10	41.21	52.00
FR3	23	**37.32**	15.00	21	37.24	19.00	22	37.28	16.00	22	37.27	17.00
FR4	21	56.98	133.00	27	**59.54**	88.00	25	59.01	110.00	27	59.45	88.00
FR5	29	63.63	87.00	28	**64.60**	95.00	26	64.31	113.00	26	64.24	112.00
FR6	29	77.21	324.00	30	78.52	270.00	33	**78.61**	223.00	33	78.41	225.00
FR7	14	64.83	442.00	18	67.37	367.00	19	**67.86**	338.00	19	67.74	343.00
FR8	44	89.21	257.00	47	90.37	236.00	48	**91.30**	211.00	48	91.27	216.00
FR9	29	65.59	100.00	35	**66.55**	69.00	35	66.53	63.00	35	66.39	64.00
MO1	7	**12.61**	1.00	7	12.56	1.00	6	12.24	1.00	7	12.59	1.00
MO2	20	31.04	3.00	22	**33.11**	2.00	22	32.99	2.00	22	33.00	3.00
MO3	19	**45.36**	28.00	18	45.23	31.00	19	45.23	27.00	18	45.24	29.00
MO5	30	60.80	56.00	30	**62.87**	69.00	28	61.69	71.00	31	62.59	56.00
MO8	42	91.57	250.00	49	92.26	158.00	46	92.46	193.00	49	**92.54**	156.00
MO9	34	72.36	110.00	36	73.47	108.00	34	**74.17**	126.00	34	74.16	124.00
SA1	4	26.45	15.00	4	26.82	13.00	4	**26.85**	12.00	4	26.83	12.00
SA2	14	**25.97**	3.00	14	25.90	3.00	14	25.89	3.00	14	25.89	4.00
SA3	22	44.66	19.00	19	44.82	20.00	20	44.93	20.00	21	**44.94**	18.00
SA4	18	57.04	173.00	20	57.71	159.00	19	**57.90**	172.00	21	57.40	142.00
SA5	19	65.91	191.00	16	65.29	226.00	19	**66.02**	203.00	19	65.88	192.00
SA6	38	70.04	158.00	42	74.10	118.00	43	74.81	113.00	44	**74.99**	104.00
SA7	38	82.68	225.00	38	83.00	237.00	40	83.31	211.00	40	**83.33**	219.00
SA8	38	77.52	304.00	45	82.61	159.00	48	**83.94**	140.00	47	83.83	159.00
SA9	34	73.56	137.00	36	74.04	123.00	35	**74.18**	125.00	35	74.04	128.00
SU1	14	**23.01**	2.00	14	22.49	2.00	14	23.00	2.00	14	**23.01**	2.00
SU2	16	23.69	4.00	16	23.59	4.00	17	24.54	4.00	17	**24.61**	3.00
SU5	32	61.47	52.00	31	61.81	57.00	33	**62.18**	48.00	33	62.05	45.00
TH2	20	35.69	5.00	22	**37.62**	2.00	22	37.62	5.00	22	37.54	4.00
TH4	24	53.26	64.00	23	53.86	71.00	22	53.94	78.00	22	53.83	75.00
TH9	34	65.47	126.00	35	66.15	99.00	35	66.24	102.00	36	**66.27**	96.00
TU1	11	24.77	4.00	9	24.76	5.00	9	**24.89**	4.00	9	**24.89**	5.00
TU2	20	34.69	3.00	20	35.39	3.00	20	**35.44**	3.00	20	35.43	3.00
TU3	22	42.52	12.00	19	42.27	19.00	22	42.51	13.00	22	**42.55**	13.00
TU8	43	83.63	218.00	44	85.24	196.00	44	85.19	199.00	45	**85.26**	173.00
TU9	33	70.01	162.00	32	70.86	171.00	32	71.05	162.00	35	**71.19**	123.00
WE1	9	15.82	1.00	8	15.63	1.00	8	15.60	1.00	9	**16.63**	0.00
WE2	22	36.09	3.00	23	**37.11**	3.00	22	36.21	3.00	23	36.94	3.00
WE3	19	43.29	36.00	17	44.34	35.00	17	44.32	37.00	20	**44.41**	28.00
WE4	29	56.59	68.00	30	58.01	72.00	30	58.00	71.00	30	**58.10**	74.00
WE5	29	**61.28**	72.00	29	61.21	66.00	28	60.90	78.00	27	61.10	83.00
WE6	38	80.36	195.00	43	**82.79**	151.00	43	82.16	148.00	40	82.00	181.00
WE7	40	72.82	86.00	40	73.34	85.00	41	74.19	81.00	41	**74.29**	86.00
WE9	16	**72.22**	355.00	17	72.12	336.00	17	72.15	328.00	18	72.20	318.00
	24.56	53.85	105.88	25.35	54.75	93.16	25.58	54.85	90.91	26.00	**54.92**	88.00

a Greedy algorithm and CPLEX in terms of computation time and solution quality.

6

Conclusions and future work

6.1 Conclusions

This thesis describes an optimisation framework that amalgamates the iterated local search method and the Lagrangian relaxation technique. Three variants of the vehicle routing problem are studied in this thesis. For each of the studied problems, a problem-specific optimisation procedure is derived under the Lagrangian ILS framework. These three optimisation procedures are common in terms of how the weights of coefficients are adjusted and how the local search algorithm is performed. Although the Lagrangian ILS framework can be directly applied to all three problems, to produce solutions with good quality, the algorithms under this framework still require problem-specific mechanisms to further enhance their performance.

6.1.1 Workforce Scheduling and Routing Problem

In Chapter 3, a new optimisation procedure for the Workforce Scheduling and Routing Problem is described. This procedure, referred to as the Lagrangian ILS, is based on the idea of an amalgamation of the iterated local search and Lagrangian relaxation, which was first introduced in Gu et al. (2019). Through various changes, the Lagrangian ILS significantly outperforms the original implementation of the idea of such amalgamation presented in Gu et al. (2019) in instances with 25 and 50 tasks. In particular, the Lagrangian ILS constantly produces the optimal solutions in almost all instances with 25 and 50 tasks. The computational experiments have also shown the superior performance of the Lagrangian ILS in comparison with CPLEX and the algorithm in Xie et al. (2017) both, in terms of the solution quality and the computational time. The computational experiments were conducted on a set of benchmark instances from the literature, regarded as standard in the publications on this topic. The exceptional performance of the

Lagrangian ILS is particularly evident in large instances, outperforming the algorithm in Xie et al. (2017) even when the Lagrangian ILS can only use half the permissible number of iterations. After applying the Lagrangian ILS to the WSRP, its performance demonstrates a great potential for solving the industry problems studied in Chapters 4 and 5.

6.2 Multi-attribute Simultaneous Pickup and Delivery Problem

In Chapter 4, a practical vehicle routing problem with simultaneous pickups and deliveries is studied. The problem considers ordered objectives where the primary objective is to maximise the number of served customers and the secondary objective is to minimise the total travel time. This problem is formulated into a three-index mathematical formulation and solved by an iterated local search that alternates between the two objectives during the application of local search. This iterated local search, called ILS2O, is based on the Lagrangian ILS and a neighbourhood reduction technique. The computational experiments were conducted on three sets of benchmarks. One is provided by a real-world transportation company and the other two are derived from the standard Solomon benchmark for vehicle routing problems. The results demonstrate the ILS2O algorithm outperforms a 2Phase algorithm, CPLEX, the iterated local search described in Xie et al. (2017) in terms of solution quality and stability within a time limit of 1 minute.

6.3 Simultaneous Pickup and Delivery Problem with Preloading under Uncertainty

In chapter 5, the preloading problem faced by the industry partner is simplified as the Simultaneous Pickup and Delivery with Preloading under Uncertainty by assuming the selection of vehicles for preloading is known and by assuming not all customers can be allocated during preloading. This problem is formulated as a 2-stage stochastic program and solved by the sample average approximation approach. An iterated local search extended from the Lagrangian ILS is designed for the sample average approximation approach.

This iterated local search, named ILS-SAA is tested on instances derived from real-world historical data. The results of the computational experiments have demonstrated that the ILS-SAA outperforms a greedy algorithm and CPLEX in terms of computational time and solution quality.

In addition, using 94 scenarios to evaluate the first-stage solution obtained by various algorithms, it has been verified that allocating as many customers as possible in the preloading may not always be a good policy.

6.4 Future work

This section outlines some potential directions for future research for the work presented in this thesis.

- Given the superior performance of the Lagrangian ILS described in Chapter 3 for the Workforce Scheduling and Routing Problem, the development of algorithms under the Lagrangian ILS framework for different vehicle routing problems such as vehicle routing problems with multi-depot, multi-trip, and multi-period, etc. This can be a promising direction for future research.

- In Chapter 4, the instances used in the computational experiments contain at most 100 customers. Due to urbanisation and the growth of e-commerce, especially during the COVID-19 pandemic, there has been an ever-increasing demand for home delivery services. It is worth investigating the performance of the ILS2O described in Chapter 4 on large-scale instances.

- The ILS-SAA described in Chapter 5 is a prototype that aims at testing whether the sample average approximation approach with the Lagrangian ILS has the potential to solve the Simultaneous Pickup and Delivery with Preloading under Uncertainty. The results of the computational experiments demonstrate the promising performance of the ILS-SAA. Therefore, it is worth investigating its performance with additional developments, for example, an advanced algorithm to generate the initial feasible solutions and new local search operators that cope with the problem features.

Furthermore, in this thesis, the presented algorithms are heuristics that focus on finding good solutions in a short time. However, after seeing the promising performance of algorithms under the Lagrangian ILS framework, it may be worth developing algorithms under this framework when a sufficient amount of computational time is permitted. For example, developing a population-based algorithm or an exact algorithm under this framework can be a promising direction.

Appendices

A Tables

A.1 Detailed computational results on performance comparison

Table A.1: Comparison between the performance of CPLEX, ILS in Xie et al. (2017), Implemented ILS and Lagrangian ILS on small and medium instances

			CPLEX		ILS		Implemented ILS		Lagrangian ILS				
Instances	$\|C\|$	$\|K\|$	Opt^*	sec^*	$\%^*$	sec_a	$\%^*$	sec_a	$\%^*$	$\|Opt\|$	sec_a	sec_w	sec_b
C101 5x4	25	4	271.70	0.05	-0.46	0.11	0.00	0.40	0.00	5	0.40	1.00	0.00
C201 5x4	25	2	863.08	0.02	0.00	0.09	0.00	0.00	0.00	5	0.00	0.00	0.00
C203 5x4	25	2	835.83	23.44	0.00	0.15	-0.09	0.20	0.00	5	0.20	1.00	0.00
R101 5x4	25	4	2195.04	0.02	0.00	0.22	0.00	0.80	0.00	5	0.60	1.00	0.00
R201 5x4	25	2	1091.07	0.14	0.00	0.03	-0.15	0.20	0.00	5	**0.00**	0.00	0.00
RC101 5x4	25	4	862.21	3.46	0.00	0.37	0.00	0.80	-0.46	1	0.40	1.00	0.00
RC201 5x4	25	3	465.25	0.28	-0.01	0.06	0.00	0.60	0.00	5	0.20	1.00	0.00
C101 6x6	25	4	927.35	0.02	0.00	0.09	0.00	0.20	0.00	5	0.20	1.00	0.00
C201 6x6	25	2	1217.10	0.01	0.00	0.01	0.00	0.40	0.00	5	0.20	1.00	0.00
C203 6x6	25	2	930.60	3.18	0.00	0.03	-1.67	0.20	0.00	5	0.20	1.00	0.00
R101 6x6	25	4	2857.05	0.03	-0.39	0.30	-0.52	0.80	**0.00**	5	0.40	1.00	0.00
R201 6x6	25	2	1377.42	0.11	-3.28	0.05	-1.01	0.40	**0.00**	5	0.20	1.00	0.00
RC101 6x6	25	4	1361.80	1.41	0.00	0.23	0.00	0.80	0.00	5	0.40	1.00	0.00
RC201 6x6	25	3	1228.89	5.63	0.00	0.14	0.00	0.20	0.00	5	0.20	1.00	0.00
C101 7x4	25	4	789.08	0.02	0.00	0.06	0.00	0.20	0.00	5	0.20	1.00	0.00
C103 7x4	25	4	671.06	186.38	0.00	0.11	-1.76	0.60	0.00	5	0.20	1.00	0.00
C201 7x4	25	2	738.35	0.02	0.00	0.02	0.00	0.00	0.00	5	0.20	1.00	0.00
C203 7x4	25	2	684.98	51.76	0.00	0.03	-1.57	0.20	0.00	5	**0.00**	0.00	0.00
R101 7x4	25	4	2447.74	0.01	0.00	0.12	0.00	0.40	0.00	5	0.40	1.00	0.00
R201 7x4	25	2	959.51	0.07	0.00	0.07	0.00	0.00	0.00	5	0.00	0.00	0.00
R203 7x4	25	2	849.47	115.23	0.00	0.03	-1.58	0.20	0.00	5	0.20	1.00	0.00
RC101 7x4	25	4	1669.63	0.12	0.00	0.09	0.00	0.60	0.00	5	0.20	1.00	0.00
RC201 7x4	25	3	967.60	0.41	0.00	0.08	-0.19	0.20	0.00	5	0.20	1.00	0.00
C101 5x4	50	6	830.00	0.69	0.00	1.09	0.00	6.60	0.00	5	2.40	3.00	2.00
C201 5x4	50	4	859.54	0.06	0.00	0.78	0.00	2.00	0.00	5	1.20	2.00	1.00
R101 5x4	50	6	4507.87	0.41	-0.08	6.84	0.00	8.40	0.00	5	**5.00**	7.00	4.00
R201 5x4	50	4	1107.51	15.97	-0.43	2.13	-1.98	6.00	**0.00**	5	2.20	3.00	2.00
C101 6x6	50	6	1154.84	57.42	0.00	1.95	0.00	8.80	0.00	5	3.60	4.00	3.00
C201 6x6	50	4	1203.93	0.05	0.00	0.66	0.00	1.40	0.00	5	1.40	2.00	1.00
R101 6x6	50	6	5190.32	1.19	0.00	2.62	0.00	8.00	0.00	5	5.00	7.00	4.00
R201 6x6	50	4	1647.70	104.03	-0.14	1.99	-0.20	4.80	**0.00**	5	2.80	3.00	2.00
C101 7x4	50	6	1356.54	0.83	-0.83	1.26	0.00	4.00	0.00	5	3.80	5.00	3.00
C201 7x4	50	4	1312.21	0.04	0.00	0.37	0.00	1.20	0.00	5	1.00	1.00	1.00
R101 7x4	50	6	4463.80	0.32	-0.12	2.01	-0.12	5.80	**0.00**	5	3.20	4.00	2.00
R201 7x4	50	4	1553.23	46.30	0.00	0.93	0.00	3.40	0.00	5	1.60	2.00	1.00
Average			1469.98	17.69	-0.16	0.72	-0.31	1.97	**-0.01**	4.89	1.10	1.77	0.74

