國立臺灣大學工學院土木工程學系

碩士論文

Department of Civil Engineering

College of Engineering

National Taiwan University

Master Thesis

考量利潤與隨機顧客之車輛途程問題

A Vehicle Routing Problem considering Profits and Stochastic Customers

趙浩雅

Hao-Ya Chao

指導教授：朱致遠 博士

Advisor: James C. Chu, Ph.D.

中華民國 109 年 6 月

June 2020

doi:10.6342/NTU202002814

誌謝

在這段求學生涯中能完成這份論文，真的收到太多人的幫助。

首先要感謝我的指導教授朱致遠老師，對我挑戰一個新的題目予以支持。這次的研究方向並不是過去許多人作為研究考量之方向，所以在研究過程中遇到相當多的困難，對此，老師總能適時的給予幫助，為我在研究中遇到的盲點提供許多協助。也是因為老師的允許，我才得以去實習，接觸台灣目前物流業營業者實務上的現況並作為研究假設參考之方向。另外，我要感謝陳柏華教授與許聿廷教授，在口試上給予我許多建議有更好的修改方向。我要謝謝同學們跟學弟妹，提供我不同方向的研究思路。

在此，我要特別謝謝映均、林達霖先生，跟實際的物流業者交談時得以知曉目前台灣實務上不同物流公司提供服務的型態差異；同時我要感謝提供我實習的 PIMCO 公司，讓我在進行業務物流協助時得以跟不同類型的送貨員直接對談，從而知曉實務上最可能觸碰到的問題及困境。

謝謝于棠、亭如等朋友在我因為陷入瓶頸而開始不斷質疑時給予安慰與鼓勵，讓我慢慢踏過瓶頸。最終，我要感謝家人在這段時間的鼓勵與支持，讓我能專心在論文的撰寫。沒有老師、業界前輩、同學、朋友、與家人的幫助支持，是很難完成此畢業論文的。望將此研究成果獻給所有幫助過我的人們。

doi:10.6342/NTU202002814

摘要

隨著網路零售營業額成長與購物環境的變化大量的送貨到府的需求使得物流業面臨極大的挑戰。其中,「最後一哩運送」攸關消費者之顧客滿意度的重要關鍵之一。然而，在實務上時常出現顧客因為各樣因素而延遲取貨、最後導致送貨失敗，甚至取消訂單，使廠商增加營運成本。因此，如何在有限的時間內針對隨機的顧客狀態合理地調度送貨員，在滿足客戶需求的情況下使得總運營最小而得到最佳利潤為物流業者重要的課題。

本研究的目標為開發出一個可考量利潤與隨機顧客之車輛模型（Vehicle routing problem with profits and stochastic customers），並針對其特性與策略進行架構。在研究方法中，本研究在模型建立中進一步提出軟時間窗與容量限制考量，並建立出一個二階段整數規劃模式進行尋求解答方案。在觀察該問題特性後，本研究另外提出兩套演算法來加速求解過程。第一套演算法在第一階段運用插入式啟發演算法去進行第一階段的路徑建立並在第二階段用迭代區域搜索法進行各情境之優化；第二套演算法中，第一階段則改以遺傳演算法進行隨機初始選擇，並同於第二階段用迭代區域搜索法優化路徑。

關鍵字：物流問題、考量利潤之車輛途程問題、軟時窗、混合整數規劃模型、啟發式演算法

doi:10.6342/NTU202002814

ABSTRACT

With the growth of internet retails and the changing of shopping environment in popularity, the large amount of home delivery brings with a new batch of ecommerce logistics challenges. "Last mile delivery" is one of the important factor to customer satisfaction and often leads to delivery failure due to the absence for customer, causing the increase of operating costs. Therefore, logistics enterprises are dedicating to the minimizing the total operation cost and optimum the profit, considering the best way to dispatch the deliveryman reasonably within a limited time.

The aim of this research was to develop an optimization model for Vehicle Routing Problem with Profits and Stochastic Customers (VRPPSC). In the thesis, soft time windows and capacity constraints are considered. A two-stage stochastic mixed-integer programming model was first proposed for the problem. Since the problem is NP-hard, two problem procedures are constructed for solving the generated problem. The first procedures is based on a combination of inserted heuristic and iterated local search algorithms, while the second using a genetic algorithm to do the randomizing of the first stage selection.

Keywords: Logistic problem, Vehicle Routing Problem with Profits, Soft Time windows, Mixed-integer Programming Model, Heuristic Algorithm

doi:10.6342/NTU202002814

CONTENTS

doi:10.6342/NTU202002814

doi:10.6342/NTU202002814

LIST OF FIGURES

doi:10.6342/NTU202002814

doi:10.6342/NTU202002814

LIST OF TABLE

doi:10.6342/NTU202002814

Chapter 1 Introduction

1.1 Background Information and Motivation

Nowadays, the businesses of online shopping and e-commerce are inflating enormously among the entire market. Fig. 1.1 shows the estimation and forecasting of the global e-commerce share of retail sales. The increasing of e-commerce sales change has remained positive since 2015. According to eMarketer (2020), the e-retail accounted for 14.1 % of all retail sales in 2019, and forecasts predict an increase of up to 22% to 2023. With the popularization of smart devices and the using of Internet, retailers open their own e-commerce platforms in order to catch on the trend. The remarkable growth of e-commerce orders leads to the challenge on logistics service.

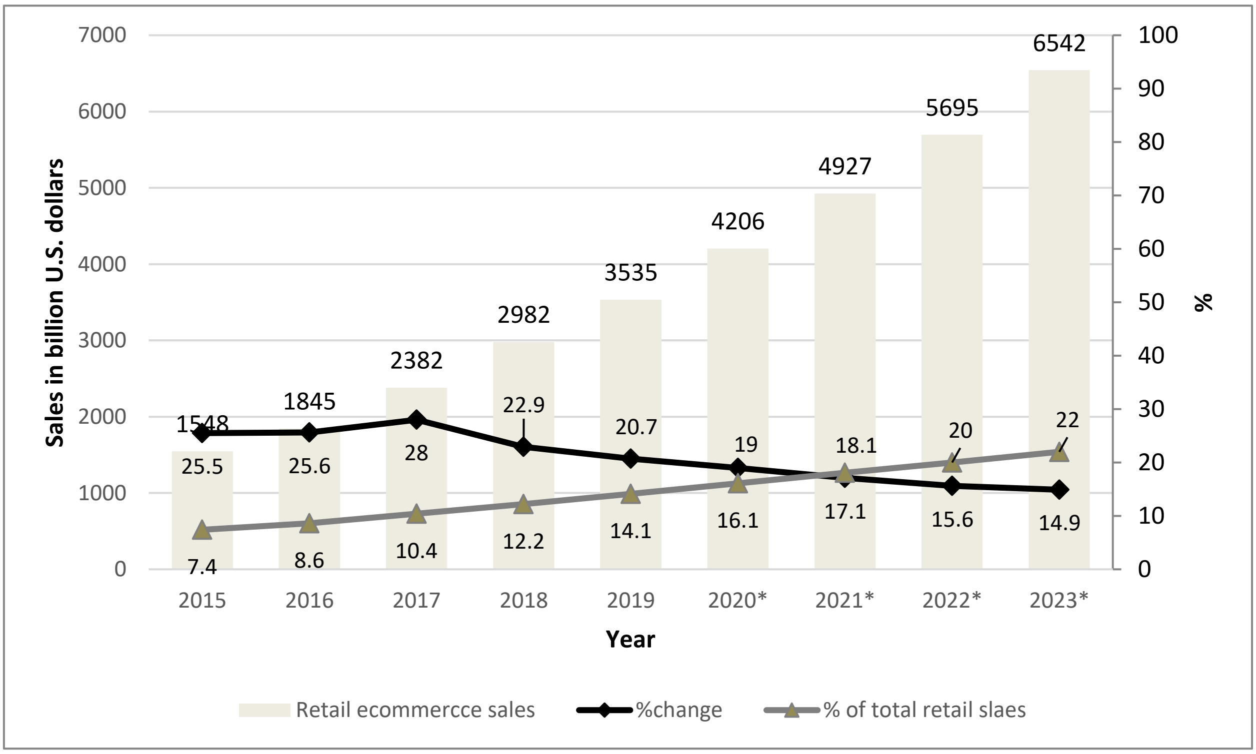

Fig. 1.1 Worldwide retail e-commerce sales (2015-2023)

doi:10.6342/NTU202002814

The last mile delivery has been viewed as the key actors in e-commerce logistics. The phase refers to the final stage of transportation in the logistics network, which the order placement and the service encounter occur. Gevaers, Voorde, and Vanelslander (2011) mention that the last mile delivery is the most cost-intensive part of the supply chain. With the rising attention on the last mile logistic, the probability of failed deliveries is focused. The failure can be caused by product returning, missing of customers, wrong sending and so forth. The unsuccessful delivery leads to the growth cost on the reverse flow. In most countries, more than half of all online shoppers have returned an online purchase. The highest incidence is where 77% of online shoppers have made a return in Germany. Blanchard (2007) mentions that the product returns are reducing profits of manufacturers and retailers by 3.8% per average. The uncertainty for the customers' behavior has caused the difficulty in the delivery dealing and has decreased the effectiveness of the route planning.

For logistic problem, the optimizing of the delivery route aims to identify the most profitable set of customers, which a set of customers to serve is selected while different profits are associated with each customer. Such problem has been referred to as the vehicle routing problem with profits (VRPP). In the context of the VRPP, it is not compulsory to visit all customers. Some previous researches have putting the probability of traveling time and service time into considerations. However, seldom literature studies the stochasticity of customers in the logistic mathematical

doi:10.6342/NTU202002814

model.

1.2 Research Objective

The aim of this thesis is to develop an optimization method for logistics network model. For the planning of the last mile delivery, some factors, such as time window and capacity are essential to consider for the delivery. Since the occurrence of the customer is uncertain, the objective of the proposed model in this research is to maximize the expected profit from different scenarios.

1.3 Research Structure

The structure of this research is outlined as follows. First, the related literature is reviewed in chapter 2. In chapter 3, a mixed-integer programming model for solving the stochastic VRPP model is formulated and two optimization algorithms are developed to solve the problem more efficiently. Next, numerical experiments are conducted in chapter 4, and the results are discussed. Finally, a comprehensive discussion of this research and suggestions for future works are made in chapter 5.

doi:10.6342/NTU202002814

Chapter 2 Literature Review

The main difference between vehicle routing problem with profits (VRPP) and classical vehicle routing problem is that not all the customers have to be served for the routing. In other words, two decisions are considered. In section 2.1, a number of typical VRPP are first reviewed. In order to provide more satisfactory and efficiency way, the enterprises have allowed the requisition from customer of the goods delivery within specific time windows. Meanwhile, in real world, couriers have to consider the capacity of vehicle. Thus, the variants of VRPP with time window constraints, capacity constraints and stochasticity consideration are discussed in section 2.2. Seldom research studied the VRPP with stochastic customers; therefore, a variety of VRP with stochastic customer (VRPSC) models are reviewed and the methods for solving the problems are compared in section 2.3.

2.1 Vehicle Routing Problem with Profits

VRPP has been studied widely. The differences between the VRPP and the regular traveling salesman problem (TSP) are the requirement of selecting customers and consideration for the profit. In the VRPP, a customer is selected based on the trade-off between its profit and the extra travel cost required to include the customer. The most interesting customer can bring the highest profitability. Vansteenwegen and Gunawan (2019) classified the routing problem with profits in two ways: One way is based on the number of vehicles or routes and another way is based on the manner where the profit and the travel cost, mostly distance or time, are modeled.

doi:10.6342/NTU202002814

The basic VRPPs and their characteristics are summarized in table 2.1. For single VRPP, three basic problems are classified. The first one is called the profitable tour problem (PTP). PTP combines both profit and travel cost in the objective function. Therefore, the objective of the PTP is to visit a subset of customers that maximizes the total collected profit minus the total travel cost. The second problem is described as the prize-collecting traveling salesperson problem (PCTSP). The objective of PCTSP is to minimize the total travel cost to reach the lower bound on the profit to be collected from a subset of customers may be visited. The third problem, usually named the orienteering problem (OP), is the other way around by which is also known as the selective traveling salesperson problem. The objective is to maximize the total collected profit by visiting a subset of customers, while not exceeding a given travel cost, typically a time constraint or a limited route length.

The routing problems with profits and multiple vehicles can be viewed as the extension of three basic problems. The multi-vehicle PTP (MVPTP) is the generation for PTP. Toth and Vigo (2014) described the multi-vehicle extension of PCTSP as capacitated prize collecting VRP (CPCVRP). The generalization for OP with multiple vehicle, known as the team orienteering problem (TOP), was first introduced by Butt and Cavalier(1994) with the name multiple tour maximum collection problem (MTMCP). TOP is by far the only one studied in depth among the routing problems with profits and multiple vehicles. Many algorithms have been proposed for the

doi:10.6342/NTU202002814

TOP. Surveys on the OP and TOP (and many variants) can be found in Vansteenwegen et al. (2011) and Gunawan et al. (2016).

