A study on lane reservation problems in transportation networks

Peng Wu

To cite this version:

Peng Wu. A study on lane reservation problems in transportation networks. Operations Research [math.OC]. Université Paris-Saclay; Université d'Evry-Val-d'Essonne; Northwestern Polytechnical University (Chine), 2016. English. NNT : . tel-01758478

HAL Id: tel-01758478
https://hal.science/tel-01758478

Submitted on 4 Apr 2018

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

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

NNT : 2016SACLE016

THESE DE DOCTORAT
DE
NORTHWESTERN POLYTECHNICAL UNIVERSITY
ET DE
L’UNIVERSITE PARIS-SACLAY
PREPAREE A L’ L’UNIVERSITE D’EVRY VAL-D’ESSONNE

ÉCOLE DOCTORALE N°580
STIC : Sciences et technologies de l’information et de la communication

Spécialité de doctorat : Mathématiques et Informatique

Par

Mr Peng WU

Études de problèmes de réservation de voie dans des réseaux de transport

Thèse présentée et soutenue à Evry, le 30 juin 2016 :

Composition du Jury :

M. J. -M. DELOSME, Professeur, Université d’Evry Val d’Essonne, Président du Jury
M. H. ALLAOUI, Professeur, Université d’Artois, Rapporteur
M. A. JOUGLET, Maître de Conférences et HDR, Université de Technologie de Compiègne, Rapporteur
M. K. BARKAOUI, Professeur, Cédric Cnam Paris, Rapporteur, Examinateur
M. S. MAMMAR, Professeur, Université d’Evry Val d’Essonne, Examinateur
M. W. ZHOU, Maître de Conférences et HDR, ESCP Europe Paris Campus, Examinateur
Mme F. CHU, Professeur, Université d’Evry Val d’Essonne, Directrice de thèse
M. A. CHE, Professeur, Northwestern Polytechnical University, Co-Directeur de thèse

A Study on Lane Reservation Problems in Transportation Networks

by

Peng WU

Laboratoire d'Informatique, Biologie Intègrative et Systèmes Complexes (IBISC)

Université d'Evry Val d'Essonne, France

Supervisors: Prof. Feng CHU and Prof. Ada CHE

June 30, 2016

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisors, Prof. Feng CHU and Prof. Ada CHE for their guidances, support and help throughout my doctoral career. I would like to thank them for guiding me into the operations research area, for leading me how to identify key scientific problems, better present and evaluate the ideas and results, and for helping me to gradually develop to be a better researcher. I also want to thank them for teaching me so much in my life.

Specially, I would like to express my sincere gratitude to Prof. Hamid ALLAOUI (Université d'Artois), Prof. Antoine JOUGLET (Maître des Conférences et HDR, Université de Technologie de Compiègne), Prof. Kamel BARKAOUI (Cédric Cnam Paris), Prof. Jean-Marc DELOSME (Université d'Evry Val d'Essonne (UEVE)), Prof. Saïd MAMMAR (UEVE), Prof. Wei ZHOU (Maître des Conférences et HDR, ESCP Europe Paris Campus) who kindly agreed to serve as jury members in my PhD. defense committee and give their insightful comments on this thesis.

I would like to show my appreciation to Prof. Chengbin CHU (Université Paris-Saclay) and Prof. MengChu ZHOU (New Jersey Institute of Technology, USA) for their insightful comments and constructive suggestions on my work. My gratitude also goes to my friends Dr. Yunfei FANG and Dr. Zhen ZHOU for their help and valuable advice.

I am grateful to Prof. Franck DELAPLACE, Prof. Hanna KLAUDEL, Madame Murielle BOURGEOIS in Lab. IBISC for their sympathetic help during my study in France. I would like to thank all my friends and talent colleagues both in Lab. IBISC, UEVE and the Management System Engineering Research Unit, Northwestern Polytechnical University for sharing the research environment and experience and their help.

Last but not the least, I would like to express special thanks to my beloved family for their unconditional caring and endless love all through these years and I would like to thank my lover and comrade Junheng CHENG for her continuous support and love.

Résumé

Aujourd'hui, le transport est devenu indispensable dans la vie quotidienne. Cependant, la congestion du trafic du fait de la forte urbanisation et de l'augmentation rapide du nombre de véhicules a réduit l'efficacité du système de transport et a causé d'énorme pollution urbaine. Dans ce contexte, pour répondre aux besoins spécifiques de transport et améliorer la performance des systèmes de transport, la réservation de voie, en tant que stratégie de gestion du trafic flexible, a été largement mise en œuvre. La majorité des études existantes sur la réservation de voie se focalisent au niveau microscopique, par exemple, un segment de route principale. Dans cette thèse, nous nous concentrons sur la réservation optimale des voies dans un réseau de transport au niveau macroscopique en minimisant son impact négatif pour deux catégories de problèmes. Nous étudions d'abord des problèmes de réservation robuste de voie et de grande taille pour les futurs poids lourds intelligents et les grands événements spéciaux. Ensuite, nous étudions la réservation de voie dans le but d'améliorer la performance du transport public avec des hypothèses spécifiques. Pour chaque problème étudié dans cette thèse, des modèles appropriés sont construits et leurs complexités sont analysées. Différentes approches de résolution sont élaborées en fonction des caractéristiques des problèmes, à savoir : une méthode exacte à deux phases, une méthode de ε-contrainte, une méthode de "cut and solve", et une méthode de "kernel search". La performance des algorithmes proposés est évaluée à l'aide de benchmarks et d'instances générées aléatoirement. Les expériences numériques montrent que les algorithmes proposés sont plusperformants que les algorithmes existant dans la littérature et le progiciel commercial CPLEX.

Mots clés: Planification et gestion des transports, Réservation de voie, Transport de marchandises, Réseaux de bus, Optimisation combinatoire, Algorithmes

Abstract

Nowadays, transportation has become an indispensable part in modern life. However, heavy traffic congestion due to high urbanization and rapid increase of vehicles has caused low transportation efficiency and huge amounts of urban pollution. In this context, to meet special transportation requirements and improve the performance of transportation systems, lane reservation, as a flexible and economic traffic management strategy, has been widely implemented in real life. The majority of studies about lane reservation in the literature focus on the impact at a microscope level, e.g., a single link or corridor. In this thesis, we focus on optimally reserving lanes at a macroscopic network level with the objective of minimizing negative impact for two categories of problems. We firstly investigate the large-size and robust lane reservation problems in the contexts of future automated truck freight transportation and large-scale special events. Then, we study lane reservation for improving the performance of bus transit system under different assumptions. For all problems studied in this thesis, appropriate models are provided and their complexities are analyzed. Different resolution approaches are developed according to the characteristics of problems, including exact two-phase method, exact ε-constraint based method, cut-and-solve method, and kernel search method. The performance of the proposed algorithms is evaluated by benchmark and randomly generated instances. Extensive numerical experiments show the proposed algorithms outperform the state-of-the-art algorithms and the commercial software CPLEX.

Keywords: Transportation planning and management; Lane reservation; Freight transportation; Bus transit systems; Combinatorial optimization; Algorithms

Contents

Notation

N: set of nodes in the network G and G'

A: set of directed arcs in the network G

K: set of tasks or OD pairs

L: set of bus lines

N_l: set of bus stops with a given passing order on bus line l, $l \in L$

A_l: set of arcs in G on bus line l with a given passing order, $A_l \subseteq A$

$L(a)$: set of bus lines containing arc $a \in A$, $L(a) \subseteq L$

A' set of arcs in G'

A'_l set of arcs on line l, $A'_l \subseteq A'$

A'^{+}_i: set of arcs coming into node $i \in N$, $A'^{+}_i \subseteq A'$

A'^{-}_i: set of arcs outgoing from node $i \in N$, $A'^{-}_i \subseteq A'$

$A'(a)$ set of arcs in A' corresponding to arc $a \in A$, $A'(a) \subseteq A'$

O: set of origin nodes

D: set of destination nodes

o_k: origin node of task or OD pair $k \in K$

d_k: destination node of task or OD pair $k \in K$

s_l: start stop of bus line $l \in L, s_l \in N_l$

d_l: terminal stop of bus line $l \in L, d_l \in N_l$

f_l: number of buses on the l-th bus line per unit of time, $l \in L$

C_{ij}: negative traffic impact due to reserving a lane on arc $(i, j) \in A$

C_a: negative impact of implementing a bus lane on arc $a \in A$

B: available bus operating budget expressed by the total transit time

D_k: amount of passengers of OD pair $k \in K$

P_T: penalty time per transfer (i.e., changing a line)

Q_{ij}: threshold of bus volume per unit time for reserving a lane on arc $(i, j) \in A$

S^l_{ij}: $S^l_{ij} = 1$: bus line $l \in L$ passes arc $(i, j) \in A$ and 0 otherwise

T_k: travel deadline to accomplish task $k \in K$

T_l: travel deadline for bus line $l \in L$

T_0: departure time of buses at start stops

τ_{ij}: travel time on a reserved lane on arc $(i, j) \in A$

τ'_{ij}: travel time on arc $(i, j) \in A$ without reserved lanes

τ''_{ij}: travel time on general-purpose lanes of arc $(i, j) \in A$ with a reserved lane

τ_a travel time on a bus lane on arc $a \in A, A'$

τ'_a travel time on arc $a \in A, A'$ without bus lanes

$T^-_{i,l}$: lower bound on arrival time at the i-th stop on bus line l, $i \in \{2, .., |N_l|\}$, $l \in L$;

$T^+_{i,l}$: upper bound on arrival time at the i-th stop on bus line l, $i \in \{2, .., |N_l|\}$, $l \in L$;

M: a large positive number

List of Figures

List of Tables

Chapter 1

Introduction

In this thesis, we investigate a new class of transportation planning problem called lane reservation problem (LRP). It mainly focuses on reconfiguring transportation networks by introducing the flexible and economic lane reservation strategy in order to meet special transportation needs or improve the performance of the existing transportation system. The objective of this thesis is to develop a methodology and tool to support optimal lane reservation decision from the macroscopic network. In this chapter, we first introduce the research background and then present the contribution and outline of this thesis.

1.1 Background

With the rapid development of economies, high urbanization has become a reality in many countries around the world. One of its resulted negative consequences is heavy traffic congestion due to the rapid increases of vehicles. Traffic situations in many large cities have been worse than ever before. The increasing traffic congestion causes many transportation problems, such as low transport efficiency, unpredictable transport time, traffic accidents, fuel waste and safety issues. These problems increasingly prevent the transportation from being operated in an efficient, reliable and safe fashion. Faced with such situations, novel transportation planning methods and tools have to be developed to aid the decision of transportation managers.

One natural and traditional approach for solving these problems is to expand the transportation network via constructing new traffic infrastructures. However, this requires a large amount of financial and personnel costs and long duration. In the meantime, it is usually restricted by limited geographic space. Therefore, reconfiguring transportation network and making full use of the existing transportation infrastructures via rationalized management ways to improve the traffic situation becomes increasingly important.

In recent years, lane reservation (LR), as a flexible, efficient, and economic traffic management strategy has been widely applied in many countries. Its core idea is to reserve lanes in an existing transportation network for specific users in order to meet their transportation needs or improve the performance of the transportation system. These reserved lanes can provide a congestion-free and safer transportation environment to special users by disallowing other general-purpose vehicles. One famous example is the bus lane reservation that is employed to free buses from traffic jams during rush hours such that the bus transit efficiency and schedule adherence is improved, thereby enhancing the attractiveness of bus transit system. Another example is that high-occupancy vehicle lanes have been used to encourage commuters to carpool and save their travel time. Besides, the LR strategy has also been applied to large-scale sport events, e.g., Olympic Games in Sydney, Athens and Beijing.

However, reserving lanes in an existing transportation network has negative impact on normal traffic because the available lanes for the general-purpose vehicles are reduced. This may further worsen already congested city's traffic. For example, travel times on the non-reserved lanes may be increased. It has shown in [96] that the travel time of general-purpose vehicles on A1 motorway in Paris was increased up to about 26% after one of three lanes was reserved. Therefore, it is necessary to decide appropriate lanes to be reserved from the transportation network to minimize the negative traffic impact. Such a transportation planning problem is called lane reservation problem. To the best of our knowledge, only a few studies in the literature have addressed LRPs, and the proposed approaches fail to solve large-size problems within reasonable times. Moreover, optimal lane reservation for improving bus transit service while simultaneously minimizing the impact of reserved lanes have not been addressed in the literature. Remarkably, LRPs differ from the existing classical transportation planning problems, such as vehicle routing problem (VRP), facility location problem (FLP), minimum-cost multi-commodity flow problem (MCMCFP), and their several special cases are classical combinatorial optimization problems. Consequently, it is of both theoretical and practical significance to further investigate such a new class of transportation planning problems. This thesis consists of developing new and efficient methodologies for the above two categories of lane reservation problems for meeting special transportation needs and improving the performance of bus transit systems, respectively.

1.2 Content and contribution

In this thesis, we mainly investigate optimal lane reservation from a macroscopic transportation network point of view to minimize the negative traffic impact due to reserved lanes. We firstly devote our attention to addressing large-size and robust lane reservation in the context of automated truck freight transportation and large-scale special events. Then, we study three new LRPs for bus transit in different contexts, which are the bus lane reservation problem (BLRP) where both bus lines and paths are predetermined, the BLRP where the bus lines are given but the paths need to be determined, and the BLRP where both bus lines and paths are not known. For each studied problem, appropriate model is formulated, the complexity is analyzed, and solution method is developed according to the characteristics of the problem.

The main contribution brought by this thesis is summarized as follows:

1) Large-size and robust lane reservation problems for automated truck freight transportation and large-scale special events are investigated. Improved mathematical models are formulated, some problem properties are derived and then two-phase exact method and exact ε-constraint method and cut-and-solve combined method are developed based on the characteristics and properties of the problems. Computational results on benchmarks and randomly generated larger-size instances indicate that the proposed algorithms outperform the state-or-the-art algorithms.

2) Study three new bus lane reservation problems in different application contexts (predetermined bus lines and paths, the bus lines are given but the paths need to be determined, and both bus lines and paths are not known). These three problems are investigated successively and the assumptions on bus lines and paths are relaxed step by step. For these problems, appropriate mathematical models are formulated, their complexities are shown and then optimal cut-and-solve method, enhanced cut-and-solve method and kernel search based heuristic are developed according to the characteristics of the problems. The results from numerical experiments on extensive randomly generated instances show the effectiveness and efficiency of the proposed algorithms.

The remainder of this thesis is organized as follows:

In Chapter 2, a review on lane reservation applications is first addressed. Then, we review lane reservation problems and their state-of-the-arts, including the lane reservation studies at microscope level and lane reservation at macroscopic network level. Then, optimization methods for combinatorial optimization problems, including single-objective and multi-objective ones, are reviewed, and the principles of the key techniques to be applied to solving the studied problems are introduced.

Chapter 3 addresses large-size lane reservation for automated truck freight transportation. We first describe the background of the problem. Then, an improved integer linear program is presented for the problem, and some properties of the model are derived, based on which a fast two-phase exact algorithm is developed for solving large-size problems. Finally, computational results on benchmark instances and larger-size instances are reported to evaluate the proposed method.

Chapter 4 investigates robust lane reservation for large-scale special events considering the uncertain traffic features. The background of the problem is first presented. Then, a bi-objective mixed-integer program is presented for the problem. An exact improved ε-constraint and cut-and-solve combined method is developed to obtain the Pareto front of the problem. Computational results on an instance based on a real network topology and randomly generated instances are finally reported.

Chapter 5 studies a bus lane reservation problem in which both the bus lines and paths are predetermined. Its background is first given and then the problem is formulated as an integer linear program. The complexity of the problem is proved. An optimal cut-and-solve method is proposed to solve the model. Finally, computational results on randomly generated instances are presented to evaluate its performance.

Chapter 6 investigates a bus lane reservation problem in which the bus lines and their stations are given but the paths need to be determined. Compared with the problem addressed in Chapter 6, the assumption that the bus paths are predetermined is relaxed. The problem is first formulated as a mixed-integer linear program and an integer linear program, respectively. The properties of the problem are explored. Then, an exact enhanced cut-and-solve algorithm and an improved kernel search based heuristic are developed for the problem, respectively. At last, computational results are reported to evaluate the performance of the proposed methods.

Chapter 7 addresses a bus lane reservation problem in which both bus lines and paths need to be determined, which further generalizes the problem studied in Chapter 6 by relaxing the assumption that bus lines are predetermined. For the problem, we first develop a bi-objective mixed-integer linear program. Several invalid inequalities are added to reduce the search space. Then, the ε-constraint method is proposed to derive Pareto optimal solutions. Computational results on a benchmark instance and randomly generated instances are finally reported.

Chapter 8 concludes this thesis and discusses some future research directions.

Chapter 2

Literature review

In this chapter, we first review the applications of lane reservation. Then, the theoretical studies on lane reservation, including lane reservation at microscopic level as well as at macroscopic network level are described, respectively. Then, we review optimization methods for single- and multi-objective combinatorial optimization problems, respectively, where the principles of the optimization methods for the studied LRPs are introduced.

2.1 Review on lane reservation applications

With the dramatic and ongoing increase of vehicles on the roadways, traffic congestion has become a common phenomenon in many cities around the world. As stated in the previous chapter, compared with the conventional and direct strategy to expand the capacity of transportation network via building road lanes or segments, appropriate traffic management strategies play a more important role. In recent years, lane reservation, as a traffic management strategy, has been widely applied in real life, including bus lane, high-occupancy vehicle (HOV) lane, and temporary reserved lanes for large-scale special events (e.g., Olympic Games).

As is widely acknowledged, the bus lane reservation, as an important bus priority strategy, has been widely implemented in many countries. It is to convert some general-purpose (GP) lanes into exclusive bus lanes in some time periods, such as morning and afternoon peak hours (see two examples in Fig. 2.1). Its objective is to improve the performance of bus transit so as to enhance bus transit attractiveness and promote passengers to shift from private cars to buses, thereby alleviating urban traffic congestion [106]. The first bus lane in the world can date back to 1940 in Chicago, American and the first European bus lane was later constructed in 1963 in Hamburg, German. After that, other European countries such as France and England

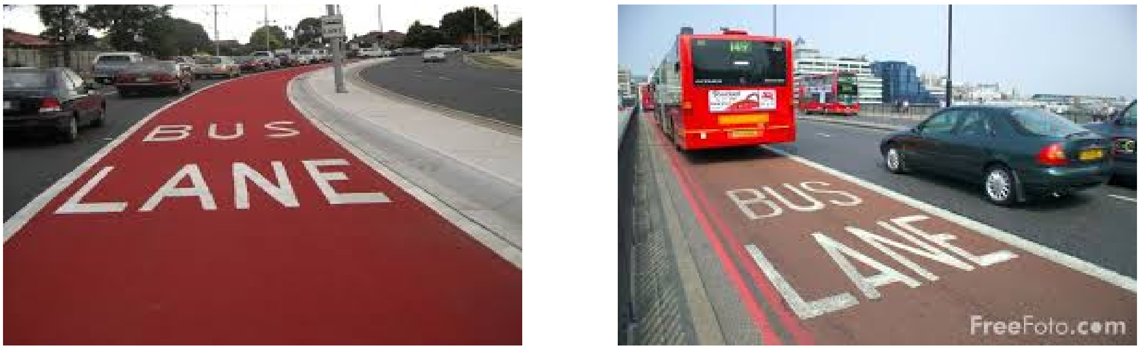


Fig. 2.1: Examples of bus lanes

also began to establish bus lanes. Later, Asian countries also implemented bus lanes to promote bus priority. The first bus lane was established on Chang'an Avenue in Beijing, China, in 1970. Nowadays, promoting bus priority via implementing bus lanes is becoming more and more popular around the globe. Only in China, many cities, e.g., Beijing, Xi'an, Kunming, Guangzhou, Chengdu, etc, have implemented bus lane reservation strategies. Obviously, bus lanes can free buses from trapping into traffic jams to achieve rapid and on-time bus transit service. Thus, many bus rapid transit (BRT) systems based on bus lanes have been developed in many cities throughout North America, Latin America, Europe, Asia during the past three decades. One of the most successful examples is Bogotá BRT system in Colombia, which is delivering over 198,000 passengers per hour during rush period and with which the average public transit speed is increased from 15 km/h to 27 km/h [119]. Hidalgo and Gutiérrez [58] reported that 120 BRT systems are implemented around the globe that cover more than 4,300 km bus lanes were serving 28,000,000 passengers each day.

Another well-known application of lane reservation strategy is the HOV lane, which is mainly reserved for the exclusive use of HOV vehicles during peak periods [1] to save their travel time. A HOV is a vehicle containing multiple occupants. The representative examples include carpools, vanpools, and buses. Like bus lanes, HOV lanes are usually marked with special signs to differentiate them from general-purpose lanes (see two examples in Fig. 2.2). By traveling on HOV lanes, HOVs can rapidly pass congested urban areas during rush hours. Consequently, travelers, especially commuters, are encouraged and drew to carpool. As stated in [111], HOV lanes mainly target at increasing the average number of persons per vehicle. The first HOV lane in the world can date back to the late 1960s in North America [112], [81] and developed rapidly during the 1980s to 1990s. HOV lanes were widely installed in Los Angeles, New York, San Francisco, Seattle, Washington DC. It was reported

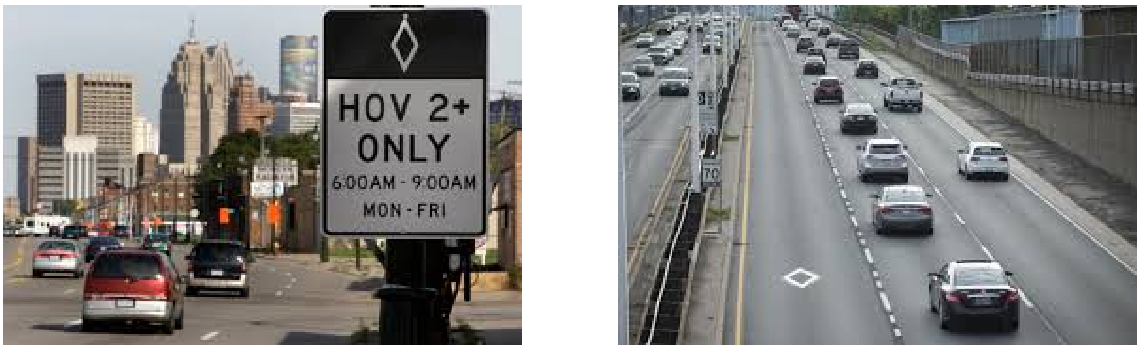


Fig. 2.2: Examples of high-occupancy vehicle (HOV) lanes

in [113] that about 2,300 miles of HOV lanes were installed in 28 metropolitan regions in the USA in 2000. Nowadays, HOV lanes have been broadly implemented not only in more than 40 cities of North America, but also in many cities around the world, such as UK, Spain, Australia, and the Netherlands [82] [99]. Since HOV lanes can encourage travelers to carpool to avoid traffic jams during rush hours, they contribute to carrying more people by fewer vehicles. As a consequence, the HOV lane strategy has been considered as an effective traffic management means of improving the road use efficiency, alleviating traffic congestion, reducing exhaust emission and saving energy. More information on the applications of HOV lanes can be found in [41].

Besides the above applications, a lane reservation strategy has also been applied to large-scale special events, such as large sport events like Olympic Games and world exposition. These special large-scale special events are held frequently in many large cites nowadays. For example, approximately 70 large sport events were held in 2009 [137]. Such events usually require organizers to ship certain people and materials from athlete villages to dispersed stadiums in a fast, safe, and reliable way. For example, the organizers of the Guangzhou Asian Games in 2010 were required to deliver athletes to dispersed stadiums within 30 min [125]. To meet these special transportation requirements, the organizers resorted to temporarily reserving lanes on certain road segments for the exclusive use of their participants. In real life, the strategy of reserving lanes for Olympic buses was adopted during the Sydney Olympic Games in 2000 [17], the Athens Olympic Games in 2004 [134], and the Beijing Olympic Games in 2008 where Olympic exclusive lanes were created and implemented for Olympic buses to deliver athletes to stadiums. Fig. 2.3 gives two examples of Olympic reserved lanes.

Fig. 2.3: Examples of exclusive lanes for Olympic Games

2.2 Review on lane reservation studies

In the previous subsection, we have described various applications of lane reservation strategy in real life, including bus lane, HOV lane and temporarily reserved lanes for large-scale special events. Due to the widespread applications of lane reservation in real life, there have been a range of theoretical studies on lane reservation strategies in the literature, which can be classified into two categories. One focuses on studying lane reservation at a microscopic road level, i.e., a single road link or corridor, and the other concerns optimal lane reservation at a macroscopic network level.

2.2.1 Lane reservation studies at microscopic road level

As stated in section 2.1, to promote bus priority and alleviate traffic congestion, bus lane reservation has been widely implemented in real life. To fully realize the potentials of bus lane, numerous studies on bus lane implementation type, bus lane install condition and form, evaluating the benefit impact of bus lane at a microscopic road level (i.e., a certain road segment or corridor) have been reported by using empirical, analytical, and simulation approaches.

Bus lane usually has two types: permanent bus lane and intermittent one. Because a permanent bus lane monopolizes a lane, on which non-bus vehicles are not allowed to pass in any time, it permanently reduce the traffic capacity of road segment for non-bus vehicles, which may cause great traffic pressure for non-bus traffic. Therefore, considerable researchers suggested implementing intermittent bus lanes to increase the utilization efficiency of road lanes [62], [77], [114], [115], [140]. Jepson and Ferreira [61] also pointed out that to implement bus lane on a certain road segment should take into account multiple factors such as traffic volume and road conditions. Seo et al. [102] established some guidelines to bus lane setting in Seoul. They concluded that bus lanes would be useful only when the condition of certain total traffic volume and bus volume levels is met. In addition to traffic volume and road conditions, Eichler and Daganzo [38] indicated that installing intermittent bus lane on a road segment should additionally consider the bus frequency and volume ratio of buses and non-bus vehicles on this road link. Zhang et al. [135] discussed the install form, size, and standards of bus lane and the corresponding station on a certain road segment. Besides, Black [16] proposed an evaluation model for bus lane reservation on a urban arterial road segment. Gan et al. [51] presented an evaluation and decision model for bus lane installing on a arterial road link.

Bus lanes reflect the priority of bus transit, which improve the travel speed of buses and reduce their travel time. Considerable positive results on bus lanes have been reported. Choi and Choi [27] pointed out that the bus transit time in South Korea was significantly reduced and about 12% car users were shifted to bus after implementing bus lane. Wei and Chong [122] reported that the average speed of buses in Kunming, China was increased up to 58%, from 9.6 to 15.2 km/h with a bus lane reservation strategy. Shalaby [103] found similar results on the effect of bus lane reservation in Toronto, Canada, that bus transit performance and attractiveness were improved. These results have indicated that bus lanes would improve the performance of bus transit, thereby enhancing bus transit attractiveness and increasing bus ridership. On the other hand, a bus lane on a road segment may result in negative traffic impact on its adjacent non-reserved lanes. Arsan and Vedagiri [8], [9] developed microscopic traffic simulation models to investigate the impact of bus lanes on non-bus vehicles under heterogeneous traffic flow. The results indicated that after installing bus lanes, the bus travel speed could significantly increase, but to maintain a minimum level of service for non-bus vehicles on adjacent lanes, the permissible ratio of non-bus traffic volume to capacity could not exceed 0.53 for 11.0m wide road and 0.62 for 14.5m wide road, respectively. Karim [65] evaluated the impact of bus lane on travel time of non-bus vehicles by adopting floating car technique. It was found that after implementing bus lane the average travel time of non-bus vehicles was significantly increased during morning and evening rush hours. Chen et al. [26] studied the impact of weaving sections on the capacity of non-bus traffic on an urban express way due to bus lanes via a microscopic traffic simulation. It was reported that the length and headway of weaving section have three different impact on non-bus traffic under different kinds of bus lane configurations. Yang and Wang [128] also applied a microscopic simulation tool to address the impact of dynamic bus lane and general bus lane in terms of traffic conflicts changes and travel times. The authors reported that both dynamic bus lane and general bus lane generate positive benefits (travel time reduction) on buses and result in negative impact (increase in travel delay and travel conflicts) on adjacent non-bus vehicles; but the former causes less impact on non-bus traffic than the latter.