The performance of the Lagrangian ILS, ILS in Xie et al. (2017), Implemented ILS, and CPLEX on the small (25 tasks) and medium (50 tasks) instances is given in Table A.1.

Table A.2: Comparison between ILS in Xie et al. (2017), Lagrangian ILS, CPLEX, Implemented ILS, and ILS-HNS in Zhou et al. (2020) on large instances from category "NoTeam Reduced"

			ILS				Lagrangian ILS							
Instances	$\|C\|$	$\|K\|$	Average	Worst	Best	sec_a	$\%_a$	$\%_w$	$\%_b$	$\|\hat{C}\|$	$\|\hat{K}\|$	sec_a	sec_w	sec_b
C101 5x4	100	8	5691.28	5780.75	5589.76	33.65	**1.22**	**2.37**	**0.04**	22.40	8.00	43.60	49.00	37.00
C103 5x4	100	8	2830.77	2921.15	2650.69	38.05	**6.18**	**6.16**	**1.79**	7.00	8.00	39.20	42.00	34.00
C201 5x4	100	4	2781.37	2800.21	2755.52	11.40	0.93	1.60	0.00	6.00	4.00	13.80	14.00	13.00
C203 5x4	100	4	2393.88	2407.41	2383.01	15.89	0.24	**0.58**	**0.30**	6.00	4.00	16.60	20.00	12.00
R101 5x4	100	12	5572.16	5590.55	5561.83	62.46	0.17	**0.19**	0.19	21.00	12.00	**46.40**	55.00	39.00
R103 5x4	100	12	1765.16	1943.30	1658.23	63.82	0.48	5.53	-0.85	1.60	12.00	**45.00**	47.00	43.00
R201 5x4	100	4	2866.39	2887.93	2839.54	14.14	**0.51**	**0.84**	**0.04**	6.00	4.00	20.60	26.00	18.00
R203 5x4	100	4	2349.24	2360.57	2335.76	12.86	0.02	-0.12	-0.16	6.00	4.00	14.20	17.00	11.00
RC101 5x4	100	11	4983.98	5093.47	4909.89	46.74	1.81	3.38	1.64	16.80	11.00	**33.00**	39.00	29.00
RC103 5x4	100	11	2421.65	2499.04	2301.86	39.24	**5.35**	**7.06**	**1.76**	4.00	11.00	**32.00**	38.00	27.00
RC201 5x4	100	5	3090.03	3100.00	3083.49	15.79	**0.34**	**0.57**	**0.42**	6.00	5.00	23.80	29.00	20.00
RC203 5x4	100	5	2530.63	2568.56	2512.64	12.54	**0.64**	**1.84**	**0.05**	6.00	5.00	18.80	21.00	18.00
C101 6x6	100	8	7695.83	7783.33	7660.86	30.17	**0.45**	**1.57**	0.00	32.00	8.00	46.20	62.00	38.00
C103 6x6	100	8	5066.61	5195.96	4971.66	37.08	**3.54**	**3.61**	**3.36**	17.40	8.00	41.00	49.00	34.00
C201 6x6	100	4	3313.45	3331.26	3298.68	21.10	**1.07**	**1.60**	0.62	9.00	4.00	36.40	44.00	32.00
C203 6x6	100	4	2479.44	2484.72	2468.53	23.56	**0.58**	**0.38**	**0.77**	6.00	3.00	36.80	41.00	28.00
R101 6x6	100	13	6005.32	6083.98	5948.71	56.05	0.14	-0.70	**0.06**	22.20	13.00	**50.80**	56.00	47.00
R103 6x6	100	13	2290.01	2383.03	2225.67	61.65	-0.90	0.69	**0.36**	4.60	12.00	**51.00**	63.00	40.00
R201 6x6	100	4	3574.46	3633.35	3510.36	32.37	**1.19**	**1.76**	**2.00**	8.80	4.00	43.00	54.00	38.00
R203 6x6	100	4	2462.68	2504.36	2443.97	23.00	**0.02**	**0.64**	-0.04	6.00	3.00	36.40	47.00	25.00
RC101 6x6	100	12	5029.94	5142.08	4975.34	43.80	**1.72**	2.40	**1.44**	16.00	12.00	**42.20**	47.00	36.00
RC103 6x6	100	12	2257.78	2337.45	2113.03	45.74	**5.44**	**3.69**	**0.60**	2.20	12.00	**36.40**	43.00	30.00
RC201 6x6	100	4	4550.99	4608.61	4490.33	31.23	**1.35**	**1.99**	**0.81**	12.00	4.00	52.20	57.00	47.00
RC203 6x6	100	4	2686.83	2719.03	2671.23	19.58	**1.08**	**1.69**	**0.91**	6.00	3.00	40.00	51.00	34.00
C101 7x4	100	9	5284.48	5360.98	5246.13	19.88	0.80	2.21	0.08	19.00	9.00	29.40	37.00	23.00
C103 7x4	100	9	2059.98	2163.05	2009.86	24.70	**2.56**	**3.56**	**1.45**	2.00	9.00	**23.00**	27.00	20.00
C201 7x4	100	4	2808.29	2830.03	2781.07	8.11	1.04	1.00	0.28	5.00	4.00	19.80	23.00	18.00
C203 7x4	100	4	2297.16	2366.11	2262.00	9.98	**1.10**	2.31	**0.03**	5.00	4.00	16.80	23.00	12.00
R101 7x4	100	14	5238.11	5283.80	5127.29	33.24	0.57	0.36	0.07	17.60	14.00	**31.20**	39.00	22.00
R103 7x4	100	14	2222.76	2333.64	2139.77	33.70	**1.41**	**1.96**	**1.63**	3.40	14.00	**21.60**	27.00	18.00
R201 7x4	100	5	2678.32	2693.43	2664.93	9.51	**0.62**	**1.01**	**0.28**	5.00	5.00	17.00	22.00	15.00
R203 7x4	100	5	2223.96	2242.78	2209.32	10.09	**0.83**	**1.26**	**0.45**	5.00	5.00	13.40	18.00	10.00
RC101 7x4	100	12	5440.59	5556.76	5373.05	29.21	-0.17	0.43	**0.13**	18.60	12.00	**22.40**	28.00	17.00
RC103 7x4	100	12	2615.16	2653.73	2591.39	24.52	**0.93**	**2.30**	**0.21**	5.00	12.00	**24.00**	29.00	19.00
RC201 7x4	100	5	2934.44	2948.77	2910.73	9.44	**0.66**	**0.99**	0.00	5.00	5.00	15.80	18.00	14.00
RC203 7x4	100	5	2308.80	2314.51	2305.62	10.14	**0.52**	**0.34**	**1.07**	5.00	4.80	18.20	22.00	15.00
Average			3466.72	3525.21	3416.16	28.18	**1.23**	**1.86**	**0.60**	9.63	7.55	30.89	36.78	25.92

Table A.3: Comparison between ILS in Xie et al. (2017), Lagrangian ILS, CPLEX, Implemented ILS, and ILS-HNS in Zhou et al. (2020) on large instances from category "NoTeam Reduced" (continued)

			CPLEX			Implemented ILS				ILS-HNS		
Instances	$\|C\|$	$\|K\|$	Obj	Gap(%)	sec^*	$\%_a$	$\%_w$	$\%_b$	sec_a	$\%_a$	$\%_w$	$\%_b$
C101 5x4	100	8	5643.50	12.06	14400.00	0.72	1.78	-0.15	101.60	-0.12	1.02	-1.34
C103 5x4	100	8	–	–	14400.00	1.59	0.43	-1.00	107.00	-14.55	-14.25	-17.02
C201 5x4	100	4	2755.20	0.00	1.46	0.93	1.60	0.00	21.40	0.93	1.60	0.00
C203 5x4	100	4	–	–	14400.00	-2.22	-2.04	-2.48	44.20	0.28	0.57	-0.07
R101 5x4	100	12	5446.89	0.00	792.03	0.17	0.11	0.20	168.00	-7.70	-8.14	-7.05
R103 5x4	100	12	–	–	14400.00	1.59	5.75	-1.08	189.60	-16.45	-11.92	-19.10
R201 5x4	100	4	–	–	14400.00	-0.76	-0.46	-1.10	40.20	-1.26	-1.51	-1.47
R203 5x4	100	4	–	–	14400.00	-2.90	-3.53	-2.50	64.00	0.72	1.19	0.15
RC101 5x4	100	11	5501.32	65.41	14400.00	2.58	3.47	2.49	116.00	-13.88	-19.01	-7.95
RC103 5x4	100	11	–	–	14400.00	1.86	0.88	-0.20	147.40	-10.68	-14.55	-11.48
RC201 5x4	100	5	3211.45	15.21	14400.00	-1.45	-2.89	-0.60	66.60	-0.33	-1.18	-0.02
RC203 5x4	100	5	–	–	14400.00	-1.41	-0.63	-1.86	85.80	0.31	1.52	-0.08
C101 6x6	100	8	7660.86	7.79	14400.00	0.32	0.89	0.00	82.20	-0.52	-0.20	-0.10
C103 6x6	100	8	–	–	14400.00	1.75	3.24	0.75	111.60	-12.41	-11.57	-11.87
C201 6x6	100	4	3278.07	0.00	1861.88	0.45	0.98	0.00	34.20	0.70	0.37	0.62
C203 6x6	100	4	–	–	14400.00	-3.83	-4.40	-3.22	40.60	-0.29	-0.40	-0.41
R101 6x6	100	13	5944.91	0.00	1189.44	0.94	2.22	0.00	194.40	-9.96	-12.00	-9.42
R103 6x6	100	13	–	–	14400.00	2.60	5.63	0.28	230.60	-26.30	-30.78	-21.33
R201 6x6	100	4	–	–	14400.00	-0.81	0.33	-2.03	72.20	-6.78	-9.73	-3.26
R203 6x6	100	4	–	–	14400.00	-4.84	-3.97	-4.22	71.80	-1.05	-0.52	-0.07
RC101 6x6	100	12	5320.99	62.43	14400.00	1.71	3.19	1.11	159.20	-23.17	-24.50	-19.55
RC103 6x6	100	12	–	–	14400.00	-0.13	0.75	-3.56	174.20	-37.80	-39.57	-40.85
RC201 6x6	100	4	–	–	14400.00	0.65	-0.18	0.79	61.00	-13.02	-14.32	-11.64
RC203 6x6	100	4	–	–	14400.00	-3.69	-4.67	-1.27	60.20	-1.00	-0.80	-0.44
C101 7x4	100	9	5246.13	6.34	14400.00	0.80	2.22	0.09	65.60	0.28	1.21	-0.09
C103 7x4	100	9	–	–	14400.00	0.27	2.83	0.02	62.60	-11.55	-13.56	-10.10
C201 7x4	100	4	2773.41	0.00	21.46	0.54	0.60	0.28	20.20	1.08	1.73	0.28
C203 7x4	100	4	–	–	14400.00	-1.16	0.68	-1.07	34.40	0.69	3.12	-0.31
R101 7x4	100	14	5079.67	0.00	481.65	1.88	0.70	0.93	105.20	-8.99	-10.09	-10.05
R103 7x4	100	14	–	–	14400.00	0.13	0.02	0.45	107.00	-4.45	-1.76	-5.12
R201 7x4	100	5	2673.28	10.25	14400.00	-0.17	-0.25	-0.04	56.00	0.03	0.11	0.00
R203 7x4	100	5	–	–	14400.00	-1.35	-1.60	-0.72	75.00	0.43	1.07	0.03
RC101 7x4	100	12	5694.57	41.86	14400.00	0.09	0.89	0.12	111.20	-15.71	-16.50	-12.94
RC103 7x4	100	12	–	–	14400.00	-0.23	-3.15	0.00	91.00	-9.84	-15.40	-2.02
RC201 7x4	100	5	3030.43	17.73	14400.00	-0.58	-0.81	-0.66	56.40	0.37	0.61	-0.23
RC203 7x4	100	5	–	–	14400.00	-1.32	-2.07	-0.34	67.80	-0.20	-0.37	-0.11
Average			–	–	12120.78	-0.15	0.24	-0.57	91.57	-6.73	-7.18	-6.23