Problem name	Objective	Vehicle
Orienteering Problem (OP) (Selective TSP, Maximum Collection Problem, Bank Robber Problem)	max profit	Single
Profitable Tour Problem (PTP)	max (profit - cost)	Single
Prize-collecting traveling salesperson problem (PCTSP)	min cost	Single
Team Orienteering Problem (TOP) (Multiple Tour Maximum Collection Problem)	max profit	Multiple
Multi-vehicle PTP (MVPTP)	max (profit - cost)	Multiple
Multi-vehicle Prize Collecting VRP (MVPCVRP)	min cost	Multiple

Table 2.1 Summary for VRPP

2.2 Variants of Vehicle Routing Problem with Profits

In this section, the variants of vehicle routing problem with profits are presented. For logistic company, mostly plural drivers are designated to accomplish the delivery; thus, the review will first focus on VRPP with multiple vehicles. Since the profitability for the customer is most considered, the review will focus on variants for TOP and MVPTP will be. Next, the literature for VRPP with the consideration of uncertainty will be reviewed.

The capacitated TOP (CTOP) is a variant of TOP that additionally considers a capacity

doi:10.6342/NTU202002814

constraint. In the problem, a demand is associated to each customer and each vehicle has a maximum capacity. The objective is to maximize the total collected profit while satisfying the capacity and duration constraint for each route. Please refer to Archetti et al. (2009), Archetti, Bianchessi, and Speranza (2013a), Luo et al. (2013) and Tarantilis, Stavropoulou, and Repoussis (2013). In theory, it can be beneficial to only serve a customer partially and receive the proportional partial profit. Archetti et al. (2013b) further extended the problem to be more beneficial by relaxing with allowing incomplete services for a customer, which is called the CTOP with incomplete service (CTOP-IS). The study proved the advantage of its advantage on profit collection ability with a branch-and-price algorithm. Another extension of the CTOP allows the customer to be served by more than one route to fulfill the service of a customer, named the split delivery CTOP (SDCTOP). Further extensions of SDCTOP are the SDCTOP with incomplete service (SDCTOP-IS) and the SDCTOP with minimum delivery amounts (SDCTOP-MDA). These variants are discussed in Archetti et al.(2013b, 2014a, 2014b) and Wang et al.(2014) respectively. The articles of CTOP and the variants reviewed are summarized in table 2.2.

doi:10.6342/NTU202002814

Problem name	Characteristic	Representative Articles
CTOP	Associated demand for each customer and the capacity constraint	Archetti et al. (2009), Archetti, Bianchessi, and Speranza (2013a) Luo et al. (2013) Tarantilis, Stavropoulou, and Repoussis (2013)
CTOP-IS	Allowing partial demand service for each customer and the capacity constraint	Archetti et al. (2013b)
SDCTOP	Split demand for each customer and the capacity constraint	Archetti et al.(2014a)
SDCTOP-IS	Allowing split demand service and partial demand service for each node and the capacity constraint	Archetti et al.(2014b)
SDCTOP-MDA	Split demand with minimum delivery amounts	Wang et al. (2014)

Table 2.2 Variants of CTOP

The TOP with time windows (TOPTW), another variant of TOP, has received considerable attention from the heuristic community in the last decades. The customers has an associated time window, which means the service for a particular customer has to start within the predefined time window. An early arrival to a particular customer leads to waiting times, while a late arrival causes an infeasibility issue. Many heuristic algorithms have been proposed and overall obtained good

doi:10.6342/NTU202002814

average results on benchmark instances, see as Vansteenwegen et al.(2009) ,Montemanni and Gambardella (2009) ,Gambardella, Montemanni, and Weyland (2012), Lin and Yu(2012), Labadie et al.(2013) and Hu and Lim(2014).

The capacitated TOP with time windows (CTOPTW) is a hybrid of TOPTW and CTOP. The problem aims to search the highest profit where opening hours of customers and the capacity of vehicle need to be considered. Due to the complexity, only very few literature study practically on the problem. Garcia et al.(2010) extended the team orienteering problem with time windows (TOPTW) by adding multiple constrains and described it as the multi-constrained team Orienteering problem with time windows (MCTOPTW). The study proposed an iterated local search (ILS) heuristic algorithm to solve the problem. Later, Aghezzaf and Fahim (2014) developed a variable neighborhood search approach for MCTOPTW. Recently, an exact algorithm is presented by Park et al.(2017) to solve the problem by applying the branch-and-price (B&P) scheme of Boussier et al.(2007) to the CTOPTW.

Contract to TOP, there is a paucity of literature on MVPTP. The capacitated and multiple-vehicle version of the PTP (CPTP) studied from Archetti et al (2009) can be viewed as the variant of MVPTP with capacity constraints. The problem is defined that each customer has a demand and the fleet of vehicles has a prefixed capacity, which must not be exceeded by the route. The study presents one exact and three heuristic algorithms. Archetti, Bianchessi, and Speranza (2013a)

doi:10.6342/NTU202002814

presented a different B&P algorithm to solve the problem. Later, Archett et al. (2018) extended CTP to the undirected capacitated general routing problem with profits (UCGRPP), which customers can be located on either vertices or edges of the graph, and constructed a two-phase exact algorithm to solve. It is noted that not all CPTP in literatures refers to MVPTP. Jepsen (2011) proposed a branch-and-cut (B&C) algorithm for the undirected version of CPTP, which only allows one tour going through the depot. Sun et al. (2018) introduced the time-dependent capacitated profitable tour problem with time windows and precedence constraints and the study considered single vehicle rather than multiple vehicles.

For the variants of VRPP with stochastic aspects ,most studies focus on stochastic traveling time, service time and waiting time, see as Campbell, Gendreau, and Thomas (2011), Papapanagiotou, Montemanni, and Gambardella (2014) and Evers et al. (2014). Ilhan, Iravani, and Daskin (2008) were the first to introduce uncertainties in the collected scores. They discussed the orienteering problem with stochastic profits (OPSP) as a variant of OP. In OPSP, the profits associated with the nodes are stochastic with a known distribution. The objective of the OPSP is to maximize the probability that the total collected score, or profit, from the route will be greater than a predefined target value. However, these researches only model single tour and lack of the aspects on stochastic customer.

doi:10.6342/NTU202002814

2.3 Vehicle Routing Problem with Stochastic Customers

In real world, parameters of the problem, such as costumers' demands, travel times, costs, or service times are often stochastic or unknown during the planning horizon. Usually, information about upcoming events is available through historical data, which can be converted into information models. The stochastic VRP (SVRP) is basically any VRP where one or more parameters are stochastic, meaning that some future events are random variables with a known probability distribution. Generally, the random variables have a probability distribution. Ritzinger, Puchinger, and Hartl (2015) provided a survey on dynamic and stochastic vehicle routing problem.

VRPSC is the problem which customers are either present or absent with a given probability. A number of models and solution procedures for VRPSC allow recourse actions to adjust a priori solution after the uncertainty is revealed. Many studies present VRPSC as a two-stage stochastic programming problem. The first stage is to determining some initial routes that adhere to the VRP constraints. After presenting the customer, the second stage solution is to follow up the routes set by the first stage, while skipping the absent customers. Waters (1989) re-optimized the route after skipping the absent customers for better result. The vehicle routing problem with stochastic customers and demands (VRPSCD) combines stochastic customers and stochastic demands, see as Bertsimas (1992), Benton and Rossetti (1993) and Gendreau, Laporte, and Séguin (1995,1996).

Recently, Sungur et al.(2010) considered the Courier Delivery Problem with uncertainty on the

doi:10.6342/NTU202002814

service times and presence of customers. Customers have soft time windows while a hard constraint is considered on the route duration. Uncertainty is represented by scenarios. To solve the large-scale problem, a two-phase approximate solution heuristic is developed.

2.4 Summary

In this chapter, works of literature regarding vehicle routing problem with profits, vehicle routing problem with stochastic customers and the extensions of these problems are reviewed. Different VRPP schemes have different concerns in applications. For instance, OP is useful for the problem with no costing concerned and aims to search the highest value from the route, while PCTSP only consider the cost, and PTP scheme is favorable for both reducing travel cost and increasing the collecting profits. Next, for VRPP with multiple vehicle, there are many studies focus on TOP and its variants but few concerned for the MVPTP and its extension. Meanwhile, VRPP with capacity constraints and time windows constraints has little researches, although it may have better simulation on real world application.

To summary, with large amount of studies on VRPP, there is still little literature considering the stochasticity of customer. In some real-life problems, the time window for the logistics allows some violation by adding an extra punishment for the delay; however, there is seldom review for the soft time window in VRPP. Moreover, existing researches concerning to logistics problem have suggested that more detail could be concerned and studied. Therefore, this researches aims to construct an

doi:10.6342/NTU202002814

optimization method for VRPP with stochastic customers, which constraints of time windows and capacity are considered in the formulation.

doi:10.6342/NTU202002814

Chapter 3 Research Methodology

3.1 Overview

In this chapter, the vehicle routing problem with stochastic customers is designed with additional constraints of capacity and soft time windows, which can be viewed as a stochastic capacitated multi-vehicle routing problem with profit and soft time window (SCMVRPPSTW). A mathematical formulation model is proposed for the problem. Then, two heuristic algorithms are developed to solve the model efficiently in a limited time.

The mathematical problem is formulated as a two-stage stochastic program. The first-stage decision variables decide the must-served customers before the realization of the uncertain data is shown. Based on the first stage variables, when the occurrence of customers become available on the second stage, the routes in each sample scenario will be optimized to find the most profitable route in the scenario.

Since the SCMVRPPSTW is a highly constrained problem and very difficult to solve, it is unlikely to solve the problem to optimality within a limited time. Therefore, the development of a high quality and fast optimization algorithm is necessary. In this study, two two-stage algorithms were proposed. In the beginning, the first algorithm presents an insertion heuristic to selected the customers and construct the initial route, while a genetic algorithm is used to selects the customer as a randomizing program in the second algorithm. In the procedure, the initial solution from first

doi:10.6342/NTU202002814

stage algorithm will be put into iterated local search algorithm and re-optimize. That is, the route in each scenario will be improve. The algorithm returned the best-found solution after certain termination criteria was met.

3.2 Basic Assumptions

According to the interviews from the manager, courier of logistic company, related news and research, pickup demands, service times and traveling times are usually reported with ambiguous words, and large amount of complexity and uncertainty may not be concerned. Therefore, the assumptions of the problems and constraints are presented:

1. Each customer can be served at most by one vehicle.
2. The load of each vehicle is restricted by its capacity.
3. The unexpected incidents on the roads are not considered.
4. The service time for each customer are static and known in advance.
5. A vehicle is allowed to arrive at a customer before the relevant time window, but the driver cannot serve the customer until the time window opens.
6. The set of the customers in each scenario may be different.

3.3 Mixed-integer Programming model

3.3.1 Network Formulation

In this section, the formulation of mixed-integer programming model is introduced. The model

doi:10.6342/NTU202002814

is a variant from the capacitated vehicle routing problem with profits and time window (CVRPPTW). The problem is formulated in the directed graph $G = (V,A)$ where V is the set of all vertices and A is the sets of arcs. V includes three sets node sets: sets of origin depots O, sets of destinations depots D and sets of customers N. Noted that not all customers must be served in the problem. For customer i, a non-negative demand c_i^q, a non-negative revenue r_i and a time window $[a_i, b_i]$ is associated. A symmetric travel time t_{ij} and distance d_{ij} are associated with each edge $(i, j) \in A$. Each vehicle k can visit any subset of customers with a total demand that does not exceed the capacity q_k. The profit of each customer can be collected by one vehicle at most.

For CVRPPTW, the problem can be categorized with the characteristic. In the CTOPTW, the subset of the potential customers available has to be selected. The objective is to maximize the total collected profit while satisfying a time limit T_{max} on the tour duration and the capacity constraint q_k for vehicle and time window limit for customers.

In the capacitated profitable tour problem with time windows (CPTPTW), a subset of the potential customers available has to be selected with the objective of maximizing the difference between the total collected profit and the cost of the total distance travelled. The tour for customer must satisfy the capacity constraint q_k for each vehicle and time window constraint for customers.

In real world, the customers usually allow a certain delay since the time interval are often described ambiguously, but the delivery may be canceled by customer if the delay is over the

doi:10.6342/NTU202002814

tolerance range. Therefore, the time window in this model is assumed to be violated barring a penalty cost in the problem if the delay time is no more than T_{exceed}.

An example of a problem instance is provided in Fig 3.1a. The origin depot and the destination depot are both set as 1. Here q_k= 6, $c_i^q = 2$, i = 2,..., 6, c_{ij}= 2 for each edge *(i, j)* except for edge (2, 3) and edge (5,6) that has cost c_{23} = 1 and $c_{56} = 3$. The time window for customer 2 and 3 is [0, 3] and the others is [0, 4]. The time limit T_{max} for the CTOPTW is equal to 6. Based on the categories for VRPP, the problem can be extended in different considerations. Fig 3.1b shows the optimal solution for CTOPTW while the optimal solution for CPTPTW is presented in fig 3.1c. For CPTPTW, the problem has no time limit for the tour; however, the time limit T_{max} can be viewed as the time window of the destination in VRPTW. Therefore, an optimal solution for the CPTPTW with time limit is presented in fig 3.1d.