In addition, there have also been considerable efforts on evaluating the impact of HOV lanes by using empirical and analytical methods. Martin et al. [84] investigated the impact of HOV lane on I-15 highway in Salt Lake through two-year empirical study. It was reported that 1) during the P.M. peak hours, the HOV lane could carry the same number of people as a non-HOV lane with only 44% of vehicles of the latter; 2) the average vehicle occupancy on HOV lane was increased by 17 %; and 3) the HOV lane saved up to 13% and 30% of travel time during the A.M. and P.M. peak periods, respectively. Sullivan and Burris [107] evaluated the two HOV projects: SR-91 in California and QuickRide in Texas in terms of the benefit-costs including travel time savings, emission reduction, fuel cost savings, operation and capital investment by using a comparison analytical method. The authors found that both projects could achieve significant travel time savings and both the benefit-cost ratios were about 1.5 to 1.7. However, there have also been negative results during the application of HOV lanes. Fuhs and Obenberger [50] reported that one HOV lane in New Jersey was closed due to lower utilization in 1998. Kwon and Varaiya [71] evaluated and analyzed the HOV system in California by using empirical data. The author found that most HOV lanes could achieve travel time savings but some were underutilized and had to be closed. Dahlgren [33] also stated that HOV lanes were not always more effective than non-HOV lanes if installing improperly.

2.2.2 Lane reservation studies at macroscopic network level

Although the studies on lane reservation at a microscopic road level provides valuable information to aid decision-makers when implementing lane reservation, the results obtained at the microscopic level cannot precisely guide optimal lane reservation decision at the macroscopic network level. Considering optimal lane reservation from a macroscopic network point of view using optimization techniques has increasingly received attention from researchers and there have been only a few studies on it reported in the literature maybe due to the complexity of the problem. Note that Lane reservation studies at macroscopic network level can also be classified into two categories according to application objectives: improving the performance of bus transit systems and meeting special transportation requirements.

To improve the performance of bus transit system through reasonable bus lane reservation at a macroscopic network level, Mesbah et al. [87] firstly considered optimal bus lane reservation at a network level as a Stackelberg leader-follower game in which traffic managers act as the leader and the system users act as the follower and the problem was formulated as a bi-level programming model. The upper level model aims to minimize the weighted sum of user travel time and vehicle operation cost via optimally selecting bus lanes while simultaneously satisfying budget constraint. The lower level can be viewed as the constraints to the upper level and it consists of three models, which involves modal spilt, traffic and transit assignment, respectively. A benders decomposition based method was proposed to obtain optimal solutions, while only one instance with up to 38 nodes and nine bus lines was tested. Mesbah et al. [86] then proposed a genetic algorithm (GA) for the problem considered in [87] and an instance with 86 nodes and ten bus lines was tested, but the quality of the obtained solution was not evaluated. Recently, Yao et al. [131] incorporated the bus frequency decision into optimal bus lane selection, and developed a bi-level programming model with the objective of minimizing the weighted sum of user travel and transit operating costs. They considered a bi-modal transportation network equilibrium model in the lower level to address the impacts of bus lanes on traffic diversion and mode shift. A GA was proposed for the problem and one instance with 13 node and six bus lines was tested. More recently, Khoo et al. [68] examined an integrated problem of bus lane selection and scheduling with the objectives of simultaneously minimizing the total travel time of cars and of buses, in which traffic diversion impact and mode shift were simulated by microscopic traffic simulation. A non-dominated sorting GA (NSGA II) embedded with a microscopic traffic simulation tool was proposed to obtain Pareto solutions. Its effectiveness was tested by an instance with the network of Malaysia and ten bus lines. In addition, Sun et al. [108] developed a tri-level programming model for optimal bus lane design in given transit network. The impact of bus lanes on traffic diversion was addressed by simultaneously solving the traffic and transit assignment sub-models. A GA and Simulated Annealing (SA) combined algorithm was proposed for the problem and a case with 51 nodes and nine bus lines was tested.

In recent years, some researchers have studied optimal lane reservation at a transportation network level to meet special and time-efficient transportation requirements for future automated truck freight transportation and large-scale special events. As stated in the previous subsection, implementing lane reservation on a road segment may cause negative impact, such as an increase of travel times, on the adjacent non-reserved lanes. The optimization objective focuses on minimizing the negative traffic impact of reserved lanes. Such a new kind of transportation planning problem is called lane reservation problem. Wu et al. [125] first developed an integer linear programming model to study an LRP motivated by completing the time-guaranteed transportation tasks arising from the 2010 Asian Games in Guangzhou, China. The authors developed a simple heuristic to obtain near-optimal solutions for problems with up to 22 nodes and 22 tasks. The work [24], [124], [126] proposed efficient meta-heuristics to obtain better solutions for the LRP in [125]. Later, Fang et al. [45] extended the LRP studied in [125] to a capacitated LRP by taking into account the residual capacity constraint on general-purpose lanes. They proposed a cut-and-solve (CS) method to exactly solve the problem instances with up to 120 nodes and 30 tasks. Fang et al. [44] further improved their proposed CS method by introducing a cutting plane technique to solve the capacitated LRP more efficiently. Larger-size problem instances with up to 120 nodes and 40 tasks were solved. Recently, Fang et al. [42] studied an LRP motivated by performing safe and time-efficient automated truck freight transportation and proposed CS based exact method as well. Computational results on instances with up to 150 nodes and 30 tasks were reported. Besides above studies, Zhou et al. [138], [139] examined the LRP for hazardous material transportation and considered a bi-objective LRP with the two objectives of minimizing the negative traffic impact due to lane reservation and the transport risk. An ε-constraint and fuzzy logic combined method was developed to recommend Pareto optimal solutions. More recently, different from the above LRPs considering constant link travel times, Fang et al. [43] studied a dynamic LRP considering dynamic travel times and an improved cut-and-solve method was presented.

By analyzing and summarizing the literature described above, the state-of-the-arts on lane reservation studies can be concluded as follows.

Firstly, most of the studies are mainly concentrated on the types, install conditions and forms, and benefits and impact of lane reservation at a microscopic road level. As stated in section 2.2.2, the obtained results at microscopic road level can provide valuable information for decision-markers when reserving lanes, but they cannot guide an optimal lane reservation decision at a macroscopic network level.

Secondly, there have been a few contributions studying optimal bus lane reservation from an existing macroscopic bus transit network to improve bus transit service [68], [76], [86], [87], [108], [131]. However, all these studies assume that the bus paths are predetermined such that the bus lanes are only selected from the given bus paths. It is understandable that the proposed theories and methods cannot be directly applied to a bus lane reservation problem in which the bus paths need to be optimally determined. In the literature, the optimization objectives mainly focus on minimizing the total users travel time or/and vehicle operating costs, while the negative impact due to reserved lanes was not considered. Guaranteeing bus arrival times at stations is one of the most important factors to evaluate bus transit service level [109], but it is ignored in the existing studies. Furthermore, the methods proposed in the existing work only solved problem instances with relatively small-size network and few bus lines. The algorithms in existing studies were tested by one or two instances and most studies adopted GA to solve their problems but the quality of the obtained solutions was not evaluated with optimal ones.

Thirdly, a few studies have considered optimally reserving lanes from an existing network to meet special transportation needs arising in future automated truck freight transportation and large-scale special events, with the objective of minimizing the negative impact of reserved lanes [24], [42]–[45], [124]–[126], [138], [139]. However, we can find that: 1) the proposed methods usually fail to solve large-size problem instances with reasonable computational times due to the NP-hard nature of the problem; 2) most of the studies assume that the road travel time is constant and does not consider its possible changes due to uncertain traffic features, such as dynamic traffic flow, traffic accidents, and fault of task vehicles; and 3) no studies consider optimally reserving lanes at a network level to minimize their negative impact in the context of bus transit.

To address these issues mentioned above, this thesis first investigates large-size lane reservation for automated truck freight transportation (Chapter 3) and robust lane reservation for large-scale special events (Chapter 4). For the two problems, efficient resolution methods are developed and their performance are evaluated on benchmark instances and large-size newly generated instances. Then, we investigate several new bus lane reservation problem at macroscopic network level minimizing negative impact of reserved lanes and guaranteeing bus arrival times at stations for different application contexts: 1) both bus stations and paths are known (Chapter 5); 2) the bus stations are given but paths need to determined (Chapter 6); and 6) both stations and paths are must be determined along with lane reservation (Chapter 7). For these new bus lane reservation problems, we propose new mathematical models, then efficient resolutions algorithms are developed according the characteristics of each problem. The proposed methods are evaluated by extensive numerical experiments. As all these studied problems are combinatorial optimization problems. In the following sections, we will review combinatorial optimization methods, including single- and multi-objective ones, where the principles of the optimization methods for the studied problems are also detailed.

2.3 Single-objective combinatorial optimization

In this section, we will briefly review the widely used methods for solving single-objective combinatorial optimization problems (generally NP-hard). These methods can be classified into three categories: exact methods, heuristic methods, and metaheuristic methods. Without loss of generality, the combinatorial optimization problems are referred to minimization problems if without special mention.

2.3.1 Exact method

The remarkable advantage of exact methods is that they can guarantee to find an optimal solution. However, the computational time usually exponentially increases with the size of the problem for NP-hard problems. In the following, we will first review the two widely applied exact methods for single-objective combinatorial optimization problems and then an exact method for the LRPs is introduced.

2.3.1.1 Dynamic programming

Dynamic programming introduced by Bellman [14] in 1956 is a commonly used exact method for a class of problems with the particular property that an optimal solution can be computed from optimal solutions of its subproblems. Specifically, the main idea is to solve a complex problem by breaking it down into a series of subproblems which have the same structure with the original problem, and such decomposition is to be made recursively. Applications of dynamic programming methods can be found in [13], [25], [28].

2.3.1.2 Branch-and-Bound

Branch-and-bound (B&B) is an enumerative technique [72]. It is usually used to solve problems, for which there exist no special ways to avoid their large sized search spaces. The B&B aims to find the best way to effectively organize an enumeration of the solutions by exploring as much as possible intrinsic properties of the studied problem, such as efficient lower and upper bounds, powerful dominance rules, such that the dominated parts of the solution space can be intelligently eliminated and only a small and reasonable number of solutions are enumerated, those that are not explored are dominated.

The procedure of branch-and-bound (B&B) has been popularized by Land and Doig [72] for general linear programming problems, by Little et al. [78] for the traveling salesman problem, and by Ignall and Schrage [60] for flowshop scheduling problems. Moreover, the B&B method was investigated by many researchers, see [11], [73], [89]. B&B method is an iterative method (generally called tree search) that contains two basic operations: branching and bounding [89].

1) Branching. The branching is to divide the original solution space (root node) into a set of subspaces (leaf nodes) corresponding to a set of subproblems. Then, by exactly solving all these subproblems that covers the original problem solution space, an optimal solution of the original problem can be obtained. The branching is repetitively applied to each parent node to generate a new set of children nodes until an optimal solution is obtained. Thus, the sets of nodes (subproblems) construct a hierarchical tree.

2) Bounding. For a node of the search tree, the bounding is to determine if it potentially contains an optimal solution by the obtained lower and upper bounds, dominance rules. If yes, the node will be maintained and be branched in its turn to find a new leaf (a feasible solution) or a new lower bound. If it proved that this node does not contain an optimal solution, then the node will be pruned.

In the literature, B&B methods have been widely applied to solving combinatorial optimization problems, see [3], [12], [23], [80].

2.3.1.3 Cut-and-solve method

For the single-objective combinatorial optimization problems to be addressed in this thesis, an exact method, cut-and-solve (CS) method, is considered, which was firstly introduced by Climer and Zhang in 2006 for integer linear programming problem and was demonstrated that it achieved excellent performance for solving the asymmetric traveling salesman problem (ATSP) [30]. CS method is a particular branch-and-bound search strategy, in which only two nodes are branched at each level during its search tree [129]. The two nodes correspond to a *sparse problem* and a *residual problem*, respectively. The former problem is exactly solved because of its small search space. Then the node corresponding to the latter problem need to be branched.

Different from classical branch-and-bound method branching on one variable at each iteration, CS method branches on a set of variables [129]. Because of these above characteristics, CS method enjoys the following advantages: 1) "wrong" choices commonly appeared in classical branch-and-bound method can be avoided in CS method;

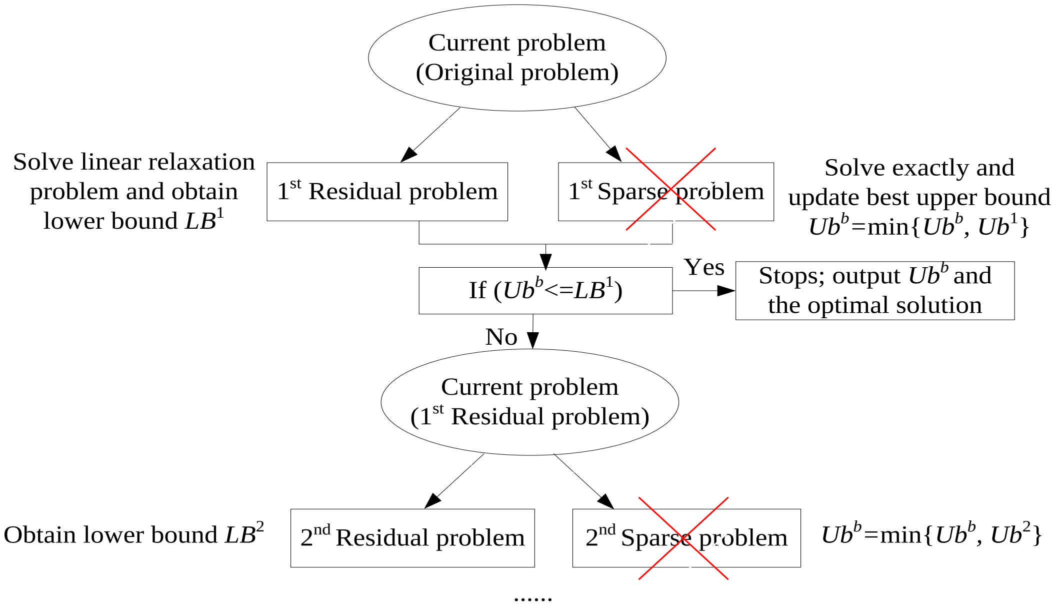


Fig. 2.4: Principle of cut-and-solve method

and 2) compared with classical branch-and-bound method in which considerable memory may be required to store all unexplored nodes in their search tree, the branching strategy of CS method can effectively maintain the tree size and is of much less memory requirement. Recently, it has also been successfully applied to solving other difficult combinatorial optimization problems, such as facility location problem [129] and lane reservation problem [42], [45]. The above remarkable advantages and successful applications of CS method motivate us to apply it to the considered LRPs in this thesis.

The general principle of CS method is depicted in Fig. 2.4. It can be further described as follows. Given an integer linear program (ILP), at the n-th level ($n \geq 1$) of CS algorithm, the branching tree has only two nodes, which correspond to a *Sparse Problem* (SP^n) and a *Residual Problem* (RP^n), respectively. Such separation is achieved using a so-called piercing cut. The SP^n has relatively small solution space such that it can be exactly solved easily. Since SP^n is a subproblem of the original problem, its optimal solution if it exists provides an upper bound of the ILP, denoted by UB^n. After exactly solving SP^n, its search space is cut off from the current solution space. If UB^n is better than the best upper bound UB^b found so far, then UB^b is replaced by UB^n. It is difficult to exactly solve the RP^n since its search space is large, and a linearly relaxed RP^n is exactly solved and a lower bound of RP^n is obtained, denoted by LB^n. If UB^b is less than or equal to LB^n that any optimal solution of RP^n will be greater than or equal to UB^b, then the UB^b is the ILP's global best value and the CS terminates. Otherwise, the RP^n is further separated into SP^{n+1} and RP^{n+1} by the piercing cut, and a new iteration starts. We note that the root node corresponds to the original problem.

To well understand the cut-and-solve method, several key points are further explained as follows:

1) The sparse problem at each iteration corresponds to a subproblem of the original problem, so its optimal objective value is an upper bound of the original problem. The best upper bound is updated when it is improved.

2) After a spare problem is solved, its corresponding solution space will be cut off from the original solution space. Hence, the size of the solution space will be iteratively reduced.

3) Because the solution space of the residual problem is large, it is difficult to solve it to optimality. Hence, only the corresponding linear relaxation problem is solved.

4) If the lower bound of the residual problem is greater than or equal to the best upper bound found so far, then the optimal objective value of the residual problem must be greater than or equal to the found best upper bound. This means that the residual problem has no better solutions than the best solution found. In this case, the iteration terminates and an optimal solution of the original problem is obtained.

The optimality and termination of CS method is guaranteed by the following two theorems owing to [30]. For more details on the proof of the two theorems, please see [30].

Theorem 1 *When the cut-and-solve method terminates, the current incumbent solution is an optimal solution.*

Theorem 2 *If the solution space for the original problem is finite, and both the method for solving the relaxed residual problem and the method for selecting and solving the sparse problem are guaranteed to terminate, then the cut-and-solve method is guaranteed to terminate.*

Piercing cut plays a crucial role for efficiently applying CS method since it drives the branching of CS method at each iteration and it should be specially designed for different optimization problems. The solution space of sparse problem should be

General procedure of cut-and-solve method

```
1: define current problem $CP_0$ as original problem and solve its linear relaxation problem. Let $n = 0$.
2: Define set $U^n = \{\text{variables with reduced costs} > \alpha^n\}$;
3: define sparse problem $SP^n$ as $CP_n$ with constraint (sum of variables in $U^n = 0$) and residual problem $RP^n$ as $CP_n$ with constraint (sum of variables in $U^n \leq 1$)
4: solve $SP^n$ exactly, and obtain its optimal solution and objective value $UB^n$. Update $UB^b$ if $UB^n$ is less than $UB^b$.
5: solve the linear relaxation problem of $RP^n$ and obtain its lower bound $LB^n$.
6: if $LB^n \geq UB^b$, return $UB^b$ and end; Otherwise, define the current problem $CP_{n+1}$ as $RP^n$, let $n = n + 1$ and goto step 2.
```

Fig. 2.5: General procedure of cut-and-solve method

small enough for easy resolution, and it should be also large enough such that it contains at least a feasible solution of the original problem; otherwise, the best upper bound cannot be updated. In fact, the efficiency of CS method is highly dependent on selecting appropriate piercing cuts. Some desirable properties of piercing cuts are suggested by Climer and Zhang [30], listed as follows:

1) The piercing cut should be able to remove the optimal solutions of the relaxed residual problem so as to prevent them from being found in next iterations.

2) The subspace cut off by the piercing cut from the solution space of the relaxed residual problem should be small enough, so that the resulted sparse problem can be relatively easily solved exactly.

3) The piercing cut should attempt to explore the "promising" solution subspace involving optimal solutions of the original problem, because the CS method will not be terminated until an optimal solution of the original problem has been found in the sparse problem.

4) The subspace cut off by the piercing cut should contain at least one feasible solution of the original problem to guarantee termination.

In [30], Climer and Zhang defined a *variable set* that is composed of the decision variables whose *reduced cost* values are greater than a given value *alpha*. Note that the reduced cost values can be derived from the resolution of the linear relaxed residual problem (each variable has a reduced cost value). The piercing cut is then defined as the sum of the decision variables being greater than or equal to one. Then a general procedure of cut-and-solve method is presented in Fig. 2.5. In this thesis, we make a contribution of developing improved cut-and-solve methods by introducing new methods of generating piercing cuts and acceleration technique to more efficiently solve the studied problems.

2.3.2 Heuristic method

A heuristic is an algorithm that quickly provides a feasible solution for a NP-hard optimization problem. Heuristics are experience-based methods, which are normally designed based on their characteristics. They can produce "good" solutions quickly, but the solutions are not guaranteed to be optimal. Heuristics can be embedded into exact methods like branch-and-bound algorithms for the optimum convergence acceleration by providing better bounds as well as metaheuristics by providing "good" initial solutions for their performance improvement. Well-known heuristics include greedy heuristic, Lagrangian based heuristic, etc. Examples of heuristics for combinatorial optimization problems can be found in [4], [79], [125]. In the following, a heuristic, kernel search (KS) method, that will be used for the resolution of the BLRP-PD is described.

2.3.2.1 Kernel search method

Kernel search method is an iterative heuristic firstly introduced by Angelelli et al. [5] in 2010 for solving ILPs, such as the multi-dimensional knapsack problem (MDKP) [5] and the portfolio selection problem (PSP) [6]. Its core idea is to identify subsets of variables and exactly solve a sequence of subproblems restricted to these subsets. Recently, the KS method has been adapted to the solution of index tracking problem (IRP), the capacitated facility location problem (CFLP) and the single-source capacitated facility location problem (SSCFLP) by [54]–[56]. These exciting results motivate us to apply KS method to solve the studied BLRP-PD. In what follows, the principle of KS method is first described, then its general procedure is presented.

For the KS method, a *kernel* is composed of a set of *promising* variables, in which each variable is likely to take a positive value in an optimal solution. An ILP restricted to a subset of variables is referred to as a *restricted* ILP. Such restriction is equivalent to setting the values of the other variables to be 0. The KS method optimally solves a sequence of restricted ILPs to obtain a near-optimal solution of the original ILP.

General procedure of kernel search method

```
solve the linear relaxation problem of the original problem.
define the kernel $K_1$ and the sequence of buckets $B_l, l = 1, 2, ..., m$.
solve the first ILP restricted to the initial kernel $K_1$ exactly. Let $l = 1$ and set parameter $\overline{m}$ ($\overline{m} \leq m$).
while $l \leq \overline{m}$ do
    update the current kernel $K_l$;
    solve ILP($K_l \bigcup B_l$);
end while
output the best feasible solution and its objective value.
```

Fig. 2.6: General procedure of kernel search method

At the first iteration of the KS method, the linear relaxation of the original problem is optimally solved, and the variables are sorted by a predefined criterion based on the relaxed solution information, such as the values of variables or/and their reduced costs. For example, the variables with positive values can be sorted in non-increasing order of their values and then for null variables (i.e., taking the value of 0) in non-decreasing order of their reduced costs. Such order aims to sort the variables in non-decreasing probability in an optimal solution of the original problem. Subsequently, the initial kernel, denoted by K_1, is built by selecting the first C (a given parameter) variables from the ordered set and the remaining variables are divided into m (a given parameter) ordered groups referred as to *bucket*s, denoted by $\{B_l\}, l = 1, ..., m$. The buckets may have different lengths. Finally, the first restricted ILP restricted to K_1 is optimally solved.

The remaining iterations of the KS method is devoted to sequentially solve $\overline{m}$ ($\overline{m} \leq m$) restricted ILPs, denoted by ILP($K_l \bigcup B_l$), restricted to $K_l \bigcup B_l, l = 1, ..., \overline{m}$, where $K_l \bigcup B_l$ is composed of variables in kernel K_l and bucket B_l. Since the search space of restricted problems is relatively small, they can be exactly solved. For $l \geq 2$, the kernel K_l is updated as follows:

$$K_l = K_{l-1} \backslash K_{l-1}^{-} \bigcup B_{l-1}^{+}, l \geq 2 \tag{2.1}$$

where $K_{l-1}^{-} \subseteq K_{l-1}$ contains variables in K_{l-1} have not been selected in the optimal solution of ILP($K_{l-1} \bigcup B_{l-1}$) as well as in h of previous iterations since they have been added to the kernel, where h is a given parameter. These excluded variables are considered no longer promising. $B_{l-1}^{+} \subseteq B_{l-1}$ contains variables in B_{l-1} taking positive values in the optimal solution of ILP($K_{l-1} \bigcup B_{l-1}$).

The solution of any restricted ILP provides an upper bound of the original ILP and the best one of these ILPs is output as the solution obtained by the KS method. The KS stops when the $\overline{m}+1$ iterations are conducted. The general procedure of KS method can be depicted in Fig. 2.6. In this thesis, we develop an improved kernel search based heuristic for the BLRP-PD to be studied in Chapter 6 according to the characteristic of the problem.

2.3.3 Metaheuristic method

A metaheuristic [92] is a special type of heuristics. Unlike traditional heuristics that are often designed for specific problems, metaheuristics normally need relatively little information on the structures of the considered problems and can explore larger solution space so as to obtain better solutions. Special mechanisms are designed in them to escape a local optimum and increase the probability of finding optimal solutions. In general, traditional heuristics are first employed to generate initial solutions. Then, the initial solutions are iteratively improved according to certain rules. Their termination often relies on a maximum number of iterations or a given computational time. Although good performance can often be achieved by metaheuristics, the optimality of the obtained solutions are also not guaranteed. There have been numerous metaheuristics in the literature, which can be classified into two categories. The first one is based on the exploration of neighbourhoods, including simulated annealing (SA) [70], tabu search (TS) [53], greedy randomized adaptive search procedure (GRASP) [47], variable neighbourhood search (VNS) [90], etc. The second one involves population based metaheuristics, including genetic algorithm (GA) [59], ant colony algorithm (ACO) [36], particle swarm optimization (PSO) [67], etc. Their applications can be found in [48], [52], [74], [93], [94], [116], [120].

2.4 Multi-objective combinatorial optimization

Without loss of generality, a general multi-objective combinatorial optimization problem (MCOP) can be formulated as

$$\min f(x) = \{f_1(x), f_2(x), \cdots, f_n(x)\}, \quad \text{s.t.} \quad x \in X \tag{2.2}$$

where X is the set of feasible solutions (also called solution space), and $x \in X$ is the vector of decision variables. $f(x)$ is the objective vector, and $Y = f(x)|x \in X$ is called the objective space.

An MCOP requires the simultaneous optimization of several objectives. Moreover, these objectives are usually conflicting. Due to the competing nature of the objectives, an optimal solution that simultaneously optimizes all the objectives may not exist. This is the main difficulty in solving MCOPs. A decision maker usually chooses a most preferred solution from a set of reference solutions that make a good trade-off among various objectives. The reference solutions are called Pareto optimal solutions. The following definitions are owing to [88].

Definition 1 (*Domination Relation*) *For any two solutions x_1 and $x_2 \in X$, x_1 dominates x_2 if and only if $f_i(x_1) \le f_i(x_2), i = 1, 2, ..., n$, where at least one inequality is strict.*

Definition 2 (*Weakly Pareto Optimality*) *A solution x^* is weakly Pareto optimal if and only if no $x \in X$ exists such that $f_i(x) < f_i(x^*), i = 1, 2, \cdots, n$, and $f(x^*)$ is called a weakly Pareto optimal objective vector or a weakly non-dominated solution (point) in the objective space.*

Definition 3 (*Pareto Optimality*) *A solution x^* is Pareto optimal if and only if no $x \in X$ exists such that $f_i(x) \le f_i(x^*), i = 1, 2, \cdots, n$ with at least one inequality being strict, and $f(x^*)$ is called a Pareto optimal objective vector or a non-dominated solution (point) in the objective space.*

The concept of Pareto optimality for MCOPs replaces the optimal solutions in a single-objective optimization problem. The set of all Pareto optimal solutions is called the Pareto optimal set. A Pareto optimal solution corresponds to a *nondominated point* in the objective space [88]. The set of all non-dominated points is called the non-dominated set. The image formed by all non-dominated points in the objective space is referred to as the $Pareto\ front$. Two particular points, namely, Ideal and Nadir points, define the lower and upper bounds on objective values over the Pareto optimal set, respectively. A bi-objective combinatorial optimization problem (BCOP) is a subclass of MCOP, where only two objectives are considered. Ideal and Nadir points of BCOPs are defined as follows [15]. Fig. 2.7 describes such two special points for a BCOP.