Table A.4: Comparison between ILS in Xie et al. (2017), Lagrangian ILS, CPLEX, Implemented ILS, and ILS-HNS in Zhou et al. (2020) on large instances from category "NoTeam Complete"

			ILS				**Lagrangian ILS**						
Instances	$\lvert C\rvert$	$\lvert K\rvert$	Average	Worst	Best	sec_a	$\%_a$	$\%_w$	$\%_b$	$\lvert\hat{K}\rvert$	sec_a	sec_w	sec_b
C101 5x4	100	17	1107.76	1117.56	1097.67	33.70	0.57	-0.01	0.07	12.40	**23.00**	27.00	21.00
C103 5x4	100	17	1026.51	1045.80	1012.86	38.63	0.64	**1.88**	-0.45	11.00	**23.20**	27.00	21.00
C201 5x4	100	8	1157.56	1157.56	1157.56	9.21	0.00	0.00	0.00	7.00	10.00	11.00	9.00
C203 5x4	100	8	1059.68	1068.72	1054.21	21.17	0.80	0.94	0.69	5.00	21.40	27.00	19.00
R101 5x4	100	25	1668.00	1676.64	1658.93	43.66	**0.41**	**0.76**	-0.01	20.20	**41.00**	49.00	34.00
R103 5x4	100	25	1243.54	1253.46	1235.05	58.66	**0.78**	**1.32**	**0.22**	15.00	63.00	81.00	52.00
R201 5x4	100	7	1436.37	1447.84	1431.16	27.61	**0.60**	**1.39**	**0.24**	6.00	**25.60**	33.00	20.00
R203 5x4	100	7	1100.75	1105.09	1097.55	23.91	**0.61**	**0.32**	0.51	6.00	**20.00**	25.00	18.00
RC101 5x4	100	22	1695.67	1710.32	1673.94	51.32	**1.35**	1.25	**0.93**	15.80	**39.00**	47.00	34.00
RC103 5x4	100	22	1355.40	1383.61	1321.66	59.19	**3.40**	**4.67**	**1.24**	11.80	**56.40**	62.00	52.00
RC201 5x4	100	9	1606.08	1620.59	1589.24	27.02	**1.38**	**2.26**	0.33	8.00	**20.60**	23.00	19.00
RC203 5x4	100	9	1165.81	1169.24	1162.95	24.84	**0.19**	**0.49**	-0.05	6.00	**24.20**	29.00	20.00
C101 6x6	100	16	988.43	1001.62	972.89	45.79	**0.53**	**0.98**	0.00	11.40	**27.20**	33.00	24.00
C103 6x6	100	16	911.62	933.90	900.82	53.15	**1.61**	**3.76**	0.57	10.20	**40.80**	45.00	35.00
C201 6x6	100	7	826.42	832.56	821.55	42.76	0.59	1.32	0.00	4.00	**38.00**	47.00	34.00
C203 6x6	100	7	693.50	699.19	689.60	44.15	0.49	1.04	0.00	4.00	**41.20**	49.00	36.00
R101 6x6	100	26	1660.15	1664.28	1657.55	67.27	**0.15**	0.11	0.24	19.60	**45.00**	53.00	40.00
R103 6x6	100	26	1225.80	1238.01	1216.08	73.23	**0.61**	**1.37**	0.00	14.00	80.80	92.00	68.00
R201 6x6	100	7	1270.25	1278.94	1265.56	55.72	**0.47**	**0.95**	**0.24**	6.00	**42.40**	51.00	35.00
R203 6x6	100	7	933.41	940.98	930.74	56.79	**1.31**	**1.91**	**1.34**	5.00	72.40	85.00	64.00
RC101 6x6	100	24	1674.61	1682.62	1663.30	61.94	**0.76**	**0.56**	**0.50**	15.40	**48.00**	60.00	39.00
RC103 6x6	100	24	1309.18	1331.28	1297.09	86.38	**3.29**	**3.61**	**3.37**	11.20	**63.40**	70.00	58.00
RC201 6x6	100	8	1380.38	1394.90	1367.89	52.16	**0.71**	**1.24**	**0.09**	6.00	**42.60**	46.00	34.00
RC203 6x6	100	8	1014.51	1024.80	1003.81	49.44	**1.14**	**2.04**	**0.47**	5.00	57.20	64.00	44.00
C101 7x4	100	17	1370.78	1381.59	1357.05	22.31	**0.15**	-0.68	0.00	14.80	**13.80**	17.00	11.00
C103 7x4	100	17	1233.60	1256.67	1220.19	27.96	**1.33**	**2.90**	**0.92**	13.00	**19.60**	26.00	16.00
C201 7x4	100	8	1263.00	1289.68	1256.30	10.56	0.53	2.59	0.00	8.00	11.20	12.00	10.00
C203 7x4	100	8	1145.36	1150.85	1137.07	17.89	**0.61**	0.00	0.32	7.60	**14.80**	21.00	9.00
R101 7x4	100	28	1787.22	1796.48	1781.13	39.28	**0.39**	**0.34**	0.43	21.60	**31.00**	40.00	25.00
R103 7x4	100	28	1349.32	1373.27	1337.92	43.54	**0.71**	**1.98**	**0.17**	16.60	**39.20**	54.00	30.00
R201 7x4	100	10	1406.55	1412.46	1401.68	19.36	**0.23**	**0.41**	0.25	9.00	**15.20**	17.00	13.00
R203 7x4	100	10	1164.89	1170.12	1160.51	18.45	-0.10	-0.41	0.00	8.20	19.20	23.00	14.00
RC101 7x4	100	23	1822.30	1837.41	1805.39	36.51	**2.10**	**2.34**	**1.37**	17.20	**24.40**	29.00	22.00
RC103 7x4	100	23	1436.23	1450.97	1427.40	38.30	**1.07**	**1.00**	**1.75**	13.20	**30.20**	38.00	23.00
RC201 7x4	100	9	1706.24	1727.00	1697.82	15.34	**0.34**	**0.94**	0.00	9.00	**11.60**	13.00	10.00
RC203 7x4	100	9	1235.76	1238.74	1230.63	16.81	**1.17**	-0.66	**1.42**	8.00	**15.40**	20.00	12.00
Average			1289.79	1301.80	1280.35	39.28	**0.86**	**1.25**	**0.48**	10.62	**33.67**	40.17	28.47

Table A.5: Comparison between ILS in Xie et al. (2017), Lagrangian ILS, CPLEX, Implemented ILS, ILS-HNS in Zhou et al. (2020) on large instances from category "NoTeam Complete" (continued)

			CPLEX			Implemented ILS				ILS-HNS		
Instances	$\|C\|$	$\|K\|$	Obj	Gap(%)	sec^*	$\%_a$	$\%_w$	$\%_b$	sec_a	$\%_a$	$\%_w$	$\%_b$
C101 5x4	100	17	1096.85	0.00	89.32	0.08	-0.13	-0.03	117.00	0.98	1.85	0.07
C103 5x4	100	17	1534.45	66.38	14400.00	-0.45	-0.56	-0.81	152.20	0.66	1.07	0.00
C201 5x4	100	8	1157.56	0.00	0.73	0.00	0.00	0.00	25.60	0.00	0.00	0.00
C203 5x4	100	8	-	-	14400.00	-0.15	-0.09	0.05	65.80	1.20	2.04	0.69
R101 5x4	100	25	1652.13	0.00	1236.85	0.25	0.43	0.15	496.40	0.22	0.27	-0.05
R103 5x4	100	25	1974.25	62.41	14400.00	0.49	0.88	0.07	586.80	-0.09	-0.13	-0.24
R201 5x4	100	7	-	-	14400.00	-0.77	-1.15	0.15	85.20	-0.14	-0.45	0.18
R203 5x4	100	7	-	-	14400.00	-1.31	-1.85	-1.20	106.40	0.52	0.18	0.51
RC101 5x4	100	22	1749.89	22.25	14400.00	0.91	1.25	0.19	330.20	0.66	0.44	0.28
RC103 5x4	100	22	1993.82	66.79	14400.00	1.88	2.08	0.25	494.00	0.72	0.75	0.07
RC201 5x4	100	9	1774.33	22.68	14400.00	-0.04	-0.17	-0.59	152.20	0.74	0.94	0.33
RC203 5x4	100	9	-	-	14400.00	-1.09	-2.14	-0.41	106.80	0.09	-0.08	0.12
C101 6x6	100	16	972.89	0.00	45.69	0.41	0.93	0.00	170.20	-0.12	-0.24	-0.05
C103 6x6	100	16	-	-	14400.00	0.58	0.36	0.54	261.60	0.98	1.80	0.76
C201 6x6	100	7	821.55	0.00	61.99	0.45	0.64	0.00	80.60	0.59	1.32	0.00
C203 6x6	100	7	-	-	14400.00	0.25	0.57	0.00	143.40	0.49	1.04	0.00
R101 6x6	100	26	1648.27	0.00	1849.44	0.11	0.21	0.17	542.60	0.01	-0.24	0.25
R103 6x6	100	26	10849.90	93.55	5339.76	0.45	1.04	0.00	713.80	0.22	0.54	-0.20
R201 6x6	100	7	-	-	14400.00	-0.61	-0.95	-0.15	150.80	-1.43	-3.56	-0.25
R203 6x6	100	7	-	-	14400.00	0.44	0.64	0.66	190.00	-0.95	-1.83	-0.11
RC101 6x6	100	24	1712.10	23.63	2833.96	0.07	-0.07	-0.04	370.40	0.03	0.18	0.04
RC103 6x6	100	24	1925.34	66.56	14400.00	1.68	1.21	2.33	632.20	0.71	0.37	3.30
RC201 6x6	100	8	-	-	14400.00	-0.17	0.29	-0.50	144.40	-0.74	-1.34	-0.56
RC203 6x6	100	8	-	-	14400.00	-0.32	-0.44	-0.29	169.00	-1.05	-1.31	-0.91
C101 7x4	100	17	1357.05	0.00	14.76	-0.82	-0.74	-1.29	82.40	0.02	0.00	0.00
C103 7x4	100	17	-	-	14400.00	0.11	1.14	-0.29	146.60	0.63	1.66	0.44
C201 7x4	100	8	1256.30	0.00	0.31	-0.79	0.00	0.00	27.20	0.53	2.59	0.00
C203 7x4	100	8	-	-	14400.00	-3.63	-4.59	-3.38	79.20	0.57	-0.15	0.32
R101 7x4	100	28	1764.78	0.00	444.14	0.26	-0.26	0.60	348.80	0.05	-0.26	0.43
R103 7x4	100	28	-	-	14400.00	-0.08	1.13	-0.39	425.00	-0.49	-0.12	-0.41
R201 7x4	100	10	1410.52	6.12	14400.00	-0.66	-1.58	0.02	83.80	0.17	0.05	0.25
R203 7x4	100	10	-	-	14400.00	-4.11	-5.01	-3.04	124.80	-0.65	-2.51	0.04
RC101 7x4	100	23	1838.86	16.06	14400.00	0.79	0.93	0.66	263.80	0.22	0.29	0.19
RC103 7x4	100	23	-	-	14400.00	0.50	0.77	0.23	321.80	0.45	0.72	0.94
RC201 7x4	100	9	-	-	14400.00	-0.26	-0.83	0.00	76.80	0.07	-0.05	0.00
RC203 7x4	100	9	-	-	14400.00	-1.02	-2.66	0.53	78.40	-1.04	-2.27	0.93
Average					10331.03	-0.18	-0.24	-0.16	231.84	0.13	0.10	0.20

The performance of the Lagrangian ILS, ILS, Implemented ILS, ILS-HNS in Zhou et al. (2020), and CPLEX on Large (100 tasks) instances from the two categories, "NoTeam Reduced" and "NoTeam Complete" is given in Tables A.2, A.3, A.4, and A.5. In these tables, the first three columns are the instance's name, the number of tasks and the number of service providers.