Fig 3.1e, 3.1f and 3.1g shows three examples for CPTP with soft time window (CPTPSTW). All problems allow a $T_{exceed} = 1$, while the unit penalty cost = 0.5 in fig 3.1e, cost = 2.5 in fig 3.1f and cost = 5 in fig 3.1g. It shows that the cost on penalty changes the optimum route. When the penalty reaches to a high amount, the optimal solution will be the same as the optimal solution of the CPTPTW.

doi:10.6342/NTU202002814

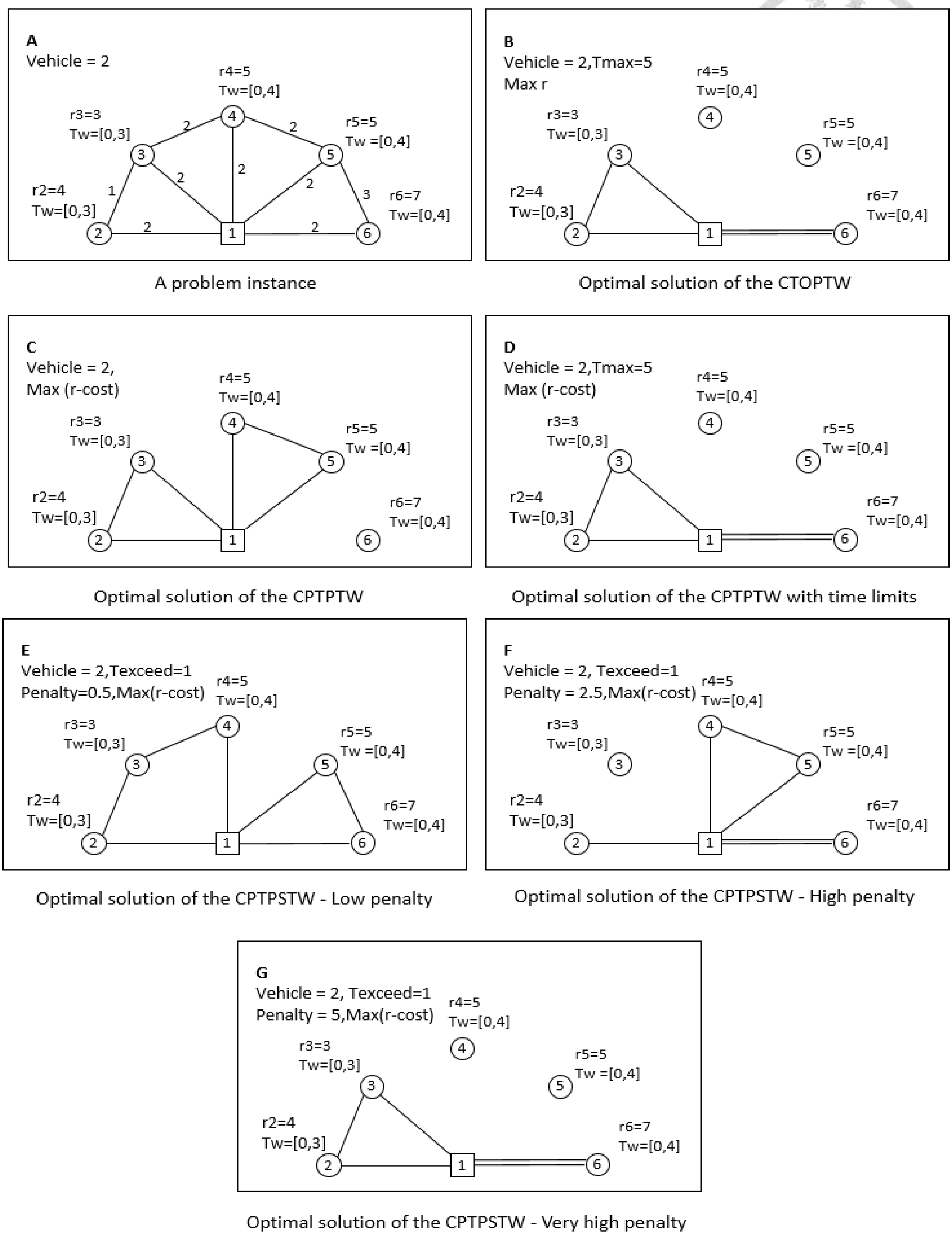

Fig. 3.1 A problem instance and optimal solutions of the CVRPPTW.

doi:10.6342/NTU202002814

3.3.2 Model Formulation

In this section, the notations in the model are first defined, than the MIP model will be described in details.

Sets:

- V: Sets of vertices, indexed by i and j.
- O: Sets of origin depots. $O \subseteq V$.
- D: Sets of destination depots. $D \subseteq V$.
- N: Sets of customers, $N \subseteq V$.
- K: Sets of vehicles, indexed by k.
- A: Sets of arcs, indexed by a tuple of two customers.
- S: Sets of scenarios, indexed by s.
- N_s: Sets of customers in scenario s.

Parameters:

- $|K|$: number of vehicles.
- n: number of customers.
- T_{max}: maximum allowable travel time for a vehicle route.
- T_{exceed}: maximum allowable exceed time for time window penalty.
- d_{ij}: traveling distance between customer i and customer j.

doi:10.6342/NTU202002814

- t_{ij}: traveling time between customer i and customer j.
- a_i: time window starting time of customer i.
- b_i: time window ending of customer i.
- τ_i: service time of customer i.
- r_i: revenue for customer i.
- l_i: selective revenue for customer i.
- c_i^q: $d$emand for customer i.
- q_k: capacity for vehicle k.
- c^1: unit vehicle operating cost.
- c^2: cost per unit distance traveled.
- c^3: cost per unit time penalty.
- c^4: cost per using penalty vehicle
- Pr_s: probability of occurrence for scenario s.
- M: a large constant.

Variables (continuous variables):

- $u_{ijk,s}$: load transported on arc (i,j) by vehicle k in scenario s.
- $T_{ik,s}$: service starting time for customer i by vehicle k in scenario s.
- $T_{ik,s}^{-}$: exceeding time for customer i by vehicle k in scenario s.

doi:10.6342/NTU202002814

Variables (binary variables):

- w_i: selection variables, 1 if the customers $i \in N$ was chosen to be served and 0 otherwise
- $x_{ijk,s}$: binary variable equal to 1 if arc *(i,j)* is traversed by the vehicle k in scenario s, and 0 otherwise;
- $y_{ik,s}$: binary variable equal to 1 if customer i is visited by the vehicle k in scenario s, and 0 otherwise;

For the model, the first-stage solution is to set the customers must be visited in the routing plan. The mathematical programming formulation is shown as the following:

$$\max \sum_{i \in N} l_i w_i + E[Q(w_i, S)] \tag{1}$$

$$w_i \in \{0,1\}, \forall i \in N \tag{2}$$

The objective function (1) consists of maximizing the selection revenue and the expected routing revenue .Constraint (2) states the binary variables in the first stage, which is used in the second stage for solving the expectation value $E[Q(w_i, S)]$:

$$E[Q(w_i, S)] = \sum_{s \in S} Pr_s Q(w_i, s) \tag{3}$$

Where $Q(w_i, s)$ is the optimal value of the second-stage problem for scenario $s \in S$. The second stage model is formulated as below.

doi:10.6342/NTU202002814

$\forall s \in S,$

$$\max Q(w_i, s) = \sum_{i \in N^s} r_i \sum_{j \in V} \sum_{k \in K} x_{ijk,s} - c^1 \sum_{(i,j) \in A} \sum_{k \in K} x_{ijk,s}$$

$$-c^2 \sum_{(i,j) \in A} d_{ij} \sum_{k \in K} x_{ijk,s} - c^3 \sum\nolimits_{i \in N} \sum\nolimits_{k \in K} T_{ik,s}^{-} \tag{4}$$

$$\sum_{j \in V} x_{ijk,s} = y_{ik,s}, \forall i \in N \cup O, k \in K, \tag{5}$$

$$\sum_{j \in V} x_{jik,s} = y_{ik,s}, \forall i \in N \cup D, k \in K, \tag{6}$$

$$\sum_{k \in K} y_{0k,s} \leq |K|, \tag{7}$$

$$\sum_{k \in K} y_{ik,s} \leq 1, \forall i \in N, \tag{8}$$

$$\sum_{k \in K} y_{ik,s} \geq w_i, \forall i \in N, \tag{9}$$

$$y_{n+1,k,s} - y_{0k,s} = 0, \forall k \in K, \tag{10}$$

$$\left(\sum_{(i,j) \in A} t_{ij} x_{ijk,s} + \sum_{i \in K} \tau_i y_{ik,s} \right) \leq T_{max}, \forall k \in K, \tag{11}$$

$$T_{ik,s} + \tau_i + t_{ij} - T_{jk,s} \leq \left(1 - x_{ijk,s}\right) M, \forall (i,j) \in A, k \in K, \tag{12}$$

doi:10.6342/NTU202002814

$$a_i y_{ik,s} \leq T_{ik,s} \leq b_i y_{ik,s} + T_{exceed}, \forall i \in N, k \in K, \quad (13)$$

$$T^-_{ik,s} \geq T_{ik,s} - b_i y_{ik,s}, \forall i \in N, k \in K, \quad (14)$$

$$\sum_{j \in V} u_{ijk,s} - \sum_{j \in V} u_{jik,s} = c^q_i, \forall i \in N, k \in K, \quad (15)$$

$$u_{ijk,s} \leq q_k, \forall i \in N, k \in K, \quad (16)$$

$$x_{ijk,s} \in \{0,1\}, \forall (i,j) \in A, k \in K, \quad (17)$$

$$y_{ik,s} \in \{0,1\}, \forall i \in V, k \in K, \quad (18)$$

$$u_{ijk,s} \geq 0, \forall i \in V, k \in K, \quad (19)$$

$$T_{ik,s} \geq 0, \forall i \in N, k \in K, \quad (20)$$

$$T^-_{ik,s} \geq 0, \forall i \in N, k \in K, \quad (21)$$

The objective function for second-stage problem (4) aims to maximize the expected profit, which is the difference between total revenue generated from customers and total cost of purchasing customers, operating the vehicles and time penalty. Constraints (5) and (6) ensure that one arc

doi:10.6342/NTU202002814

enters and one arc leaves each visited customers. Constraints (7) limits the number of routes to be at most |K|, while constraints (8) and (9) guarantee that the customers selected in the first stage must be served. Constraints (10) ensures that the origin and destination customers of a request are visited by the same vehicle. Constraint (11) ensures the total transporting time on each route will not exist the maximum allowable transporting time. Constraint (12) guarantees the consistency of time. Precedence constraints are imposed through inequalities constraint (13) and constraint (14) ensures schedule feasibility with respect to time windows. Note that for a given k, the value of T_{ik} is meaningless whenever customer i is not visited by vehicle k. Constraints (15) is the capacity constraints for the route. Finally, constraints (17)-(21) states the binary and non-negativity properties of the decision variable.

doi:10.6342/NTU202002814

3.4 Heuristic Algorithm

Time is always a managerial issued for an industry. Normally for real logistics, the problem setting for a vehicle may up to 70 or more customers. The solving time increases rapidly as the scale of networks grows. Since the problem can be viewed as an integration of two NP-hard optimization problems where each separate problem is by its own difficult to solve, two kinds of two-stage solving frameworks are built for the problem.