Definition 4 (*Ideal point*) *The vector $f^I = (f_1^I, f_2^I)$ with $f_1^I = \min f_1(x), s.t. x \in X$, and $f_2^I = \min f_2(x),\ s.t.\ x \in X$, represents the $Ideal\ point$.*

Definition 5 (*Nadir point*) *The vector $f^N = (f_1^N, f_2^N)$ with $f_1^N = \min f_1(x), s.t. f_2(x) = f_2^I, x \in X$, and $f_2^N = \min f_2(x),\ s.t.\ f_1(x) = f_1^I, x \in X$, represents the Nadir point.*

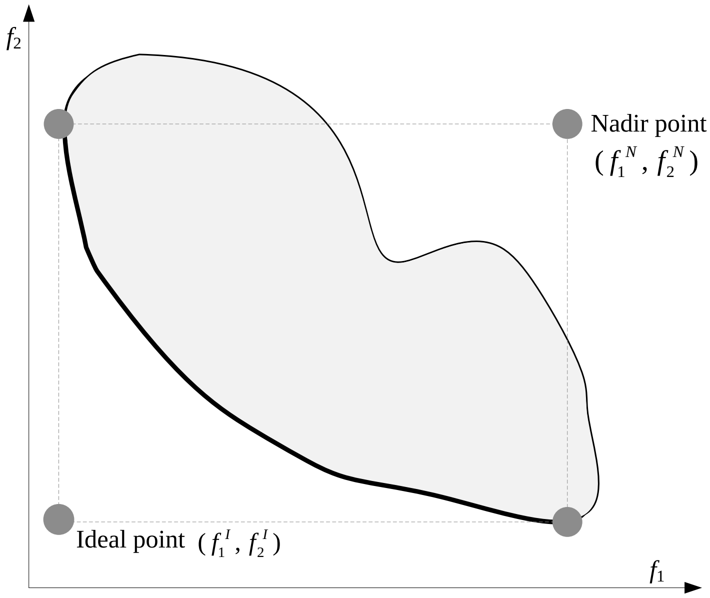


Fig. 2.7: Ideal and Nadir points of a BCOP

There have been many approaches in the literature to solve MCOPs over the past few decades. They can be mainly classified into two categories based on the strategies for handling objectives: Pareto-based evolutionary algorithm and scalarization method. The former one introduces Pareto-based ranking schemes into evolutionary algorithms such as GA. The individuals in evolution population are usually classified based on their domination relationships. The core idea of the latter is to transform an MCOP to be a single-objective optimization one.

2.4.1 Pareto-based evolutionary algorithm

Evolutionary algorithms are popular approaches to quickly generate Pareto solutions of MCOPs. Currently, most multi-objective evolutionary optimization algorithms use Pareto-based ranking schemes to classify the individuals in the evolution population and a mechanism to assign suitable fitness to promote the individual dispersion in the population. Two representative Pareto-based evolutionary algorithms: non-dominated sorting genetic algorithm-II (NSGA-II) [34] and strength Pareto evolutionary algorithm 2 (SPEA-2) [141] have been widely used to solve MCOPs. Other Pareto-based evolutionary algorithms include multi-objective particle swarm optimization algorithm [98] and multi-objective differential evolutionary algorithm [2].

The main advantage of Pareto-based evolutionary algorithms is that multiple solutions can be generated at each iteration, allowing to quickly compute an approximation of the Pareto front. However, although their solution efficiencies are usually very high, their solution effectiveness is highly dependent on the selection of parameters and initial population generation. Another main disadvantage is the fact that their obtained solutions are not guaranteed be Pareto optimal. In fact, it is only known that the obtained solutions cannot be dominated among each other. Applications of Pareto-based evolutionary algorithms for MCOPs can refer to [49], [83], [110], [118].

2.4.2 Scalarization method

Scalarization methods are usually applied to generating the Pareto front (i.e., all nondominated solutions) of an MCOP. In the following, two most popular and widely used scalarization techniques: weighted sum method and ε-constraint method are described, respectively.

2.4.2.1 weighted sum method

The popular and straightforward scalarization method is the weighted sum method, which was first introduced by Zadeh [133]. It aims to convert an MCOP to a single-objective optimization problem by using a linear weighted sum formulation that combines all the objectives, that is

$$\min \sum_{i=1}^{n} \gamma_i f_i(x) \tag{2.3}$$

$$\text{s.t.} \quad x \in X \tag{2.4}$$

The optimal solution of the above single-objective optimization problem would be Pareto optimal if a proper weight vector $(\gamma_1, \gamma_2, ..., \gamma_n)$ is set. The converted single objective is an aggregation of all objectives of MCOP via a linear weighted sum. Hence, this method is not appropriate if not all objectives can be reasonably represented by the linear combination. Moreover, the weighted sum method is ill-suited for an MCOP with nonconvex objective space [37]. Furthermore, considerable redundant runs may be caused by inappropriate weight settings.

2.4.2.2 ε-constraint method

Another well-known scalarization technique is the ε-constraint method, which was first proposed by Haimes et al. [57]. The method aims to optimize only the primary objective, called the most preferred one, while restricting the others by allowable values ε's, called ε-constraints. Suppose that the objective m in (2.2) is selected as the most preferred one, with ε-constraint method the MCOP can be transformed into the single-objective one as follows:

$$\begin{aligned} &\min f_m(x) \\ \text{s.t.}\quad & f_i(x) \leq \varepsilon_i, i \in \{1, ..., n\}\backslash\{m\}, \\ & x \in X \end{aligned} \tag{2.5}$$

where the value of ε_i is limited by the ideal and nadir points. For problem $P(\varepsilon)$, the following theorem holds owning to [88].

Theorem 3 *If a vector $\varepsilon = (\varepsilon_1, \varepsilon_2, ..., \varepsilon_{m-1}, \varepsilon_{m+1}, ..., \varepsilon_n \in \mathbb{R}^{n-1})$ exists such that x^* is an optimal solution of problem $P(\varepsilon)$, then x^* is (at least weakly) Pareto optimal.*

By systematically modifying the ε vector, a set of Pareto optimal solutions are obtained by exactly solving a sequence of ε-constraint problems. In theory, the Pareto front can be derived if the ε vector is mannered in an appropriate way [117]. Compared with the weighted sum method, the ε-constraint method can avoid the difficulties of setting appropriate weights and solve MCOP with nonconvex objective space.

In this thesis, two of the studied LRPs are BCOPs, and we are intended to exactly solve them (i.e., finding the Pareto front) and the ε-constraint method is considered. For a BCOP, without loss of generality, assume that the first objective is selected as the primary objective, then it can be transformed into the following single-objective one.

$$\begin{aligned} P(\varepsilon):\quad & \min f_1(x) \\ \text{s.t.}\quad & f_2(x) \leq \varepsilon, \\ & x \in X \end{aligned} \tag{2.6}$$

where the value of ε is bounded by $[f_2^I, f_2^N]$ obtained by computing the ideal and nadir points (see Definitions 4 and 5). Pareto optimal solutions can be always found by optimally solving ε-constraint problems $P(\varepsilon)$'s with given ε values [15]. The Pareto front can be obtained by solving ε-constraint problems as long as we know how to systematically modify ε to find at least one solution for each non-dominated point of the Pareto front [117].

For BCOPs, a traditional method of modifying ε is to uniformly divide the interval of ε into a number of subintervals and take each interval limit value as the value of ε. This method is referred to as the equidistant ε-constraint method [139]. Due to its simplicity of implementation for BCOPs, it has been applied to solve many BCOPs.

Exact ε-constraint method for BCOPs with integer objective values

```
Compute $f^I = (f_1^I, f_2^I)$ and $f^N = (f_1^N, f_2^N)$
Set $Y_N' = \{(f_1^I, f_2^N)\}, \varepsilon_j = f_2^N - \delta$ and $\delta = 1$, respectively. Let $j = 2$.
while ($\varepsilon_j \geq f_2^I$) do
    Solve problem $P(\varepsilon_j)$ exactly and obtain an optimal solution $x^*$. Add the corresponding objective vector of the optimal solution $x^*$: $f(\varepsilon_j) = (f_1(\varepsilon_j), f_2(\varepsilon_j))$ to $Y_N'$.
    $\varepsilon_{j+1} = f_2(\varepsilon_j) - \delta$ and $j = j + 1$.
end while
Remove dominated points from $Y_N'$ to obtain its Pareto front $Y_N$ if existing.
```

Fig. 2.8: Exact ε-constraint method for BCOPs with integer objective values

However, it cannot guarantee that all the non-dominated points are found. Recently, Bérubé et al. [15] defined a sequence of ε-constraint problems based on a progressive reduction of the values of ε and proposed an exact method to obtain the Pareto front of BCOPs with integer objective values [15]. This method overcomes the drawback of the equidistant ε-constraint method, and we refer to it as the exact ε-constraint method. Its procedure is outlined in Fig. 2.8.

Bérubé et al. first showed the correctness of the exact ε-constraint method to generate the Pareto font for BCOPs with integer objective values and applied it to solving the bi-objective traveling-salesman problem with profits [15]. It solves a sequence of ε-constraint problems through a progressive reduction of the values of ε and then obtains the whole non-dominated set after removing dominated solutions. It has been successfully applied to find the Pareto fronts of some BCOPs, such as the prize-collecting Steiner tree problem [75], vehicle routing problem [97].

Throughout the exact ε-constraint method for BCOPs, the ε value at iteration $j + 1$ is the optimal value of f_2 at iteration j minus parameter δ that is set as 1. This is because the minimum difference of the values of f_2 between any two non-dominated solutions for a bi-objective integer linear program cannot be smaller than 1 due to integer objective values, and thus, no non-dominated points would be lost by this method. In this thesis, we generalize the exact ε-constraint method for the BCOPs with either integer objective values or fractional objective values and further improve the exact ε-constraint method to exactly solve the studied bi-objective LRPs by introducing acceleration techniques according to the characteristics of the studied problems.

2.5 Conclusions

In this chapter, we first review the literature on lane reservation that, as a traffic management strategy, has many applications in real life, such as bus lane, high-occupancy lane and temporary reserved lanes for large-scale special events (e.g., Olympic Games). Then, we review the literature on lane reservation studies. Considerable studies have been conducted on lane reservation, most of which are concentrated on the types, install conditions and forms, and benefits and impact of lane reservation at a microscopic road level via empirical, analytical and simulation approaches. On the other hand, there have been a few studies considering optimally reserving lanes at a macroscopic network level; however, the proposed methods usually fail to solve large-size problems and most of the studies do not consider the uncertain traffic features. Moreover, no studies consider optimal bus lane reservation for improving bus transit system at a network level while minimizing its negative impact. These above issues motivate us to study the following problems in this thesis.

Chapter 3

Large-size lane reservation for automated truck transportation

3.1 Introduction

As stated in Chapter 2, lane reservation, as a flexible traffic management strategy, has been widely applied in real life, and it has been paid attention by both researchers and practitioners. Meanwhile, implementing lane reservation may cause negative impact on normal traffic. Only limited studies in the literature has considered optimally reserving lanes at a transportation network level to minimize the negative impact of reserved lanes. Due to the NP-hard nature of the problem, the existing methods failed to solve large-size problems within acceptable computational time. In this chapter, we study large-size lane reservation for future automated truck freight transportation.

As is widely acknowledged, freight transportation constitutes an important component of the supply chain. It supports economic activities by achieving the efficient movement of raw materials and finished products [31]. As pointed out by [32], transportation is responsible for a significant part of the final cost of products and constitutes an important part of the expenditures of a country. As a consequence, the efficient shipments of cargos has attracted much attention. However, increasing travel demand results in increasingly severe traffic congestion, which causes many problems in freight transportation, such as low transport efficiency, unpredictable transport time, traffic accidents, fuel waste and safety issues. These problems increasingly prevent the freight transportation from being operated in an efficient, reliable and safe fashion [42]. As such, transport managers are urging to appropriate solutions for them. With the development of advanced driver assistance system (ADAS), automated driving is being brought into our daily life [105]. The automated trucks could provide remarkable advantages such as improving transport safety, increasing transportation efficiency, decreasing drivers' stress, and reducing fuel consumption. Therefore, introducing automated driving for trucks would be a promising solution for efficient, reliable, economic and safe freight transportation. One of the key concerns of using automated trucks in freight transportation is the transport safety. Therefore, effective solutions for ensuring safe and time-efficient automated truck transportation is needed, as unlike manually driven trucks, automated ones must have the ability of detecting possible dangers and responding to them correctly and promptly. Providing automated trucks a preferable transportation environment, such as dedicated truck lanes, would be ideal.

It is also predicable that the automated vehicles and manually driven ones will coexist during a long time in the future; however, constructing new transportation network dedicated to future automated trucks is an infeasible way because of the high costs and limited geographic space. Thus, transportation network reconfiguration and to make full use of the existing one is an important opinion. A lane reservation strategy, to convert existing general-purpose lanes in the existing transportation network to dedicated ones (e.g., dedicated truck lanes), provides a good opportunity for safe and time-efficient automated truck transportation. However, because of the exclusive use of the reserved lanes by the automated trucks, the available lanes in the network for general-purpose vehicles are reduced and thus negative traffic impact will be generated on the remaining general-purpose lanes. Therefore, it is necessary to decide appropriate lanes to be reserved so as to satisfy the safe and time-guaranteed automated truck freight transportation, while minimizing the negative traffic impact at the same time.

In this chapter, we study a lane reservation problem for automated truck freight transportation (called LRP in short hereafter), which aims at designing reserved paths for a set of automated truck transportation tasks such that they can be completed within given travel deadlines safely. These reserved lanes ensure the time-guaranteed and safe automated truck transportation but cause negative impact on normal traffic, such as an increase in travel times on adjacent non-reserved lanes. The objective of the LRP is to minimize the negative impact of all reserved lanes. To the best of our knowledge, the LRP has only been addressed by [42]. However, it is found that the method proposed in [42] becomes difficult to solve large-size problems within acceptable computational time due to its NP-hardness. Therefore, more efficient solution approaches are desired to efficiently solve large-size LRP, which is the main focus of this chapter. In comparison with [42], we first present several valid inequalities for the integer linear program proposed by [42]. Experimental comparison results show that these valid inequalities are effective in saving computational time. Moreover, we identify that several special cases of the LRP are classical combinatorial optimization problems. Then, to efficiently solve the LRP, especially for large-size instances, we develop a novel fast two-phase exact algorithm based on the derived properties. Computational experiments based on 120 benchmark ones and 285 newly generated larger-size ones with up to 700 nodes and 55 tasks show that the proposed algorithm is much more efficient as compared with the state-of-art algorithm.

The rest of the chapter is organized as follows. In Section 3.2, problem description is recalled, its improved integer program is provided and its NP-hardness is formally proved. In Section 3.3, we derive several optimal properties for the LRP. Based on the derived properties, a new fast two-phase exact algorithm is developed in Section 3.4. Then, in Section 3.4, numerical experiments on benchmark and newly generated instances are conducted to evaluate the efficiency of the proposed algorithm by comparing with the state-of-art algorithm. Section 3.6 concludes this chapter.

3.2 Problem formulation

The LRP is formulated as an integer linear program (ILP), which is defined on a transportation network that can be represented by a directed graph $G(N, A)$ with a node set N and an arc set A. A node (resp. an arc) represents a road intersection (resp. a road segment). Given a set of automated truck transportation tasks to be accomplished and their corresponding origin-destination (OD) pair, the LRP consists of optimally selecting lanes from the existing network to be reserved and designing a reserved path for each task in order to ensure that it can be completed within its travel deadline safely. However, such lane reservation reduces the available lanes of general-purpose vehicles such that the negative traffic impact, such as the increase of travel time on the remaining general-purpose lanes, may be caused. The objective of the LRP is to minimize the total negative impact caused by all reserved lanes.

As stated in [42], some assumptions are made to facilitate the formulation of the LRP. Firstly, at most one reserved lane is allowed on each road segment. Secondly, there is only one path for each task from its origin to destination in order to ensure the transport safety and the path only consists of reserved lanes. Thirdly, there are at least two lanes on each road segment allowing one reserved lane. We first summarize the parameters and decision variables as follows.

Sets and parameters

N: set of nodes, $i \in N$

A: set of arcs, $(i, j), i, j \in N$
K: set of transportation tasks with $|K|$ tasks, $k \in K$
O: set of origin nodes, $O \subseteq N$
D: set of destination nodes, $D \subseteq N$
o_k: origin node of task $k \in K$, $o_k \in O$
d_k: destination node of task $k \in K$, $d_k \in D$
T_k: prescribed travel duration to complete task $k \in K$
τ_{ij}: travel time on a reserved lane on arc $(i, j) \in A$
C_{ij}: negative impact caused by a reserved lane on arc $(i, j) \in A$

Decision variables
z_{ij} $z_{ij} = 1$, if arc (i, j) is reserved, and 0 otherwise, $(i, j) \in A$
x_{ij}^k $x_{ij}^k = 1$, if the path of task k pass arc (i, j) which is reserved, and 0 otherwise, $(i, j) \in A, k \in K$

Before giving an improved formulation, we first recall the ILP for the LRP provided by [42] shown below.

$$\mathcal{P}_l : \quad \min \sum_{(i,j)\in A} C_{ij} z_{ij} \tag{3.1}$$

$$\text{s.t.} \sum_{(o_k,i)\in A} x_{o_k i}^k = 1, \forall k \in K, \tag{3.2}$$

$$\sum_{(id_k)\in A} x_{i,d_k}^k = 1, \forall k \in K, \tag{3.3}$$

$$\sum_{j:(i,j)\in A} x_{ij}^k = \sum_{j:(j,i)\in A} x_{ji}^k, \forall j \in N \backslash \{o_k, d_k\}, \forall k \in K, \tag{3.4}$$

$$\sum_{(i,j)\in A} x_{ij}^k \tau_{ij} \leq T_k, \forall k \in K, \tag{3.5}$$

$$x_{ij}^k \leq z_{ij}, \forall (i, j) \in A, \forall k \in K, \tag{3.6}$$

$$z_{ij} \in \{0, 1\}, \forall (i, j) \in A, \tag{3.7}$$

$$x_{ij}^k \in \{0, 1\}, \forall (i, j) \in A, \forall k \in K. \tag{3.8}$$

Objective (3.1) is to minimize the total negative impact caused by reserved lanes. Constraints (3.2)-(3.4) guarantee that there exists a feasible path for each OD pair. To be more specific, constraint (3.2) (resp. (3.3)) implies that there exists only one arc outgoing from (resp. coming into) origin node o_k (resp. destination node d_k). Constraint (3.4) ensures the flow conservation for intermediate nodes between origin and destination for each task $k \in K$. Constraint (3.5) indicates that the total travel duration for task k from its origin to destination should not exceed its travel deadline. Constraint (3.6) ensures that task k can pass a reserved lane on arc $(i, j) \in A$ only if this arc is reserved. Constraints (3.7) and (3.8) enforce the bounds of all decision variables.

In the following, we improve this formulation based on the following observations.

Observation 1 *For any task $k \in K$, there will be no arcs entering into (resp. outgoing from) its origin o_k (resp. destination d_k) on its transportation path.*

With Observation 1, we add the following valid inequalities into $\mathcal{P}_l$ without excluding optimal solutions.

$$x^k_{io_k} = 0, \forall (i, o_k) \in A, \forall k \in K \tag{3.9}$$

$$x^k_{d_k i} = 0, \forall (d_k, i) \in A, \forall k \in K \tag{3.10}$$

Obviously, constraints (3.9) and (3.10) reduce the search space of the original problem since part of variables are prefixed.

Observation 2 *For any task $k \in K$, each node in the network will be passed at most once.*

Note that if a node in the network is passed more than once by a task (i.e., cycles exist on the transport path), this obviously generates larger negative impact compared with the case without cycles. With Observation 2, we also add the following constraints into $\mathcal{P}_l$.

$$\sum_{j:(i,j)\in A} x^k_{ij} \leq 1, \forall j \in N \backslash \{o_k, d_k\}, \forall k \in K \tag{3.11}$$

$$\sum_{j:(j,i)\in A} x^k_{ji} \leq 1, \forall j \in N \backslash \{o_k, d_k\}, \forall k \in K \tag{3.12}$$

Constraints (3.11) and (3.12) are also valid inequalities, which tighten the search space of the original problem. With the newly obtained constraints, we derive the following improved program.

$$\begin{aligned} \mathcal{P}'_l: \quad & \min \sum_{(i,j)\in A} C_{ij} z_{ij} \\ \text{s.t.} \quad & \text{Constraints (3.2)-(3.12)} \end{aligned}$$

Remark 1 *The improved formulation employs $\sum_k(|A^-_k|+|A^+_k|)+2|K|(|N|-2)$ more constraints as compared with [42]'s formulation, where $|A^-_k|$(resp. $|A^+_k|$ denotes the number of arcs entering into origin o_k(resp. outgoing from destination d_k).*

Remark 2 *The prefixing of $\sum_k(|A^-_k|+|A^+_k|)$ variables and relatively more constraints help reduce search space. Computational results in Section 5.1 show that $\mathcal{P}'_l$ can save 19.11% average time compared with $\mathcal{P}_l$.*

Fang et al. [42] stated that the considered LRP is NP-hard based on their observation but the proof was not given. In this chapter, the NP-hardness of the studied LRP is shown below.

Theorem 4 *The LRP is NP-hard.*

Proof: The proof is based on the fact that constrained shortest path problem (CSPP) reduces to the special case of the LRP with only one task. Let us consider the following instance of CSPP. Given a graph $G' = (N', A')$ associated with two weights $c_{ij} > 0$ (say, cost) and $d_{ij} > 0$ (say, delay), $(i, j) \in A'$, two distinguished nodes s and t, $s, t \in N'$ and a positive value T. The CSPP consists of finding a minimum cost s-t path with its total delay being equal to or less than T.

We then show how to transform the above CSPP instance into an instance of the special case of the LRP with only one task. Denote the only task as task 1. Let us map (N', A') to (N, A) (i.e., N' and A' are mapped to N and A, respectively), s to o_1 and t to d_1. C_{ij}, τ_{ij}, and T_1 are equal to c_{ij}, d_{ij}, and T, respectively. Through such linear transformation, the CSPP is reduced into the special case of the LRP with only one task. As the CSPP is NP-hard even for acyclic networks [127], the LRP with one task is consequently NP-hard. Certainly, the LRP in general case is NP-hard. □

3.3 Property analysis

In this section, we first investigate several special cases for the LRP. Note that these special cases correspond to classical combinatorial optimization problems and can be tackled using existing techniques. The potential benefits are that if an instance is recognized as one special case of them, then it can be efficiently solved accordingly. Then, the LRP in the general case is analyzed.

3.3.1 Special cases

Case 1: The LRP with only one task and large task travel deadline.

When the travel deadline of the task is large enough, then the travel deadline constraint can be relaxed. Obviously, the special LRP in Case 1 is equivalent to finding a reserved path with minimum impact. Then, the following proposition is straightforward.

Proposition 1 *The special LRP in Case 1 is equivalent to a shortest path problem.*

Remark 3 *The special LRP in Case 1 is polynomially solvable as the shortest path problem that can be efficiently solved by the Dijkstra shortest path algorithm [35] whose time complexity is $O(|N|^2)$.*

Case 2: The single-source LRP with large task travel deadline.

For this case, we have the following proposition.

Proposition 2 *The special LRP in Case 2 is equivalent to a directed Steiner tree problem (DSTP).*

Proof: Let us consider an instance of DSTP as follows. Given a directed graph $G = (V, E)$ with positive weights on the edges, a set of terminal vertices $V' \subseteq V$, a root node vertex r. $i, j \in V$, edge $(i, j) \in E$ and its weight is denoted by w_{ij}. The instance of DSTP consists of finding a minimum weight out-branching tree T rooted at r, such that the vertices in V' are included in T.

We now show how to transform the DSTP instance into an instance of the special LRP in Case 2. Let us map graph $G = (V, E)$ to graph $G = (N, A)$ (i.e., V and E are mapped to N and A, respectively), V' to D and r to the only origin node. The weight C_{ij} is equal to w_{ij}. Then, designing the task paths with minimal impact of lane reservation is equivalent to finding a minimum weight out-branching tree T. With such linear transformation, the DSTP is reduced into the special Case 2 of the LRP. □

Remark 4 *The special LRP in Case 2 is NP-hard, as the DSTP is known to be NP-hard [66], and it can be efficiently solved by the dual ascent approach [123].*

Case 3: The LRP with only one task.

Based on the complexity proof before, we have the following proposition.

Proposition 3 *The special LRP in Case 3 is equivalent to a constrained shortest path problem.*

Remark 5 *The special LRP in Case 3 is NP-hard and can be efficiently solved by Lagrangian relaxation algorithm [127].*

3.3.2 General case

Generally, an LRP contains multiple tasks and different tasks are allowed to share reserved lanes. Since the path design of any task may be influenced by other tasks' paths due to the requirement of minimizing negative impact of reserved lanes. Consequently, the path of any task may be not an optimal path determined by solving its corresponding constrained shortest path problem. That is to say, solving an LRP with $|K|(|K| > 1)$ tasks may be not equivalent to independently solving $|K|$ constrained shortest path problems. In fact, for the LRP in general case, the following proposition is straightforward.

Proposition 4 *Solving an LRP with multiple tasks is equivalent to finding an optimal loopless path respecting travel deadline constraint for each task to form the best path combination such that the total negative impact of reserved lanes is minimized.*

As previously analyzed, we know that our considered problem with only one task is NP-hard even for acyclic networks. This means that the LRP with multiple tasks in the network with cycles is even harder to handle. In the following section, a solution approach to efficiently solve the LRP in general case is developed.

3.4 Solution approach

For the considered LRP in general case, Fang et al. [42] proposed a cut-and-solve (CS) based optimal algorithm, which can solve problem instances with up to 150 nodes and 30 tasks within 18000 CPU seconds. In this chapter, a new fast two-phase exact algorithm is developed to efficiently solve the larger-size LRP. The algorithm is composed of two major phases. In the first phase, all feasible paths respecting the travel deadline constraint are enumerated for each task $k \in K$. An optimal lane reservation scheme and task paths are then determined in the second phase. We detail our new optimal algorithm in what follows.

3.4.1 Task path enumeration

As indicated by Proposition 3, the LRP with one task is a constrained shortest path problem that consists of finding a loopless reserved path respecting the travel deadline constraint with minimal negative impact. Thus, for any task $k \in K$, let P_k denote the set of all loopless paths connecting its origin and destination and respecting its travel deadline (i.e., the total travel duration is equal to or less than T_k).