In Table A.1, the columns Opt^* and sec^* give the optimal objective value and the computational time required by CPLEX. Each column $\%^*$ gives the percentage differences (3.45) of the objective value with respect to the objective value obtained by CPLEX. Each column sec_a reports the average computational time required by the respective algorithm. The worst and best times required by the Lagrangian ILS are shown in columns sec_w and sec_b. Furthermore, the column $|Opt|$ gives the number out of 5 groups that the Lagrangian ILS obtains the optimal solution. The values obtained by the Lagrangian ILS are in bold if they are better than the values obtained by ILS and Implemented ILS.

In Tables A.2, A.3, A.4, and A.5, the columns Average, Worst, Best give the average, worst, best objective values produced by the ILS in Xie et al. (2017). The columns Obj and Gap(%) show the objective values and the optimality gap obtained by CPLEX. 0% in column Gap(%) means the objective value is optimal. The columns $\%_a$, $\%_w$, $\%_b$ are the percentage difference of average, worst, best objective values relative to the average, worst, best values reported in Xie et al. (2017). In addition, the columns $|\hat{C}|$ and $|\hat{K}|$ show the average number of outsourced tasks and the average number of service providers used. The objective values obtained by the Lagrangian ILS are in bold if they are better than the values obtained by ILS, Implemented ILS, and ILS-HNS whereas the computational times required by the Lagrangian ILS are in bold if they are smaller than the time required by ILS and Implemented ILS.

A.2 Detailed computational results on sensitivity analysis

Tables A.6 and A.7 present the analyses on the performance of the Lagrangian ILS with $\omega \in \{0.5, 7, 15\}$ on large (100 tasks) instances from the categories "NoTeam Reduced" and "NoTeam Complete" using $\psi = 50$ and $\gamma = 2$. All percentage differences are referenced to the corresponding values obtained by the Lagrangian ILS with $\omega = 0.5$. Furthermore,

Table A.6: Sensitivity analysis on the performance of the Lagrangian ILS with ω when $\psi = 50$ and $\gamma = 2$ for large instances from category "NoTeam Reduced"

	$w = 0.5$				$w = 7$				$w = 15$			
Instances	Average	Worst	Best	sec_a	$\%_a$	$\%_w$	$\%_b$	sec_a	$\%_a$	$\%_w$	$\%_b$	sec_a
C101 5x4	5651.34	5693.86	5590.79	27.00	0.37	0.88	-0.32	250.80	0.82	0.80	0.06	657.20
C103 5x4	2735.81	2861.35	2648.41	20.20	2.66	2.28	2.08	223.80	4.71	7.91	2.19	463.80
C201 5x4	2755.52	2755.52	2755.52	8.20	0.00	0.00	0.00	72.00	0.00	0.00	0.00	155.00
C203 5x4	2388.39	2394.82	2376.48	9.80	-0.15	-0.36	-0.34	84.00	0.30	0.06	0.02	149.20
R101 5x4	5573.35	5594.99	5551.43	25.20	0.28	0.65	-0.11	306.00	0.67	0.53	1.88	487.60
R103 5x4	1874.80	1939.20	1768.58	26.80	5.48	6.21	6.24	204.40	5.03	0.85	5.82	466.60
R201 5x4	2860.05	2883.04	2838.50	11.60	0.61	0.82	0.00	117.80	0.75	1.54	0.00	193.80
R203 5x4	2341.70	2347.77	2335.90	8.40	0.17	0.11	0.16	80.20	0.40	0.66	0.16	147.40
RC101 5x4	4938.50	5068.07	4880.45	19.60	1.63	3.35	2.19	168.60	1.75	3.24	2.19	389.20
RC103 5x4	2340.87	2505.81	2281.10	17.20	6.61	9.91	5.91	184.40	6.52	9.63	6.82	331.80
RC201 5x4	3084.00	3097.32	3076.10	12.20	0.27	0.48	0.18	110.60	0.27	0.52	0.13	251.60
RC203 5x4	2520.16	2540.27	2511.29	10.40	0.35	1.14	0.00	96.60	0.17	0.31	0.00	198.80
C101 6x6	7660.86	7660.86	7660.86	26.40	0.00	0.00	0.00	240.20	0.00	0.00	0.00	511.60
C103 6x6	4919.85	4979.85	4818.55	24.00	1.53	1.07	0.29	243.40	1.89	2.16	0.42	553.40
C201 6x6	3288.42	3303.94	3278.07	19.20	0.31	0.78	0.00	213.60	0.31	0.78	0.00	451.00
C203 6x6	2460.54	2468.17	2454.02	18.80	0.29	0.39	0.18	190.40	0.26	0.25	0.18	366.60
R101 6x6	6102.83	6162.33	5970.73	29.00	2.55	3.47	0.43	333.20	2.59	3.53	0.43	596.40
R103 6x6	2297.62	2367.90	2215.10	31.20	2.85	3.58	0.15	230.00	3.61	6.44	0.05	580.00
R201 6x6	3568.41	3609.33	3551.83	19.60	2.34	1.74	3.14	254.80	1.75	1.76	2.98	692.40
R203 6x6	2460.59	2489.00	2437.28	19.80	0.77	1.44	0.00	157.80	0.43	0.44	0.00	412.40
RC101 6x6	4999.87	5102.17	4924.98	20.60	2.22	3.73	1.82	266.20	2.29	3.58	1.82	634.00
RC103 6x6	2189.73	2271.85	2101.68	21.80	0.09	1.66	-0.87	208.40	2.71	1.98	0.07	535.80
RC201 6x6	4506.70	4563.49	4450.47	28.40	1.06	1.88	0.17	332.60	1.70	2.64	0.70	814.80
RC203 6x6	2656.09	2675.32	2646.91	21.60	0.31	0.88	0.00	202.40	0.31	0.91	0.00	436.80
C101 7x4	5274.75	5311.78	5246.13	12.60	0.64	1.31	0.15	171.80	0.63	1.31	0.09	361.80
C103 7x4	2024.68	2110.19	1977.30	14.80	2.75	5.91	1.02	118.60	2.75	5.86	1.02	235.80
C201 7x4	2773.41	2773.41	2773.41	12.00	0.00	0.00	0.00	127.60	0.00	0.00	0.00	266.80
C203 7x4	2274.81	2286.93	2261.33	9.20	0.59	1.09	0.00	107.20	0.59	1.12	0.00	196.00
R101 7x4	5238.77	5290.99	5133.88	17.00	2.65	3.16	1.06	171.00	2.06	0.94	0.60	325.00
R103 7x4	2211.23	2274.09	2133.68	12.60	-0.08	0.14	1.14	118.20	2.44	1.64	1.38	233.60
R201 7x4	2664.93	2669.15	2661.68	8.20	0.19	0.28	0.16	83.80	0.26	0.34	0.16	152.20
R203 7x4	2209.64	2217.39	2201.33	8.20	0.40	0.45	0.10	68.40	0.40	0.45	0.10	140.00
RC101 7x4	5503.19	5548.14	5477.84	13.20	1.58	1.27	1.90	146.40	2.37	3.08	2.04	317.20
RC103 7x4	2657.10	2751.24	2591.39	12.20	2.54	5.77	0.21	108.60	2.63	5.77	0.21	223.20
RC201 7x4	2918.10	2922.88	2912.39	8.20	0.16	0.19	0.20	88.00	0.29	0.23	0.25	199.20
RC203 7x4	2302.18	2309.86	2291.57	8.20	0.60	0.16	0.61	100.60	0.98	1.06	0.61	185.60
Average	3450.80	3494.51	3410.75	17.04	1.24	1.83	0.77	171.73	1.52	2.01	0.90	369.82

on large instances from category "NoTeam Reduced", Table A.8 presents the results obtained from the Lagrangian ILS with $\psi \in \{5, 50, 150, 400\}$ when $\gamma = 2$ and $\omega = 1$, and Table A.9 presents the results obtained from the Lagrangian ILS with $\gamma \in \{0.2, 2, 10, 100\}$ when $\psi = 50$ and $\omega = 1$. In Tables A.8 and A.9, the Average, Best, sec_a give the average, best objective values and the average computational time.

Table A.7: Sensitivity analysis on the performance of the Lagrangian ILS with ω when $\psi = 50$ and $\gamma = 2$ on large instances from category "NoTeam Complete"

	$w = 0.5$				$w = 7$				$w = 15$			
Instances	Average	Worst	Best	sec_a	$\%_a$	$\%_w$	$\%_b$	sec_a	$\%_a$	$\%_w$	$\%_b$	sec_a
C101 5x4	1100.07	1104.89	1096.85	13.40	0.29	0.73	0.00	123.40	0.29	0.73	0.00	224.40
C103 5x4	1018.71	1026.46	1012.86	14.60	0.17	0.88	-0.24	133.40	0.25	0.84	0.00	229.20
C201 5x4	1157.56	1157.56	1157.56	6.00	0.00	0.00	0.00	56.60	0.00	0.00	0.00	116.20
C203 5x4	1050.06	1057.84	1046.93	11.60	0.30	1.03	0.00	105.80	0.30	1.03	0.00	202.00
R101 5x4	1665.35	1681.02	1659.39	19.80	0.49	1.25	0.34	173.20	0.71	1.40	0.44	433.20
R103 5x4	1236.76	1240.19	1233.57	38.20	0.37	0.56	0.26	288.40	0.41	0.63	0.30	520.00
R201 5x4	1432.19	1441.87	1427.75	12.60	0.31	0.98	0.00	120.20	0.31	0.98	0.00	213.60
R203 5x4	1097.81	1102.87	1092.29	13.00	0.42	0.54	0.00	103.00	0.31	0.45	0.00	169.40
RC101 5x4	1673.35	1684.33	1654.80	20.60	0.65	0.69	-0.05	197.60	0.85	1.06	0.11	445.20
RC103 5x4	1326.72	1343.62	1317.55	32.00	1.66	2.36	1.28	274.40	1.65	2.47	1.28	595.20
RC201 5x4	1591.16	1598.68	1583.97	12.60	0.45	0.92	0.00	117.80	0.45	0.92	0.00	216.40
RC203 5x4	1166.91	1173.74	1163.47	13.80	0.45	0.99	0.17	157.40	0.46	1.04	0.17	288.80
C101 6x6	984.65	1002.37	972.89	14.80	1.19	2.94	0.00	121.00	0.86	1.31	0.00	260.40
C103 6x6	903.93	923.82	895.14	21.40	1.04	3.04	0.21	193.00	1.02	2.97	0.21	393.40
C201 6x6	821.55	821.55	821.55	23.40	0.00	0.00	0.00	254.40	0.00	0.00	0.00	504.40
C203 6x6	691.61	693.45	689.60	28.60	0.29	0.55	0.00	252.20	0.29	0.55	0.00	469.00
R101 6x6	1656.66	1664.10	1650.71	28.60	0.43	0.56	0.15	234.80	0.38	0.48	0.15	549.40
R103 6x6	1218.87	1221.17	1216.08	46.80	0.23	0.42	0.00	406.60	0.23	0.42	0.00	743.40
R201 6x6	1267.63	1270.60	1262.36	25.00	0.38	0.54	-0.01	212.00	0.38	0.57	0.03	390.00
R203 6x6	929.21	932.88	924.57	31.80	1.03	1.07	0.62	334.80	0.94	1.06	0.62	615.80
RC101 6x6	1664.95	1669.53	1661.24	24.60	0.60	0.44	0.59	223.80	0.60	0.38	0.63	488.40
RC103 6x6	1276.48	1286.60	1268.59	48.40	1.27	1.47	1.33	452.80	1.50	1.33	1.33	726.80
RC201 6x6	1373.74	1385.81	1366.72	25.00	0.32	0.90	0.00	256.60	0.41	0.90	0.00	421.00
RC203 6x6	1007.75	1023.01	1003.87	30.80	0.48	1.87	0.47	238.40	0.39	1.87	0.01	490.00
C101 7x4	1365.47	1381.59	1357.05	8.80	-0.19	0.32	0.00	79.80	0.62	1.78	0.00	151.60
C103 7x4	1225.43	1237.30	1218.50	11.00	0.95	1.05	0.79	83.20	1.01	1.52	0.79	185.80
C201 7x4	1256.30	1256.30	1256.30	6.80	0.00	0.00	0.00	65.20	0.00	0.00	0.00	131.20
C203 7x4	1135.90	1137.14	1133.47	8.80	0.21	0.32	0.00	80.20	0.21	0.32	0.00	154.60
R101 7x4	1779.33	1785.68	1775.08	18.60	0.51	0.46	0.53	169.40	0.56	0.80	0.45	300.20
R103 7x4	1350.84	1365.78	1338.45	24.40	0.80	1.45	0.57	198.80	0.53	1.09	0.57	364.00
R201 7x4	1399.20	1403.42	1398.14	11.60	0.08	0.38	0.00	82.80	0.01	0.20	0.00	170.20
R203 7x4	1175.46	1183.55	1166.71	12.00	1.17	1.75	0.57	92.40	1.26	1.75	0.57	202.20
RC101 7x4	1795.16	1807.24	1783.57	14.00	0.51	0.02	0.16	127.80	0.72	1.33	0.16	297.20
RC103 7x4	1418.46	1429.47	1402.46	15.80	0.73	0.65	0.15	139.60	1.03	0.84	0.15	331.20
RC201 7x4	1701.94	1715.93	1697.82	8.60	0.24	1.06	0.00	64.00	0.24	1.06	0.00	114.20
RC203 7x4	1224.99	1251.66	1213.14	11.40	0.91	2.90	0.03	90.60	0.40	0.45	0.03	195.00
Average	1281.73	1290.64	1275.58	19.70	0.52	0.97	0.22	175.15	0.54	0.96	0.22	341.75