The first two-stage framework is a hybridization of two local search model, as shown in Fig. 3.2 and 3.3. For fig 3.2, the first stage is a simple insertion heuristic. Customers are selected from the score calculated based on certain factors. Considering the formulation with improving and switching for the initial route, another first two-stage framework, replacing first local search model to genetic algorithm (GA), is constructed. Fig. 3.3 shows the framework. The procedure combines genetic operators, selection, and crossover with an efficient local search to be the first stage selection. Next, the second stage applies the iterated local search scheme from Pieter Vansteenwegen et al. (2009) in order to re-optimize the route planning based on the selection in first stage, shown in Fig. 3.4. The detail of heuristic stage will be described in the following sections.

doi:10.6342/NTU202002814

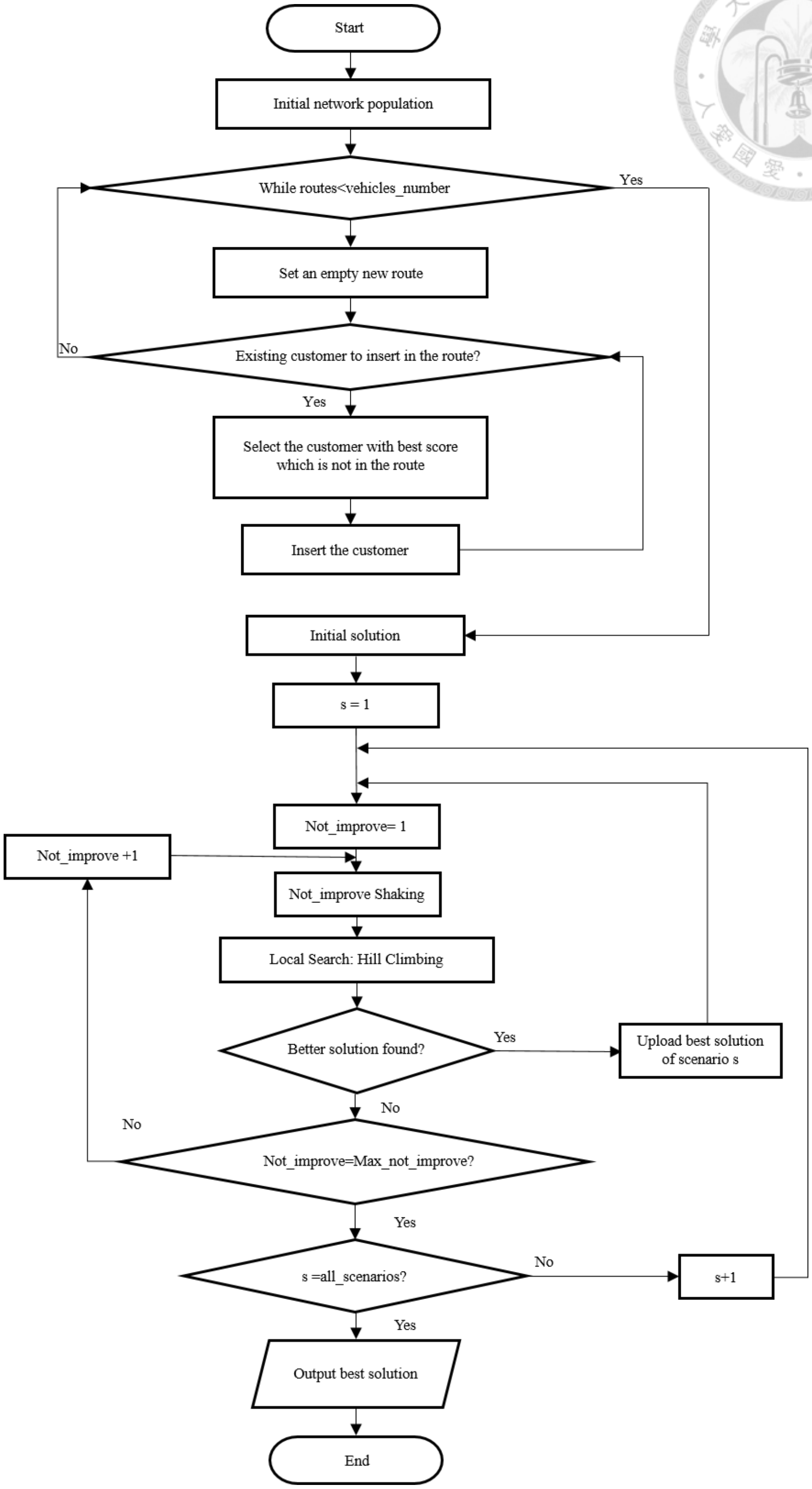

Fig. 3.2 Two-Stage Structure for insertion heuristic and ILS

doi:10.6342/NTU202002814

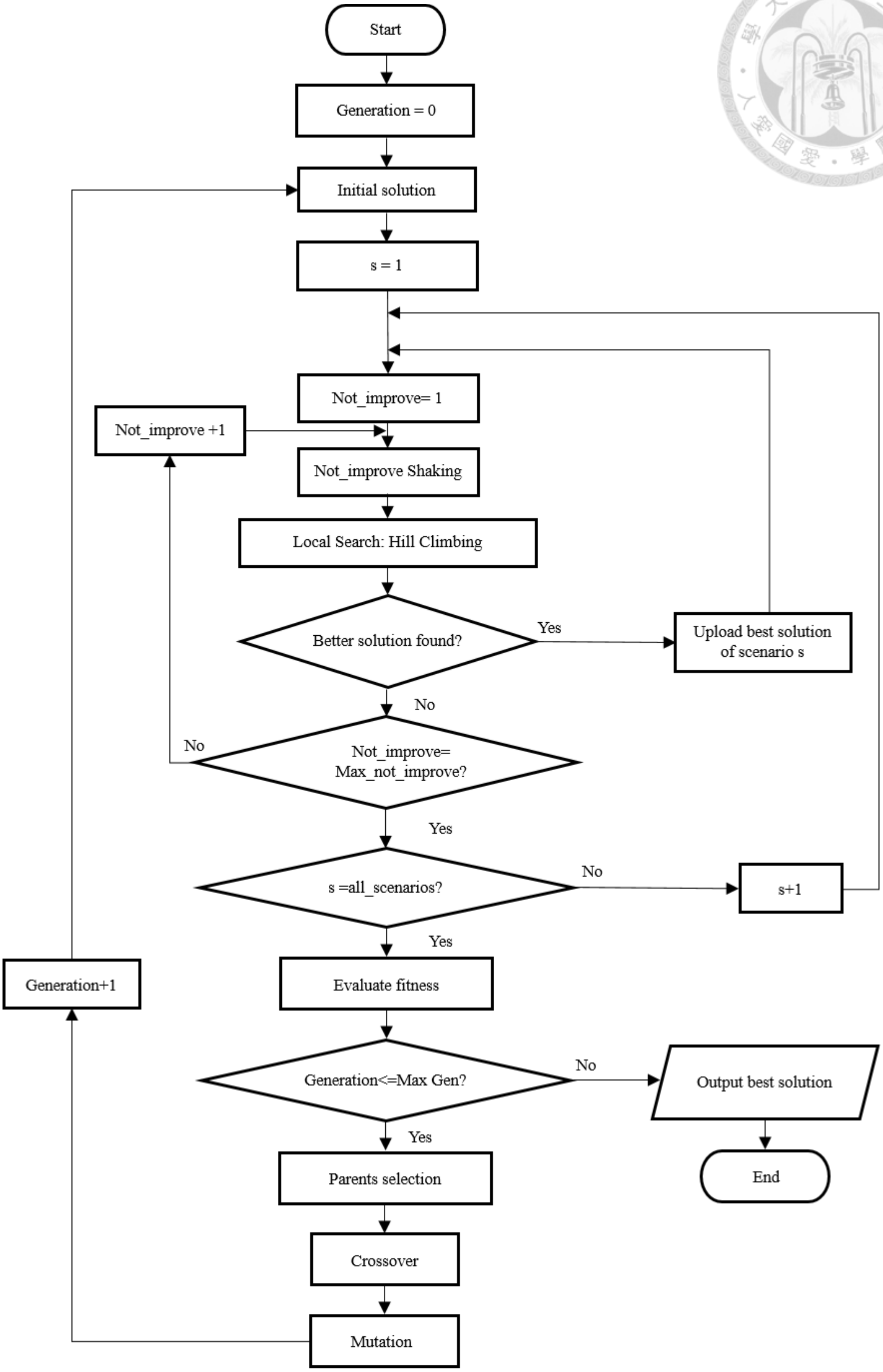

Fig. 3.3 Two-Stage Structure for genetic algorithm and ILS

doi:10.6342/NTU202002814

3.4.1 First-stage method

For the two-stage model, the first stage of the algorithm consist of the maximizing the number of customer must served. An initial route is constructed based on the selected customers.

3.4.1.1 Insertion method

For the insertion method, the initial route will be constructed based on the selected customers. Let $(i_0,\ i_1 \dots i_e)$ be the current initial route, with $i_0 = i_e = 0$. For each unrouted customer, the best feasible insertion place in the emerging set is set as

$$best\big(i(u), u, j(u)\big) = \max\big[best\big(i_{p-1}, u, i_p\big)\big], p = 1, \dots, e, u\ \ unrouted\ \ and\ \ feansilble.$$

The insertion for u must follows the time window constraints and the capacity constraints. Since the only uncertainty for the problem is the occurrence of customers, other parameters are assumed to be known in advance. For the routing, the profit, costing and time most considered for building the route. Therefore, the approach is described,

$$best_1(i, u, j) = m_1 r_u + m_2 l_u - m_3(t_{i,u} + t_{u,j} - t_{i,j}) - m_4 punish_u - m_5 a_u$$

$$m_1, m_2, m_3, m_4, m_5 \geq 0,$$

The insertion aims to maximize the benefit from the insertion, therefore, if m_1 and m_2 are set as 0, the calculation will have an addition check for the adding score to avoid negative-score insertion. After no more customer with feasible insertions can be found, the method starts a new route, unless the total number of vehicle routing reaches the vehicle numbers. It is clear that

doi:10.6342/NTU202002814

different values of m_1, m_2, m_3, m_4 and m_5 lead to different possible criteria for selecting the customer for insertion and its best insertion. This method can have a simple and fast selection for the first step; however, the heuristic cannot contrast the goodness of the initial routes constructed from the proposed heuristic. Therefore, a genetic algorithm is served as a randomized mutation of the first-stage method.

3.4.1.2 Genetic algorithm

In this thesis, GA, a powerful and common used optimized heuristic methodology, is introduced to make a random selection of must-served customers for the later algorithm. The main elements of GA includes chromosome coding, population initialization, fitness function, selection procedure, crossover procedure and mutation procedure. Size of population (pop), crossover rate (cross_rate), mutation rate (mutate_rate), and the number of generations (gen numbers) will be the main parameters in the proposed algorithm. In a GA, the chromosomes or individuals are represented as strings which encode candidate solutions for an optimization problem, that later evolve towards better solutions.

The general schema for the GA algorithm is as following:

1. Generate a population of successive solutions. The chromosome for selection is demonstrated as Fig. 3.5.
2. While the number of generations $\leq$ than maximum number of generation:

doi:10.6342/NTU202002814

a. If a random generated probability $\leq$ crossover rate, a crossover operation is performed on a pair of chromosomes to obtain a new solution that reflects aspects of both parents.

b. Else, for one or some randomized chromosomes, perform the mutation operation in them to obtain a new solution.

c. Apply the local search to improve the solution obtained from the evaluation operation. As can be seen from Fig. 3.4, the binary stream only represents the selection. The selection will be put into further ILS heuristics to get the best solution.

1	0	0	1	0	1	0	1	1	1	1	0	0	1	1	1	0	1	0	1	1	0	1

Fig. 3.4 A solution representation as a chromosome of the problem

3.4.2 Second-stage method

To simulate the reality, the scenarios are generated into the problem. Since VRP problems are proofed to be NP-hard, it may takes time to make the optimization of the routing in each scenario. Meanwhile, the visit should be verified that the new visit scheduled in the tour must satisfy time window and vehicle's capacity limitation. To provide a higher ability heuristic, which may escape local optima and ensure the quality of the result, an iterated local search (ILS) scheme of Vansteenwegen et al. (2009) is applied. The heuristic combines an insertion step and a shaking step to escape from local optima.

- Insertion step

doi:10.6342/NTU202002814

The insertion step aims to provide a visit insertion for the original route. To decrease time on checking the feasibility on time window, the time of waiting at the location and maximum allowable shifting time for each included location were recorded. Since the service can only start when the time window opens, let v_i be the arriving-time at location *i* and $Wait_i$ be the waiting time. $Wait_i$ equals 0 if the arrival takes place during the time window.

$$Wait_i = \ max[0, a_i - v_i] \tag{22}$$

Let $Maxshift_i$ be the maximum allowable shifting time for location *i* without causing infeasible visit in the route. $Maxshift_i$ of location *i* is equal to the sum of $Wait_i$ and $Maxshift_i$ of the next location *i+1*, unless the service time is limited by its own time window. It is noted that only the maximum allowable shifting time for origin depot and destination depot has no allowable exceed time, which other location has a tolerance on exceeding time.

$$Maxshift_i = \min[b_i + T_{exceed} - (v_i + Wait_i), Wait_{i+1} + Maxshift_{i+1}], i \in N \tag{23}$$

$$Maxshift_i = \min[b_i - (v_i + Wait_i), Wait_{i+1} + Maxshift_{i+1}], i \in (O \cup D)$$

The total time consumption ($Shift_j$) to insert an extra visit *j* between the further constructed visits *i* and *k* is defined as

$$Shift_j = t_{ij} + Wait_j + \tau_i + t_{jk} - t_{ik} \tag{24}$$

To make a feasible insertion for visit *j* between *i* and *k* for vehicle *v*, $Shift_j$ should be limited to the sum of $Wait_k$ and $Maxshit_k$ for location k. Meanwhile, $Shift_j$ should be limited by the

doi:10.6342/NTU202002814

total time limit of the route. The following formulas were defined to check the feasible time:

$$Shift_j = t_{ij} + Wait_j + \tau_i + t_{jk} - t_{ik} \leq Wait_k + Maxshit_k$$

$$Shift_j \leq Maxshit_{destination,v}$$

Service *j* should be followed by the time window constraints of customer *j* and the capacity constraint for the routing.