Remark 6 *The optimal path of the task $k \in K$ of the LRP must be in the set P_k.*

It is not hard to find that the problem determining the set P_k for each task $k \in K$ is equivalent to finding all loopless paths with their travel duration being equal to or less than T_k in the direct graph $G(N, A)$. We note that such problem can be efficiently solved using the well-known Yen's $\mathcal{K}$-shortest loopless path algorithm [132] with its time complexity $O(\mathcal{K}|N|(|A| + |N|\log|N|)$ [18], which belongs to a kind of deviation algorithms ranking the first $\mathcal{K}$ loopless paths for given pair of nodes. To more efficiently obtain the set $P_k, \forall k \in K$, the search space for each $k \in K$ is reduced before using Yen's $\mathcal{K}$-shortest loopless path algorithm. For $k \in K, a \in A$, we define the possibly passed arc set A_k as follows.

$$A_k = \{(i,j)|\varphi(o_k,i) + \tau_{ij} + \varphi(j,d_k) \leq T_k\}, k \in K \tag{3.13}$$

where $\varphi(o_k,i)$(resp. $\varphi(j,d_k)$ denotes the shortest path from o_k to i(resp. j to d_k) when all the arcs in the network are reserved. Note that arcs belonging to $A\backslash A_k$ would not be passed by task k, otherwise the travel deadline constraint will be violated. In other words, set A_k includes all arcs in the network that may be passed by task $k \in K$. Since set A_k is a subset of A, obviously the resolution of the path enumeration for task k will be accelerated.

Remark 7 *From Proposition 4, we can observe that the path selected for task $k \in K$ in the optimal solution of the LRP may not be the travel deadline constrained path with minimal negative impact.*

As the above remark, after obtaining the set P_k for each $k \in K$, the path for any task cannot be simply determined to be the one in the set P_k with the minimal negative impact. According to Proposition 4, the LRP is to find the best task path combination in order to minimize the total negative impact.

3.4.2 Lane reservation and task path determination

The candidate path sets $P_k, \forall k \in K$ are determined in the above phase. In the second phase, we propose an integer programming method to determine the optimal lane reservation scheme and task path from P_k for each task $k \in K$. To formulate the model of determining optimal lane reservation and all task paths, we need to additionally define parameter δ^k_{pij}, if path $p \in P_k$ passes arc $(i,j) \in A_k$, $\delta^k_{pij} = 1$, and 0 otherwise, and a new variable y^k_p as follows.

y^k_p: $=1$ if the path $p \in P_k$ is selected; and 0 otherwise, $\forall k \in K$.

Then, a new ILP for determining optimal lane reservation and task path is developed as follows.

$$\mathcal{P}''_l : \quad \min \sum_{(i,j)\in A} C_{ij} z_{ij}$$

$$\text{s.t.} \quad \sum_{p \in P_k} y^k_p = 1, \forall k \in K \tag{3.14}$$

Algorithm for large-size LRP

```
1: Initialize $P_k = \varnothing$ for all $k \in K$, $k = 0$
2: **while** $k \leq |K|$, **do**
3: let $l = 1$
4: for task $k$, define set $A_k$ based on formula (3.13)
5: for OD pair $(o_k, d_k)$, compute the $l$-th shortest path using Yen's $\mathcal{K}$-shortest loopless path algorithm, and record the travel duration $d_l^k$ and its path $p_l^k$
6: **if** $d_l^k \leq T_k$, $P_k = P_k \cup \{p_l^k\}$, $l = l + 1$ and **goto** step 5; **otherwise** $k = k + 1$ and **goto** step 2
7: **end while**
8: Output $P_k$ for all $k \in K$
9: Construct model $\mathcal{P}_l''$ with sets $P_k, \forall k \in K$ obtained in the first phase
10: Solve $\mathcal{P}_l''$ exactly using CPLEX ILP solver
11: Output an optimal lane reservation scheme and the corresponding paths for tasks.
```

Fig. 3.1: Algorithm LRP: algorithm for large-size LRP

$$\sum_{p \in P_k} \delta_{pij}^k y_p^k \leq z_{ij}, \forall k \in K, \forall (i,j) \in A \tag{3.15}$$

$$y_p^k, z_{ij} \in \{0,1\}, \forall p \in P_k, \forall k \in K, \forall (i,j) \in A \tag{3.16}$$

where constraint (3.14) indicates that only one path is selected from the candidate path set P_k. Constraint (3.15) ensures that the path of task $k, \forall k \in K$ involving arc $(i,j), \forall (i,j) \in A$ only if this arc is reserved. Constraint (3.16) gives the ranges of decision variables. Note that the model $\mathcal{P}_l''$ is a linear program which can be tackled by the optimization software such as CPLEX ILP solver. The overall algorithm for the LRP, called Algorithm LRP, can be outlined in Fig.3.1.

3.5 Computational results

In this section, we conduct numerical computational experiments to evaluate the performance of the proposed algorithm. Our algorithm is coded in C++ language and combined with Yen's $\mathcal{K}$-shortest loopless path algorithm [132] and CPLEX (version 12.6) ILP solver with default settings. All the experiments are conducted on a PC with 2.5 GHz and 2.95 GB RAM under Windows 7.

The performance of the proposed algorithm is evaluated on 81 sets of instances with five instances each set, including 120 benchmark instances [42] and 285 newly generated instances. The new instances are randomly generated based on the way in [42] described as follows. Waxman's network model [121] is used to generate the network $G(N, A)$. $\rho = 2|A|/|N|$ is called the average node degree. To be more specific, the nodes of $G(N, A)$ are randomly distributed in a square area $[0, 100] \times [0, 100]$, the existence of arc (i, j) between nodes i and j is dependent by a probability function α exp $(-L_{ij}/\beta L_{max})$, where L_{ij} and L are the Euclidean distance between nodes i and j and the maximum Euclidean distance between any pair of nodes, respectively, and $0 \leq \alpha \leq 1$ and $0 \leq \beta \leq 1$. The origin and destination nodes o_k and $d_k, k \in K$ are randomly generated from set N. Parameter τ_{ij} is computed by L_{ij}/V, where V denotes the average travel speed on a reserved lane of arc (i, j) and is set as 60. Parameter τ'_{ij} represents the travel time on arc (i, j) without reserved lanes. It is computed by $L_{ij}/V\phi_{ij}$, where ϕ_{ij} is randomly generated in the interval [0.5, 0.8]. The impact of a reserved lane on arc (i, j) is defined as $C_{ij} = r_{ij}\tau'_{ij}$, where r_{ij} is a given number. The travel deadline for task k is defined as $T_k = L_k + \lambda(L'_k - L_k)$, where L_k (resp. L'_k) is the shortest travel time from origin o_k to destination d_k when all arcs in the network are reserved (resp. no lanes in the network are reserved) and λ is a given parameter. In the default case, r_{ij} is set as $r_{ij} \in [0.2, 0.3]$ and λ is randomly generated in $[0, 1]$. Besides, sensitive analysis for the performance of our algorithm is conducted with different values of r_{ij} and λ.

For brevity, let CT_0, CT_{cs} and CT_{tp} denote the average computational time (CPU seconds) spent by CPLEX ILP solver solving $\mathcal{P}'_l$ (i.e., solving the LRP with a direct use of CPLEX), the CS algorithm [42] and by our two-phase algorithm for five instances of each set, respectively. Besides, let $CT_{tp\text{-}f}$ (resp. $CT_{tp\text{-}s}$) denote the average CPU time (seconds) spent by the first (resp. second) phase of our proposed algorithm. Note that the computational times of both methods are limited to 18000s (i.e., five hours), as is the case in [42].

Table 3.1: Comparison results for the instances with $|N| = 60$-90

Set	$\|N\|$	$\|K\|$	ρ	CT'_0	CT_0	$(CT'_0 - CT_0)/CT'_0(\%)$
1	60	25	7	9.95	6.75	32.09
2	60	30	7	26.73	22.89	14.37
3	70	25	7	40.99	36.25	11.56
4	70	30	7	93.11	64.78	30.43
5	80	25	7	46.62	36.19	22.36
6	80	30	7	124.76	102.64	17.73
7	90	25	7	171.08	99.24	41.99
8	90	30	7	317.12	302.96	4.47
Average				103.80	83.96	19.11

To demonstrate that the proposed improved model (see Section 2.2) is more efficient, we compare it with the previous model proposed by [42] by solving a number of instances. The two models for these instances are both solved by CPLEX. The comparison results are summarized in Table 3.1. Let CT_0' denote the computational time spent by CPLEX for solving $\mathcal{P}_l$. We can observe from Table 3.1 that the computational time of the improved model is less than that of the existing model and the former saves an average 19.11% time compared with the latter. This indicates that the model $\mathcal{P}_l'$ is more efficient than the previous model proposed by [42].

To evaluate the performance of the proposed algorithm, we first compare it with benchmark instances with the state-of-the-art algorithm (i.e., the CS algorithm proposed by [42]) in terms of computational time in obtaining optimal solutions. The computational results are reported in Tables 3.2 and 3.3 and Fig.'s 3.2 and 3.3.

Table 3.2: Comparison results for the instances with $|N| = 100$

Set	$\|N\|$	$\|K\|$	ρ	CT_{cs}	$CT_{tp\text{-}f}$	$CT_{tp\text{-}s}$	CT_{tp}	CT_{cs}/CT_{tp}
9	100	10	5	1.59	3.42	0.39	3.80	0.42
10	100	15	5	1.56	1.52	0.35	1.87	0.84
11	100	20	5	41.82	8.68	0.95	9.64	4.34
12	100	25	5	37.21	57.98	7.05	65.03	0.57
13	100	30	5	63.45	3.80	1.09	4.89	12.98
14	100	10	7	6.41	0.52	0.09	0.62	10.38
15	100	15	7	62.32	2.46	0.26	2.72	22.88
16	100	20	7	315.98	2.87	0.32	3.19	99.10
17	100	25	7	1193.88	1.75	0.24	1.98	601.65
18	100	30	7	1288.72	1.75	0.30	2.05	627.73
19	100	10	12	18.67	2.45	0.22	2.67	7.00
20	100	15	12	78.30	5.79	0.36	6.14	12.75
21	100	20	12	496.89	67.25	16.23	83.48	5.95
22	100	25	12	1590.22	10.61	0.77	11.38	139.72
23	100	30	12	8900.67	197.38	883.58	1080.95	8.23
Average				939.85	24.55	60.81	85.36	11.01

Table 3.2 and Fig. 3.2 report the computational results for instances with fixed number of nodes $|N| = 100$ and average node degree ρ ranging from five to twelve. It can be observed from Table 3.2 that the average CPU time of the CS algorithm CT_{cs} varies from 1.59s to 8900.67s with its average value being 873.18s, while that of our algorithm CT_{tp} varies from 0.62s to 1080.95s with its average value being 85.36s.

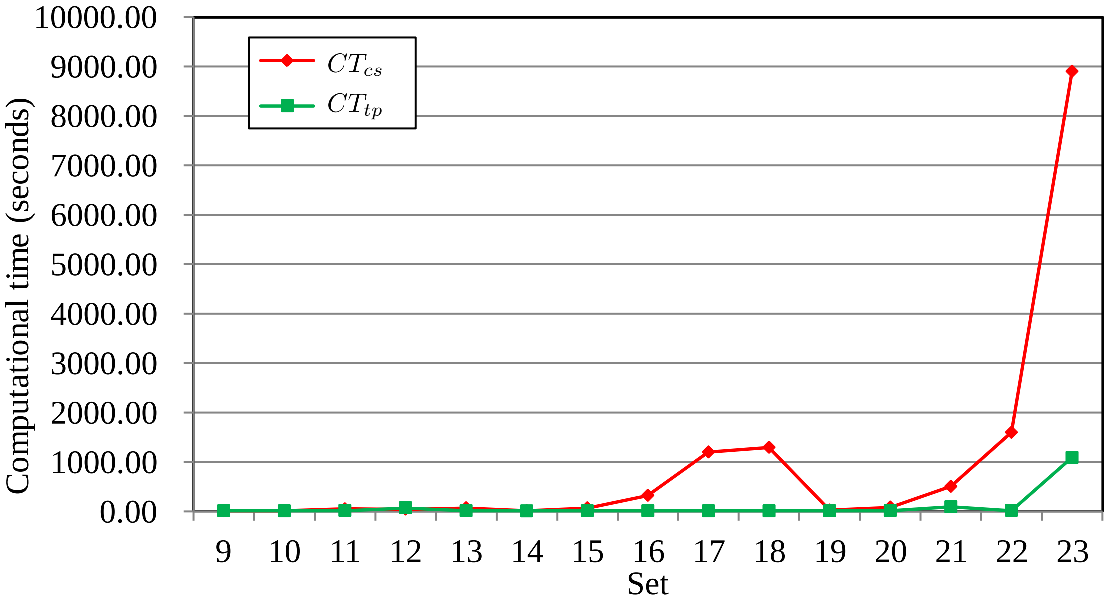


Fig. 3.2: Comparison results for the instances with $|N| = 100$

CT_{tp} is less than CT_{cs} over all sets 9-23 except the small-size sets 9, 10 and 12. On average, CT_{cs} is eleven times more than CT_{tp}. This indicates that our algorithm is much more efficient than the CS algorithm in terms of computational time. Moreover, CT_{cs} increases sharply with $|K|$ for a given average node degree, especially for larger $\rho = 7$ and 12, whereas CT_{tp} varies slightly and it does not necessarily increase with $|K|$. For example, CT_{tp} for sets 8 and 9 decreases from 3.19s to 1.98s when $|K|$ increases from 20 to 25. As shown in Fig. 3.2, CT_{cs} increase much sharply for sets 21-23, whereas CT_{tp} increases gradually.

On the other hand, we observe that $CT_{tp\text{-}s}$ are positively correlated with $CT_{tp\text{-}f}$ over sets 1-15 (i.e., the larger $CT_{tp\text{-}f}$ is, the larger $CT_{tp\text{-}s}$ is). For example, $CT_{tp\text{-}f}$'s are 8.68s and 57.98s for sets 3 and 4, respectively, and $CT_{tp\text{-}s}$'s are 0.95s and 7.05s, respectively. The main reason is that a larger $CT_{tp\text{-}f}$ implies that more task path candidates exist for the tasks, which means more integer variables y_p^k in $\mathcal{P}_l''$ in step 9 of our algorithm such that it is more difficult to be solved. Besides, it can be seen that generally $CT_{tp\text{-}f}$ is larger than $CT_{tp\text{-}s}$, which shows the first phase consumes more time. An exception is set 23 (i.e., the largest-size set) where $CT_{tp\text{-}s}$ is larger than $CT_{tp\text{-}f}$. The reason is that there exist many task path candidates and it makes the model formed in second phase much difficult to be solved.

Table 3.3 and Fig. 3.3 present the results when $|N|$ is increased from 110 to 150 and $|K|$ ranges from 10 to 30 for a given average node degree. In Table 3.3, we

Table 3.3: Comparison results for the instances with $|N| = 110$-150

Set	$\|N\|$	$\|K\|$	ρ	CT_{cs}	$CT_{tp\text{-}f}$	$CT_{tp\text{-}s}$	CT_{tp}	CT_{cs}/CT_{tp}
24	110	10	7	6.27	6.07	0.58	6.65	0.94
25	110	15	7	18.02	7.93	0.85	8.78	2.05
26	120	15	7	103.29	16.20	1.70	17.90	5.77
27	120	20	7	121.68	4.56	0.55	5.11	23.81
28	130	20	7	299.71	7.68	0.59	8.28	36.21
29	130	25	7	1406.91	9.36	0.61	9.97	141.09
30	140	25	7	1575.25	5.19	0.51	5.70	276.35
31	140	30	7	1686.95	17.05	1.31	18.36	91.89
32	150	30	7	1878.07	100.21	8.09	108.31	17.34
Average				788.46	19.36	1.64	21.01	37.54

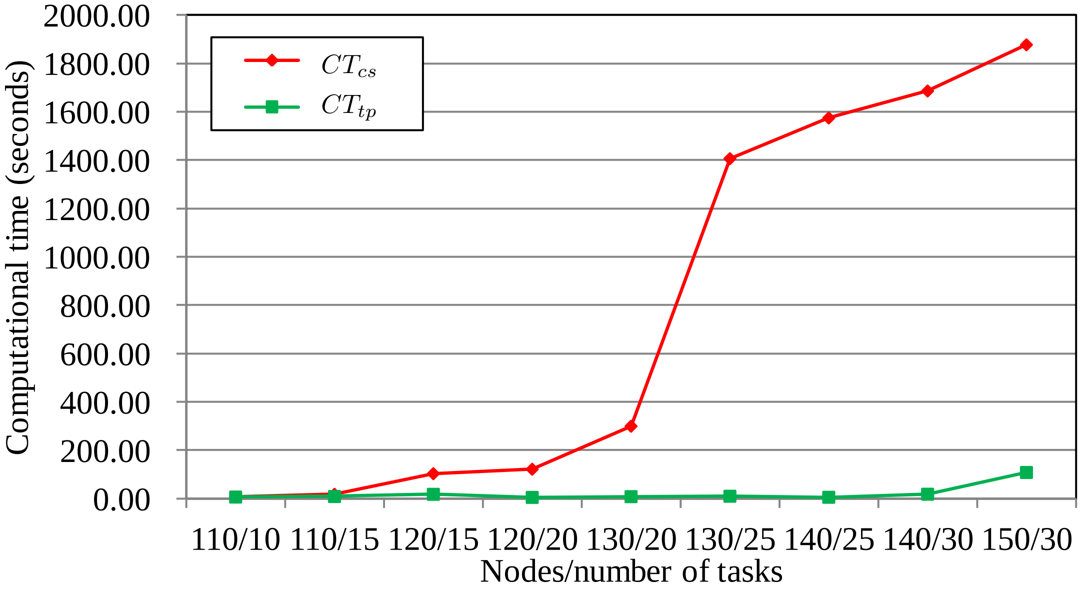


Fig. 3.3: Comparison results for the instances with $|N| = 110$-150

can see that CT_{cs} varies from 6.27s to 1878.07s with its average value being 788.46, whereas CT_{tp} ranges from 5.70s to 108.31s with it average value being 21.01s. CT_{tp} is less than CT_{cs} over sets 24-32, except the smallest-size set 24. The CS algorithm spends more than 37 times average time of that by our algorithm. This further shows the efficiency of our algorithm as compared with the CS algorithm proposed by [42]. Moreover, we can observe from Fig. 3.3 that CT_{cs} exponentially increase with the size of the problem, while CT_{tp} varies very slightly. This shows that the newly proposed algorithm is more robust for the LRP.

In order to further evaluate the performance of the proposed method for solving larger-size problems, 57 newly generated larger-size problem sets are tested. Results are reported in Tables 3.4-3.7.

Table 3.4: Comparison results for the instances with $|N| = 160$-200

Set	$\|N\|$	$\|K\|$	CT_0	CT_{cs}	$CT_{tp\text{-}f}$	$CT_{tp\text{-}s}$	CT_{tp}
33	160	20	876.44	388.84	9.01	1.05	10.14
34	160	30	4409.98	2294.33	10.72	0.67	11.40
35	170	30	12999.62	6758.82	2.66	0.21	2.87
36	170	35	15864.33	10997.67	2.78	0.23	3.01
37	180	35	-	17341.28	2.45	0.30	2.75
38	180	40	-	-	2.50	0.36	2.85
39	190	40	-	-	7.05	0.85	7.90
40	190	45	-	-	8.03	0.93	9.23
41	200	45	-	-	4.43	0.35	4.78
42	200	50	-	-	5.72	0.44	6.16
Average			>14215.04	>12778.09	5.54	0.54	6.11

Table 3.4 presents the results for large-size instances with $|N|$ increasing from 160 to 200, $|K|$ varying from 20 to 50 and fixed average node degree being seven. We can observe in Table 3.4 that the CPU time consumed by our algorithm is far less than those by CPLEX and the CS algorithm over all sets 33-42. As the problem size increases, CT_0 and CT_{cs} increase exponentially, whereas CT_{tp} varies slightly. It is worthwhile to note that CPLEX and the CS algorithm in [42] can only exactly solve the four (resp. five) of ten sets within 18000s, whereas our algorithm can exactly solves all the sets with an average time 6.11s. Besides, we can observe that $CT_{tp\text{-}f}$ is larger than $CT_{tp\text{-}s}$ over all the sets 33-42.

Table 3.5 reports the computational results for larger-size instances with $|N|$ increasing from 300 to 700, $|K|$ varying from 30 to 60 under two scenarios $\rho = 5$ and 10. It can be found that CPLEX and the CS algorithm in [42] cannot generate optimal

Table 3.5: Comparison results for the instances with $|N|$ = 300-700

Set	$\lvert N\rvert$	$\lvert K\rvert$	ρ	CT_0	CT_{cs}	$CT_{tp\text{-}f}$	$CT_{tp\text{-}s}$	CT_{tp}
43	300	30	5	-	-	23.18	1.70	24.89
44	300	40	5	-	-	25.46	2.37	27.82
45	400	40	5	-	-	30.20	1.79	31.99
46	400	45	5	-	-	30.53	1.73	32.27
47	500	45	5	-	-	42.70	17.66	60.36
48	500	50	5	-	-	60.07	21.32	81.38
49	600	50	5	-	-	298.68	459.74	758.42
50	600	55	5	-	-	733.51	620.91	1354.42
51	700	55	5	-	-	1740.83	6553.05	8293.88
Average				-	-	331.68	853.36	1185.05
52	700	60	5	-	-	-	-	-
53	300	30	10	-	-	73.20	6.33	79.53
54	300	40	10	-	-	241.91	19.92	261.83
55	400	40	10	-	-	199.15	15.30	214.46
56	400	45	10	-	-	175.20	7.70	182.90
57	500	45	10	-	-	872.13	100.21	972.34
58	500	50	10	-	-	755.42	84.01	839.43
Average				-	-	386.17	38.91	425.08
59	600	50	10	-	-	-	-	-

solutions for any set within 18000s (i.e., they lose their power for these large-size instances totally), whereas the proposed algorithm is able to exactly solve the instances with up to 700 nodes and 55 tasks with average node degree $\rho = 5$ and 500 nodes and 50 tasks with average node degree $\rho = 10$, respectively. Due to the NP-hardness of the problem, we find that the proposed algorithm rapidly increases with the problem size for each given average node degree. For example, CT_{tp} for set 43 is 24.89s, while for set 51 is 8293.88s. Moreover, it is not hard to find that the increase of CT_{tp} is mainly caused by the increase of $CT_{tp\text{-}s}$. We also observe that our algorithm consumes more time to solve instances with a larger average node degree when $|N|$ and $|K|$ are given. For example, CT_{tp} for set 48 is 81.38s, while for set 58 is 839.43s. The possible reason is that a larger average node degree may result in more task path candidates for each task, which requires more computational effort for both phases in the proposed algorithm. We note that the proposed algorithm loses power for solving sets 52 and 59 due to the lack of memory in the first phase.

Table 3.6: Computational results for sensitive analysis of different T_k

Set	λ	$\lvert N \rvert$	$\lvert K \rvert$	$CT_{tp\text{-}f}$	$CT_{tp\text{-}s}$	CT_{tp}
60	0.2	200	20	1.57	0.41	1.98
61	0.2	300	20	2.60	2.00	4.59
62	0.2	400	20	3.61	0.51	4.12
63	0.2	500	20	3.16	0.53	3.69
64	0.2	600	20	4.66	0.61	5.27
Average				3.12	0.81	3.93
65	0.5	200	20	4.50	0.70	5.19
66	0.5	300	20	14.72	2.02	16.74
67	0.5	400	20	11.95	1.22	13.17
68	0.5	500	20	18.70	1.14	19.84
69	0.5	600	20	45.95	2.80	48.75
Average				19.16	1.58	20.74
70	0.8	200	20	23.55	6.42	29.97
71	0.8	300	20	58.38	9.06	67.43
72	0.8	400	20	81.40	7.03	88.44
73	0.8	500	20	326.63	45.59	372.23
74	0.8	600	20	601.74	25.52	627.26
Average				218.34	18.72	237.07

Finally, we conduct sensitive analysis for the input parameters T_k and C_{ij}. Table 3.6 and Fig. 3.4 depict the results for three scenarios regarding the travel deadline

T_k. It is defined as $T_k = L_k + \lambda(L'_k - L_k)$. The values of λ are set as 0.2. 0.5, and 0.8, respectively, that attempts to generate small, medium, and large values of T_k. We can see in Table 3.6 that our algorithm is able to solve all the problem sets within 11 minutes (i.e., 660s). It can be observed that for the three scenarios CT_{tp} varies from 1.98s to 5.27s, 5.19s to 48.75s, and 29.97s to 627.26s, respectively with its average values being 3.93s, 29.97s, and 237.07s, respectively. The larger λ is the larger CT_{tp} is. For example, the values of CT_{tp} are 5.27s, 48.75s, and 627.26s for sets 55, 60, and 65, respectively. The main reason is that a tighter T_k results in less feasible path candidates for task k, which naturally requires less computational effort. In addition, for each scenario, CT_{tp} increases with $|N|$ and it increases more quickly for a large T_k as shown in Fig. 3.4.

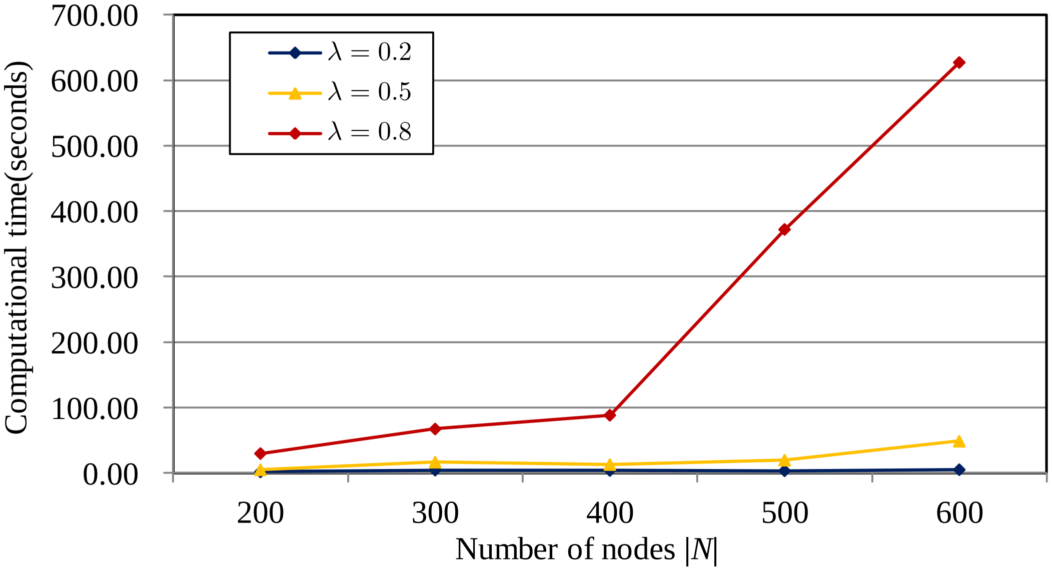


Fig. 3.4: Computational results of sensitive analysis of T_k

Table 3.7 and Fig. 3.5 show the results of sensitive analysis for impact parameters C_{ij}, which is defined as $C_{ij} = r_{ij}\tau'_{ij}$. It can be found from Table 3.7 that the ranges of CT_{tp} under the three scenarios are 4.08-72.22s, 2.82-75.04s, and 4.53-89.35s, respectively. Moreover, it can be observed in Fig. 3.5 that the changing trends of CT_{tp} for the three scenarios are almost the same. Moreover, CT_{cs}'s under the three scenarios are 26.37s, 21.63s, and 26.83s, respectively. These results indicate that our algorithm is insensitive to the changes of C_{ij}.