Table A.8: Sensitivity analysis on the performance of the Lagrangian ILS with ψ when $\gamma = 2$ and $w = 1$ for large instances from category "NoTeam Reduced"

	$\psi = 5$			$\psi = 50$			$\psi = 150$			$\psi = 400$		
Instances	Average	Best	sec_a	$\%_a$	$\%_b$	sec_a	$\%_a$	$\%_b$	sec_a	$\%_a$	$\%_b$	sec_a
C101 5x4	5647.95	5620.91	52.80	0.46	0.59	43.60	0.55	0.54	35.20	-0.10	0.59	21.40
C103 5x4	2710.20	2641.89	54.60	2.01	1.46	39.20	-1.45	-1.22	32.20	-1.80	-0.39	26.00
C201 5x4	2755.52	2755.52	23.40	0.00	0.00	13.80	0.00	0.00	14.80	0.00	0.00	12.80
C203 5x4	2383.07	2375.99	26.20	-0.21	0.00	16.60	-0.58	-0.43	14.20	-0.31	0.00	16.20
R101 5x4	5634.11	5561.83	66.00	1.27	0.19	46.40	0.72	0.12	41.20	1.00	0.06	28.80
R103 5x4	1885.96	1663.30	62.20	6.86	-0.54	45.00	0.74	-9.34	40.40	3.37	-0.22	38.80
R201 5x4	2838.89	2838.50	48.20	-0.45	0.00	20.60	-0.65	0.00	18.40	-1.19	-0.63	15.20
R203 5x4	2339.05	2335.76	22.80	-0.42	-0.16	14.20	-0.05	0.15	12.60	-0.10	0.15	14.40
RC101 5x4	4893.14	4789.81	46.40	-0.01	-0.83	33.00	-0.14	-0.24	27.80	-0.04	0.05	24.60
RC103 5x4	2301.10	2245.02	38.80	0.40	-0.73	32.00	0.74	-0.41	31.80	-1.86	-1.40	25.20
RC201 5x4	3080.56	3072.11	40.00	0.03	0.05	23.80	-0.09	-0.30	15.00	-0.03	-0.09	14.60
RC203 5x4	2521.63	2512.64	29.40	0.28	0.05	18.80	0.05	0.00	15.80	0.11	0.05	17.80
C101 6x6	7660.86	7660.86	60.80	0.00	0.00	46.20	0.00	0.00	36.20	-0.26	0.00	19.40
C103 6x6	4914.46	4818.71	54.40	0.56	0.29	41.00	-0.81	-2.26	33.80	-0.43	-0.40	25.20
C201 6x6	3284.40	3278.07	48.40	0.19	0.00	36.40	0.04	0.00	31.20	-0.44	0.00	22.80
C203 6x6	2460.25	2451.92	55.80	-0.20	0.09	36.80	0.19	0.07	27.00	-0.26	-0.32	26.80
R101 6x6	5966.29	5948.79	73.80	-0.51	0.06	50.80	-0.50	-0.37	50.20	-0.89	0.07	28.20
R103 6x6	2268.38	2223.06	77.80	-1.86	0.24	51.00	-0.50	0.27	51.00	-0.97	-0.36	39.80
R201 6x6	3536.10	3440.32	70.80	0.12	0.00	43.00	-0.65	-3.23	38.00	-0.49	-2.00	26.20
R203 6x6	2452.33	2445.06	44.60	-0.41	0.00	36.40	-0.45	0.08	25.00	0.11	0.08	23.80
RC101 6x6	4999.90	4926.10	53.80	1.13	0.46	42.20	0.37	0.46	34.40	1.03	0.29	25.80
RC103 6x6	2259.48	2192.21	55.00	5.51	4.19	36.40	0.43	2.15	31.60	1.33	2.15	36.60
RC201 6x6	4481.43	4448.62	79.00	-0.18	-0.12	52.20	0.51	0.58	41.40	0.32	0.58	29.20
RC203 6x6	2664.52	2649.69	61.20	0.25	0.10	40.00	0.23	0.00	25.60	-0.46	-0.49	27.40
C101 7x4	5245.47	5241.64	35.20	0.06	0.00	29.40	0.02	-0.01	22.60	-0.31	0.00	12.60
C103 7x4	2033.07	1971.26	41.60	1.27	-0.48	23.00	0.91	0.02	19.80	-1.67	-0.19	17.80
C201 7x4	2773.41	2773.41	37.00	-0.20	0.00	19.80	-0.47	0.00	11.60	-0.21	0.00	11.80
C203 7x4	2272.64	2261.33	34.20	0.03	0.00	16.80	-0.33	0.00	13.20	-0.78	0.00	13.80
R101 7x4	5326.33	5269.67	46.20	2.22	2.77	31.20	2.29	3.16	28.80	0.44	0.07	17.60
R103 7x4	2239.31	2130.03	36.20	2.14	1.18	21.60	-0.40	-4.44	20.80	0.80	-0.33	20.00
R201 7x4	2659.95	2657.34	29.60	-0.07	0.00	17.00	-0.28	-0.16	11.60	-0.39	-0.16	11.00
R203 7x4	2204.45	2199.10	22.00	-0.04	-0.02	13.40	-0.27	-0.16	12.40	-0.35	-0.49	10.60
RC101 7x4	5495.16	5366.12	31.80	0.82	0.00	22.40	1.19	-0.13	21.40	1.36	-0.13	16.80
RC103 7x4	2663.83	2586.03	31.20	2.74	0.00	24.00	1.82	0.00	20.60	-0.89	-0.21	18.40
RC201 7x4	2916.85	2912.38	25.60	0.06	0.06	15.80	-0.14	0.06	11.80	-0.25	0.06	10.20
RC203 7x4	2293.54	2279.74	26.40	-0.14	-0.06	18.20	-0.77	-1.29	11.80	-0.49	0.09	15.80
Average	3446.21	3404.02	45.64	0.66	0.25	30.89	0.06	-0.45	25.87	-0.14	-0.10	21.21

Table A.9: Sensitivity analysis on the performance of the Lagrangian ILS with γ when $\psi = 50$ and $w = 1$ for large instances from category "NoTeam Reduced"

	$\gamma = 0.5$			$\gamma = 2$			$\gamma = 10$			$\gamma = 100$		
Instances	Average	Best	sec_a	$\%_a$	$\%_b$	sec_a	$\%_a$	$\%_b$	sec_a	$\%_a$	$\%_b$	sec_a
C101 5x4	5675.01	5627.26	52.20	0.93	0.70	43.60	0.86	0.70	37.40	0.57	0.52	42.40
C103 5x4	2675.17	2628.91	44.60	0.72	0.97	39.20	-2.84	-1.63	32.60	-1.97	-0.08	35.00
C201 5x4	2755.52	2755.52	19.20	0.00	0.00	13.80	0.00	0.00	13.60	0.00	0.00	13.80
C203 5x4	2379.96	2375.99	17.80	-0.34	0.00	16.60	-0.49	-0.42	16.80	-0.85	-0.68	16.60
R101 5x4	5585.98	5570.15	68.80	0.42	0.34	46.40	0.90	2.21	35.00	0.46	0.35	32.00
R103 5x4	1852.00	1766.49	55.00	5.15	5.34	45.00	-0.45	-1.32	35.00	-0.99	-0.23	36.60
R201 5x4	2849.98	2838.50	28.00	-0.06	0.00	20.60	0.27	0.00	23.60	0.15	0.00	22.60
R203 5x4	2340.71	2332.23	14.60	-0.35	-0.31	14.20	-0.54	-0.15	15.60	-0.40	-0.31	14.80
RC101 5x4	4928.17	4903.95	54.00	0.70	1.52	33.00	-0.54	-0.25	28.60	-0.39	-0.05	22.40
RC103 5x4	2317.98	2288.88	46.60	1.12	1.20	32.00	1.18	1.42	27.00	-2.89	-0.98	28.40
RC201 5x4	3084.78	3082.46	23.60	0.17	0.39	23.80	-0.18	0.00	20.60	-0.31	0.00	20.40
RC203 5x4	2527.03	2511.29	17.40	0.49	0.00	18.80	0.23	0.00	19.20	0.28	0.00	17.80
C101 6x6	7660.86	7660.86	64.00	0.00	0.00	46.20	0.00	0.00	55.40	0.00	0.00	48.00
C103 6x6	4851.47	4798.15	50.80	-0.73	-0.14	41.00	-2.18	-2.78	49.60	-1.85	-0.61	44.80
C201 6x6	3293.59	3278.07	38.20	0.47	0.00	36.40	0.47	0.00	39.60	0.35	0.00	38.60
C203 6x6	2456.72	2450.24	33.00	-0.34	0.02	36.80	-0.23	-0.06	43.00	0.06	0.00	40.80
R101 6x6	5972.56	5970.73	84.80	-0.41	0.43	50.80	-1.41	0.43	44.80	-0.61	0.43	42.00
R103 6x6	2339.89	2258.82	62.80	1.25	1.82	51.00	3.10	1.92	45.60	3.19	1.89	41.40
R201 6x6	3542.39	3445.87	51.20	0.30	0.16	43.00	-0.14	-2.90	48.40	-0.13	-0.84	42.80
R203 6x6	2445.72	2437.28	32.20	-0.68	-0.32	36.40	-0.72	-0.29	30.20	-0.78	-0.29	25.20
RC101 6x6	4993.84	4917.09	62.40	1.01	0.28	42.20	0.59	-0.56	31.80	1.38	1.67	31.20
RC103 6x6	2252.08	2213.38	45.20	5.20	5.11	36.40	1.82	3.78	37.00	-1.01	1.14	36.40
RC201 6x6	4512.52	4477.23	55.20	0.51	0.52	52.20	0.03	0.15	59.20	-0.55	0.15	51.20
RC203 6x6	2658.28	2646.91	33.80	0.01	0.00	40.00	-0.37	-0.57	39.20	-0.50	-0.79	41.20
C101 7x4	5262.16	5242.08	39.00	0.38	0.01	29.40	0.33	0.00	24.80	0.30	0.01	23.40
C103 7x4	1984.48	1968.08	28.40	-1.14	-0.64	23.00	-5.19	-1.40	25.60	-3.33	-0.06	21.40
C201 7x4	2774.01	2773.41	25.60	-0.18	0.00	19.80	-0.03	0.00	19.60	0.02	0.00	19.60
C203 7x4	2277.70	2262.00	21.60	0.25	0.03	16.80	-0.54	-0.70	15.80	0.12	0.03	15.80
R101 7x4	5225.79	5103.09	45.40	0.34	-0.40	31.20	-0.38	0.46	24.60	-0.11	-0.40	25.80
R103 7x4	2166.33	2110.80	37.60	-1.15	0.28	21.60	-7.19	-4.71	20.20	-6.11	-6.04	16.20
R201 7x4	2662.69	2657.34	17.00	0.04	0.00	17.00	0.05	0.00	17.20	-0.09	0.00	17.40
R203 7x4	2204.81	2199.10	14.40	-0.03	-0.02	13.40	0.00	0.00	12.40	-0.15	-0.08	15.20
RC101 7x4	5494.07	5477.84	36.20	0.80	2.04	22.40	0.33	1.70	18.20	0.08	0.00	15.60
RC103 7x4	2611.57	2586.03	28.40	0.80	0.00	24.00	-1.20	-0.21	16.20	-1.33	0.00	20.00
RC201 7x4	2913.48	2905.21	21.60	-0.06	-0.19	15.80	-0.20	-0.19	16.00	-0.03	-0.19	17.00
RC203 7x4	2285.22	2277.62	18.60	-0.50	-0.15	18.20	-0.67	-0.34	15.20	-0.70	-0.09	14.80
Average	3439.29	3411.08	38.59	0.42	0.53	30.89	-0.43	-0.16	29.29	-0.50	-0.15	28.02

$\mathcal{B}$

Mathematical models

B.1 Weighted sum three-index model for MASPDP

The mixed integer program presented below is used to test the performance of CPLEX with the three-index model. It is modified from the problems (4.1)−(4.23) and (4.24)−(4.25) using weighted sum where λ_1 is the weight for objective function (4.1) and λ_2 is the weight for objective function (4.24).

$$\max \lambda_1 \sum_{i \in T} \sum_{j \in C} \eta_j^i - \lambda_2 \left(\sum_{k \in T} \sum_{(i,j) \in A_C} t_{i,j} x_{i,j}^k + \sum_{i \in T} \sum_{j \in C} t_{0,j} \gamma_j^i \right) \tag{B.1}$$

subject to:

$$(4.2) - (4.23)$$

References

Europe's demographic future: facts and figures on challenges and opportunities. *Office for Official Publications of the European Communities, Luxembourg*, 2007.