For a vehicle routing problem with soft time window, it allows the visit to have a exceed time for flexibility in moving. The exceeded time from the time window of customer *i* is record as $Punish_i$. If the visit take place in the time window, $Punish_i$ will equal zero.

$$Punish_i = \max[0, v_i - b_i] \tag{25}$$

$Realexist_{i,s}$ is represented as the existence of the which will equal 1 if customer *i* exist and 0 otherwise. The revenue $score_{j,s}$ for a visit *j* inserted is defined as:

$$score_{j,s} = Realexist_{j,s} r_j - c^1 - c^2\left(t_{ij} + t_{jk} - t_{ik}\right) - c^3(Punish_i + influence) \tag{26}$$

$shift_i$ and $score_{i,s}$ are the values to determine to be the better possible insertion, which *influence* is the punishment changing due to the insertion for customer *i*. For each visit the heuristic aims to find the highest possible $score_{i,s}$, while the lowest $score_{i,s}$ presents the better insertion spaces for further improvement. In order to determine the best selection in all feasible visit, a ratio is calculated as following.

$$Ratio_{i,s} = (score_{j,s})^3 / shift_i$$

doi:10.6342/NTU202002814

The visit with the highest ratio will be selected for the insertion. Since the objective of the model is to optimal the revenue and the visit is constrained by the time window, the relevant of the time consumption should be less than the scoring in the decision for visiting insertion. Due to the possibility for the negative revenue with high cost, a triple of the score is applied in the calculation of ratio.

After the procedure, all other visits should be updated. The procedure for the heuristic algorithm is presented in Fig. 3.5. The total score is first update, and the visits after the insertion should update the waiting time, the arrival time, the service time, $Maxshift$ and $Punish$. The allowable shifting time for the service starting and the following services will gradually decreased due to the waiting time for the visit. If $Punish$ changes in the procedure, the total score calculated should also be update. This gives the following formulas to update the visits after the insertion position, when customer j is inserted between i and k :

$$Shift_j = t_{ij} + Wait_j + \tau_i + t_{jk} - t_{ik}$$

$$Wait_{k*} = max[0, Wait_k - Shift_j]$$

$$v_{k*} = v_k + Shift_j$$

$$Shift_k = max[0, Shift_j - Wait_k]$$

$$Totalscore \ = Totalscore - c^3(max[0, v_{k*} - b_k] - Punish_{k*})$$

$$Punish_{k*} = max[0, v_{k*} - b_k]$$

doi:10.6342/NTU202002814

$$Maxshift_{k*} = Maxshift_k - Shift_k$$

$Shift_k$ and the same formulas are then used to update the visit after k. The procedure will continue until $Shift_k$ is reduced to zero. Meanwhile, $MaxShift$ of the visits before the insertion j should be updated using formula (23) mentioned above.

```
Insertion step:

For each non included visit in customer:
    Calculate possible insert position Wait, Arrive, Punish, Ratio;
    Determine best possible insert position and shift
Insert visit with highest ratio(j);
Update TotalScore;
For each visit after j(until Shift == 0):
    Update Wait, Arrive, Maxshift, Shift;
    If Punish change then:
        Update TotalScore, Punish
Visit j: Update Maxshift;
For each visit before j(until Maxshift remains the same):
    Update Maxshift;
```

Fig. 3.5 Pseudo code for the insertion

- Shaking step

The shaking step aims to escape from the local optimum when the solution makes no improvement after a number of iterations. Oone or more visits will be removed from the original tour during this step. For every shaking step, $Remove$ and $Start\_shake$ are set as the inputs of number to remove from the single tour and the place starts the removing. If the end location is

doi:10.6342/NTU202002814

reached during the removal, the process continues after the start location. Meanwhile, the shaking step only allows the visits not selected in the chromosome from GA to move.

After the removal, all visits following the removed visits are shifted to the beginning. The shifted visit should be updated similar to the process shown in the insertion stop. For the visits before the remove. The pseudo for the shaking step can be seen in Fig. 3.6.

```
Shaking step:

For each route:
    if set of visits i=>j not in must:
        Delete visits:
    Calculate Shift;
    For each visit after j(until Shift == 0):
        Update Wait, Arrive, Maxshift, Shift;
        If Punish change then:
            Update TotalScore, Punish
    For each visit before i(until Maxshift remains the same):
        Update Maxshift;

Update score
```

Fig. 3.6 Pseudo code for the shaking

Fig. 3.7 presents the iterated local search heuristics pseudo code for a single scenario. Since the selection in GA does not represent a real network, a local search with time oriented applied to construct a first stage solution, and the route for each scenarios will be optimal from first stage solution. Vehicles not used in the first stage will be set as empty routes. The heuristic follows a loop until no improvements are identified for the best solution determined during a fixed number NI of

doi:10.6342/NTU202002814

times. Firstly, the insertion step is applied to the route. The insertion stop when it reaches a local optimum. If the score of the solution is better than the best score recorded, the score and the solution are recorded and $Remove$ is reset to one for the next shake step. Secondly, the shake step is applied. After each shake step, $Start\_shake$ is increased by the value $Remove$ and $Remove$ is increased by one for the next shake step. If $Start\_shake$ is equal to or greater than the size of the smallest tour, this size is subtracted to determine the new position. If $Remove$ equals the maximum number of removable locations ML, the number is reset to one. By using the shaking parameters as described following, almost all customers, except the selected ones in the GA, are removed at least once in the entire procedure.

doi:10.6342/NTU202002814

```
Iterated local search:

Input: Initial_route,
Output: BestFound

Start_shake←1
Remove←1
NumberOfTimesNoImprovement = 0;
while NumberOfTimesNoImprovement < NI do
        while not local optimum do
                Insert;
        If Solution better than BestFound then
                BestFound←Solution;
                R←1;
                NumberOfTimesNoImprovement = 0;
        Else
                NumberOfTimesNoImprovement←NumberOfTimesNoImprovement+1;
        Shake Solution (R, S);
        Start_shake←Start_shake +Remove;
        Remove = Remove+1;
        If Start_shake >=Size of smallest Tour then
                Start_shake←Start_shake- Size of smallest Tour;
        If Remove== ML then
                Remove←1;
Return BestFound;
```

Fig. 3.7 Pseudo code for the ILS

doi:10.6342/NTU202002814

Chapter 4 Computational Result

In this section, some test instances are designed. All the computational experiments in this research were conducted on a desktop computer with Linux operating system, Intel i7-7700 with CPU @3.6 GHz and 16 GB Random Access Memory (RAM). Results of comparison tested data were solved at the Gurobi Mathematical Programming Solver 8.1 with Python interface, and the optimization algorithm was coded with Python Programming Language 2.7.

4.1 Test instances

Since no test problems for SCMVRPPSTW are available, and the algorithms are modified specifically to deal with this variant, a small test set and a large test set are generated by revising Solomon’s VRPTW benchmarks (refer to Solomon (1987)). The small test set checks the accuracy of two heuristic algorithms while the large test set is aimed to test the calculating time of the algorithms with a more practical size. The data sets uses the original instances from C1 and C2 types while the small test problems contain first 20 customers and the large test problems contain all 100 customers. The descriptions for the problem sets refers to Table 4.1.

Type	**Sets**	**Capacity**	**Maximum Time**	**Service time**	**Customers**	**Width of the time windows**
C1	9	200	1236	90	Clustered	Tight
C2	8	700	3390	90	Clustered	Soft

Table 4.1 Description for Solomon’s problem sets.

doi:10.6342/NTU202002814

In real world, the profit of a good depends on the size and its weight; therefore, the profit earned from each customer is designed to be proportional to the customer's demand. The revenue r_i for customer i is set as quintuple of c_i^q. The operating cost for the vehicle c^1 is set as 10 while the distance traveling cost c^2 is 2.5. Since the time penalty should be avoid, the cost for time penalty c^3 is set as 20. Mostly, customers want to have a successful transaction for the order; thus, the possibility of the customer in the test instances is presented as a number between 0.5 to 0.99. It can be seen that the scenarios have different probability since the customers are associated to uneven probability.

The number of scenarios grows exponentially with the rising number of customers, which means that for each increasing customer, the number of scenarios set will be twice as the original, and it will cost large amount of time to calculate the expected revenue of all scenarios. Since the influence on the expectation of revenue for the scenario is correlated to its probability, the scenario with low possibility have little effect on the outcome; thus, in the problem sets, the scenarios are selected from 10000 random samples. The sample scenarios with the top five highest possibility are selected in small problem test and the top thirty highest possibility are selected in the large problem test for the later approach.

Three groups of different value combination $m_1, m_2, m_3, m_4,$ and m_5 for insertion heuristic are set. The first group sets the profitability of the customer as the priority; therefore, for the first

doi:10.6342/NTU202002814

insertion heuristic, the parameters are set as following: $m_1 = 1, m_2 = 1, m_3 = 2, m_4 = 20, m_5 = 0$. The second type considered the time consuming between each customer; thus, the parameters for the second type are set as: $m_1 = 0, m_2 = 0, m_3 = 1, m_4 = 0, m_5 = 0$. The third type considers both insertion timing and the moving time for customers to avoid the insertion with short distance but large waiting time, which the parameters are $m_1 = 0, m_2 = 0, m_3 = 0.5, m_4 = 0, m_5 = 0.5$. The maximum number of iteration without improvement *NI* and the maximum number of removable locations *ML* are the parameters predetermined in the heuristic. *NI* is set as 150 for initial. For *ML*, a percentage of n/|*K*| (number of customers/ number of vehicles) is used for the second stage test. The initial generation number is set as 3000 for small problem sets and 10000 for large problem sets. The mutation number is set as 0.05 and the reverse number is set as 1. The performance of the heuristic in small problem sets is compared with the result from Gurobi solver with the preset to terminate when running time reached three hours (10800 seconds).

4.2 Results

Table 4.1 shows the result from the solver with different number of vehicles. The column LB represents the best solution found in three hours and UB represents the best upper bound. The gap between the LB and UB is between 0% to 7%. Table 4.2-4.4 compare the outcome obtained from the solver and two heuristic model. 'gap' denotes the difference between result from heuristic and solver. The negative gap means the heuristic has a result that is closer to the optimum solution than

doi:10.6342/NTU202002814

the result from the solver.

Fig. 4.1- 4.3 are the comparisons for the gap in different problem sets. From the figures, it clearly illustrates that the vehicle number has no apparent influence on the gap for the problem sets with wider time windows (c2), which means both heuristic have similar gap in c2-type problem sets whether the allowing vehicle number is large or small. In contrast, the gap for the problem sets with short scheduling horizon (c1) using GA and ILS is much smaller than using the insertion heuristic and ILS with single vehicle and the difference decreases when the vehicle number rises. The largest gap and smallest gap to UB for the heuristic using insertion algorithm and ILS are 37% and 2%, while for the heuristic using GA and ILS are 15 % and 0%. Fig 4.4 and 4.5 depict the computational times for small problem sets and testing time of two heuristics for large problem sets are illustrated in Fig 4.6 and 4.7. It appears that insertion and ILS heuristic have significant advantage of calculation time to GA and ILS, while both heuristics run shorter time to have the result than the solver for small number. There is no significant relation shows between the computational time and the vehicle number, since for small number of vehicle the allowable route changing and improvement is limited, and the adding vehicles are vacant when the best routing plan is found.