Table 3.7: Computational results for sensitive analysis of different impact

Set	r_{ij}	$\lvert N\rvert$	$\lvert K\rvert$	$CT_{tp\text{-}f}$	$CT_{tp\text{-}s}$	CT_{tp}
75	[0.1, 0.2]	200	20	3.78	0.30	4.08
76	[0.1, 0.2]	300	20	8.21	0.61	8.82
77	[0.1, 0.2]	400	20	7.82	0.59	8.41
78	[0.1, 0.2]	500	20	13.15	1.46	14.60
79	[0.1, 0.2]	600	20	68.33	3.90	72.22
Average				20.26	1.37	26.37
80	[0.2, 0.3]	200	20	2.51	0.31	2.82
81	[0.2, 0.3]	300	20	3.08	0.42	3.49
82	[0.2, 0.3]	400	20	17.50	1.33	18.83
83	[0.2, 0.3]	500	20	29.40	2.27	31.67
84	[0.2, 0.3]	600	20	69.15	5.89	75.04
Average				24.33	2.04	21.63
85	[0.3, 0.5]	200	20	4.16	0.37	4.53
86	[0.3, 0.5]	300	20	5.08	0.45	5.53
87	[0.3, 0.5]	400	20	15.82	0.97	16.79
88	[0.3, 0.5]	500	20	16.68	1.28	17.97
89	[0.3, 0.5]	600	20	84.43	4.91	89.35
Average				25.23	1.60	26.83

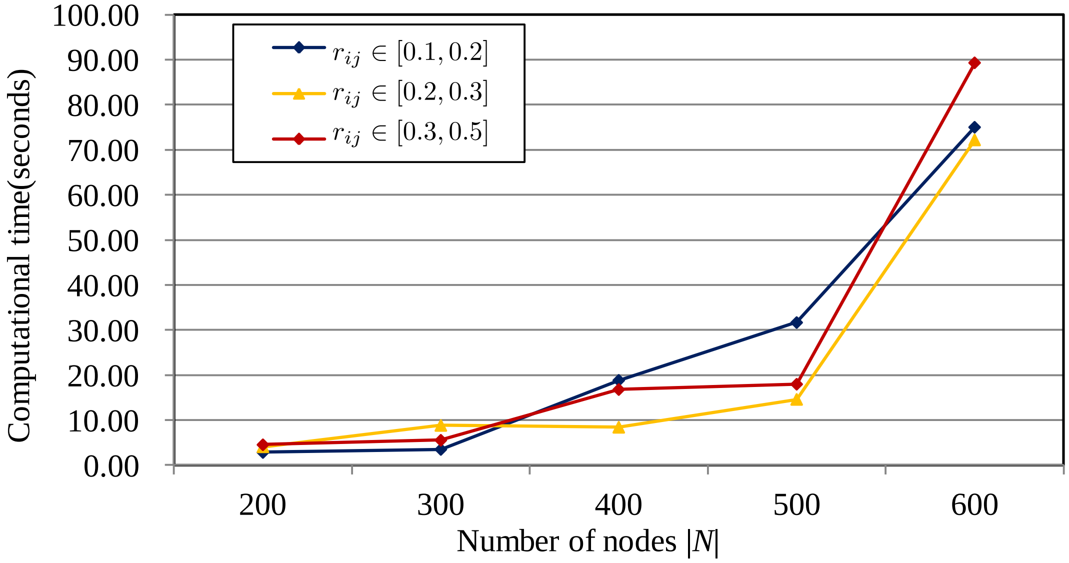


Fig. 3.5: Computational results of sensitive analysis of different impact

3.6 Conclusions

In this chapter, we have studied large-size lane reservation for automated truck freight transportation. For the problem, we first proposed valid inequalities for the integer linear program proposed by [42]. Computational comparison results indicated that these valid inequalities are effective in saving computational time. Furthermore, we investigated several special cases of the considered problem, which were identified to be classical combinatorial optimization problems. To efficiently solve the problem exactly, especially large-size problem instances, we devised a new fast two-phase exact algorithm based on the characteristics of the problem. Computational results demonstrated that the proposed algorithm significantly outperforms the state-of-the-art algorithm (i.e., the CS method proposed by [42]) and it can solve large-size instances with up to 700 nodes and 55 tasks. Furthermore, sensitive analysis experiments for input parameters was conducted to show their impact on the proposed algorithm.

Chapter 4

Robust lane reservation for large-scale special events

4.1 Introduction

In this chapter, we investigate a robust lane reservation problem for large special events. Such events have the following characteristics [125]: 1) many people participate; 2) many activities take place at different venues. In particular, such events usually require organizers to deliver certain people and materials from athlete villages to geographically dispersed venues within a given travel duration. For example, the organizers of the Guangzhou Asian Games in 2010 were committed to deliver athletes to any stadium within 30 min [125]. However, it is not so easy to meet such special transportation needs due to the host city's congested traffic situation. A lane reservation strategy in an existing transportation network may solve this problem in a flexible and efficient way. With this strategy, a lane on some road segments is temporarily reserved for these special transportation tasks such that they can be completed within the given travel duration. As stated in Chapter 2, lane reservation strategy has been successfully applied to some large sport events [17], [134]. Nevertheless, such lane reservation will generate negative impact on normal traffic. Hence, it is critical to optimally reserve lanes so as to minimize the impact of lane reservation. To address the above issues, a lane reservation problem for large-scale special events was firstly studied by Wu et al. [125] and the work [24], [124], [126] further investigated the problem by proposing efficient meta-heuristics. Fang et al. [45] extended the LRP in [125] to a capacitated LRP by additionally considering the issue of road segment residual capacity. We note that the LRP for large-scale special events is a generalization of the LRP studied in Chapter 3, because the path of a task for the former problem is not necessary to be entirely reserved.

One common assumption of all the previous studies on an LRP is that the link travel time is a constant average value. In real life, uncertain traffic features, such as dynamic traffic flow, traffic accidents, and fault of task vehicles, may make the expected link travel time vary. If the travel time of some links on a path exceeds their average values, the total travel duration on the path may also exceed its deadline, i.e., the lane reservation solution obtained may still be infeasible. That is to say, these uncertain features may make an already-obtained solution impractical. Although Fang et al. [43] considered dynamic link travel time for the LRP, in which the whole time period was divided into four intervals, and then, constant average link travel time for each interval was considered. In reality, the link travel time in such intervals may also dynamically change due to uncertain traffic features. This means that the link travel time in such intervals may exceed its constant average value, and the obtained solution may become infeasible. Therefore, it is necessary to handle the possible increase in link travel time due to the uncertainties mentioned above, which was ignored by all the previous studies.

In this chapter, the concept of lane reservation robustness is introduced to handle the uncertainty in link travel times. The robustness of solutions can be seen as the ability to cope with the possible increase in link travel times due to the uncertain traffic features. That is, we try to generate a robust lane reservation solution such that the tasks may still be completed within their deadlines even if the actual travel times of tasks increase due to uncertain traffic conditions. We study a robust lane reservation problem for large-scale special events (called RLRP in short hereafter). Its aim is to optimally choose lanes to be reserved and design paths for special tasks to minimize the total negative impact of reserved lanes and maximize the robustness of the lane reservation solution. Because the lane reservation robustness optimization is introduced, the problem becomes a multi-objective optimization problem. The solution methods introduced by the previous studies can not be directly applied to solving it. For the RLRP, we firstly define a lane reservation robustness. Then, a bi-objective mixed-integer linear program is presented. An enhanced version of the exact ε-constraint method is proposed to find its Pareto front. Moreover, we develop an improved exact ε-constraint and a cut-and-solve combined method, in which several techniques are developed to improve its computational efficiency.

The remainder of this chapter is organized as follows. Section 4.2 gives the definition of lane reservation robustness and formulates a bi-objective mixed-integer linear program. An improved exact ε-constraint method combined with a cut-and-solve method is developed to find the Pareto front in Section 4.3. Section 4.3 reports computational results. Finally, Section 4.5 concludes this chapter.

4.2 Problem formulation

The RLRP can be described as follows. Let a directed graph $G = (N, A)$ represent a transportation network that is composed of a set of nodes N and a set of directed arcs A connecting pairs of nodes. The nodes and arcs can be viewed as road intersections and road links in a transportation network, respectively. Given a set of transportation tasks and their corresponding to origin-destination (OD) pairs, RLRP aims to select some lanes from a transportation network to be reserved and design a time-guaranteed path in the network for each OD pair such that the task will be completed within its given deadline. The objectives are to minimize the total traffic impact of reserved lanes and to maximize the lane reservation robustness. We define the robustness of a given lane reservation solution as follows.

Definition 6 *For task $k, k \in K = \{1, 2, \cdots, |K|\}$, the free slack, S_k, is the difference between its deadline and its total travel duration on the designed task path. The robustness R of a lane reservation solution is defined as the minimal value among all its free slacks $S_k, k \in K (i.e., R = min_{k \in K} S_k)$.*

The robustness of a lane reservation solution represents its ability against the increase in link travel times due to uncertain traffic conditions. To well formulate the problem, we make some assumptions as follows: First, at least two lanes exist on each road link such that a lane can be reserved; otherwise, the impact due to lane reservation will be too heavy. Second, the traffic capacity of reserved lanes is large enough to allow any special task to use by noting that the special tasks are of limited quantity. Third, at most one lane on a road link is allowed to be reserved for the tasks, and each reserved lane can be shared by multiple tasks because the vehicle flow of special transportation tasks is relatively low. Fourth, the path of each task can be composed of reserved and non-reserved lanes. That is to say, the task paths can be partially reserved. The notations for the formulation are listed as follows:

Sets and parameters

N:	set of nodes, $i \in N$
A:	set of arcs, $(i, j), i, j \in N$
K:	set of transportation tasks with $\|K\|$ tasks, $k \in K$
O:	set of origin nodes, $O \subseteq N$
D:	set of destination nodes, $D \subseteq N$

o_k: origin node of task $k \in K$, $o_k \in O$

d_k: destination node of task $k \in K$, $d_k \in D$

T_k: prescribed travel duration to complete task $k \in K$

τ_{ij}: travel time on a reserved lane on arc $(i,j) \in A$

τ'_{ij}: travel time on arc $(i,j) \in A$ without reserved lanes

C_{ij}: negative impact caused by a reserved lane on arc $(i,j) \in A$

Decision variables

z_{ij} : $z_{ij} = 1$, if arc (i,j) is reserved; otherwise $z_{ij} = 0$, $(i,j) \in A$

x_{ij}^k : $x_{ij}^k = 1$, if there is a reserved lane on arc $(i,j) \in A$, and the path of task $k \in K$ pass the arc; otherwise, $x_{ij}^k = 0$

y_{ij}^k : $y_{ij}^k = 1$, if there is no reserved lane on arc $(i,j) \in A$, and the path of task $k \in K$ pass the arc; otherwise, $y_{ij}^k = 0$

R : the lane reservation robustness

A bi-objective MILP for the RLRP is formulated as

$$\mathcal{P}_r: \quad f_1: \min \sum_{(i,j)\in A} C_{ij} z_{ij} \tag{4.1}$$

$$f_2: \max \quad R \tag{4.2}$$

$$\text{s.t.} \sum_{(i,j)\in A} (x_{ij}^k + y_{ij}^k) = 1, i = o_k, \forall k \in K, \tag{4.3}$$

$$\sum_{(i,j)\in A} (x_{ij}^k + y_{ij}^k) = 1, j = d_k, \forall k \in K, \tag{4.4}$$

$$\sum_{(i,j)\in A} (x_{ij}^k + y_{ij}^k) = 0, j = o_k, \forall k \in K, \tag{4.5}$$

$$\sum_{(i,j)\in A} (x_{ij}^k + y_{ij}^k) = 0, i = d_k, \forall k \in K, \tag{4.6}$$

$$\sum_{j:(i,j)\in A} (x_{ij}^k + y_{ij}^k) = \sum_{j:(j,i)\in A} (x_{j,i}^k + y_{j,i}^k), \forall j \in N \backslash \{o_k, d_k\}, \forall k \in K, \tag{4.7}$$

$$\sum_{j:(i,j)\in A} (x_{ij}^k + y_{ij}^k) \leq 1, \forall j \in N \backslash \{o_k, d_k\}, \forall k \in K, \tag{4.8}$$

$$\sum_{j:(j,i)\in A} (x_{ji}^k + y_{ji}^k) \leq 1, \forall j \in N \backslash \{o_k, d_k\}, \forall k \in K, \tag{4.9}$$

$$\sum_{(i,j)\in A} (\tau_{ij} x_{ij}^k + \tau'_{ij} y_{ij}^k) \leq T_k, \forall k \in K, \tag{4.10}$$

$$S_k = T_k - \sum_{(i,j)\in A} (\tau_{ij} x_{ij}^k + \tau'_{ij} y_{ij}^k), \forall k \in K, \tag{4.11}$$

$$R \leq S_k, \forall k \in K, \tag{4.12}$$

$$x_{ij}^k \leq z_{ij}, \forall (i,j) \in A, \forall k \in K, \tag{4.13}$$

$$y_{ij}^k \leq 1 - z_{ij}, \forall (i,j) \in A, \forall k \in K, \tag{4.14}$$

$$z_{ij}, x_{ij}^k, y_{ij}^k \in \{0,1\}, \forall (i,j) \in A, \forall k \in K, \tag{4.15}$$

$$R \geq 0. \tag{4.16}$$

Objective function (4.1) is to minimize the total impact caused by reserved lanes and objective function (4.2) is to maximize the robustness of a lane reservation solution. Constraints (4.3)-(4.9) guarantee that a feasible path for each task from the origin node o_k to its destination d_k exists. To be more specific, constraints (4.3) and (4.4) ensure that there is only one arc outgoing from the origin node o_k and one arc coming into the destination node d_k on the travel path of task k, respectively. Constraints (4.5) and (4.6) guarantee that there are no arcs coming into o_k and no arcs outgoing from d_k on the travel path of task k, respectively. Constraint (4.7) is the flow balance constraint for all the nodes except o_k and d_k. Constraints (4.8) and (4.9) mean that any node $j, j \in N, j \neq ok$ and $j \neq d_k$, can be visited by task $k, k \in K$, at most once. Constraint (4.10) ensures that the travel duration of task $k, k \in K$, does not exceed its deadline T_k. Constraint (4.11) defines the free slack of task $k, S_k, k \in K$, which represents the difference between its deadline and its total travel duration with a lane reservation strategy. Constraint (4.12) ensures that the lane reservation robustness is the minimal value among all the free slacks. Constraint (4.13) guarantees that task k cannot pass a reserved lane on arc (i,j) if no lane of arc (i,j) is reserved. Constraint (4.14) ensures that task k can pass a non-reserved lane on arc (i,j) only if there is no reserved lane on this arc. Constraint (4.15) enforces the binary restrictions on the decision variables. Constraint (4.16) spells a non-negative R.

Theorem 5 *The RLRP is NP-hard.*

Proof: Consider a special case of the RLRP that only objective f_1 is to be optimized and the path of a task is entirely reserved. Obviously, such special case of the RLRP corresponds to the LRP studied in Chapter 3, which have been proved to be NP-hard. Therefore, the RLRP is also NP-hard. □

There are two strongly conflicting objectives in model $\mathcal{P}_r$. To minimize the total impact of reserved lanes f_1, the number of lanes to be reserved needs to be minimized. To maximize the robustness of a lane reservation solution f_2, more lanes need to be reserved to reduce the completion time of the special tasks.

$$\begin{aligned} \mathcal{P}_r': \quad & f_1 : \min \quad \sum_{(i,j)\in A} C_{i,j} z_{ij} \\ & f_2 : \min \quad -R \end{aligned}$$

$$\text{s.t.} \quad \text{Constraints}(4.1) - (4.16) \tag{4.17}$$

In the following section, an approach to solve $\mathcal{P}_r'$ is developed.

4.3 Solution approach

As previously mentioned, for an MCOP, the concept of Pareto optimality replaces the optimality in a single-objective optimization problem. The resolution of an MCOP means to find a set of Pareto optimal (or non-dominated) solutions. For the studied bi-objective RLRP, we aim to propose an exact approach to find all the non-dominated points, i.e., the Pareto front. Scalarization techniques have been proposed to exactly solve MCOPs. The popular and straightforward way is the weighted sum method [133]. It aims to convert an MCOP to a single-objective optimization problem by using a linear weighted sum formulation that combines all the objectives. Its optimal solution would be Pareto optimal if proper weight combinations are used. Note that the converted single objective is an aggregation of all objectives of MCOP via a linear weighted sum. Hence, this method is not appropriate if not all objectives can be represented by the linear combination. Moreover, the weighted sum method is ill-suited for an MCOP with nonconvex objective space [37].

Another well-known technique to solve MCOPs is the ε-constraint method introduced by Haimes et al. [57]. It aims to optimize only the primary objective, called the most preferred one, and the others are transformed into constraints. In theory, the Pareto front of an MCOP can be obtained with the ε-constraint method [117]. Compared with the weighted sum method, this method avoids the drawbacks described above [85]. Moreover, since the ε-constraint method was first introduced to solve the bi-objective shortest path problem in 1982 [29], it has been applied to solve most of BCOPs [139], [15], [46], [64], [75], [97]. The successful applications of the ε-constraint method and its capability to obtain the Pareto front of BCOPs encourage us to use it to solve our problem.

4.3.1 An enhanced version of exact ε-constraint method for BCOPs

As stated in Chapter 2, the algorithm in Fig. 2.8 can find the exact Pareto front of BCOPs with integer objective values. We have observed that these problems solved by the algorithm in Fig. 2.8 have the following two characteristics, which guarantee that the objective values of these problems are always integer:

i) Integer linear programming problem (i.e., all decision variables are integers).

ii) The coefficients of decision variables in objective functions are all integers.

However, for BCOPs with fractional objective values, non-dominated points may be lost by directly using it. Recently, Feng et al. [46] proposed an iterative ε-constraint method to obtain the Pareto front for a bi-objective scheduling problem formulated as an MILP with fractional objective values. However, their method did not consider the so-called minimum unit value defined below, and it needs to solve an MILP for lexicographic optimization at each iteration. Hence, its computational burden may be heavy. To address this issue, we propose an enhanced version of the exact ε-constraint method by redefining parameter δ.

Definition 7 (*Minimum unit value*) *For a BCOP, the minimum unit value is the minimal objective unit value of f_2.*

The enhanced version of the exact ε-constraint method is defined as the algorithm in Fig. 2.8 by setting δ as the minimum unit value of a BCOP.

Theorem 6 *If δ is set as the minimum unit value in the algorithm in Fig. 2.8, the Pareto front of a BCOP can be found by the enhanced version of the exact ε-constraint method.*

Proof: The correctness of Theorem 6 can be proved similar to the proof of Theorem 3 in the work of Bérubé et al. [15] with the redefined δ. For more details, please see [15]. □

Note that the enhanced version can be considered as a generalization of the exact ε-constraint method introduced by Bérubé et al. [15] to solve BCOPs. With the definition of a minimum unit value, the enhanced version of the exact ε-constraint method can find the Pareto fronts of some BCOPs with fractional objective values in some special cases, such as special integer linear programs and MILPs, as follows:

a) integer linear programming problems, where the coefficients of decision variables in the objective functions are fractional;

b) MILPs, where the objective functions contain integer variables only and their coefficients are fractional;

c) MILPs, where the objective functions contain integer and real variables, but the minimum unit value can be determined.

Obviously, the minimum unit values of problems in cases a) and b) are the minimal unit value of the coefficients in f_2. If we set ε as the minimum unit value, then their Pareto fronts can be obtained by using the enhanced version of the exact ε-constraint method. For the problem in case c), generally, the minimum unit value of a BCOP may be determined by the values of real decision variables, coefficients in the function objectives, and coefficients in the constraints. We need to analyze each case to decide the value of δ. We have not found a general method yet for this case even though we have made many attempts, thereby demanding more studies in the future. RLRP addressed in this study belongs to case b) where the minimum unit value can be determined. In this chapter, we apply the enhanced exact ε-constraint method in which δ is defined as the minimum unit value to solve the LRP to demonstrate its applicability.

4.3.2 Improved exact ε-constraint method for the RLRP

An improved exact ε-constraint method is proposed for RLRP. As described above, RLRP has two conflicting objectives. Its first objective is considered as the primary one in this study, because the total traffic impact is concerned by more stakeholders (i.e., all the general-purpose passengers). With the ε-constraint method, the bi-objective model $\mathcal{P}'_r$ can be transformed into the following ε constraint problem:

$$\begin{aligned} \mathcal{P}_C(\varepsilon): \quad & \min \sum_{(i,j)\in A} C_{ij} z_{ij} \\ \text{s.t.} \quad & -R \leq \varepsilon \qquad\qquad (4.18) \\ & \text{and constraints} (4.3) - (4.16) \end{aligned}$$

To simplify the expression, let $C(\varepsilon)$ denote the obtained optimal objective value by solving $\mathcal{P}_C(\varepsilon)$. Note that if the value of the ε is large enough, the reduced problem corresponds to the LRP studied in Chapter 4, which is NP-hard. Hence, the $\mathcal{P}_C(\varepsilon)$ is also NP-hard.

In the following, we propose an improved exact ε-constraint method for RLRP. By analyzing the problem's characteristic, we adapt the enhanced version of the exact ε-constraint method to $\mathcal{P}_1$. A cut-and-solve method is proposed to exactly solve $\mathcal{P}_C(\varepsilon)$ at each iteration of the ε-constraint method. Moreover, the optimal value of $-R$ in the ideal point can be quickly obtained by a simple polynomial algorithm, and a strengthening technique is proposed to reduce redundant runs in the exact ε-constraint method.

4.3.2.1 Computation of ideal and nadir points

For the exact ε-constraint method, the first step is to compute the ideal and Nadir points $(C^I, -R^I)$ and $(C^N, -R^N)$ of $\mathcal{P}'_r$ by optimally solving the following four MILPs:

$$\begin{aligned} \mathcal{P}_{CI}: \quad & C^I = \min \sum_{(i,j)\in A} C_{ij} z_{ij} \\ \text{s.t.} \quad & \text{Constraints } (4.3) - (4.16) \\ \mathcal{P}_{RI}: \quad & -R^I = \min -R \\ \text{s.t.} \quad & \text{Constraints } (4.3) - (4.16) \end{aligned}$$

The problem $\mathcal{P}_{CN}$ is formed by adding constraint (4.19) that fixes the optimal value of R, i.e.,

$$\begin{aligned} \mathcal{P}_{CN}: \quad & C^N = \min \sum_{(i,j)\in A} C_{ij} z_{ij} \\ \text{s.t.} \quad & R = R^I \qquad (4.19) \\ & \text{and constraints } (4.3) - (4.16) \end{aligned}$$

The final problem is formed by adding constraint (4.20) that fixes the optimal value of C, i.e.,

$$\begin{aligned} \mathcal{P}_{RN}: \quad & -R^N = \min -R \\ \text{s.t.} \quad & \sum_{(i,j)\in A} C_{ij} z_{ij} = C^I \qquad (4.20) \\ & \text{and constraints } (4.3) - (4.16) \end{aligned}$$

The following corollary obviously holds by Definition 3.

Corollary 1 *(C^I, R^N) and (C^N, R^I) are two non-dominated solutions of RLRP.*

In this chapter, to speed up the exact ε-constraint method, we compute the optimal value of R (i.e., R^I) by using the following polynomial shortest path algorithm. Let $dis(o_k, d_k), k \in K$, denote the shortest travel duration from o_k and d_k.

In 4.1, we use Dijkstra's shortest path algorithm with its complexity $O(|N|^2)$ to compute the shortest travel duration. Via preliminary tests for 20 randomly generated instances with 100 nodes and 30 tasks for problem T_k, the average computation time by using the algorithm in 4.1 is only 0.07% of that by CPLEX. The following corollary is straightforward.

Corollary 2 *R^I of $\mathcal{P}_{RI}$ is equal to $\min_{k\in K}\{T_k - dis(o_k, d_k)\}$ by the algorithm described in Fig. 4.1.*

Calculation of the optimal value of robustness

```
1: Transform the non-reserved network to an entirely reserved network by replacing travel time $\tau'_{ij}$ with $\tau_{ij}$.
2: Find the shortest path from $o_k$ to $d_k$ and calculate the shortest travel duration $dis(o_k, d_k)$ for all tasks by Dijkstra's shortest path algorithm.
3: Calculate $R^I = min_{k\in K}T_k - dis(o_k, d_k)$.
```

Fig. 4.1: Calculation of the optimal value of robustness

4.3.2.2 Definition of parameter δ

As previously described, the enhanced version of the exact ε-constraint method can be used to find the Pareto front of an MILP if its minimum unit value can be determined. Here, we show that the minimum unit value of $\mathcal{P}'_r$ can be found. By Definition 7, the minimum unit value of $\mathcal{P}'_r$ is the minimal unit value of f_2.

By Definition 6, the second objective function has the following equivalent form: $f_2 = \max\min_{k\in K}\{S_k\} = \max\min_{k\in K}\{T_k - \sum_{(i,j)\in A}(\tau_{ij}x^k_{ij} + \tau'_{ij}y^k_{ij})\}$. Objective function f_2 is a linear combination of integer variables x^k_{ij}, y^k_{ij} and their coefficients τ_{ij} and τ'_{ij}. Based on the above analysis, the minimum unit value of RLRP is the minimal unit value of τ_{ij} and $\tau'_{ij}, \forall(i,j) \in A$. Hence, δ can be set as the minimal unit value of τ_{ij} and τ'_{ij}. For this reason, the enhanced version of the exact ε-constraint method can be applied to exactly solve RLRP.

4.3.2.3 Strengthening technique

The exact ε-constraint method may generate dominated solutions at some iterations that are redundant runs. These redundant runs may be time-consuming, particularly for large-size problems. Via preliminary tests, we have observed that dominated solutions have the following characteristic: optimal solutions of consecutive iterations may have the same total traffic impact (f_1) and different robustness of lane reservation (f_2). To reduce the number of dominated solutions, the following strengthening technique is proposed at each iteration of the exact ε-constraint method to improve the obtained optimal solution of $\mathcal{P}_C(\varepsilon)$.

Let vectors (z^*, x^*, y^*) and $(C^*, -R^*)$ denote an optimal solution of $\mathcal{P}_C(\varepsilon)$ and the corresponding objective vector, respectively. Note that a reserved network is known if x^* is fixed. The principal idea of the strengthening technique is to search a shortest path for any task k in such a known reserved network such that $Sk, k \in K$, and the

Strengthening technique

```
1: Obtain the reserved network with $x^*$ from the optimal solution of $\mathcal{P}_C(\varepsilon)$, i.e., $(z^*, x^*, y^*)$.
2: Calculate the shortest travel time for all tasks in the reserved network by Dijkstra's shortest path algorithm.
3: Record all the arcs passed by each task, denoted by $(x', y')$.
4: $(z^*, x', y')$ and $(C', -R')$ are a new optimal solution and its corresponding objective vector of $\mathcal{P}'_r$, respectively.
```

Fig. 4.2: Strengthening technique

robustness of the lane reservation are all maximized and the number of redundant runs is reduced.

An optimal solution with the best robustness of problem $\mathcal{P}_C(\varepsilon)$, denoted as an improved solution (z^*, x', y'), and the new objective vector $(C', -R')$ may be obtained by the following strengthening technique.

Note that for two solutions (z^*, x^*, y^*) and (z^*, x', y'), we have $C' = C^*$ due to the same variable value of Z and $R' \leq R^*$ from Step 2 of the algorithm in Fig. 4.2. If $R' > R^*$, (z^*, x^*, y^*) is dominated by (z^*, x', y'). Computation results reported in Section 4.4 show that the strengthening technique is effective in reducing redundant runs.