Saurabh Agrawal, Rajesh K Singh, and Qasim Murtaza. A literature review and perspectives in reverse logistics. *Resources, Conservation and Recycling*, 97:76–92, 2015.

Juan J Alcaraz, Luis Caballero-Arnaldos, and Javier Vales-Alonso. Rich vehicle routing problem with last-mile outsourcing decisions. *Transportation Research Part E: Logistics and Transportation Review*, 129:263–286, 2019.

Haneen Algethami and Dario Landa-Silva. Diversity-based adaptive genetic algorithm for a workforce scheduling and routing problem. *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 1771–1778. IEEE, 2017.

Haneen Algethami, Rodrigo Lankaites Pinheiro, and Dario Landa-Silva. A genetic algorithm for a workforce scheduling and routing problem. *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 927–934. IEEE, 2016.

Haneen Algethami, Anna Martínez-Gavara, and Dario Landa-Silva. Adaptive multiple crossover genetic algorithm to solve workforce scheduling and routing problem. *Journal of Heuristics*, 25(4):753–792, 2019.

Samira Almoustafa, Said Hanafi, and Nenad Mladenović. New exact method for large asymmetric distance-constrained vehicle routing problem. *European Journal of Operational Research*, 226(3):386–394, 2013.

Claudia Archetti and M Grazia Speranza. A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*, 2(4):223–246, 2014.

Claudia Archetti and Maria Grazia Speranza. The split delivery vehicle routing problem: A survey. In *The vehicle routing problem: Latest advances and new challenges*, pages 103–122. Springer, 2008.

Claudia Archetti, Dominique Feillet, Alain Hertz, and Maria Grazia Speranza. The capacitated team orienteering and profitable tour problems. *Journal of the Operational Research Society*, 60:831–842, 2009.

Florian Arnold, Michel Gendreau, and Kenneth Sörensen. Efficiently solving very large-scale routing problems. *Computers & Operations Research*, 107:32–42, 2019.

Mustafa Avci and Seyda Topaloglu. A hybrid metaheuristic algorithm for heterogeneous vehicle routing problem with simultaneous pickup and delivery. *Expert Systems with Applications*, 53:160–171, 2016.

Roberto Baldacci, Maria Battarra, and Daniele Vigo. Routing a heterogeneous fleet of vehicles. In *The vehicle routing problem: latest advances and new challenges*, pages 3–27. Springer, 2008.

Roberto Baldacci, Maria Battarra, and Daniele Vigo. Valid inequalities for the fleet size and mix vehicle routing problem with fixed costs. *Networks: An International Journal*, 54(4):178–189, 2009.

Tolga Bektaş and Gilbert Laporte. The pollution-routing problem. *Transportation Research Part B: Methodological*, 45(8):1232–1250, 2011.

Gerardo Berbeglia, Jean-François Cordeau, Irina Gribkovskaia, and Gilbert Laporte. Static pickup and delivery problems: a classification scheme and survey. *Top*, 15(1): 1–31, 2007.

Andrea Bettinelli, Alberto Ceselli, and Giovanni Righini. A branch-and-price algorithm for the multi-depot heterogeneous-fleet pickup and delivery problem with soft time windows. *Mathematical Programming Computation*, 6(2):171–197, 2014.

Andre Bevilaqua, Diego Bevilaqua, and Keiji Yamanaka. Parallel island based memetic algorithm with lin–kernighan local search for a real-life two-echelon heterogeneous vehicle routing problem based on brazilian wholesale companies. *Applied Soft Computing*, 76:697–711, 2019.

John R Birge and Francois Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.

Yossi Borenstein, Nazaraf Shah, Edward Tsang, Raphael Dorne, Abdullah Alsheddy, and Christos Voudouris. On the partitioning of dynamic workforce scheduling problems. *Journal of Scheduling*, 13(4):411–425, 2010.

Andreas Bortfeldt. A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints. *Computers & Operations Research*, 39(9):2248–2257, 2012.

Khaoula Bouanane, Mohammed El Amrani, and Youssef Benadada. The vehicle routing problem with simultaneous delivery and pickup: a taxonomic survey. *International Journal of Logistics Systems and Management*, 41(1-2):77–119, 2022.

Mouaouia Cherif Bouzid, Hacene Aït Haddadene, and Said Salhi. An integration of lagrangian split and vns: The case of the capacitated vehicle routing problem. *Computers & Operations Research*, 78:513–525, 2017.

Nils Boysen, Stefan Fedtke, and Stefan Schwerdfeger. Last-mile delivery concepts: a survey from an operational research perspective. *OR Spectrum*, 43(1):1–58, 2021.

Kris Braekers, Richard F Hartl, Sophie N Parragh, and Fabien Tricoire. A bi-objective home care scheduling problem: Analyzing the trade-off between costs and client inconvenience. *European Journal of Operational Research*, 248(2):428–443, 2016a.

Kris Braekers, Katrien Ramaekers, and Inneke Van Nieuwenhuyse. The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99:300–313, 2016b.

José Brandão. A tabu search algorithm for the open vehicle routing problem. *European Journal of Operational Research*, 157(3):552–564, 2004.

Olli Bräysy and Michel Gendreau. Vehicle routing problem with time windows, part i: Route construction and local search algorithms. *Transportation Science*, 39(1):104–118, 2005a.

Olli Bräysy and Michel Gendreau. Vehicle routing problem with time windows, part ii: Metaheuristics. *Transportation Science*, 39(1):119–139, 2005b.

Jose Caceres-Cruz, Pol Arias, Daniel Guimarans, Daniel Riera, and Angel A Juan. Rich vehicle routing problem: Survey. *ACM Computing Surveys (CSUR)*, 47(2):1–28, 2014.

J Arturo Castillo-Salazar, Dario Landa-Silva, and Rong Qu. Workforce scheduling and routing problems: literature survey and computational study. *Annals of Operations Research*, 239(1):39–67, 2016.

Juan Castro-Gutierrez. *Multi-objective tools for the vehicle routing problem with time windows*. PhD thesis, University of Nottingham, 2012.

Juan Castro-Gutierrez, Dario Landa-Silva, and José Moreno Pérez. Nature of real-world multi-objective vehicle routing with evolutionary algorithms. In *2011 IEEE International Conference on Systems, Man, and Cybernetics*, pages 257–264. IEEE, 2011.

Diego Cattaruzza, Nabil Absi, and Dominique Feillet. Vehicle routing problems with multiple trips. *Annals of Operations Research*, 271(1):127–159, 2018.

Alberto Ceselli, Giovanni Righini, and Matteo Salani. A column generation algorithm for a rich vehicle-routing problem. *Transportation Science*, 43(1):56–69, 2009.

Binhui Chen, Rong Qu, Ruibin Bai, and Wasakorn Laesanklang. A variable neighborhood search algorithm with reinforcement learning for a real-life periodic vehicle routing problem with time windows and open routes. *RAIRO-Operations Research*, 54(5): 1467–1494, 2020.

Cen Chen, Zachary Rubinstein, Stephen Smith, and Hoong Chuin Lau. Tackling large-scale home health care delivery problem with uncertainty. *27th International Conference on Automated Planning and Scheduling (ICAPS), AAAI Press*, 27:358–366, 2017.

Wen-Chyuan Chiang and Robert A Russell. A metaheuristic for the vehicle-routeing problem with soft time windows. *Journal of the Operational Research Society*, 55(12): 1298–1310, 2004.

Geoff Clarke and John W Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581, 1964.

Leandro C Coelho, Jean-Philippe Gagliardi, Jacques Renaud, and Angel Ruiz. Solving the vehicle routing problem with lunch break arising in the furniture delivery industry. *Journal of the Operational Research Society*, 67(5):743–751, 2016.

Carlos A Coello Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, 1(3):269–308, 1999.

William Jay Conover. Practical Nonparametric Statistics. *John Wiley & Sons, New York*, 1999.

Jean François Cordeau and Mirko Maischberger. A parallel iterated tabu search heuristic for vehicle routing problems. *Computers & Operations Research*, 39(9):2033–2050, 2012.

Jean François Cordeau, Michel Gendreau, and Gilbert Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks: An International Journal*, 30(2):105–119, 1997.

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.

Jean François Cordeau, Gilbert Laporte, Federico Pasin, and Stefan Ropke. Scheduling technicians and tasks in a telecommunications company. *Journal of Scheduling*, 13(4): 393–409, 2010.

G.B. Dantzig and J.H. Ramser. The truck dispatching problem. *Management Science*, 6 (1):80 – 91, 1959.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

Ai-min DENG, MAO Chao, and Yan-ting ZHOU. Optimizing research of an improved simulated annealing algorithm to soft time windows vehicle routing problem with pick-up and delivery. *Systems Engineering-Theory & Practice*, 29(5):186–192, 2009.

Jan Dethloff. Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *OR Spectrum*, 23(1):79–96, 2001.

Karl F Doerner and Verena Schmid. Survey: matheuristics for rich vehicle routing problems. In *International Workshop on Hybrid Metaheuristics*, pages 206–221. Springer, 2010.

Anders Dohn, Esben Kolind, and Jens Clausen. The manpower allocation problem with time windows and job-teaming constraints: A branch-and-price approach. *Computers & Operations Research*, 36(4):1145–1157, 2009.

Jing Fan. The vehicle routing problem with simultaneous pickup and delivery based on customer satisfaction. *Procedia Engineering*, 15:5284–5289, 2011.

AM Fathollahi-Fard, M Hajiaghaei-Keshteli, and R Tavakkoli-Moghaddam. A Lagrangian relaxation-based algorithm to solve a home health care routing problem. *International Journal of Engineering*, 31(10):1734–1740, 2018.

Christian Fikar and Patrick Hirsch. Home health care routing and scheduling: A review. *Computers & Operations Research*, 77:86–95, 2017.

Peter C Fishburn. Exceptional paper-lexicographic orders, utilities and decision rules: A survey. *Management science*, 20(11):1442–1471, 1974.

Marshall L Fisher. The Lagrangian relaxation method for solving integer programming problems. *Management Science*, 27(1):1–18, 1981.

Filippo Focacci, François Laburthe, and Andrea Lodi. Local search and constraint programming. In *Handbook of Metaheuristics*, pages 369–403. Springer, 2003.

Véronique François, Yasemin Arda, and Yves Crama. Adaptive large neighborhood search for multitrip vehicle routing with time windows. *Transportation Science*, 53(6):1706–1730, 2019.

Antonio Frangioni. About lagrangian methods in integer optimization. *Annals of Operations Research*, 139:163–193, 2005.

Zhuo Fu, Richard Eglese, and Leon YO Li. A new tabu search heuristic for the open vehicle routing problem. *Journal of the Operational Research Society*, 56(3):267–274, 2005.

Zhuo Fu, Richard Eglese, and Leon YO Li. A unified tabu search algorithm for vehicle routing problems with soft time windows. *Journal of the Operational Research Society*, 59(5):663–673, 2008.

Y. Gajpal and P. Abad. Saving-based algorithms for vehicle routing problem with simultaneous pickup and delivery. *Journal of the Operational Research Society*, 61(10): 1498–1509, 2010.

Xiaobing Gan, Yan Wang, Shuhai Li, and Ben Niu. Vehicle routing problem with time windows and simultaneous delivery and pick-up service based on mcpso. *Mathematical Problems in Engineering*, 2012, 2012.