doi:10.6342/NTU202002814

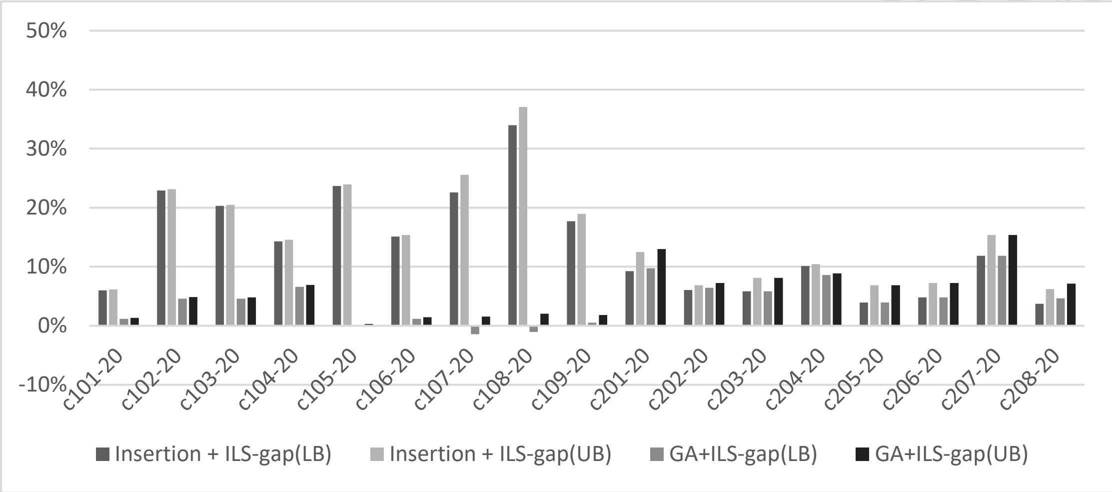

Fig. 4.1 Gap of heuristics for vehicle = 1

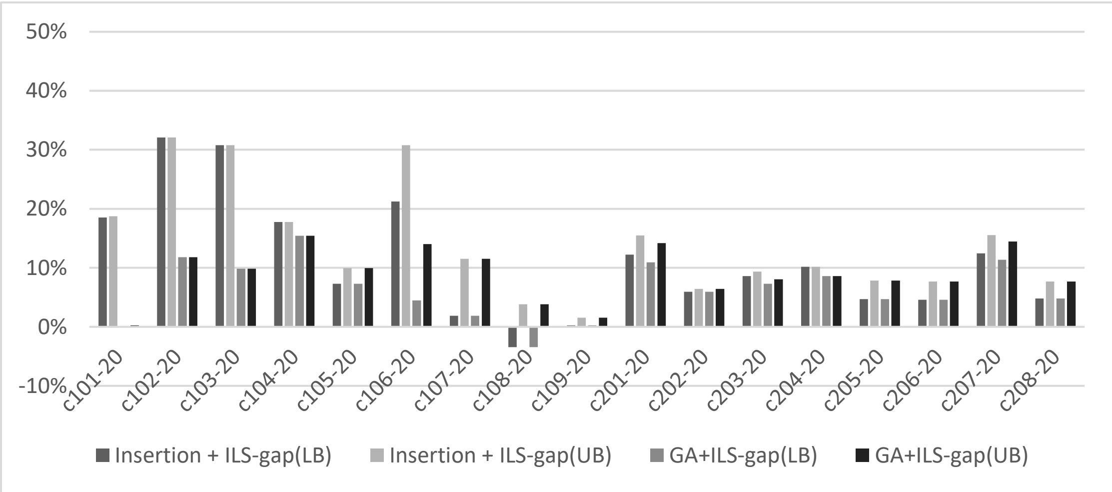

Fig. 4.2 Gap of heuristics for vehicle = 2

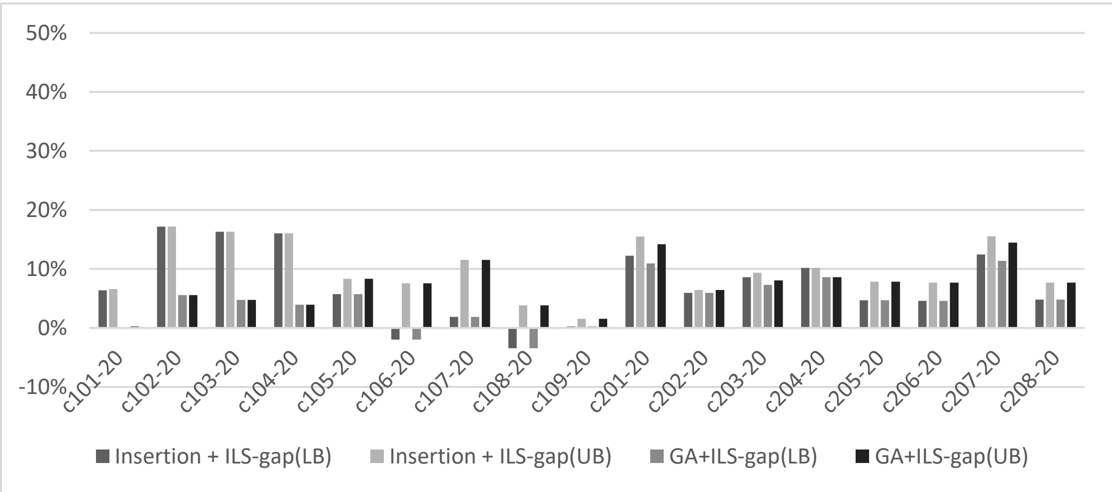

Fig. 4.3 Gap of heuristics for vehicle = 3

doi:10.6342/NTU202002814

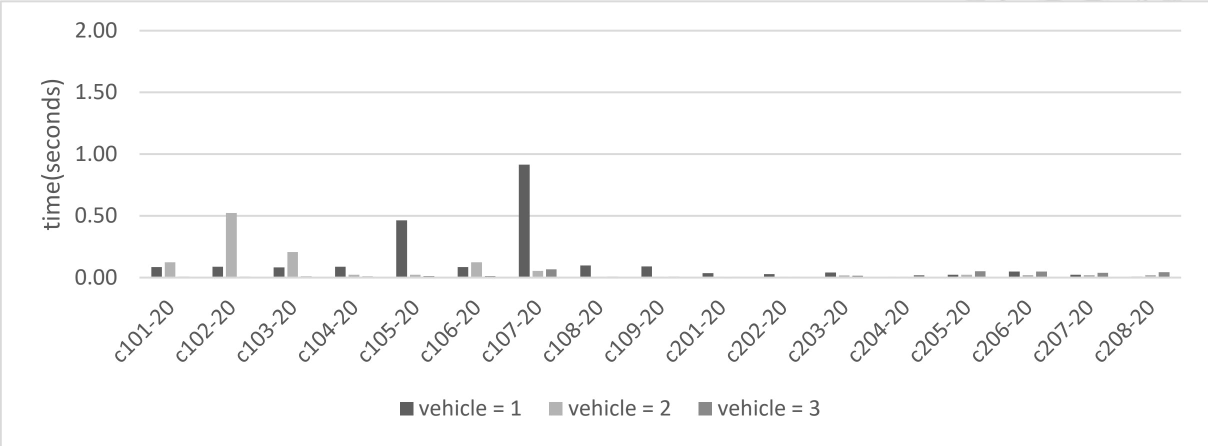

Fig. 4.4 Computational time of Insertion + ILS for small testing

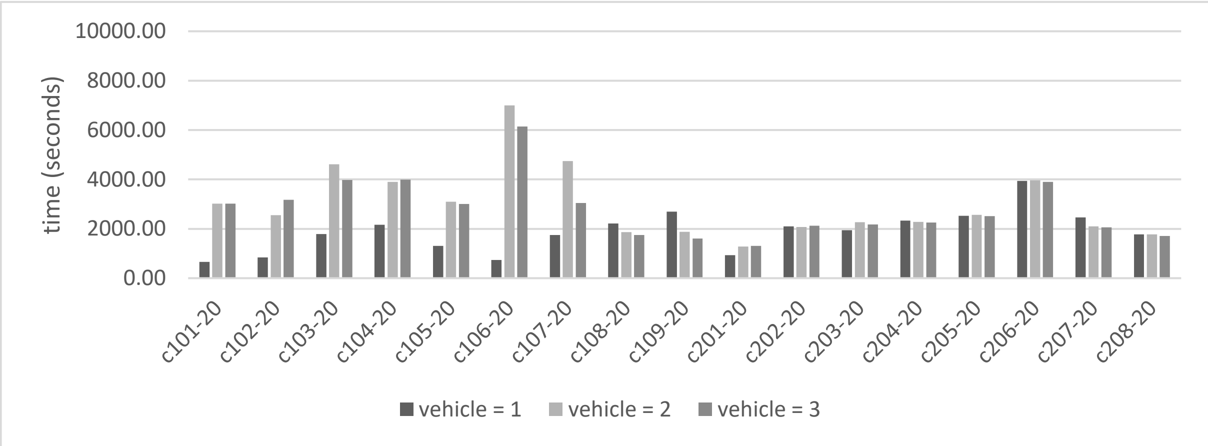

Fig. 4.5 Computational time for GA + ILS for small testing

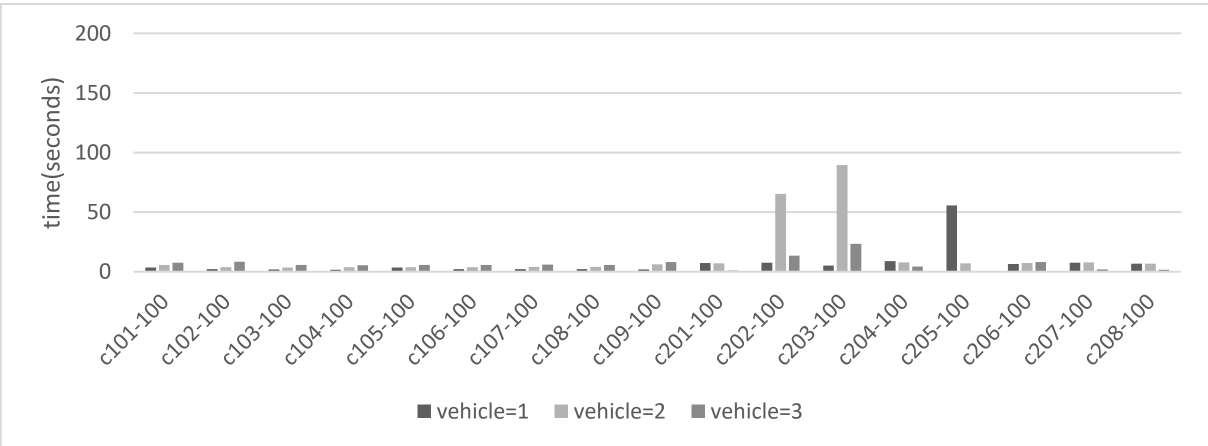

Fig. 4.6 Computational time of Insertion + ILS for large testing

doi:10.6342/NTU202002814

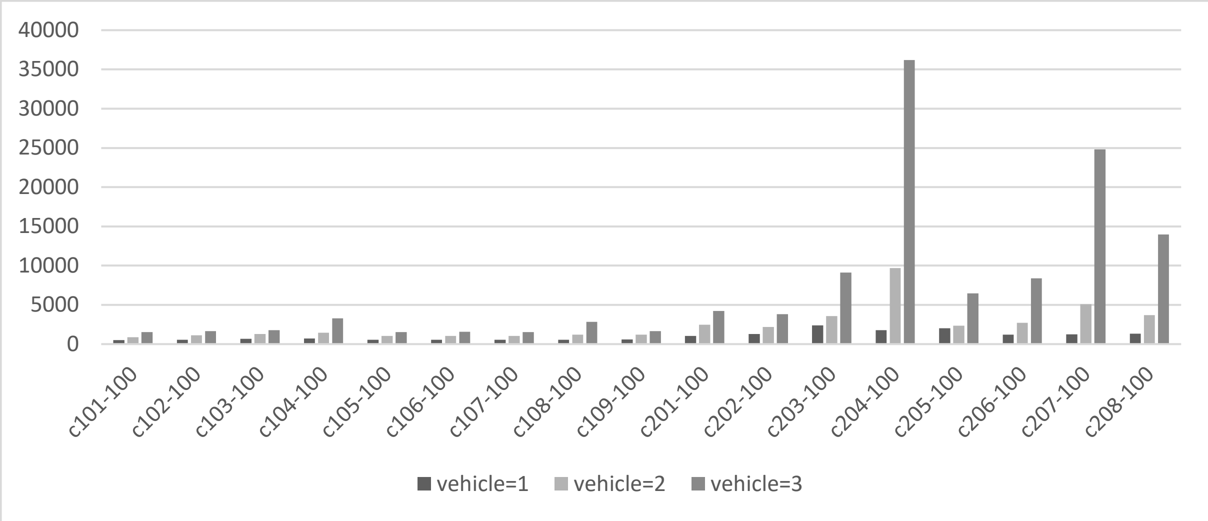

Fig. 4.7 Computational time of GA+ ILS for large testing

doi:10.6342/NTU202002814

	vehicle = 1				vehicle = 2				vehicle=3			
	customer selection	LB	UB	gap	customer selection	LB	UB	gap	customer selection	LB	UB	gap
c101-20	7	741.38	742.59	0.16%	19	1275.67	1278.55	0.22%	19	1275.67	1275.79	0.01%
c102-20	10	757.76	759.77	0.26%	19	1294.78	1294.78	0.26%	19	1294.78	1294.90	0.26%
c103-20	10	757.76	759.17	0.19%	19	1301.59	1301.59	0.19%	19	1301.59	1301.70	0.19%
c104-20	10	757.76	760.01	0.30%	19	1306.83	1306.83	0.30%	19	1306.83	1306.95	0.30%
c105-20	10	746.04	748.27	0.30%	18	1267.10	1300.24	0.30%	19	1275.67	1293.01	0.30%
c106-20	7	741.52	743.74	0.30%	15	1171.27	1282.95	0.30%	19	1275.67	1278.65	0.30%
c107-20	8	735.53	757.59	3.00%	16	1196.92	1312.55	3.00%	19	1235.36	1317.39	3.00%
c108-20	7	738.36	761.12	2.99%	17	1223.18	1312.14	6.78%	18	1235.42	1320.24	6.42%
c109-20	9	750.48	759.97	2.13%	18	1293.50	1310.34	2.13%	19	1274.60	1318.64	2.13%
c201-20	16	1188.84	1227.40	2.53%	19	1204.15	1243.39	2.53%	19	1208.96	1252.64	2.53%
c202-20	19	1245.68	1255.70	2.93%	19	1250.75	1257.42	2.93%	19	1250.56	1261.59	2.93%
c203-20	18	1229.96	1257.80	3.33%	19	1249.62	1259.26	3.33%	19	1248.49	1259.56	3.33%
c204-20	19	1261.84	1265.63	3.73%	19	1262.36	1262.36	3.73%	19	1262.36	1266.15	3.73%
c205-20	17	1213.25	1249.33	4.13%	18	1223.31	1261.79	4.13%	19	1220.76	1260.50	4.13%
c206-20	19	1224.49	1254.30	2.38%	18	1221.58	1259.41	3.00%	19	1222.61	1257.55	2.78%
c207-20	17	1208.26	1250.85	4.53%	18	1216.45	1254.03	4.53%	18	1215.85	1257.02	4.53%
c208-20	19	1222.70	1252.91	4.93%	19	1224.58	1259.65	4.93%	19	1228.54	1258.58	4.93%