4.3.2.4 Cut-and-solve method

As previously presented, a sequence of NP-hard $P(\varepsilon)$'s need be exactly solved in the exact ε-constraint method. The resolution efficiency of $\mathcal{P}_C(\varepsilon)$ seriously influences the efficiency of the ε-constraint method. In order to speed it up for solving the problem, we propose a cut-and-solve method to exactly solve $\mathcal{P}_C(\varepsilon)$ instead of directly using an optimization software, such as CPLEX and Gurobi. Numerical results presented in Section 4.4 show that the proposed method is more efficient than CPLEX. Specially, $\mathcal{P}_{CI}$ and $\mathcal{P}_{CN}$ are viewed as ε-constraint problems with $\varepsilon = +\infty$ and $-R^I$, respectively, and they are also solved by the proposed cut-and-solve method.

a) Preprocessing

In order to speed up the resolution of $PC(\varepsilon)$, we first conduct a preprocessing proposed in the previous chapter to reduce the search solution space. For any $k \in K$, sets A_k and Ω are defined as follows:

$$A_k = \{(i,j)|l(o_k, i) + \tau_{ij} + l(j, d_k) > T_k, \forall (i,j) \in A\}, \forall k \in K \tag{4.21}$$

$$\Omega = \{(i,j)|(i,j) \in A_k, \forall k \in K\} \tag{4.22}$$

where $l(o_k, i)$(resp., (j, d_k)) denotes the shortest travel time from o_k to $i(j$ to $d_k)$ when all arcs in the network are reserved. If an arc in A_k is passed by task k, its deadline constraint will be violated. Hence, any arc in A_k would not be used by task k. Moreover, any arc (i,j) in set Ω would not be passed by any task in a feasible solution. Then, the corresponding decision variables can be fixed to zeros in the optimal solutions of $\mathcal{P}_C(\varepsilon)$, and an equivalent and tighter model $\mathcal{P}'_C(\varepsilon)$ is defined as

$$\begin{aligned}
\mathcal{P}'_C(\varepsilon): \quad & \min \sum_{(i,j)\in A} C_{ij} z_{ij} \\
\text{s.t.} \quad & x_{ij}^k + y_{ij}^k = 0, \forall k \in K, (i,j) \in A_k && (4.23) \\
& z_{ij} = 0, \forall (i,j) \in \Omega && (4.24) \\
& \text{and constraints}(4.3) - (4.16), (4.18)
\end{aligned}$$

Constraints (4.23) and (4.24) can reduce the search space of $\mathcal{P}_C(\varepsilon)$'s without excluding any feasible solutions.

b) Cut-and-solve iteration

As discussed previously, the cut-and-solve method is an iterative search strategy for combinatorial optimization problems. Briefly speaking, a piercing cut (PC^n) is generated at the nth iteration ($n \leq 1$) of the cut-and-solve method, and it divides the solution space of current problem (CP_n) into two subspaces. The small solution space corresponds to a sparse problem (SP^n), and the large solution space corresponds to a residual problem (RP^n). Note that CP_1 is defined as the original problem. SP^n can be exactly solved easily, and its optimal solution if existing provides an upper bound (for a minimization problem) of the original problem. The current best upper bound UB_{best} is updated in case of improvement of the upper bound and the solution space of SP^n is cut off. Meanwhile, we solve a linear relaxed problem of RP^n to obtain a lower bound LB_n because its solution space is so large that it is difficult to be exactly solved. Obviously, if LB_n is greater than or equal to UB_{best}, then the solution corresponding to UB_{best} is an optimal solution of the original problem, and the algorithm is terminated. Otherwise, CP_{n+1} is defined as RP^n, and a new iteration repeats. The reader is referred to Chapter 2 for more details.

PC^n critically influences the efficiency of a cut-and-solve method. The solution space of SP^n should be small enough for easy resolution, and it should be also large enough such that it contains at least a feasible solution of the original problem; otherwise, the best upper bound cannot be updated. Climer and Zhang [30] and Fang et al. [42], [45] defined PC^n by the reduced cost information and achieved good performance. In this chapter, we also define PC^n via reduced cost as

$$PC^n(n \leq 1) : \sum_{z_{ij} \in \phi_n} z_{ij} \leq 1 \tag{4.25}$$

where set $\phi_n = \{z_{ij} | \psi(z_{ij}) > \alpha_n, \forall (i,j) \in A\}, \psi(z_{ij})$ is the reduced cost of z_{ij} obtained by solving CP_n's linear relaxation problem, and α_n is a given positive value.

At iteration $n(n \leq 1)$, CP_{n+1} is defined as $RP^n(CP_1$ is the original problem, i.e., $\mathcal{P}'_C(\varepsilon))$. Then, for the considered problem, SP^n and RP^n are defined, respectively, as

$$\begin{aligned} SP^n: \quad & \min \sum_{(i,j) \in A} C_{ij} z_{ij} \\ \text{s.t.} \quad & \sum_{z_{ij} \in \phi_t} \leq 1, t = 1, 2, ..., n-1 && (4.26) \\ & \sum_{z_{ij} \in \phi_n} = 0 && (4.27) \\ & \text{and constraints}(4.3) - (4.16), (4.18), (4.23), (4.24) \end{aligned}$$

$$\begin{aligned} RP^n: \quad & \min \sum_{(i,j) \in A} C_{ij} z_{ij} \\ \text{s.t.} \quad & \text{Constraints}(4.3) - (4.16), (4.18), (4.23) - (4.26) \end{aligned}$$

The overall algorithm for $P_C(\varepsilon)$ is shown in 4.3.

The improved exact ε-constraint and cut-and-solve combined method to find the Pareto front of the RLRP is outlined as Algorithm RLRP.

4.4 Computational results

The performance of Algorithm RLRP is evaluated by using an instance based on a real network topology and 44 randomly generated problem sets with five instances for each set (i.e., 220 instances). The proposed method is coded in C++ with Visual Studio 2008 embedded with CPLEX (version 12.4) in default setting mode for the resolution of $\mathcal{P}_{RN}$, sparse problems, and linear relaxation of residual problems at each

Cut-and-solve method for $\mathcal{P}_C(\varepsilon)$

```
Implement preprocessing and obtain an equivalent integer linear program $\mathcal{P}'_C(\varepsilon)$.
Initialize $n := 0$ and current best upper bound $UB_{best} := +\infty$ and $CP_1 := \mathcal{P}'_C(\varepsilon)$.
Relax all integer variables of $CP_1$ to be continuous ones and solve the linear relaxation problem, and obtain the reduced cost of variables $z_{ij}$.
Let $n = n + 1$. If $n > 1, CP_n = RP^{n-1}$. Define set $\phi_n$ and $PC^n$.
Cut $CP_n$ into two spaces and obtain problems $RP^n$ and $SP^n$.
Solve $SP^n$ exactly and obtain the optimal objective value $UB_n$. Let $UB_{best} = \min\{UB_n, UB_{best}\}$.
Solve $RP^n$'s linear relaxation problem and obtain the optimal objective value $LB_n$ and the reduced cost of variables $z_{ij}$.
$UB_{best} \leq LB_n$, go Step 5, and otherwise go to Step 4.
Output $UB_{best}$ and the corresponding solution as the optimal objective value and an optimal solution, respectively
```

Fig. 4.3: Cut-and-solve method for $\mathcal{P}_C(\varepsilon)$.

Algorithm RLRP

```
Initial input: set $\delta$ as the minimal unit value of $\tau'_{ij}$ and $\tau_{ij}$, and $Y'_N = \varnothing$
Determine $-R^I$ directly for $\mathcal{P}_{RI}$ by the algorithm described in Fig. 4.1.
Solve $\mathcal{P}_C$ and $\mathcal{P}_{CI}$ by the algorithm described in Fig. 4.3 to determine $C^I$ and $C^N$, respectively. Solve $\mathcal{P}_{RN}$ by CPLEX to obtain $-R^N$.
Set $Y'_N = \{(C^I, R^N)\}$. Let $j = 2$ and $\varepsilon_j = -R^N - \delta$, respectively.
while ($\varepsilon_j \leq -R^I$) do
Solve $\mathcal{P}_C(\varepsilon_j)$ exactly by using the algorithm described in Fig. 4.3, and obtain the optimal solution and its corresponding objective vector $(z^*, x^*, y^*)$ and $(C^*, -R^*)$, respectively.
Obtain an improved solution $(z^*, x', y')$ and its objective vector $(C^*, -R')$ by using the algorithm described in Fig. 4.2.
$Y'_N = Y'_N \cup (C^*, -R')$
$\varepsilon_{j+1} = -R' - \delta$, and $j = j + 1$
end while
Remove dominated points from $Y'_N$ to obtain the exact Pareto front $Y_N$ if existing.
```

Fig. 4.4: Algorithm RLRP: algorithm for the RLRP.

iteration of the cut-and-solve method (i.e., the algorithm described in Fig. 4.3). All experiments are carried out on a PC with a 3.4 GHz processor and 8.0 GB RAM under windows 7.

For simplicity, Algorithm RLRP$'$ is denoted as a simple version of Algorithm 5 without the strengthening technique and in which $\mathcal{P}_C(\varepsilon)$ is directly solved by CPLEX instead of the cut-and-solve method. Let $|N|, |K|$, and $|A|$ denote the number of nodes, tasks, and arcs, respectively; $|F|$ denote the average size of the Pareto front of five instances in a set; CT_r and CT'_r denote the average computation time (CPU seconds) for finding the Pareto front by Algorithms RLRP and RLRP$'$ respectively; and $|J|$ and $|J'|$ denote the average number of ε-constraint problems solved by Algorithms RLRP and RLRP$'$, respectively. R_p is defined as $(|J'| - |J|)/|F| \times 100$, which is the reduction rate of redundant runs by the strengthening technique and measures its performance. Note that the computation time of each instance is limited to 18000s.

4.4.1 Instance based on a real network topology

We first test an instance based on a real network topology from the city of Ravenna, Italy. The instance has 270 arcs, 105 nodes, and 12 transportation tasks, i.e., 12 OD pairs [139]. The number of nodes, arcs, OD pairs, and all related parameters of the network are generated according to [139]. The results of this instance are shown in Table 4.1.

Table 4.1: Computational results for instance based on a real network topology

$\|A\|$	$\|N\|$	$\|K\|$	$\|F\|$	$\|J\|$	$R_p(\%)$	CT_r	CT'_r	CT_r/CT'_r
270	105	12	18	18	33.33	59.54	80.01	0.74

It can be observed from Table 4.1 that our algorithm can find all the non-dominated solutions within 59.34s, which is only 74% of that spent by Algorithm RLRP$'$. Moreover, the number of ε-constraint problems solved in the proposed algorithm is equal to the number of the non-dominated points of the Pareto front, i.e., 18. For this instance, our algorithm finds a non-dominated point at its each iteration. In addition, the rate R_p is 33.33% which means that Algorithm RLRP can avoid 33.33% redundant runs compared with Algorithm RLRP$'$. This indicates that the proposed strengthening technique is effective.

The Pareto front for this instance is shown in Fig. 4.5, where vertical ordinate and horizontal abscissa represent the value of the first and second objective functions, respectively. Decision makers (DMs) can choose a preferred solution from the Pareto

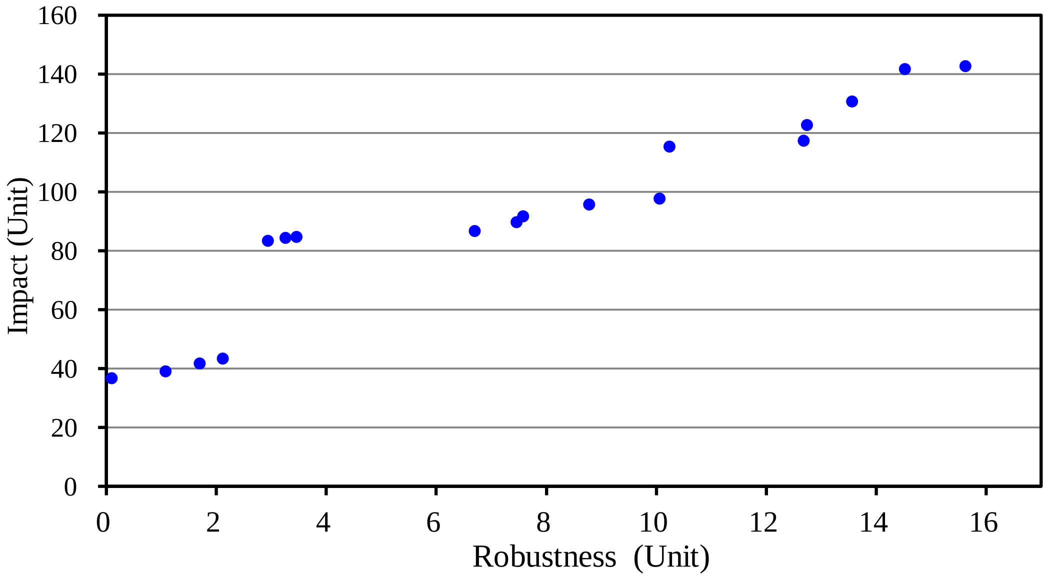


Fig. 4.5: Distribution of the Pareto front of the instance based on a real network topology

front. For example, if DMs prefer a solution with the lowest impact of lane reservation, they can choose the first point from the Pareto front, but the selected solution will be impractical if the travel time on the task paths slightly increases (0.1 unit time) due to uncertain traffic features; on the contrary, if DMs prefer the most reliable solution, they can choose the last point from the Pareto front, which means that the selected solution will be still practical even if the travel time of any task path increases 15.62 unit time due to uncertain traffic conditions, but it will cause the largest impact. In reality, DMs usually choose their preferred solution by some methods such as the fuzzy-logic based method [139], and the selected solution will be of lower impact compared with the last point and bigger robustness compared with the first point, i.e., the obtained solution has not only relatively low impact but also relatively high reliability.

4.4.2 Randomly generated test instances

To further evaluate the performance of the proposed algorithm, randomly generated problem instances are tested. They are generated in the following way. The network graph is generated by Waxman's method [121]. All the nodes are randomly distributed in a square area, and the existence probability of an arc (i, j) is decided by a probability function. The ratio $|A|/|N|$ is called as the average node degree of the network. To simulate the practical transportation network, the ratio $|A|/|N|$ is in the interval [3.5, 4]. The OD pairs are randomly generated from the graph. The given travel deadline for special task T_k is randomly generated in $[l'(o_k, d_k), l(o_k, d_k)]$, where $l'(o_k, d_k)$(resp. $l(o_k, d_k)$) is the shortest travel duration of a path with no reserved lanes (resp., with reserved lanes only). The parameters of link travel times and impact of reserved lanes are estimated as follows.

Estimation of link travel time

To better simulate the practical transportation situation, in this chapter the widely used Bureau of Public Roads (BPR) function [19] is employed to approximately estimate the link travel times $\tau_{ij}, \tau'_{ij}, \tau''_{ij}$, where τ''_{ij} denotes the travel time of general-purpose vehicles on arc (i, j) when a lane of this arc is reserved, given as:

$$\tau'_{ij} = t^0_{ij} \cdot (1 + \alpha(v^g_{ij}/m_{ij}c_{ij})^\beta), \forall(i, j) \in A \tag{4.28}$$

$$\tau_{ij} = t^0_{ij} \cdot (1 + \alpha(v^{task}_{ij}/c_{ij})^\beta), \forall(i, j) \in A \tag{4.29}$$

$$\tau''_{ij} = t^0_{ij} \cdot (1 + \alpha(v^g_{ij} - /(m_{ij} - 1)c_{ij})^\beta), \forall(i, j) \in A \tag{4.30}$$

where t^0_{ij} denotes the free-flow travel time on arc (i, j); v^g_{ij} and c_{ij} are the traffic flow per unit time in a general traffic situation on arc (i, j), the vehicle flow of special tasks per unit time on arc (i, j), and the capacity of one lane on arc (i, j), respectively; m_{ij} is the number of lanes on arc (i, j); $m_{ij}c_{ij}$ and $(m_{ij} - 1)c_{ij}$ are the total capacity of arc (i, j) without and with a reserved lane, respectively; and α and β are two coefficients.

Like most previous studies [24],[42],[44],[45],[124]–[126],[138],[139], the link travel time before implementing lane reservation, i.e., τ'_{ij}, is assumed to be known in advance and takes an average value in a general traffic situation. Hence, the volume-to-capacity (v/c) ratio in the BPR function (4.28) is considered as an average value in a general traffic situation. The data can be estimated by available programs or obtained from the historical data or simulation results. Because the vehicle flow for special tasks is relatively low compared with the capacity of the reserved lanes, even all the tasks simultaneously pass the same reserved lane, τ_{ij} is approximately estimated as the free-flow travel time, as is the case in [86]. τ'_{ij} is approximately estimated as (4.30) because the remaining $m_{ij} - 1$ lanes will burden all the general-purpose traffic demand.

The parameters used in (4.28)-(4.30) are listed as follows. The free-flow travel speed is set as 70 km/h [96], and t^0_{ij} is calculated as the Euclidean distance between nodes i and j divided by the free-flow speed. Capacity c_{ij} is assumed to be 900 veh/h [86]. The ratio $v^g_{ij}/m_{ij}c_{ij}$ is randomly generated in [0.5, 1.2]. The interval is used to generate different traffic situations from a near free-flow situation to an over saturated one [96]. Integer m_{ij} is randomly generated in [2, 4] and $\alpha = 0.15$ and $\beta = 4$.

Evaluation of negative impact due to lane reservation

In all previous studies, the negative impact of a reserved lane on arc C_{ij} is defined as the increased time on the adjacent non-reserved lanes due to lane reservation. In this chapter, the negative traffic impact is estimated as $C_{ij} = P_{ij}(\tau''_{ij} - \tau'_{ij})$, where P_{ij} denotes the number of general-purpose travelers on arc (i, j). With the new formula, the total increased time of all genera-purpose travelers caused by a reserved lane is estimated.

Parameter P_{ij} is assumed to be known in advance and takes an average value in a general traffic situation. To some extent, the more the vehicles, the more the passengers. Hence, P_{ij} is set as $b_{ij}v^g_{ij}$, where integer b_{ij} is defined as the average passenger count inside each vehicle, and it is randomly generated in [1, 20]. In general, a number of heterogeneous general-purpose vehicles may exist, and they have different capacities. Vehicles with larger capacities usually contain more passengers. In China, for example, a public vehicle usually contains more than 20 passengers, whereas a private car usually contains one person. Consequently, for brevity, we randomly generate an average value in the interval [1, 20]. In reality, the data of the number of general-purpose passengers per unit time passing each road link can be estimated by available programs to the DMs or obtained from historical data.

The increased link travel time on adjacent lanes due to lane reservation is calculated as $\tau''_{ij} - \tau'_{ij}$. As reported in [96], statistical results show that the travel time on adjacent lanes increases about 26% after one of three lanes is converted to be a reserved lane in A1 motorway in Paris. This is very close to the computational result (27.63%) calculated by $(\tau''_{ij} - \tau'_{ij})/\tau'_{ij}$ with an average v/c ratio of this highway available in [96]. This means that $\tau''_{ij} - \tau'_{ij}$ is applicable to approximately estimate the increased time on general-purpose lanes due to lane reservation. Moreover, the formula can also reflect the impact degree for special traffic situations. For example, if a two-lane highway is operating in a near-saturated traffic situation (suppose its v/c ratio is 0.95), then the increased percentage in travel time on the remaining lane after implementing lane-reservation will be up to 163.31% according to $(\tau''_{ij} - \tau'_{ij})/\tau'_{ij}$. That is, a reserved lane on this highway will cause a bottleneck. In fact, if reserving a lane on some links will generate bottlenecks on them or have a high impact, then the links may not be selected to be reserved since one objective of the considered problem is to minimize the total impact due to lane-reservation. Moreover, we have conducted computational tests to evaluate the sensitivity of different impact parameters on the performance of our algorithm.

Results for random instances

Table 4.2: Computational results for various types of impact

Set	Impact	$\|A\|$	$\|N\|$	$\|K\|$	CT_r
1	Type 1	240	60	15	19.95
2	Type 1	240	60	20	48.28
3	Type 1	278	70	20	74.59
4	Type 1	278	70	25	119.74
5	Type 1	318	80	20	262.59
6	Type 1	318	80	25	340.56
Average					144.29
7	Type 2	240	60	15	23.73
8	Type 2	240	60	20	43.07
9	Type 2	278	70	20	87.08
10	Type 2	278	70	25	92.52
11	Type 2	318	80	20	171.89
12	Type 2	318	80	25	418.72
Average					139.50
13	Type 3	240	60	15	20.97
14	Type 3	240	60	15	40.17
15	Type 3	278	70	15	117.86
16	Type 3	278	70	15	137.92
17	Type 3	318	80	15	242.16
18	Type 3	318	80	15	377.68
Average					156.13

Table 4.2 gives the computational results for three scenarios regarding the types of impact, called Types 1, 2, and 3, due to lane reservation. Type 1 impact is calculated by $P_{ij}(\tau''_{ij} - \tau'_{ij})$ The other two impacts are calculated as $R_r P_{ij}(\tau''_{ij} - \tau'_{ij}), r =$ 1 and 2, where R_1 and R_2 are randomly generated from the intervals [0.5, 1] and [1, 1.5], respectively. The two intervals are used to simulate smaller impact and larger impact, respectively. The computational time for these three types of impact ranges between 19.95s and 340.56s, 23.73s and 418.72s, and 20.97s and 377.68s, respectively. Moreover, we can see in Fig. 4.6 that the changing trends of CT_r for the three scenarios are almost the same. Furthermore, the average computation times spent by our algorithm for all scenarios are 144.29s, 139.50s, and 156.13s, respectively, which

are almost the same. These results show that the performance of our algorithm is stable to the changes of impact parameters.

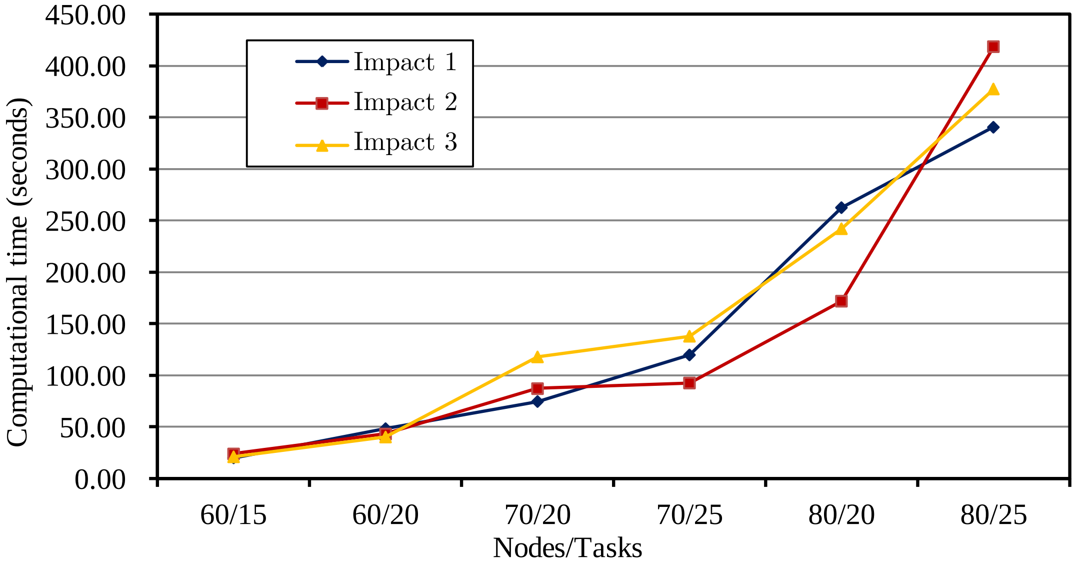


Fig. 4.6: Computational results of sensitive analysis of different impact

Table 4.3: Comparison results for the instances with $|N|$ = 60-90

Set	$\|A\|$	$\|N\|$	$\|F\|$	$\|J\|$	$R_p(\%)$	CT_r	CT'_r	CT_r/CT'_r
19	236	60	11.2	11.2	19.64	55.69	74.67	0.75
20	252	65	14.4	14.4	23.61	94.22	127.79	0.74
21	274	70	17.2	17.2	18.60	134.98	200.18	0.67
22	296	75	19.8	19.8	24.24	205.78	380.54	0.54
23	314	80	23.8	23.8	17.65	288.29	481.34	0.60
24	334	85	27.0	27.2	27.41	369.75	753.29	0.49
25	350	90	30.6	30.8	16.99	445.16	921.50	0.48
Average			20.57	20.63	21.11	227.7	419.9	0.54

Table 4.3 reports the results for instances with $|K| = 20$ and $|N|$ varying from 60 to 90. We first analyze the performance of the proposed strengthening technique. We can observe from Table 4.3 that the average number of ε-constraint problems solved by Algorithm BLRP is less than BLRP′ for sets 19-25. R_p varies from 16.99% to 27.41%, and the average R_p is 21.11%, which means that on average 21.11% redundant runs are avoided by Algorithm BLRP compared with Algorithm BLRP′. This demonstrates that the proposed strengthening technique is useful for reducing the number of dominated solutions and the number of ε-constraint problems solved.

Moreover, in Table 4.3, the gap between $|J|$ and $|F|$ is zero for sets 19-23 and very small for sets 24 and 25. Hence, a Pareto optimal solution is found at almost each iteration of the proposed method. By comparing CT_r and CT'_r from Table 4.3, we find that CT_r is less than CT'_r all the seven sets. In particular, CT_r gradually increases, whereas CT'_r rapidly increases as the number of nodes $|N|$ increases. The average value of CT_r is only 227.7s, which is only 54% of CT'_r. This implies that the cut-and-solve method is more efficient in solving $\mathcal{P}_C(\varepsilon)$'s compared with CPLEX.

Table 4.4: Comparison results for the instances with $|N| = 100$

Set	$\|A\|$	$\|K\|$	$\|F\|$	$\|J\|$	$R_p(\%)$	CT_r	CT'_r	CT_r/CT'_r
26	398	5	21.6	21.6	10.19	31.87	43.23	0.74
27	406	10	22.0	22.0	14.55	70.52	125.41	0.56
28	404	15	25.0	25.0	17.60	170.71	358.05	0.48
29	398	20	31.0	31.0	18.06	304.18	565.59	0.54
30	396	25	34.6	34.6	16.18	440.99	701.16	0.63
31	390	30	35.0	35.0	14.29	511.04	1052.84	0.49
32	394	35	45.0	46.2	17.78	819.39	2066.75	0.40
33	396	40	32.0	32.0	14.38	863.08	3410.29	0.25
34	408	50	27.6	27.6	13.04	1221.77	4371.91	0.28
Average			30.42	30.56	15.38	492.62	1410.58	0.35

Table 4.4 presents the results on the instances with a fixed number of nodes $|N| = 100$, whereas the number of tasks $|K|$ varies from 5 to 50. From Table 4.4, we can see that the value of R_p varies from 10.19% to 18.06%, and the average value is 15.38%. This means that the strengthening technique is effective when the number of tasks varies. It can be seen that $|J|$ is very close to $|F|$, and its gap is zero over all sets except for set 32, and the average $|F|/|J|$ is 99.54% (i.e., 30.42/30.56). This indicates that a Pareto optimal solution is found at almost each iteration of Algorithm BLRP. On the other hand, we can see from Table 4.4 and Fig. 4.7 that CT_r and CT'_r increase with the number of tasks $|K|$, but CT_r increases more slowly than CT'_r. The ratio CT_r/CT'_r varies from 0.25 to 0.74, and the average computation time spent by Algorithm BLRP is only 35% of that spent by Algorithm BLRP′. They imply that Algorithm BLRP is more efficient than Algorithm BLRP′ in solving instances with fixed nodes and varying $|K|$. Take set 33 as an example, Algorithm BLRP spends less than 900s to find the exact Pareto front, whereas Algorithm BLRP′ spends more than 3400s. Moreover, Algorithm BLRP is more efficient for large $|K|$ because the value of CT_r/CT'_r has a decreasing trend.