Margaretha Gansterer, Murat Küçüktepe, and Richard F Hartl. The multi-vehicle profitable pickup and delivery problem. *Or Spectrum*, 39:303–319, 2017.

Michael R Garey and David S Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. *Freeman, San Francisco*, 1979.

Michel Gendreau and Jean Yves Potvin. Tabu search. *Handbook of Metaheuristics*, pages 41–59. Springer, 2019.

Michel Gendreau, Gilbert Laporte, and René Séguin. An exact algorithm for the vehicle routing problem with stochastic demands and customers. *Transportation Science*, 29 (2):143–155, 1995.

Michel Gendreau, Ola Jabali, and Walter Rei. Chapter 8: Stochastic vehicle routing problems. In *Vehicle Routing: Problems, Methods, and Applications, Second Edition*, pages 213–239. SIAM, 2014.

Michel Gendreau, Ola Jabali, and Walter Rei. 50th anniversary invited articlefuture research directions in stochastic vehicle routing. *Transportation Science*, 50(4):1163–1173, 2016.

Arthur M Geoffrion. Lagrangean relaxation for integer programming. In *Approaches to integer programming*, pages 82–114. Springer, 1974.

Roel Gevaers, Eddy Van de Voorde, and Thierry Vanelslander. Characteristics and typology of last-mile logistics from an innovation perspective in an urban context. In *City distribution and urban freight transport*. Edward Elgar Publishing, 2011.

Billy E Gillett and Leland R Miller. A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2):340–349, 1974.

Asvin Goel and Frank Meisel. Workforce routing and scheduling for electricity network maintenance with downtime minimization. *European Journal of Operational Research*, 231(1):210–228, 2013.

Marc Goetschalckx and Charlotte Jacobs-Blecha. The vehicle routing problem with backhauls. *European Journal of Operational Research*, 42(1):39–51, 1989.

Bruce Golden, Arjang Assad, Larry Levy, and Filip Gheysens. The fleet size and mix vehicle routing problem. *Computers & Operations Research*, 11(1):49–66, 1984.

Bruce L Golden, Subramanian Raghavan, Edward A Wasil, et al. *The vehicle routing problem: latest advances and new challenges*. Springer, 2008.

Kannan Govindan, Hamed Soleimani, and Devika Kannan. Reverse logistics and closed-loop supply chain: A comprehensive review to explore the future. *European Journal of Operational Research*, 240(3):603–626, 2015.

Hanyu Gu, Yefei Zhang, and Yakov Zinder. Lagrangian relaxation in iterated local search for the workforce scheduling and routing problem. *International Symposium on Experimental Algorithms (SEA)*, pages 527–540. Springer, 2019.

Hanyu Gu, Lucy MacMillan, Yefei Zhang, and Yakov Zinder. Iterated local search with neighbourhood reduction for the pickups and deliveries problem arising in retail industry. In *Optimization and Learning: 4th International Conference, OLA 2021, Catania, Italy, June 21-23, 2021, Proceedings*, pages 190–202. Springer, 2021.

Hanyu Gu, Hue Chi Lam, and Yakov Zinder. A hybrid genetic algorithm for scheduling jobs sharing multiple resources under uncertainty. *EURO Journal on Computational Optimization*, 10:100050, 2022a.

Hanyu Gu, Yefei Zhang, and Yakov Zinder. An efficient optimisation procedure for the workforce scheduling and routing problem: Lagrangian relaxation and iterated local search. *Computers & Operations Research*, 144:105829, 2022b.

G Guastaroba, J F Côté, and LC Coelho. The multi-period workforce scheduling and routing problem. *Omega*, 102:102302, 2021.

Monique Guignard. Lagrangean relaxation. *Top*, 11:151–200, 2003.

Elin E Halvorsen-Weare and Martin WP Savelsbergh. The bi-objective mixed capacitated general routing problem with different route balance criteria. *European Journal of Operational Research*, 251(2):451–465, 2016.

Pierre Hansen, Nenad Mladenović, Raca Todosijević, and Saïd Hanafi. Variable neighborhood search: basics and variants. *EURO Journal on Computational Optimization*, 5(3):423–454, 2017.

Pierre Hansen, Nenad Mladenović, Jack Brimberg, and José A Moreno Pérez. Variable neighborhood search. In *Handbook of Metaheuristics*, pages 57–97. Springer, 2019.

Gerhard Hiermann, Jakob Puchinger, Stefan Ropke, and Richard F Hartl. The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *European Journal of Operational Research*, 252(3):995–1018, 2016.

Richard P Hornstra, Allyson Silva, Kees Jan Roodbergen, and Leandro C Coelho. The vehicle routing problem with simultaneous pickup and delivery and handling costs. *Computers & Operations Research*, 115:104858, 2020.

Toshihide Ibaraki, Shinji Imahori, Koji Nonobe, Kensuke Sobue, Takeaki Uno, and Mutsunori Yagiura. An iterated local search algorithm for the vehicle routing problem with convex time penalty functions. *Discrete Applied Mathematics*, 156(11):2050–2069, 2008.

Leonie M Johannsmann, Emily M Craparo, Thor L Dieken, Armin R Fügenschuh, and Björn O Seitner. Stochastic mixed-integer programming for a spare parts inventory management problem. *Computers & Operations Research*, 138:105568, 2022.

Nicolas Jozefowiez, Frédéric Semet, and El-Ghazali Talbi. Multi-objective vehicle routing problems. *European Journal of Operational Research*, 189(2):293–309, 2008.

Brian Kallehauge, Jesper Larsen, Oli BG Madsen, and Marius M Solomon. Vehicle routing problem with time windows. In *Column generation*, pages 67–98. Springer, 2005.

Kittitach Kamsopa, Kanchana Sethanan, Thitipong Jamrus, and Liliana Czwajda. Hybrid genetic algorithm for multi-period vehicle routing problem with mixed pickup and delivery with time window, heterogeneous fleet, duration time and rest area. *Engineering Journal*, 25(10):71–86, 2021.

Sally Kassem and Mingyuan Chen. Solving reverse logistics vehicle routing problems with time windows. *The International Journal of Advanced Manufacturing Technology*, 68 (1):57–68, 2013.

Bariş Keçeci, Fulya Altıparmak, and Imdat Kara. A mathematical formulation and heuristic approach for the heterogeneous fixed fleet vehicle routing problem with simultaneous pickup and delivery. *Journal of Industrial & Management Optimization*, 17(3):1069, 2021.

Merve Keskin, Bülent Çatay, and Gilbert Laporte. A simulation-based heuristic for the electric vehicle routing problem with time windows and stochastic waiting times at recharging stations. *Computers & Operations Research*, 125:105060, 2021.

Gerard A. P. Kindervater and Martin W. P. Savelsbergh. Vehicle routing: handling edge exchanges. *Local Search in Combinatorial Optimization*, pages 337–360. Princeton University Press, 2018.

Anton J Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

John Kloke and Joseph W McKean. Nonparametric Statistical Methods Using R. *CRC Press, Boca Raton*, 2015.

Çağrı Koç, Tolga Bektaş, Ola Jabali, and Gilbert Laporte. Thirty years of heterogeneous vehicle routing. *European Journal of Operational Research*, 249(1):1–21, 2016.

Çağrı Koç, Gilbert Laporte, and İlknur Tükenmez. A review of vehicle routing with simultaneous pickup and delivery. *Computers & Operations Research*, 122:104987, 2020.

Grigorios D Konstantakopoulos, Sotiris P Gayialis, and Evripidis P Kechagias. Vehicle routing problem and related algorithms for logistics distribution: a literature review and classification. *Operational Research*, 22:2033–2062, 2020.

Attila A Kovacs, Sophie N Parragh, Karl F Doerner, and Richard F Hartl. Adaptive large neighborhood search for service technician routing and scheduling problems. *Journal of Scheduling*, 15(5):579–600, 2012.

Raphael Kramer, Anand Subramanian, Thibaut Vidal, and F Cabral Lucídio dos Anjos. A matheuristic approach for the pollution-routing problem. *European Journal of Operational Research*, 243(2):523–539, 2015.

Raphael Kramer, Jean-François Cordeau, and Manuel Iori. Rich vehicle routing with auxiliary depots and anticipated deliveries: An application to pharmaceutical distribution. *Transportation Research Part E: Logistics and Transportation Review*, 129: 162–174, 2019.

Wasakorn Laesanklang, Dario Landa-Silva, and J Arturo Castillo-Salazar. Mixed integer programming with decomposition to solve a workforce scheduling and routing problem. *4th International Conference on Operations Research and Enterprise Systems (ICORES)*, pages 283–293. Springer, 2015.

Wasakorn Laesanklang, Dario Landa-Silva, and J Arturo Castillo-Salazar. Mixed integer programming with decomposition for workforce scheduling and routing with time-dependent activities constraints. *5th International Conference on Operations Research and Enterprise Systems (ICORES)*, pages 330–339. Springer, 2016.

Rahma Lahyani, Mahdi Khemakhem, and Frédéric Semet. Rich vehicle routing problems: From a taxonomy to a definition. *European Journal of Operational Research*, 241(1): 1–14, 2015.

Gilbert Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, 2009.

Gilbert Laporte, Michel Gendreau, Jean Yves Potvin, and Frédéric Semet. Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research*, 7(4-5):285–300, 2000.

Gilbert Laporte, François V Louveaux, and Luc Van Hamme. An integer l-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research*, 50(3):415–423, 2002.

Hongtao Lei, Gilbert Laporte, and Bo Guo. A generalized variable neighborhood search heuristic for the capacitated vehicle routing problem with stochastic service times. *Top*, 20(1):99–118, 2012.

Claude Lemaréchal. Lagrangian relaxation. *Computational Combinatorial Optimization: Optimal Or Provably Near-optimal Solutions*, pages 112–156. Springer, 2001.

Adam N Letchford, Jens Lysgaard, and Richard W Eglese. A branch-and-cut algorithm for the capacitated open vehicle routing problem. *Journal of the Operational Research Society*, 58(12):1642–1651, 2007.

Feiyue Li, Bruce Golden, and Edward Wasil. The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. *Computers & Operations Research*, 34(10):2918–2930, 2007.

Haibing Li and Andrew Lim. A metaheuristic for the pickup and delivery problem with time windows. *International Journal on Artificial Intelligence Tools*, 12(02):173–186, 2003.

Jian Li, Panos M Pardalos, Hao Sun, Jun Pei, and Yong Zhang. Iterated local search embedded adaptive neighborhood selection approach for the multi-depot vehicle routing problem with simultaneous deliveries and pickups. *Expert Systems with Applications*, 42(7):3551–3561, 2015.

Xiangyong Li, Stephen CH Leung, and Peng Tian. A multistart adaptive memory-based tabu search algorithm for the heterogeneous fixed fleet open vehicle routing problem. *Expert Systems with Applications*, 39(1):365–374, 2012.

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.

Stanley Frederick WT Lim, Xin Jin, and Jagjit Singh Srai. Consumer-driven e-commerce: A literature review, design framework, and research agenda on last-mile logistics models. *International Journal of Physical Distribution & Logistics Management*, 48(3):308–332, 2018.

Ran Liu, Xiaolan Xie, Vincent Augusto, and Carlos Rodriguez. Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. *European Journal of Operational Research*, 230(3):475–486, 2013.

Ran Liu, Biao Yuan, and Zhibin Jiang. Mathematical model and exact algorithm for the home care worker scheduling and routing problem with lunch break requirements. *International Journal of Production Research*, 55(2):558–575, 2017.

Helena Ramalhinho Lourenço, Olivier C Martin, and Thomas Stützle. Iterated local search: Framework and applications. *Handbook of Metaheuristics*, pages 129–168. Springer, 2019.

Hong Ma, Brenda Cheang, Andrew Lim, Lei Zhang, and Yi Zhu. An investigation into the vehicle routing problem with time windows and link capacity constraints. *Omega*, 40(3):336–347, 2012.

Rafael Martí. Multi-start methods. In *Handbook of Metaheuristics*, pages 355–368. Springer, 2003.

Sara Martins, Manuel Ostermeier, Pedro Amorim, Alexander Hübner, and Bernardo Almada-Lobo. Product-oriented time window assignment for a multi-compartment vehicle routing problem. *European Journal of Operational Research*, 276(3):893–909, 2019.

Hokey Min. The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General*, 23(5):377–386, 1989.

Lai Mingyong and Cao Erbao. An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows. *Engineering Applications of Artificial Intelligence*, 23(2):188 – 195, 2010.

Fermín Alfredo Tang Montané and Roberto Diéguez Galvao. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research*, 33(3):595–619, 2006.

IlKyeong Moon, Jeong-Hun Lee, and June Seong. Vehicle routing problem with time windows considering overtime and outsourcing vehicles. *Expert Systems with Applications*, 39(18):13202–13213, 2012.