Table 4.2 Testing outcome for Gurobi solver for 3hours

doi:10.6342/NTU202002814

	Gurobi solver(3hr)			Insertion + ILS				GA+ILS			
	customer selection	LB	UB	customer selection	value	gap(LB)	gap(UB)	customer selection	value	gap(LB)	gap(UB)
c101-20	7	741.38	742.59	9	696.76	6%	6%	11	732.87	1%	1%
c102-20	10	757.76	759.77	8	584.28	23%	23%	9	672.91	11%	11%
c103-20	10	757.76	759.17	11	603.89	20%	20%	11	675.65	11%	11%
c104-20	10	757.76	760.01	8	649.51	14%	15%	12	707.64	7%	7%
c105-20	10	746.04	748.27	11	569.48	24%	24%	11	746.04	0%	0%
c106-20	7	741.52	743.74	10	629.66	15%	15%	11	732.87	1%	1%
c107-20	8	735.53	757.59	11	569.48	23%	26%	11	746.04	-1%	2%
c108-20	7	738.36	761.12	11	487.36	34%	37%	11	746.04	-1%	2%
c109-20	9	750.48	759.97	12	617.57	18%	19%	11	746.46	1%	2%
c201-20	16	1188.84	1227.40	17	1078.55	9%	13%	18	1072.73	10%	13%
c202-20	19	1245.68	1255.70	18	1170.06	6%	7%	19	1165.25	6%	7%
c203-20	18	1229.96	1257.80	18	1158.28	6%	8%	19	1158.28	6%	8%
c204-20	19	1261.84	1265.63	20	1133.90	10%	10%	19	1153.50	9%	9%
c205-20	17	1213.25	1249.33	19	1165.78	4%	7%	20	1165.78	4%	7%
c206-20	19	1224.49	1254.30	19	1165.78	5%	7%	20	1165.78	5%	7%
c207-20	17	1208.26	1250.85	19	1065.19	12%	15%	20	1065.19	12%	15%
c208-20	19	1222.70	1252.91	19	1177.22	4%	6%	20	1165.78	5%	7%
average gap						14%	15%			5%	7%

Table 4.3 Solution value and gap for vehicle = 1

doi:10.6342/NTU202002814

	Gurobi solver(3hr)			Insetion+ILS				GA+ILS			
	customer selection	LB	UB	customer selection	value	gap(LB)	gap(UB)	customer selection	value	gap(LB)	gap(UB)
c101-20	19	1275.67	1278.55	17	1039.63	19%	19%	19	1275.67	0%	0%
c102-20	19	1294.78	1294.78	14	879.48	32%	32%	17	1142.01	12%	12%
c103-20	19	1301.59	1301.59	14	900.87	31%	31%	18	1173.26	10%	10%
c104-20	19	1306.83	1306.83	18	1074.39	18%	18%	18	1105.11	15%	15%
c105-20	18	1267.10	1300.24	19	1174.26	7%	10%	20	1174.26	7%	10%
c106-20	15	1171.27	1282.95	17	922.70	21%	31%	18	1118.72	4%	14%
c107-20	16	1196.92	1312.55	19	1174.26	2%	12%	20	1174.26	2%	12%
c108-20	17	1223.18	1312.14	20	1265.28	-3%	4%	19	1265.28	-3%	4%
c109-20	18	1293.50	1310.34	20	1290.48	0%	2%	18	1290.48	0%	2%
c201-20	19	1204.15	1243.39	20	1056.82	12%	15%	18	1072.73	11%	14%
c202-20	19	1250.75	1257.42	20	1176.70	6%	6%	19	1176.70	6%	6%
c203-20	19	1249.62	1259.26	19	1142.26	9%	9%	19	1158.28	7%	8%
c204-20	19	1262.36	1262.36	20	1133.90	10%	10%	19	1153.50	9%	9%
c205-20	18	1223.31	1261.79	19	1165.78	5%	8%	20	1165.78	5%	8%
c206-20	18	1221.58	1259.41	19	1165.78	5%	8%	20	1165.78	5%	8%
c207-20	18	1216.45	1254.03	19	1065.19	12%	16%	19	1078.17	11%	14%
c208-20	19	1224.58	1259.65	19	1165.78	5%	8%	20	1165.78	5%	8%
average gap						11%	14%			6%	9%

Table 4.4 Solution value and gap for vehicle = 2

doi:10.6342/NTU202002814

	Gurobi solver(3hr)			Insetion+ILS				GA+ILS			
	customer selection	LB	UB	customer selection	value	gap(LB)	gap(UB)	customer selection	value	gap(LB)	gap(UB)
c101-20	19	1275.67	1278.55	20	1194.55	6%	7%	19	1275.67	0%	0%
c102-20	19	1294.78	1294.78	20	1072.69	17%	17%	20	1222.69	6%	6%
c103-20	19	1301.59	1301.59	20	1089.69	16%	16%	20	1239.69	5%	5%
c104-20	19	1306.83	1306.83	20	1097.33	16%	16%	18	1255.11	4%	4%
c105-20	18	1267.10	1300.24	20	1194.55	6%	8%	20	1194.55	6%	8%
c106-20	15	1171.27	1282.95	20	1194.55	-2%	8%	20	1194.55	-2%	8%
c107-20	16	1196.92	1312.55	19	1174.26	2%	12%	20	1174.26	2%	12%
c108-20	17	1223.18	1312.14	20	1265.28	-3%	4%	20	1265.28	-3%	4%
c109-20	18	1293.50	1310.34	20	1290.48	0%	2%	20	1290.48	0%	2%
c201-20	19	1204.15	1243.39	20	1056.82	12%	15%	18	1072.73	11%	14%
c202-20	19	1250.75	1257.42	20	1176.70	6%	6%	20	1176.70	6%	6%
c203-20	19	1249.62	1259.26	19	1142.26	9%	9%	19	1158.28	7%	8%
c204-20	19	1262.36	1262.36	20	1133.90	10%	10%	19	1153.50	9%	9%
c205-20	18	1223.31	1261.79	19	1165.78	5%	8%	20	1165.78	5%	8%
c206-20	18	1221.58	1259.41	19	1165.78	5%	8%	20	1165.78	5%	8%
c207-20	18	1216.45	1254.03	19	1065.19	12%	16%	19	1078.17	11%	14%
c208-20	19	1224.58	1259.65	19	1165.78	5%	8%	20	1165.78	5%	8%
average gap						7%	10%			4%	7%

Table 4.5 Solution value and gap for vehicle = 3

doi:10.6342/NTU202002814

The small test problems with different soft time window allowable intervals are examined with the insertion heuristic. The trend of objective value for the soft time window interval for both problem sets are displayed in Fig.4.8-4.11. It appears that only c108-20, c201-100,202-100, c204-100 and c208-100 show little improvement with the rising of interval. The penalty for arriving-late time and the limitation for the total time T_{max} restricts the influence for the soft time window.

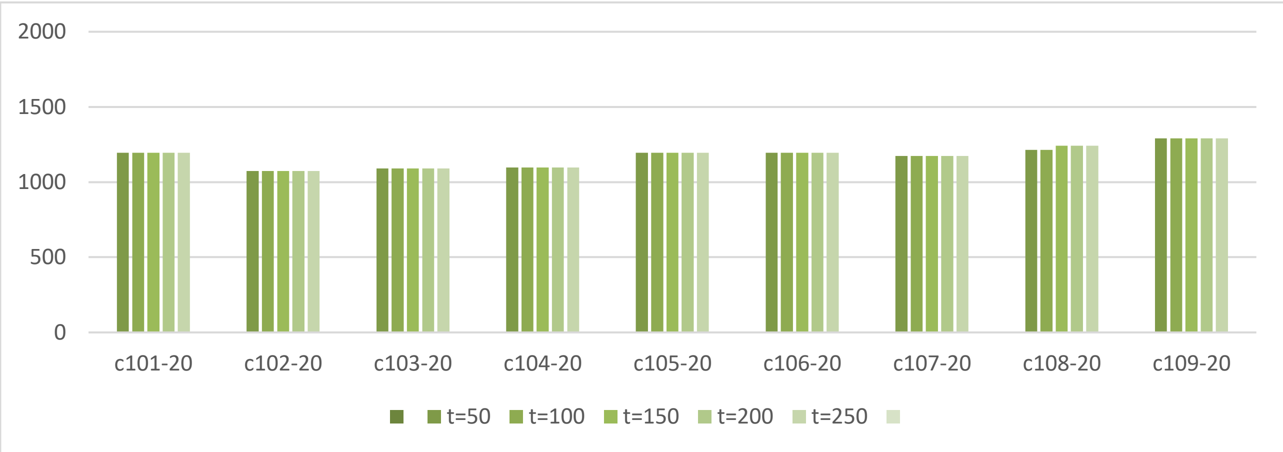

Fig. 4.8 Objective revenue for small test instances c1 in different allowable soft time window limit t

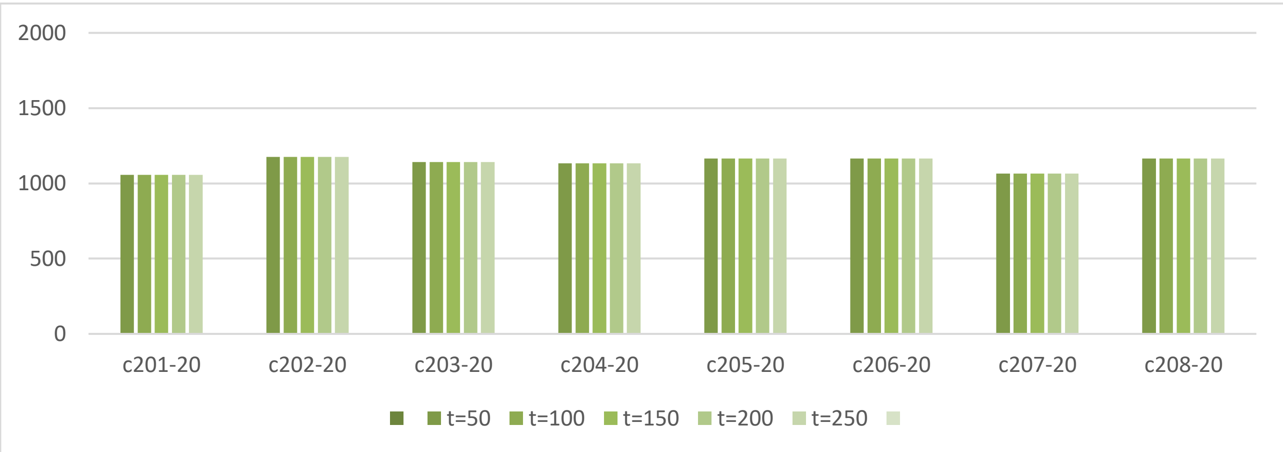

Fig. 4.9 Objective revenue for small test instances c2 in different allowable soft time window limit t

doi:10.6342/NTU202002814

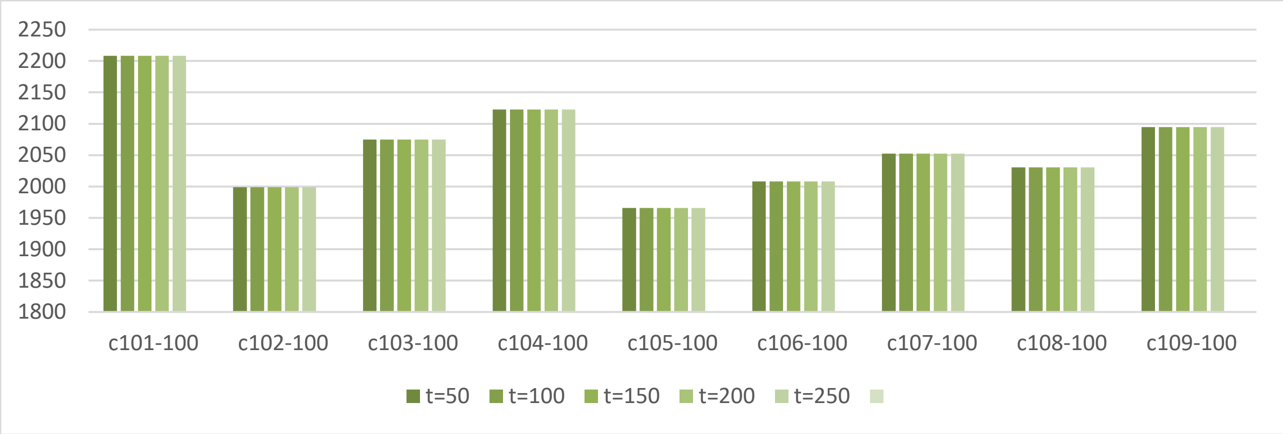

Fig. 4.10 Objective revenue for large test instances c1 in different allowable soft time window limit