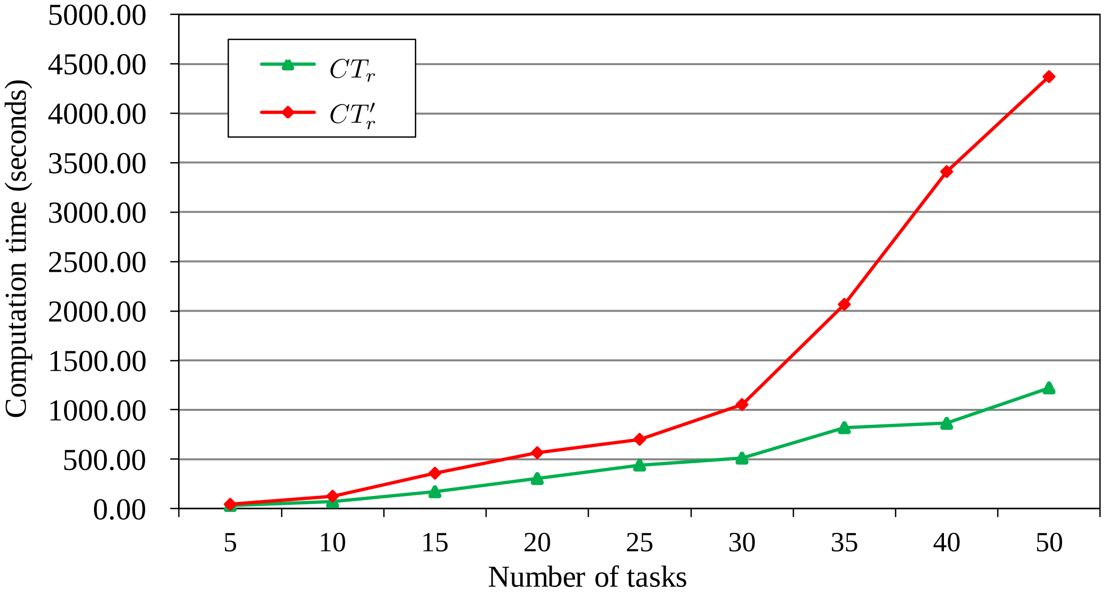


Fig. 4.7: Comparison results for the instances with $|N| = 100$

Table 4.5 presents the results for the instances with $|N|$ varying from 110 to 150 and $|K|$ varying from 20 to 50. $CT'_r/|J'|$ and $CT_r/|J|$ represents the average computational time for solving each $\mathcal{P}_C(\varepsilon)$ by CPLEX and the cut-and-solve method, respectively. It can be seen from Table 4.5 that the gap between $|J|$ and $|F|$ is very small, and the average $|F|/|J|$ is 99.82% (i.e., 39.8/39.87). This indicates that Algorithm BLRP can find a non-dominated solution at almost each iteration. Moreover, 8.57% to 24.17% redundant runs and 14.74% on average can be efficiently avoided by our algorithm compared with Algorithm BLRP$'$ for sets 35-43. This shows that the strengthening technique is also useful for these cases. It can be observed from Table 4.5 and Fig. 4.8 that CT_r and CT'_r increase with the numbers of tasks and nodes; however, CT'_r increases much more quickly than CT_r, and the average CT_r/CT'_r is 37%. This indicates that our algorithm is more efficient than Algorithm BLRP$'$ instances with varying $|N|$ and $|K|$. In addition, due to the NP-hardness of $\mathcal{P}_C(\varepsilon)$'s, we can see from Table 4.5 that $CT'_r/|J'|$ and $CT_r/|J|$ both increase with the number of tasks and nodes, whereas the former increases slower than the latter. This shows that the proposed cut-and-solve method is more efficient than CPLEX in solving $\mathcal{P}_C(\varepsilon)$'s. It is worthwhile to note that Algorithm BLRP$'$ cannot solve four of five instances of set 44 within 18000s, whereas our algorithm only spends an average computation time of 9538.81s to find the Pareto fronts for all of them.

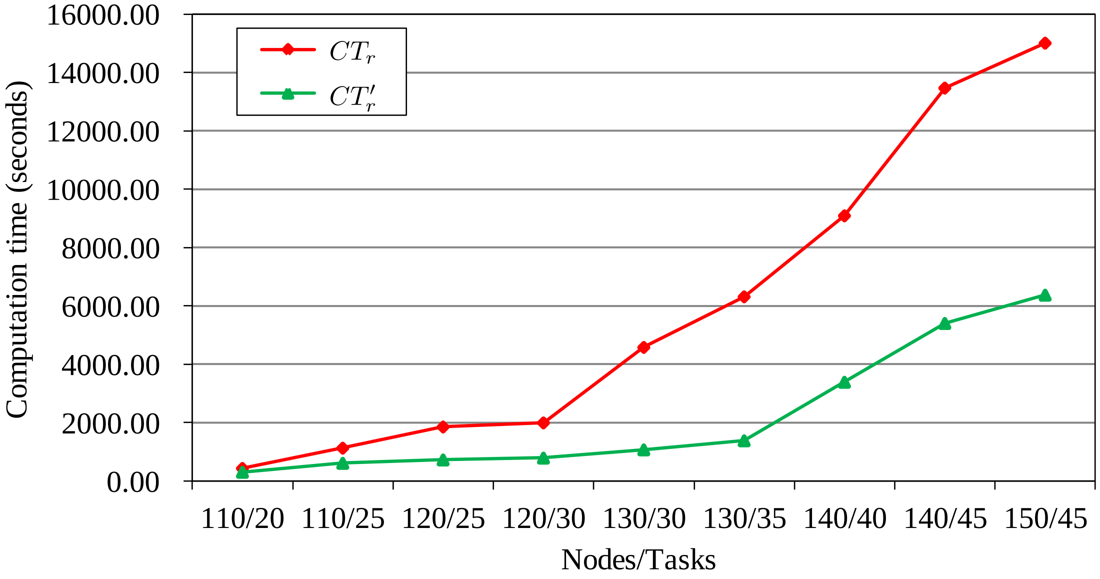


Fig. 4.8: Comparison results for the instances with $|N| = 110$-150

4.5 Conclusions

In this chapter, we have investigated a bi-objective robust lane reservation problem whose optimization goals are to minimize the total impact of reserved lanes and to maximize robustness of the lane-reservation solution. We developed an improved exact ε-constraint and cut-and-solve combined method to find its Pareto front. Computational results on an instance based on a real network topology and 220 randomly generated instances showed the efficiency of the proposed approach. In addition, we generalized the exact ε-constraint method that was initially designed for BCOPs with integer objective values. We showed that the enhanced version of the exact ε-constraint method is able to find the Pareto front for the considered bi-objective MILP. The corresponding work in this chapter has been published in the following paper.

Table 4.5: Comparison results for the instances with $|N| = 110\text{-}150$

Set	$\|A\|$	$\|N\|$	$\|K\|$	$\|F\|$	$\|J\|$	$\|J'\|$	$R_p(\%)$	CT_r	CT'_r	CT_r/CT'_r	$CT_r/\|J\|$	$CT'_r/\|J'\|$
35	416	110	20	35.0	35.0	38.0	8.57	303.39	434.08	0.70	8.67	11.42
36	422	110	25	37.2	37.2	42.4	13.98	609.28	1128.65	0.54	16.38	26.62
37	468	120	25	42.4	42.4	48.0	13.21	725.57	1855.68	0.39	17.11	38.66
38	470	120	30	38.0	38.0	44.6	17.37	788.76	1994.51	0.40	20.76	44.72
39	512	130	30	36.8	36.8	42.0	14.13	1071.62	4582.61	0.23	29.12	109.11
40	510	130	35	40.0	40.0	46.8	17.00	1382.67	6316.60	0.22	34.57	134.97
41	554	140	40	43.0	43.0	48.2	12.09	3388.31	9094.26	0.37	78.80	188.68
42	556	140	45	42.2	42.4	52.6	24.17	5404.62	13472.37	0.40	127.47	256.13
43	596	150	45	43.6	44.0	49.0	11.17	6373.66	15011.97	0.42	144.86	306.37
Average				39.8	39.87	45.73	14.74	2227.54	5987.86	0.37	55.87	130.93
44	600	150	50	46	46.2	–	–	9538.81	–	<0.53	206.47	–

Chapter 5

Bus lane reservation problem

5.1 Introduction

In this chapter, we investigate a bus lane reservation problem (BLRP) motivated by improving the performance of bus transit system. As mentioned in Chapter 1, traffic congestion is one of the major challenges all over the world. To alleviate urban traffic congestion, the most direct means: building new roads is restricted by the geographic space, high construction cost and long duration. The development of public transport has been widely accepted as a potential practical solution if it is efficient, effective, reliable and comfortable. Bus transit as one of the oldest public transport modes has great advantages in the high flexibility and low fare [68]. However, bus transit appears to be less attractive due to its inefficient transit caused by heavy traffic congestion, especially in morning and afternoon pear hours. Bus priority strategy emerged in this circumstance has been widely used to improve bus transit service.

The bus lane reservation strategy, to convert some general lanes on some road links (in some time periods, such as peak hours) into be reserved for buses, is an important bus-priority strategy that has been widely employed in our real life. Its main aim is to help achieve time-efficient bus transit to enhance its attractiveness. The major advantage of bus lane reservation is to keep the bus transit from trapping into the congested traffic and provide a congestion-free transit environment on them. Thus, bus transit time-efficient can usually be ensured and better schedule adherence can usually be achieved as well [109]. Despite the potential benefits brought by reserved bus lanes, once they are implemented, they may make non-bus vehicles on non-reserved lanes spend more time, i.e., negative impact will be generated by reserved lanes. Princeton and Cohen [96] concluded that the average travel time on general-purpose lanes was increased up to 26% after a lane of A1 motorway in Paris was reserved. Therefore, it is necessary to optimally reserve bus lane since improper bus lane reservation may worsen the already congested urban traffic instead of improving it.

As stated in Chapter 2, there have been a few studies concerning optimal bus lane reservation at the transportation network level through optimization methods. These studies provide valuable tools for decision-makers to perform optimal bus lane reservation. However, some limitations below exist: 1) all the proposed algorithms were evaluated by only one study case; 2) the size of study cases are limited in relatively small networks; 3) the proposed non-linear programming models are difficult to solve due to their non-linearity nature for large-size instances; and 4) negative impact of bus lanes on normal traffic have not been considered yet.

Different from the existing studies, this chapter studies a new bus lane reservation problem from the perspective of minimizing negative traffic impact of reserved bus lanes and guaranteeing time-efficient bus transit. It intends to optimally choose lanes from an existing bus transit network to be bus lanes for time-guaranteed bus transit. For the considered problem, we suppose that the total travel time of a bus line from its origin to terminal must be completed within a given deadline. This aims to improve the service level of a bus transit system and increase bus transit attractiveness. Different from the previous LRPs addressed in [41], [125] and Chapters 3 and 4, we suppose that a lane on a road segment can be reserved only when the bus volume per unit time on it reaches a given volume level. This assumption attempts to maximize the effectiveness of bus lanes. Similar to the LRPs studied in Chapters 3 and 4, the objective of the BLRP is to minimize the total negative traffic of reserved lanes.

The remainder of this chapter is constructed as follows. In Section 5.2, we first describe the BLRP and formulate an integer linear program for it. Then its complexity is demonstrated. Section 5.3 sketches an optimal algorithm based on cut-and-solve method. Computational results are presented in Section 5.4. This chapter is concluded in Section 5.5.

5.2 Problem formulation

The BLRP is described as follows. A bus transit network can be represented by a graph $G = \{N, A\}$, where N(resp. A) is a set of nodes (resp. arcs). A node represents a road intersection or bus station, and an arc represents a road segment connecting pairs of nodes in the network. Given a set of bus lines K, the BLRP is to select lanes in the transit network to be reserved such that the total travel time on each bus line is completed within given deadline. The reserved bus lanes reduces bus transit time while it may make the adjacent non-reserved lanes more congested. The objective of the BLRP is to minimize the total negative impact of all reserved bus lanes.

The following assumptions are made for the BLRP as follows: 1) bus lines are predetermined and there are at least two lanes on each arc; 2) road segment travel time can be decreased on a reserved lane; and 3) the bus path passed by each bus line can be composed of non-reserved and reserved lanes, i.e., the bus paths can be partially reserved. The parameters and decision variables used for the formulation is given as follows:

Sets and parameters
N: set of nodes, $i \in N$
A: set of arcs, $(i, j), i, j \in N$
L: set of bus lines, $l \in L$
τ_{ij}: travel time on a reserved lane on arc $(i, j) \in A$
τ'_{ij}: travel time on arc $(i, j) \in A$ without reserved lanes
T_l: given travel deadline for the bus line $l \in L$
C_{ij}: negative impact caused by a reserved lane on arc $(i, j) \in A$
f_l: number of buses on the bus line $l, l \in L$
Q_{ij}: threshold of bus volume per unit time for reserving a lane on arc $(i, j) \in A$
S^l_{ij}: $S^l_{ij} = 1$: bus line $l \in L$ passes arc $(i, j) \in A$ and 0 otherwise

Decision variables
z_{ij} : $z_{ij} = 1$, if arc (i, j) is reserved; otherwise $z_{ij} = 0$, $(i, j) \in A$

Then, the BLRP is formulated as the following integer linear program:

$$\mathcal{P}_b : \quad \min \sum_{(i,j)\in A} C_{ij} z_{ij} \tag{5.1}$$

$$\text{s.t.} \sum_{(i,j)\in A} S^l_{ij}(\tau_{ij} z_{ij} + \tau'_{ij}(1 - z_{ij})) \leq T_l, \forall l \in L \tag{5.2}$$

$$\sum_{l\in L} S^l_{ij} f_l \geq z_{ij} Q_{ij}, \forall (i, j) \in A \tag{5.3}$$

$$z_{ij} \in \{0, 1\}, \forall (i, j) \in A \tag{5.4}$$

Objective function (5.1) is to minimize total negative traffic impact of all reserved lanes. Constraint (5.2) represents that the total travel time of the l-th bus line should not exceed its given travel deadline. Constraint (5.3) guarantees that the bus volume on each arc a should exceed a certain level if it is reserved. Constraint (5.4) is a binary constraint on the decision valuables. The above formulation can be further simplified as follows.

Let A' denote the set of arcs on which the bus volume does not exceed a certain level, i.e., for $\forall(i,j) \in A', \sum_{l \in L} S_{ij}^l f_l < Q_{ij}$. Thus, the proposed model $\mathcal{P}_b$ can be easily simplified as the following model $\mathcal{P}_b'$.

$$P_B': \quad \min \quad \sum_{(i,j)\in A} C_{ij} z_{ij}$$

$$\text{s.t. Constraints (5.2) and (5.4)}$$

$$z_{ij} = 0, \forall(i,j) \in A' \tag{5.5}$$

The complexity of the BLRP is shown as follows.

Theorem 7 *The BLRP is NP-hard.*

Proof: First, we will show that the special case of the BLRP, where there is only one bus line, i.e., $|L| = 1$, and Q_{ij} is small enough such that constraint (5.3) can be relaxed, is NP-hard. The special case of the BLRP (we call it SBLRP) that can be represented by the following integer linear program:

$$SP_b: \quad \min \quad \sum_{(i,j)\in A_s} C_{ij} z_{ij}$$

$$\text{s.t.} \sum_{(i,j)\in A_s} (\tau_{ij} z_{ij} + \tau_{ij}'(1 - z_{ij})) \leq TC \tag{5.6}$$

$$z_{ij} \in \{0,1\}, \forall(i,j) \in A_s \tag{5.7}$$

where A_s denotes the set of arcs of the path of the only bus line, and TC denotes its given deadline. Let $|A_s|$ denote the number of arcs in A_s. SP_B can be easily to the following equivalent form.

$$SP_b': \quad \min \quad \sum_{(i,j)\in A_s} C_{ij} z_{ij}$$

$$\text{s.t.} \sum_{(i,j)\in A_s} (\tau_{ij}' - \tau_{ij}) z_{ij} \geq \sum_{(i,j)\in A_s} \tau_{ij}' - TC \tag{5.8}$$

$$z_{ij} \in \{0,1\}, \forall(i,j) \in A_s \tag{5.9}$$

Then, we demonstrate the NP-hardness of SP_b' through a reduction from the 0-1 Knapsack Problem, which is known to be NP-hard [95]. The 0-1 Knapsack Problem (0-1 KP) is defined as follows: there are n items that have to be packed in a knapsack, each item j has an associated *profit* p_j and *weight* $w\ j$, given a knapsack with *capacity* c, the objective is to maximize the total profit. Its integer linear programming model can be constructed as follows:

$$\text{0-1 } KP: \quad \max \quad \sum_{j=1}^{n} p_i x_i \tag{5.10}$$

$$\text{s.t.} \sum_{i=1}^{n} w_i x_i \leq c \tag{5.11}$$

$$x_i \in \{0, 1\}, i = 1, 2, ..., n \tag{5.12}$$

Finally, we explain how to transform 0-1 KP to SBLRP. First, n corresponds to $|A_s|$, $i(i = 1, 2, ..., n)$, p_i, w_i and c correspond to C_{ij}, $\tau'_{ij} - \tau_{ij}$, and $TC - \sum_{(i,j)\in A_s} \tau_{ij}$, respectively. Second, x_i is replaced by $1 - z_{ij}$. With such linear transformation, the 0-1 KP is reduced into the SBLRP.

As analyzed above, the 0-1 Knapsack Problem known to be NP-hard (5.3) is reducible to a special case of the BLRP, i.e., SBLRP. Hence, the BLRP is also NP-hard. □

5.3 Solution approach

In this section, an optimal algorithm based on cut-and-solve method is employed to solve the BLRP. The principle of cut-and-solve method has been presented in Chapter 2. As stated previously, it is important to define the *piercing cut* (PC^n), *sparse problem* (SP^n) and *residual problem* (RP^n) when applying the cut-and-solve method. In the following, these main components designed for the studied BLRP are detailed, respectively.

5.3.1 Definition of piercing cut, sparse problem and residual problem

For a cut-and-solve method, an appropriate PC^n is critical. It combines a set of decision variables U_n. U_n is defined according to the *reduced costs* of valuables [30], which is composed of valuables whose reduced costs are greater than a given positive value, denoted as α^n. Due to that the variables in U_n are all binary, the sum of them is either greater than 1 or equal to 0. With such a piercing cut, CP^n is separated into RP^n (with the sum of all valuables in U_n is greater than 1) and SP^n (with the sum of all valuables in U_n is equal to 0).

As indicated in [30], the way defining PC^n above achieved good performance for solving the ATSP. Fang et al. [42], [45] also adopted such a PC^n to solve LRP. Moreover, the objective of the BLRP is directly related with valuables $z_{ij}, (i, j) \in A$ like that in [30]. This inspires us to use reduced costs of variables to define the piercing cut. Therefore, U_n is defined as follows:

$$U_n = \{z_{ij} | \varphi(z_{ij}) > \alpha^n, \forall (i, j) \in A\} \tag{5.13}$$

where $\varphi(z_{ij})$ is the reduced cost of z_{ij} and the choice of the value of α^n depends on the distribution of reduced cost of z_{ij}, e.g., $0.1 \times \max\{\varphi(z_{ij})|\forall(i,j) \in A$ as like that in Fang et al. [42]. The reduced costs of variables $z_{ij}, (i,j) \in A$, are obtained by solving the linear relaxation problem of the *current problem* CP^n. The PC^n is defined as follows:

$$(PC^n) \sum_{z_{ij} \in U_n} \geq 1 \tag{5.14}$$

With the piercing cut PC^n, two sub-problems SP^n and RP^n are defined as follows, respectively.

$$\begin{aligned}
(SP^n) \quad \min \quad & \sum_{(i,j)\in A} C_{ij} z_{ij} \\
\text{s.t. Constraints } & (5.2), (5.4) \text{ and } (5.5) \\
& \sum_{z_{ij} \in U_t} z_{ij} \geq 1, t = 1, 2, ..., n-1 && (5.15) \\
& \sum_{z_{ij} \in U_n} z_{ij} = 0 && (5.16) \\
(RP^n) \quad \min \quad & \sum_{(i,j)\in A} C_{ij} z_{ij} \\
\text{s.t. Constraints } & (5.2), (5.4), (5.5), (5.14), (5.15)
\end{aligned}$$

5.3.2 Improved piercing cut technique

In order to further improve the performance of the algorithm, the improved piercing cut PC'_n proposed by Fang et al. [45] based on the basis of PC^n is adapted here. A new definition of $U_n (n \geq 2)$ used for the definition of PC'_n, denoted as U'_n, is given as follows:

$$U'_n = \{z_{ij} | \varphi(z_{ij}) > \alpha^n, z_{ij} \in U'_{n-1}, \forall (i,j) \in A\} \tag{5.17}$$

When $n = 1, U'_1 = U_1$.

$$(PC'_n) \sum_{z_{ij} \in U'_n} z_{ij} \geq 1 \tag{5.18}$$

With PC'_n, two new sub-problems $SP'_n (n \geq 2)$ and $RP'_n (n \geq 2)$ can be defined as follows:

$$\begin{aligned}
(SP'_n) \quad \min \quad & \sum_{(i,j)\in A} C_{ij} z_{ij} \\
\text{s.t. Constraints } & (5.2), (5.4) \text{ and } (5.5) \\
& \sum_{z_{ij} \in U'_{n-1} \setminus U'_n} z_{ij} \geq 1 && (5.19)
\end{aligned}$$

Algorithm BLRP

```
1: Initialize $n = 1, UB_b = +\infty,$ and $CP_1 = \mathcal{P}'_b$.
2: Solve $CP's$ linear relaxation problem to obtain the reduced costs of all variables.
3: Define $PC'_n$ by (5.18) and obtain $SP'_n$ and $RP'_n$.
4: Solve problem $SP'_n$ exactly to obtain $UB_n$. If $UB_n < UB_b$, $UB_b = UB_n$.
5: Solve the linear relaxation problem of $RP'_n$ to obtain $LB_n$. If $LB_n \geq UB_b$, output
   $UB_b$ and the corresponding solution as the optimal objective function value and
   the optimal solution, respectively, and end;
6: Set $CP_{n+1} = RP'_n, n = n + 1$, go back to Step 2.
```

Fig. 5.1: Algorithm BLRP: algorithm for the BLRP

$$\sum_{z_{ij} \in U'_n} z_{ij} = 0 \tag{5.20}$$

$$(RP'_n) \quad \min \quad \sum_{(i,j) \in A} C_{ij} z_{ij}$$

$$\text{s.t. Constraints } (5.2), (5.4), (5.5), (5.18)$$

When $n = 1, SP'_1 = SP_1$, and $RP'_1 = RP_1$.

Theorem 8 *For $n \geq 2$, if $U'_1 \supseteq U'_2 \supseteq \cdots U'_{n-1} \supseteq U'_n$ holds, SP'_n and RP'_n are equal to SP^n and RP^n, respectively.*

Proof: Its correctness can be proved similar to Theorem 3 in [45], for more details, please see [45]. □

The overall algorithm for the BLRP is summarized as Algorithm BLRP.

5.4 Computational results

This section reports experiment computational results on 15 randomly generated problems sets with five instances in each, i.e., 75 instances in total, tested on a PC with 2.5 GHz CPU and 2.95 GB RAM. The proposed method is coded in C++ embedded with the optimization software CPLEX (version 12.4) in default setting used to optimally solve SP'_n and the relaxation problem of RP'_n. Its performance is evaluated by comparing computation time (CPU seconds) with CPLEX. The instances are randomly generated as follows.

The graph $G = (N, A)$ is generated based on the network model proposed by [121]. The existence of an arc for a pair of nodes depends on a probability function associated with the distances of nodes. The parameters of link travel times τ'_{ij}, τ_{ij}, and impact of lane reservation C_{ij} are estimated as those in Chapter 4. To avoid a trivial problem, the deadline T_l is randomly and uniformly generated between the total travel time on an entire reserved bus path and on a non-reserved path. Integer f_l is randomly generated in [5, 12], which means that bus service frequency ranges from 5 to 12 minutes, i.e., 5 to 12 vehicles/hour. As pointed out by Seo et al. [102], a reserved lane with a bus volume in the range of 20-400 vehicles/ hour is useful, so Q_{ij} is set as 20 vehicles/hour, which is the lowest level. Let CT_b, and CT_0 denote the computation time spent by the proposed algorithm and CPLEX, respectively. The computational results are summarized in Tables 5.1 and 5.2, and Fig.'s 5.2 and 5.3.

Table 5.1: Comparison results for the instances with $|N|$ = 50-90

Set	$\|N\|$	$\|L\|$	CT_b	CT_0	CT_b/CT_0
1	50	20	0.37	0.39	0.94
2	60	20	1.19	1.50	0.79
3	70	20	1.41	1.62	0.87
4	80	20	1.75	2.07	0.85
5	90	20	3.57	4.20	0.85
Average			1.66	1.96	0.85

Table 5.1 reports the computational results for instances with 20 fixed number of bus lines and number of nodes increasing from 50 to 90. We can see from Table 5.1 that both the proposed algorithm and CPLEX can exactly solve all instances within relatively short computation time. The computation time by the proposed algorithm, i.e., CT_b is less than CT_0, for all sets 1-5. CT_0 increases from 0.39s to 4.2s, and its average value is 1.96s, whereas the CT_b increases from 0.37s to 3.57s, and its average value is 1.66s. We can also see that the value of ratio CT_b/CT_0 changes between 0.79 and 0.94, and its average value is 0.85. This indicates that the proposed algorithm only spends 85% average computation time of that spent by CPLEX. This shows that the proposed algorithm is more efficient than CPLEX.

In addition, it can be seen from Table 5.1 and Fig. 5.2 that CT_b and CT_0 both increase with the number of nodes $|N|$, while CT_b increases more gradually than CT_0. This also indicates that the proposed algorithm is more efficient than CPLEX for instances with fixed number of bus lines $|L|$ and varying number of nodes $|N|$.