Gur Mosheiov. Vehicle routing with pick-up and delivery: tour-partitioning heuristics. *Computers & Industrial Engineering*, 34(3):669–684, 1998.

Yuichi Nagata, Olli Bräysy, and Wout Dullaert. A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research*, 37(4):724–737, 2010.

Gabor Nagy and Saıd Salhi. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research*, 162(1):126–141, 2005.

Gábor Nagy, Niaz A Wassan, M Grazia Speranza, and Claudia Archetti. The vehicle routing problem with divisible deliveries and pickups. *Transportation Science*, 49(2): 271–294, 2013.

Napoleão Nepomuceno, Ricardo Barboza Saboia, and Plácido Rogério Pinheiro. A fast randomized algorithm for the heterogeneous vehicle routing problem with simultaneous pickup and delivery. *Algorithms*, 12(8):158, 2019.

Peter Nesbitt, Lewis R Blake, Patricio Lamas, Marcos Goycoolea, Bernardo K Pagnoncelli, Alexandra Newman, and Andrea Brickey. Underground mine scheduling under uncertainty. *European Journal of Operational Research*, 294(1):340–352, 2021.

Jorge Oyola, Halvard Arntzen, and David L Woodruff. The stochastic vehicle routing problem, a literature review, part i: models. *EURO Journal on Transportation and Logistics*, 7(3):193–221, 2018.

Tayfun Öztaş and Ayşegül Tuş. A hybrid metaheuristic algorithm based on iterated local search for vehicle routing problem with simultaneous pickup and delivery. *Expert Systems with Applications*, 202:117401, 2022.

Binbin Pan, Zhenzhen Zhang, and Andrew Lim. A hybrid algorithm for time-dependent vehicle routing problem with time windows. *Computers & Operations Research*, 128: 105193, 2021.

Dimitris C Paraskevopoulos, Gilbert Laporte, Panagiotis P Repoussis, and Christos D Tarantilis. Resource constrained routing and scheduling: Review and research prospects. *European Journal of Operational Research*, 263(3):737–754, 2017.

Sophie N Parragh, Karl F Doerner, and Richard F Hartl. A survey on pickup and delivery problems. part 1: Transportation between customers and depot. *Journal für Betriebswirtschaft*, 58:21–51, 2008a.

Sophie N Parragh, Karl F Doerner, and Richard F Hartl. A survey on pickup and delivery problems. part 2: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft*, 58:81–117, 2008b.

Fan Peng, Yanfeng Ouyang, and Kamalesh Somani. Optimal routing and scheduling of periodic inspections in large-scale railroad networks. *Journal of Rail Transport Planning & Management*, 3(4):163–171, 2013.

Puca Huachi Vaz Penna, Anand Subramanian, and Luiz Satoru Ochi. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics*, 19(2):201–232, 2013.

Dilson Lucas Pereira, Júlio César Alves, and Mayron César de Oliveira Moreira. A multiperiod workforce scheduling and routing problem with dependent tasks. *Computers & Operations Research*, 118:104930, 2020.

Victor Pillac, Christelle Gueret, and Andrés L Medaglia. A parallel matheuristic for the technician routing and scheduling problem. *Optimization Letters*, 7(7):1525–1535, 2013.

Olcay Polat. A parallel variable neighborhood search for the vehicle routing problem with divisible deliveries and pickups. *Computers & Operations Research*, 85:71–86, 2017.

Mateusz Polnik, Annalisa Riccardi, and Kerem Akartunalı. A multistage optimisation algorithm for the large vehicle routing problem with time windows and synchronised visits. *Journal of the Operational Research Society*, 72(11):2396–2411, 2021.

Alireza Rahimi-Vahed, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei. A path relinking algorithm for a multi-depot periodic vehicle routing problem. *Journal of Heuristics*, 19(3):497–524, 2013.

Sebastian Reil, Andreas Bortfeldt, and Lars Mönch. Heuristics for vehicle routing problems with backhauls, time windows, and 3d loading constraints. *European Journal of Operational Research*, 266(3):877–894, 2018.

Julia Rieck and Jürgen Zimmermann. A new mixed integer linear model for a rich vehicle routing problem with docking constraints. *Annals of Operations Research*, 181(1):337–358, 2010.

Julia Rieck and Jürgen Zimmermann. Exact solutions to the symmetric and asymmetric vehicle routing problem with simultaneous delivery and pick-up. *Business Research*, 6 (1):77–92, 2013.

Ulrike Ritzinger, Jakob Puchinger, and Richard F Hartl. A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*, 54 (1):215–231, 2016.

Robert Russell, Wen-Chyuan Chiang, and David Zepeda. Integrating multi-product production and distribution in newspaper logistics. *Computers & Operations Research*, 35 (5):1576–1588, 2008.

Robert A Russell. Hybrid heuristics for the vehicle routing problem with time windows. *Transportation Science*, 29(2):156–166, 1995.

Nasser R Sabar, Ashish Bhaskar, Edward Chung, Ayad Turky, and Andy Song. An adaptive memetic approach for heterogeneous vehicle routing problems with two-dimensional loading constraints. *Swarm and Evolutionary Computation*, 58:100730, 2020.

Said Salhi, Arif Imran, and Niaz A Wassan. The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation. *Computers & Operations Research*, 52:315–325, 2014.

Dimitrios Sariklis and Susan Powell. A heuristic method for the open vehicle routing problem. *Journal of the Operational Research Society*, 51(5):564–573, 2000.

Carlo S Sartori and Luciana S Buriol. A study on the pickup and delivery problem with time windows: Matheuristics and new instances. *Computers & Operations Research*, 124:105065, 2020.

Michael Schneider and Maximilian Löffler. Large composite neighborhoods for the capacitated location-routing problem. *Transportation Science*, 53(1):301–318, 2019.

Michael Schneider, Bastian Sand, and Andreas Stenger. A note on the time travel approach for handling time windows in vehicle routing problems. *Computers & Operations Research*, 40(10):2564–2568, 2013.

Michael Schneider, Andreas Stenger, and Dominik Goeke. The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4): 500–520, 2014.

Michel Povlovitsch Seixas and André Bergsten Mendes. Column generation for a multitrip vehicle routing problem with time windows, driver work hours, and heterogeneous fleet. *Mathematical Problems in Engineering*, 2013, 2013.

Yong Shi, Toufik Boudouh, and Olivier Grunder. A hybrid genetic algorithm for a home health care routing problem with time window and fuzzy demand. *Expert Systems with Applications*, 72:160–176, 2017.

Yong Shi, Yanjie Zhou, Toufik Boudouh, and Olivier Grunder. A lexicographic-based two-stage algorithm for vehicle routing problem with simultaneous pickup–delivery and time window. *Engineering Applications of Artificial Intelligence*, 95:103901, 2020.

Lina Simeonova, Niaz Wassan, Said Salhi, and Gábor Nagy. The heterogeneous fleet vehicle routing problem with light loads and overtime: Formulation and population variable neighbourhood search with adaptive memory. *Expert Systems with Applications*, 114: 183–195, 2018.

Lina Simeonova, Niaz Wassan, Naveed Wassan, and Said Salhi. Recent developments in real life vehicle routing problem applications. *Green Transportation and New Advances in Vehicle Routing Problems*, pages 213–228. Springer, 2020.

Marius M Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987.

Anand Subramanian, Eduardo Uchoa, Artur Alves Pessoa, and Luiz Satoru Ochi. Branch-and-cut with lazy separation for the vehicle routing problem with simultaneous pickup and delivery. *Operations Research Letters*, 39(5):338–341, 2011.

Anand Subramanian, Eduardo Uchoa, Artur Alves Pessoa, and Luiz Satoru Ochi. Branch-cut-and-price for the vehicle routing problem with simultaneous pickup and delivery. *Optimization Letters*, 7(7):1569–1581, 2013.

Ilgaz Sungur, Yingtao Ren, Fernando Ordóñez, Maged Dessouky, and Hongsheng Zhong. A model and algorithm for the courier delivery problem with uncertainty. *Transportation Science*, 44(2):193–205, 2010.

CD Tarantilis and CT Kiranoudis. Distribution of fresh meat. *Journal of Food Engineering*, 51(1):85–91, 2002.

Christos D Tarantilis, George Ioannou, Chris T Kiranoudis, and Gregory P Prastacos. Solving the open vehicle routeing problem via a single parameter metaheuristic algorithm. *Journal of the Operational Research Society*, 56(5):588–596, 2005.

Paolo Toth and Daniele Vigo. A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls. *European Journal of Operational Research*, 113(3):528–543, 1999.

Paolo Toth and Daniele Vigo. An overview of vehicle routing problems. *The vehicle routing problem*, pages 1–26. SIAM, 2002a.

Paolo Toth and Daniele Vigo. *The vehicle routing problem*. SIAM, 2002b.

Paolo Toth and Daniele Vigo. *Vehicle routing: problems, methods, and applications*. SIAM, 2014.

Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, 40(1):475–489, 2013.

Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins. A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research*, 234(3):658–673, 2014.

Chao Wang, Dong Mu, Fu Zhao, and John W Sutherland. A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup–delivery and time windows. *Computers & Industrial Engineering*, 83:111–122, 2015.

Hsiao-Fan Wang and Ying-Yen Chen. A genetic algorithm for the simultaneous delivery and pickup problems with time window. *Computers & Industrial Engineering*, 62(1): 84–95, 2012.

Jiahai Wang, Ying Zhou, Yong Wang, Jun Zhang, CL Philip Chen, and Zibin Zheng. Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: formulation, instances, and algorithms. *IEEE transactions on cybernetics*, 46(3):582–594, 2016.

David Wolfinger. A large neighborhood search for the pickup and delivery problem with time windows, split loads and transshipments. *Computers & Operations Research*, 126: 105110, 2021.

David Wolfinger and Juan-José Salazar-González. The pickup and delivery problem with split loads and transshipments: A branch-and-cut solution approach. *European Journal of Operational Research*, 289(2):470–484, 2021.

Johannes Wollenburg, Andreas Holzapfel, Alexander Hübner, and Heinrich Kuhn. Configuring retail fulfillment processes for omni-channel customer steering. *International Journal of Electronic Commerce*, 22(4):540–575, 2018.

Fulin Xie, Chris N Potts, and Tolga Bektaş. Iterated local search for workforce scheduling and routing problems. *Journal of Heuristics*, 23(6):471–500, 2017.

Jiyang Xu and Steve Y Chiu. Effective heuristic procedures for a field technician scheduling problem. *Journal of Heuristics*, 7(5):495–509, 2001.

Hande Yaman. Formulations and valid inequalities for the heterogeneous vehicle routing problem. *Mathematical Programming*, 106(2):365–390, 2006.

Hande Yaman, Oya Ekin Karasan, and Bahar Y Kara. Release time scheduling and hub location for next-day delivery. *Operations Research*, 60(4):906–917, 2012.

Zhiwei Yang, Jan-Paul van Osta, Barry van Veen, Rick van Krevelen, Richard van Klaveren, Andries Stam, Joost Kok, Thomas Bäck, and Michael Emmerich. Dynamic vehicle routing with time windows in theory and practice. *Natural Computing*, 16(1):119–134, 2017.

Baozhen Yao, Bin Yu, Ping Hu, Junjie Gao, and Mingheng Zhang. An improved particle swarm optimization for carton heterogeneous vehicle routing problem with a collection depot. *Annals of Operations Research*, 242(2):303–320, 2016.

Alkin Yurtkuran, Betul Yagmahan, and Erdal Emel. A novel artificial bee colony algorithm for the workforce scheduling and balancing problem in sub-assembly lines with limited buffers. *Applied Soft Computing*, 73:767–782, 2018.

Sandra Zajac and Sandra Huber. Objectives and methods in multi-objective routing problems: a survey and classification scheme. *European Journal of Operational Research*, 290(1):1–25, 2021.

Defu Zhang, Sifan Cai, Furong Ye, Yain-Whar Si, and Trung Thanh Nguyen. A hybrid algorithm for a vehicle routing problem with realistic constraints. *Information Sciences*, 394:167–182, 2017.

Tao Zhang, W Art Chaovalitwongse, and Yuejie Zhang. Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries. *Computers & Operations Research*, 39(10):2277–2290, 2012.

Yalan Zhou, Manhui Huang, Hong Wu, Guoming Chen, and Zhijian Wang. Iterated local search with hybrid neighborhood search for workforce scheduling and routing problem. *2020 12th International Conference on Advanced Computational Intelligence (ICACI)*, pages 478–485. IEEE, 2020.

Lin Zhu and Jiuh-Biing Sheu. Failure-specific cooperative recourse strategy for simultaneous pickup and delivery problem with stochastic demands. *European Journal of Operational Research*, 271(3):896–912, 2018.