t

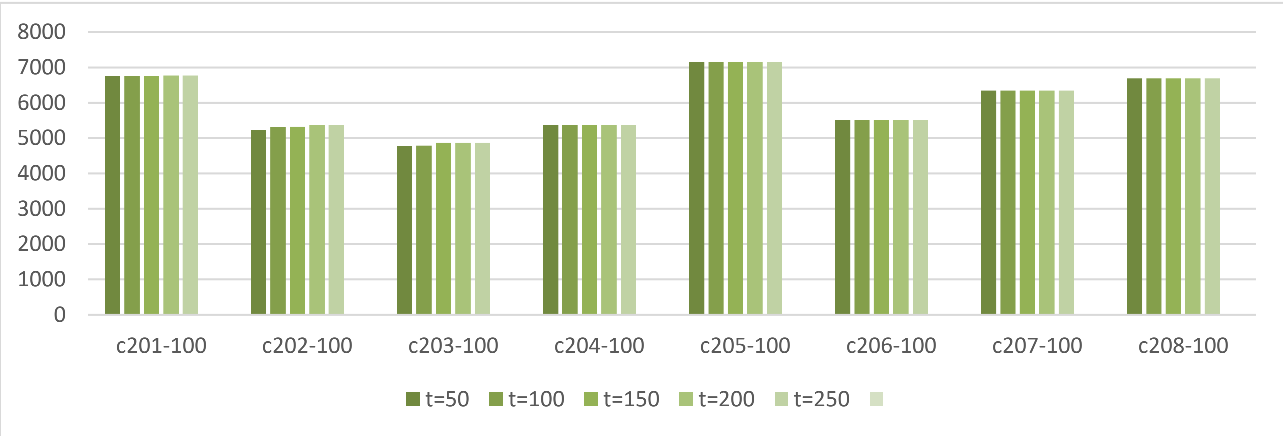

Fig. 4.11 Objective revenue for large test instances c2 in different allowable soft time window limit

t

doi:10.6342/NTU202002814

Chapter 5 Conclusions and Future research

Nowadays, the e-commerce has brought a profound change for economy and society. With the further development of e-commerce, the uncertainty for the customers with the last-mile delivery has brought more and more attention for the planning in logistic. In this research, a SCMVRPPSTW model is constructed to make an application for logistic routing planning problem. Considering the stochasticity of the problem, two two-stage heuristic algorithms are proposed. For the first stage method, the first algorithm simply used an insertion heuristic to build the initial route while the second algorithm adding the genetic algorithm to escape the local optima and search for better result. The second stage solving method is based on an existing metaheuristic developed for the TOPTW, which can proposed a fast re-optimizing for the adjusted routing.

The approximate test problems are generated by revising the Solomon's benchmarks test problems. The results from the algorithms were compared with the result from MIP model using Gurobi Solver running for 10800 seconds. In the small testing, the GA with ILS has more standard presents than insertion with ILS, while both heuristics are able to obtain optimal or near optimal solutions for the tested problems in an acceptable computational time.

There are several directions for the future study. In this study, routes are assumed to start as a distribution center and end at a collection center. Therefore, it is a single depot problem. For large companies, the goods may be stored in more than one collection center which vehicles can be

doi:10.6342/NTU202002814

stationed. Thus, how to extend one depot to multiple depots is a direction of the further study.

Meanwhile, each vehicle is assumed to perform at most one route in the same planning period in this research. In some practical applications, the vehicle capacity is small or the planning period is large, performing more than one route per vehicle may be more appropriate for practical implementation. In urban areas, where travel times are rather small, it is often the case that after performing short tours vehicles are reloaded and used again. Hence, how to extend one trip to multiple trips is also a direction of the future works.

Moreover, the study only consider the stochasticity of the customer for logistics problem and the traveling time for each routes is set as known. In real practice, the traveling time is usually not certain due to other time dependent properties of the network such as congestion levels, incident location, and construction zone on certain road segments. Sometimes when facing a traffic jam, the deliverymen change the routing while the delivery. Therefore, how to extend the problem with a dynamic planning is also a direction of the future works.

doi:10.6342/NTU202002814

REFERENCE

[1] Aghezzaf, B., & Fahim, H. E. (2014, June). The multi-constraint team orienteering problem with time windows in the context of distribution problems: A variable neighborhood search algorithm. In *2014 International Conference on Logistics Operations Management* (pp. 155-160). IEEE.

[2] Archetti, C., Bertazzi, L., Laganà, D., & Vocaturo, F. (2017). The undirected capacitated general routing problem with profits. *European Journal of Operational Research*, *257*(3), 822-833.

[3] Archetti, C., Feillet, D., Hertz, A., & Speranza, M. G. (2009). The capacitated team orienteering and profitable tour problems. *Journal of the Operational Research Society*, *60*(6), 831-842.

[4] Archetti, C., Bianchessi, N., & Speranza, M. G. (2013a). Optimal solutions for routing problems with profits. *Discrete Applied Mathematics*, *161*(4-5), 547-557.

[5] Archetti, C., Bianchessi, N., & Speranza, M. G. (2013b). The capacitated team orienteering problem with incomplete service. *Optimization letters*, *7*(7), 1405-1417.

[6] Archetti, C., Bianchessi, N., Speranza, M. G., & Hertz, A. (2014a). The split delivery capacitated team orienteering problem. *Networks*, *63*(1), 16-33.

[7] Archetti, C., Bianchessi, N., Speranza, M. G., & Hertz, A. (2014b). Incomplete service

doi:10.6342/NTU202002814

and split deliveries in a routing problem with profits. *Networks*, *63*(2), 135-145.

[8] Blanchard, D. (2007). Supply chains also work in reverse. *Industry Week*, *1*(1), 48-49.

[9] Boussier, S., Feillet, D., & Gendreau, M. (2007). An exact algorithm for team orienteering problems. *4or*, *5*(3), 211-230.

[10] Butt, S. E., & Cavalier, T. M. (1994). A heuristic for the multiple tour maximum collection problem. *Computers & Operations Research*, *21*(1), 101-111.

[11] Campbell, A. M., Gendreau, M., & Thomas, B. W. (2011). The orienteering problem with stochastic travel and service times. *Annals of Operations Research*, *186*(1), 61-81.

[12] Morganti, E., Seidel, S., Blanquart, C., Dablanc, L., & Lenz, B. (2014). The impact of e-commerce on final deliveries: alternative parcel delivery services in France and Germany. *Transportation Research Procedia*, *4*(0), 178-190.

[13] eMarketer. (2020). E-commerce share of total global retail sales from 2015 to 2023. Retrieved from https://www.statista.com/statistics/534123/e-commerce-share-of-retail-sales-worldwide/

[14] eMarketer. (2020). Retail e-commerce sales worldwide from 2014 to 2023(in billion U.S. dollars). Retrieved from https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/

[15] eMarketer. (2020). Annual retail e-commerce sales growth worldwide from 2017 to 2023.

doi:10.6342/NTU202002814

Retrieved from https://www.statista.com/statistics/288487/forecast-of-global-b2c-e-commerce-growth/

[16] Alex Rolfe (2017, August 2). Worldwide retail e-commerce sales reach $1.9 trillion in 2016. *Payments Cards & Mobile.* Retrieved from: https://www.paymentscardsandmobile.com/retail-e-commerce-sales-reach-1-9-trillion/

[17] Evers, L., Glorie, K., Van der Ster, S., Barros, A.I. and Monsuur, H. (2014). A two-stage approach to the orienteering problem with stochastic weights. *Computers & Operations Research*, *43*, 248-260.

[18] Gambardella, L. M., Montemanni, R., & Weyland, D. (2012). Coupling ant colony systems with strong local searches. *European Journal of Operational Research*, *220*(3), 831-843.

[19] Garcia, A., Vansteenwegen, P., Souffriau, W., Arbelaitz, O., & Linaza, M. (2009). *Solving multi constrained team orienteering problems to generate tourist routes.*

[20] Gevaers, R., Van de Voorde, E., & Vanelslander, T. (2011). Characteristics and typology of last-mile logistics from an innovation perspective in an urban context. *City Distribution and Urban Freight Transport: Multiple Perspectives, Edward Elgar Publishing*, 56-71.

[21] Golden, B. L., Levy, L., & Vohra, R. (1987). The orienteering problem. *Naval Research Logistics (NRL)*, *34*(3), 307-318.

doi:10.6342/NTU202002814

[22] Gunawan, A., Lau, H. C., & Vansteenwegen, P. (2016). Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, *255*(2), 315-332.

[23] Hu, Q., & Lim, A. (2014). An iterative three-component heuristic for the team orienteering problem with time windows. *European Journal of Operational Research*, *232*(2), 276-286.

[24] Ilhan, T., Iravani, S. M., & Daskin, M. S. (2008). The orienteering problem with stochastic profits. *Iie Transactions*, *40*(4), 406-421.

[25] Jepsen, M. K. (2011). *Branch-and-cut and branch-and-cut-and-price algorithms for solving vehicle routing problems*.

[26] Labadie, N., Mansini, R., Melechovský, J., & Calvo, R. W. (2012). The team orienteering problem with time windows: An lp-based granular variable neighborhood search. *European Journal of Operational Research*, *220*(1), 15-27.

[27] Lin, S. W., & Vincent, F. Y. (2012). A simulated annealing heuristic for the team orienteering problem with time windows. *European Journal of Operational Research*, *217*(1), 94-107.

[28] Luo, Z., Cheang, B., Lim, A., & Zhu, W. (2013). An adaptive ejection pool with toggle-rule diversification approach for the capacitated team orienteering problem. *European*

doi:10.6342/NTU202002814

Journal of Operational Research, *229*(3), 673-682.

[29] LM, M. R. G. (2009). Ant colony system for team orienteering problem with time windows. *Foundations of Computing and Decision Sciences*, *344*, 287-306.

[30] Papapanagiotou, V., Montemanni, R., & Gambardella, L. (2014). Objective function evaluation methods for the orienteering problem with stochastic travel and service times. *Journal of Applied Operations Research*, *6*(1), 16-29.

[31] Park, J., Lee, J., Ahn, S., Bae, J., & Tae, H. (2017). Exact algorithm for the capacitated team orienteering problem with time windows. *Mathematical Problems in Engineering*, *2017*.

[32] Ritzinger, U., Puchinger, J., & Hartl, R. F. (2016). A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*, *54*(1), 215-231.

[33] Sun, P., Veelenturf, L. P., Dabia, S., & Van Woensel, T. (2018). The time-dependent capacitated profitable tour problem with time windows and precedence constraints. *European Journal of Operational Research*, *264*(3), 1058-1073.

[34] Tarantilis, C. D., Stavropoulou, F., & Repoussis, P. P. (2013). The capacitated team orienteering problem: a bi-level filter-and-fan method. *European Journal of Operational Research*, *224*(1), 65-78.

[35] Tarantilis, C. D., Stavropoulou, F., & Repoussis, P. P. (2013). The capacitated team

doi:10.6342/NTU202002814

orienteering problem: a bi-level filter-and-fan method. *European Journal of Operational Research*, *224*(1), 65-78.

[36] Toth, P., & Vigo, D. (Eds.). (2014). *Vehicle routing: problems, methods, and applications*. Society for Industrial and Applied Mathematics.

[37] Tsiligirides, T. (1984). Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, *35*(9), 797-809.

[38] Vansteenwegen, P., Souffriau, W., Berghe, G. V., & Van Oudheusden, D. (2009). Iterated local search for the team orienteering problem with time windows. *Computers & Operations Research*, *36*(12), 3281-3290.

[39] Vansteenwegen, P., Souffriau, W., & Van Oudheusden, D. (2011). The orienteering problem: A survey. *European Journal of Operational Research*, *209*(1), 1-10.

[40] Vansteenwegen, P., & Gunawan, A. (2019). *Orienteering problems: Models and algorithms for vehicle routing problems with profits*. Springer Nature.

[41] Weber, C. L., Koomey, J. G., & Matthews, H. S. (2010). The energy and climate change implications of different music delivery methods. *Journal of Industrial Ecology*, *14*(5), 754-769.

[42] Wang, X., Golden, B., & Gulczynski, D. (2014). A worst-case analysis for the split delivery capacitated team orienteering problem with minimum delivery

doi:10.6342/NTU202002814

amounts. *Optimization letters*, *8*(8), 2349-2356.

[43] Li, Z., & Hu, X. (2011, August). The team orienteering problem with capacity constraint and time window. In *The 10th International Symposium on Operations Research and its Applications (ISORA 2011)* (pp. 157-163).

[44] 瘋時尚（民 108 年 8 月 14 日）。德國網購退貨率居歐洲之首，Zalando、亞馬遜等電商紛紛採取應對措施。取自： https://ifashiontrend.com/zalando-amazon-return-policy/

doi:10.6342/NTU202002814