Table 5.2 presents the comparison results of the proposed algorithm and CPLEX for instances with $|N|$ increasing from 100 to 500 and $|L|$ varying from 20 to 60. It can be found from Table 5.2 that both methods can obtain optimal solutions for all

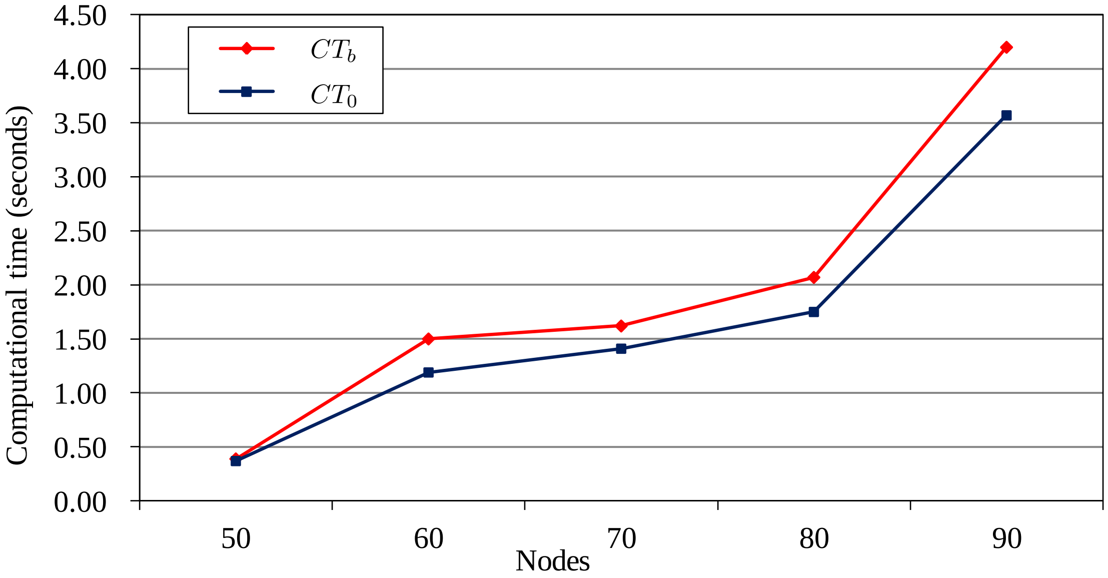


Fig. 5.2: Comparison results for the instances with $|N| = 50$-90

Table 5.2: Comparison results for the instances with $|N| = 100$-500

Set	$\|N\|$	$\|L\|$	CT_b	CT_0	CT_b/CT_0
6	100	20	4.07	5.56	0.73
7	100	25	14.57	16.68	0.87
8	200	30	20.52	24.44	0.84
9	200	35	197.28	236.57	0.83
10	300	40	228.11	286.11	0.80
11	300	45	379.84	471.16	0.81
12	400	50	537.99	841.95	0.64
13	400	55	1182.38	1761.56	0.67
14	500	60	1438.45	1877.62	0.77
Average			444.80	613.52	0.73

instances in 1900s. The computation time spent by CPLEX increases from 5.56s to 1877.62s and its average value for all instances is 613.52s, whereas the computation time spent by the proposed cut-and-solve algorithm increases from 4.07s to 1438.45s and its average value for all instances is 444.8s. We can find that CT_b is less than CT_0 over all sets 6-14 in Table 5.2. In addition, it can be seen in Table 5.2 that CT_b/CT_0 ranges between 0.64 and 0.87, and its average value for all the sets is 0.73. This means that the proposed algorithm can find optimal solution by spending 73% the computational time of that spent by CPLEX for all the instances on the average. This indicates that the proposed algorithm is more efficient than CPLEX.

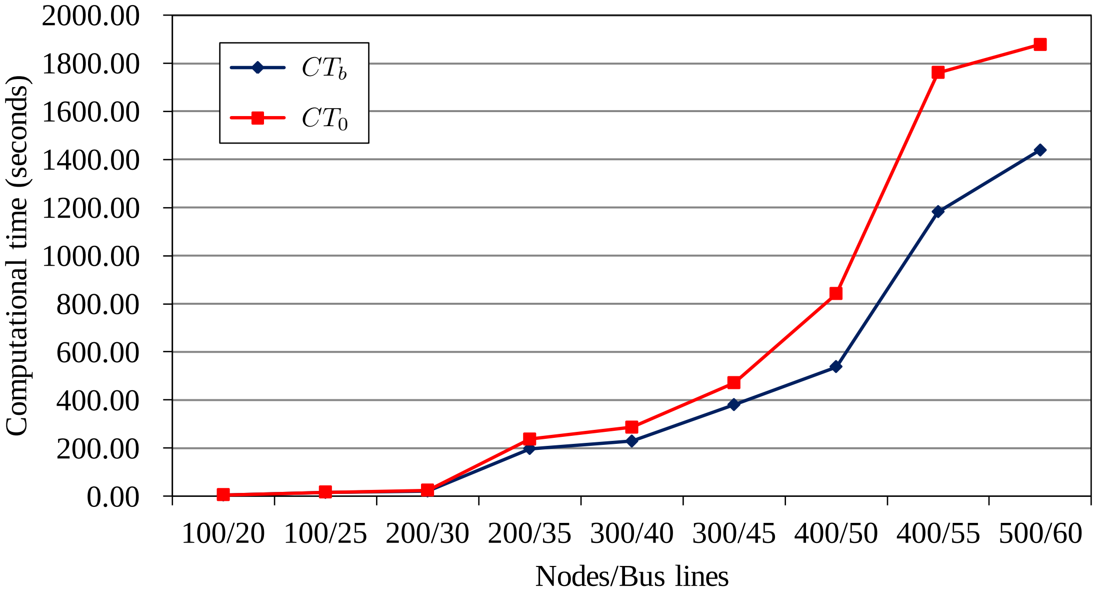


Fig. 5.3: Comparison results for the instances with $|N| = 100$-500

Moreover, from Table 5.2 and Fig. 5.3, we can find that the computational time by the proposed algorithm and CPLEX both increases with the number of bus lines $|L|$, but CT_b increases more gradually than CT_0 from sets 6-11. Note that CT_b increases a bit faster than CT_0 from sets. This may because that the solution space for these larger-size problem sets is larger and the generated piercing cuts in the proposed algorithm may be not very efficient that influences the convergence of the algorithm. Although so, it can be found that the computation time spent by the proposed algorithm is less than that spent by CPLEX for the larger-size problem sets 12-14.

5.5 Conclusions

In this chapter, we investigate a new bus lane reservation problem. It is to optimally choose some existing general lanes in a bus transit network and convert them into bus lanes via lane reservation so that the total travel time of buses on each bus line is less than a given deadline. This aims to improve the service level of a bus transit system. Meanwhile, the bus volume on a reserved lane should exceed a certain bus volume level to maximize its effectiveness. The bus lane reservation problem aims to minimize the impact caused by reserved lanes. An integer linear programming model was formulated and the problem was demonstrated to be NP-hard. Then, a cut-and-solve algorithm was adapted to exactly solve the BLRP. The computational results on randomly generated instances show that the proposed algorithm is more efficient than the commercial optimization software CPLEX.

Chapter 6

Bus lane reservation problem with path design

6.1 Introduction

One common characteristics of all the existing studies on optimal bus lane reservation at the macroscopic network level [68], [76], [86], [87], [108], [131] as well as the BLRP studied in Chapter 5 assume that the bus paths are predetermined, and thus the lanes to be reserved are only selected from the known bus paths. Clearly, the proposed theories and methods in the above studies cannot be directly applied to a bus lane reservation problem in which the bus transit paths need to be optimally determined. Moreover, although guaranteeing bus station arrival-time is one of the most important indicators for evaluating a bus transit system, all previous studies do not consider such issue. As reported by [7], [109], rapidness and reliability have been widely recognized as the key measures for the service quality of a bus transit system as well as passenger satisfaction that justifies a bus lane reservation scheme.

Achieving the station arrival-time guaranteed bus transit will improve the service level of a bus transit system because the setting and guaranteeing of arrival time of bus at station will clearly reduce the travel time of passengers and avoids or significantly reduces the excess waiting time. Moreover, as pointed out by [130], when the arrival times at bus stations are unreliable or not guaranteed, bus passengers, especially commuters need to accommodate much extra time to their schedules for arriving on time. The station arrival-time guaranteed bus transit improves bus transit reliability such that such extra time could be avoided or reduced. On the other hand, for the bus operators, station arrival-time guaranteed bus transit will not only help them develop reliable timetable but also reduce the bus operating costs since the total travel time on each bus line is reduced. Because the bus arrival-times at stations usually are not guaranteed due to the increasingly congested urban traffic environment, appropriate bus lane reservation may achieve this in a flexible and economic way.

In this chapter, we investigate a new bus lane reservation problem in which the bus paths need to be designed (BLRP-PD). Compared with the existing studies, the assumption that bus paths are predetermined is relaxed and guaranteeing bus station arrival-time issue is considered. The BLRP-PD aims to optimally select lanes to be reserved for bus use in a transportation network and determine the bus path for each bus line such that the rapid and station arrival-time guaranteed bus transit can be ensured, thereby achieving rapid and reliable bus transit service. But negative impact on non-bus vehicles may be caused by bus lanes. The objective is to minimize the total negative impact of reserved lanes. Due to the introduction of bus path design, the resulted new bus lane reservation problem become much more difficult to solve. The existing solution approaches cannot be directly applied to solving it. An exact enhanced cut-and-solve based method is developed for it. New piercing cut strategy and acceleration technique are developed for the CS method. Moreover, to faster solve large-size problems, a kernel search based heuristic is developed to yield optimal or near-optimal solutions. Improvement techniques are developed for the KS algorithm according to the characteristics of the problem.

The remainder of this chapter is organized as follows. In Section 6.2, we present a problem description and its two formulations. The solution approaches are described in Section 6.3. Section 6.4 summarizes and discuss the computational results. Finally, Section 6.4 concludes this chapter.

6.2 Problem formulation

The BLRP-PD is described as follows. Let a directed graph G=$\{N$,$A\}$ represent an urban road network, where N and A are the set of nodes representing road intersections or bus stations, and arcs connecting pairs of nodes, respectively. Let l, L, and N_l denote the l-th bus line, the set of bus lines, and the set of the bus stations on bus line l, respectively. We have $l \in L$, and $N_l \subseteq N$. Given a set of bus lines and their corresponding bus stations, the BLRP-PD consists of optimally selecting lanes in the network to be reserved for exclusive use of buses and designing bus paths such that rapid and station arrival-time guaranteed bus transit is ensured. However, bus travel time is reduced on bus lanes but the travel time of non-bus road users on adjacent non-reserved lanes may be increased, i.e., negative impact may be caused by bus lanes. The objective is to minimize the total negative impact generated by all the bus lanes.

To well study the problem, some assumptions are made as follows: 1) transportation network layout and bus stations and the passing order of stations on each bus line are assumed to be known; 2) there exist at least two lanes on each road segment such that one lane can be reserved; and 3) the capacity of a reserved bus lane is assumed to be large enough and a bus lane can be shared by multiple bus lines to increase its effectiveness. In addition, the negative impact caused by a reserved lane is defined by Definition 8.

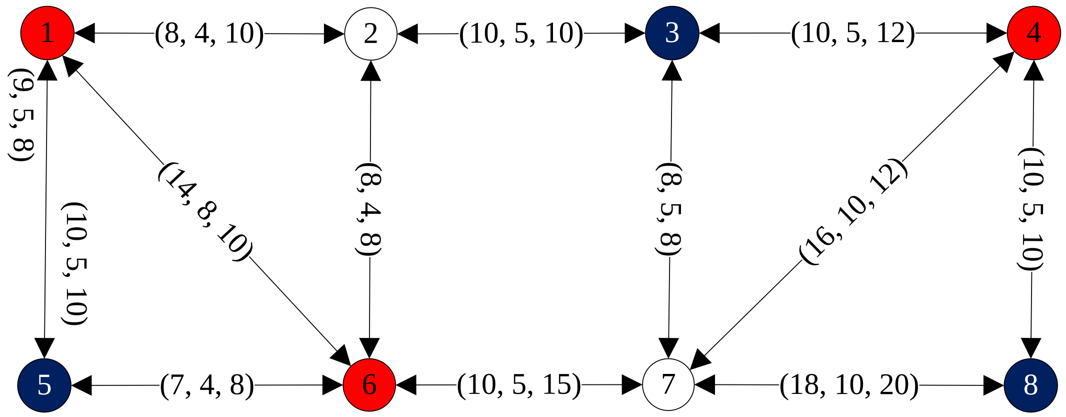
Definition 8 *For arc $(i, j) \in A$, the impact caused by a reserved lane on this arc is defined as the total increase of travel time of all non-bus users on its non-reserved lanes caused by this reserved lane (i.e., $P_{ij}(\tau''_{ij} - \tau'_{ij})$), where P_{ij} and τ''_{ij} are estimated number and travel time of of non-bus users on arc $(i, j) \in A$ with a reserved lane, respectively.*

Theorem 9 *The BLRP-PD is NP-hard.*

Proof: If there is only origin and terminal station (i.e., only two stations) for each bus transit line and the path is required to be entirely reserved, then such a special case of the BLRP is reducible to the LRP in Chapter 3 that is proved to be NP-hard. Therefore, the BLRP is also NP-hard. □

To clearly show the optimal solution obtained for the considered BLRP-PD, a small example, denoted by Example 1, is presented and analyzed which allows for easy inspection of the optimal solution. This instance consists of two bus transit lines with three stations for each in a simple road network with 8 nodes and 24 arcs, as shown in Fig. 6.1. Each arc is characterized by a three-tuple (τ', τ, C), where τ', τ and C are the bus travel time without reserved lanes, the bus travel time on a reserved lane, and the negative impact of a reserved lane on the arc, respectively. For instance, from node 1 to 2, (8, 4, 10) means that the bus travel time without reserved lanes, the bus travel time on a reserved lane, and the negative impact of a reserved lane on the arc are 8 unit time, 4 unit time, and 10 unit impact, respectively.

In general, buses usually travel between two consecutive stations on its shortest path in order to ensure the bus transit efficiency. Consequently, the bus paths of the two bus lines are 1→ 6 → 7 → 4 and 5 → 6 → 7 → 3 → 4 → 8, respectively. The bus arrival times at stations 6 and 4 (resp. 3 and 8) for bus line 1 (resp. bus line 2) are 15 and 42 (resp. 26 and 47) including both in-vehicle and bus dwell times, respectively. The bus transit network without reserved lanes is shown in Fig. 6.2.



The bus lines 1 and 2 sequentially involve stations 1, 6, 4 and 5, 3, 8, respectively. The arrival times at bus stations 6 and 4 for bus line 1 (resp. 3 and 8) are guaranteed to be less than or equal to 10 and 34 (resp. 20 and 35), respectively.

Fig. 6.1: Network of Example 1

The arcs marked in red (resp. blue) are passed by bus line 1 (resp. bus line 2) and arcs marked in green are passed by both bus lines. We can find that the required bus arrival times at stations are not satisfied. we intend to optimally select lanes to be reserved lanes and design reserved lane based path for each bus line to achieve the required rapid and station arrival-time guaranteed bus service.

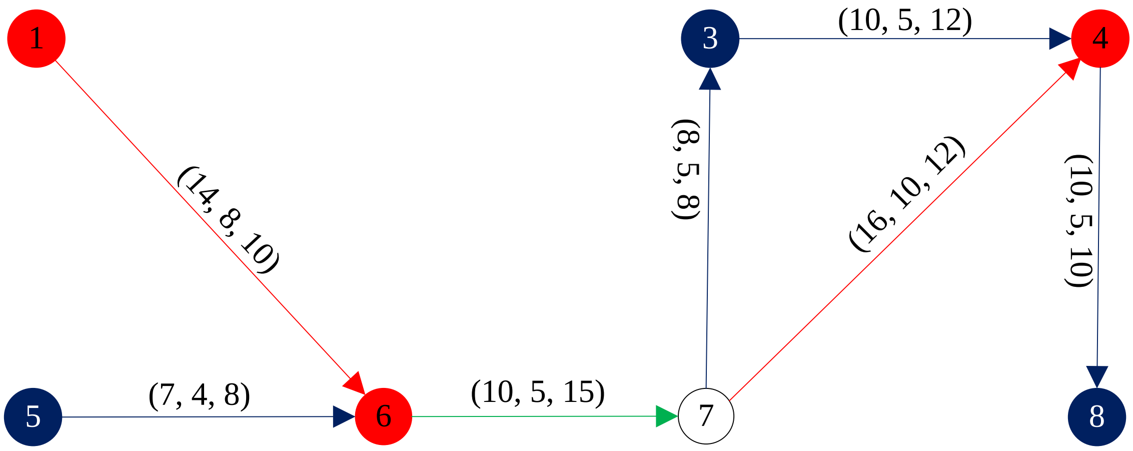


Fig. 6.2: Bus paths without reserved lanes

For Example 1, the outputs of the problem include: 1) an optimal lane reservation scheme is (1, 2), (2, 3), (2,6) and (3, 4); 2) the bus arrival-time guaranteed paths for bus lines 1 and 2 with reserved lanes are 1→ 2 → 6 → 7 → 3 → 4 and 5 → 1 → 2 → 3 → 4 → 8, respectively; and 3) the minimal negative impact of reserved lanes is 40. The arrival times at stations 6 and 4 (resp. 3 and 8) for bus transit line 1 (resp.

bus line 2) are 9 and 33 (resp. 19 and 35), respectively, which satisfy the required the rapid and arrival-time guaranteed bus transit needs. In addition, the average arrival time at station for lines 1 and 2 are reduced 30.72% and 31.19%, respectively. The bus transit network with reserved lanes is shown in Fig. 6.3. Note that arcs marked in red (resp. blue) are passed by bus line 1 (resp. bus line 2) and reserved lanes (1, 2) and (3, 4) marked in green are shared by both bus lines.

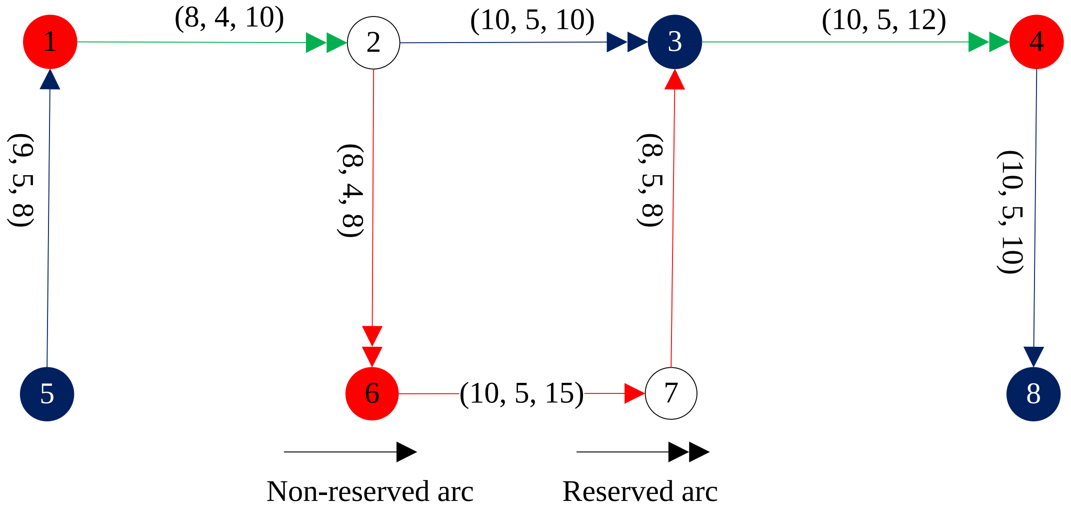


Fig. 6.3: Bus paths with reserved lanes

By comparing the results in Fig. 6.2 with Fig. 6.3, we can observe that the bus paths for both bus transit lines without and with reserved lanes are different. Besides, the objective of minimizing the negative impact may lead to shared reserved lanes so as to reduce their negative impact.

6.2.1 MIP formulation

To formulate the problem, the input parameters and decision variables are given and defined as follows.

Sets and parameters

N: set of nodes, $i \in N$

A: set of arcs, $(i, j), i, j \in N$

L: set of bus lines, $l \in L$

N_l: set of bus stops included in bus line l, $l \in L$;

s_l: start stop of bus line $l \in L, s_l \in N_l$

d_l: terminal stop of bus line $l \in L, d_l \in N_l$

τ_{ij}: travel time on a reserved lane on arc $(i, j) \in A$

τ'_{ij}: travel time on arc $(i, j) \in A$ without reserved lanes

T_0: departure time of buses at start stops
C_{ij}: negative impact caused by a reserved lane on arc $(i,j) \in A$
$T^-_{i,l}$: lower bound on arrival time at the i-th stop on bus line l, $i \in \{2,..,|N_l|\}$, $l \in L$;
$T^+_{i,l}$: upper bound on arrival time at the i-th stop on bus line l, $i \in \{2,..,|N_l|\}$, $l \in L$;

Decision variables
t^l_i: arrival time on bus line l at node i; t^l_i is set as 0 if node i is not passed, $\forall i \in N$, $\forall l \in L$;
z_{ij}: $z_{ij} = 1$, if a lane is reserved on arc (i,j); and otherwise $z_{ij} = 0$, $\forall (i,j) \in A$;
x^l_{ij}: $x^l_{ij} = 1$, if the path of bus line $l \in L$ passes arc (i,j) and a lane on this arc is reserved for buses; and otherwise $x^l_{ij} = 0$, $\forall (i,j) \in A$;
y^l_{ij}: $y^l_{ij} = 1$, if the path of bus line $l \in L$ passes arc (i,j) and no lanes on this arc are reserved for buses; and otherwise $y^l_{ij} = 0$, $\forall (i,j) \in A$;

With the notations, definition and assumptions given above, the BLRP-PD can be formulated as the following mixed-integer program $\mathcal{P}_1$.

$$\mathcal{P}_1: \quad \min \sum_{(i,j)\in A} C_{ij} z_{ij} \tag{6.1}$$

$$\text{s.t.} \sum_{j:(i,j)\in A} (x^l_{ij} + y^l_{ij}) = 1, i \neq d_l, \forall i \in N_l, \forall l \in L \tag{6.2}$$

$$\sum_{i:(i,j)\in A} (x^l_{ij} + y^l_{ij}) = 1, j \neq s_l, \forall j \in N_l, \forall l \in L \tag{6.3}$$

$$\sum_{i:(i,j)\in A} \left(x^l_{ij} + y^l_{ij}\right) = \sum_{i:(j,i)\in A} \left(x^l_{ji} + y^l_{ji}\right), j \in N \backslash N_l, \forall l \in L \tag{6.4}$$

$$t^l_{s_l} = T_0, \forall l \in L \tag{6.5}$$

$$t^l_i = \sum_{j:(j,i)\in A} (x^l_{ji}(t^l_j + \tau_{ji}) + y^l_{ji}(t^l_j + \tau'_{ji})), \forall i \in N, \forall i \neq s_l, \forall l \in L \tag{6.6}$$

$$T^-_{i,l} \leq t^l_i - t^l_{s_l} \leq T^+_{i,l}, \forall i \neq s_l, \forall i \in N_l, \forall l \in L \tag{6.7}$$

$$x^l_{ij} \leq z_{ij}, \forall (i,j) \in A, \forall l \in L \tag{6.8}$$

$$y^l_{ij} + z_{ij} \leq 1, \forall (i,j) \in A, \forall l \in L \tag{6.9}$$

$$t^l_i \geq 0, \forall i \in N, \forall l \in L \tag{6.10}$$

$$x^l_{ij}, y^l_{ij} \in \{0,1\}, \forall (i,j) \in A, \forall l \in L \tag{6.11}$$

$$z_{ij} \in \{0,1\}, \forall (i,j) \in A \tag{6.12}$$

Objective function (6.1) is to minimize the total negative impact caused by reserved lanes. Constraints (6.2)-(6.4) ensure a feasible bus transit path for each bus line $l \in L$.

To be more specific, constraint (6.2) (resp. constraint (6.3)) guarantees that there is one and only one arc outgoing from (resp. entering into) all bus stops except start stop s_l (resp. terminal stop d_l) for bus line $l, \forall l \in L$. The intermediate nodes on bus line $l, \forall l \in L$, is guaranteed by flow conservation constraint (6.4). Constraint (6.5) represents the departure time at start bus stop. Constraint (6.6) indicates the arrival time at node i on bus line $l, \forall l \in L$. Constraint (6.7) guarantees the arrival time at bus stops on each bus line $l, \forall l \in L$. Constraint (6.8) ensure that the path of bus line l cannot pass a reserved lane on arc (i, j) if this arc is not reserved. Constraint (6.9) implies that z_{ij} and y_{ij}^l cannot both take the value of 1. According to their definitions, if $y_{ij}^l = 1$, then bus line l passes arc (i, j) and this arc is not reserved, and otherwise bus line l can pass a reserved one; consequently $z_{ij} = 0$. Constraints (6.10)-(6.12) impose the bounds of decision variables.

6.2.1.1 Model linearisation

It is not difficult to find that the above MIP $\mathcal{P}_1$ is non-linear due to the existence of the non-linear constraint (6.6). In this subsection, we will transform non-linear model $\mathcal{P}_1$ into an equivalent linear one by reformulating constraint (6.6). For any node $i \in N, i \neq s_l$ and $\forall l \in L$, there exist two cases to compute the arrival time at node i, which are (a) the path of bus line l passes node i; and (b) the path of bus line l does not pass node i. The reformulation based on the above analysis is given as follows.

a) The path of bus line l passes node i. This case further includes two subcases, i.e., bus line l passes node i via a non-reserved lane or reserved lane. For the former subcase, constraint (6.6) can be reformulated as follows:

$$\sum_{j:(j,i)\in A} (y_{ji}^l(t_i^l - t_j^l - \tau'_{ji})) = 0, \forall i \neq s_l, \forall l \in L \tag{6.13}$$

With (6.13), it can found that each item $y_{ji}^l(t_i^l - t_j^l - \tau'_{ji})$ in (6.13) is equal to 0. Thus, we can easily obtain the following equation.

$$y_{ji}^l(t_i^l - t_j^l - \tau'_{ji}) = 0, \forall (j, i) \in A, \forall i \neq s_l, \forall l \in L \tag{6.14}$$

From (6.14), we can find that if $y_{ji}^l = 1$, then $t_i^l - t_j^l - \tau'_{ji} = 0$; if $y_{ji}^l = 0$, (6.14) is still satisfied. Thus, constraint (6.6) can represented using the following two inequalities.

$$t_i^l - t_j^l - \tau'_{ji} \geq M(y_{ji}^l - 1), \forall (j, i) \in A, \forall i \in N, \forall i \neq s_l, \forall l \in L \tag{6.15}$$

$$t_i^l - t_j^l - \tau'_{ji} \leq M(1 - y_{ji}^l), \forall (j, i) \in A, \forall i \in N, \forall i \neq s_l, \forall l \in L \tag{6.16}$$

Similarly, for the later subcase, i.e., bus line l passes node i via a reserved lane, constraint (6.6) can be reformulated using the following two inequalities:

$$t_i^l - t_j^l - \tau_{ji} \geq M(x_{ji}^l - 1), \forall (j,i) \in A, \forall i \in N, \forall i \neq s_l, \forall l \in L \tag{6.17}$$

$$t_i^l - t_j^l - \tau_{ji} \leq M(1 - x_{ji}^l), \forall (j,i) \in A, \forall i \in N, \forall i \neq s_l, \forall l \in L \tag{6.18}$$

b) The path of bus line l does not pass node i. Thus, $\sum_{j:(j,i)\in A} x_{ji}^l = 0$ and $\sum_{j:(j,i)\in A} y_{ji}^l = 0$. Then, constraint (6.6) can be reformulated as follows:

$$t_i^l \leq M \sum_{j:(j,i)\in A} (x_{ji}^l + y_{ji}^l), \forall i \in N, \forall i \neq s_l, \forall (j,i) \in A, \forall l \in L \tag{6.19}$$

With the analysis above, non-linear constraint (6.6) can be equivalently replaced by constraints(6.15)-(6.19). Then, the non-linear model $\mathcal{P}_1$ can be transformed into the following mixed-integer linear program $\mathcal{P}_1'$.

$$\begin{aligned} \mathcal{P}_1': \quad \min \quad & \sum_{(i,j)\in A} C_{ij} z_{ij} \\ \text{s.t. Constraints } & (6.2) - (6.5), (6.7) - (6.12), (6.15) - (6.19) \end{aligned}$$

6.2.1.2 Property analysis of MIP formulation

In order to more efficiently solve the considered problem, a preprocessing is proposed to reduce the search space for its optimal solutions and several additional constraints are added to tighten $\mathcal{P}_1$ as well.

a) Preprocessing

Note that if the travel time on each road segment in the network is known, the shortest travel time between any two nodes in the network can be easily calculated by the well-known Floyd shortest path algorithm with the complexity of $O(N^3)$. Let $\varphi(i,j)$ denote the shortest travel from node i to j when the network is entirely reserved, i.e., each road link has a bus lane. Then, for each bus line $l \in L$, two sets $\mathcal{N}_{s_m^l, s_{m+1}^l}$ and $\mathcal{A}_{s_m^l, s_{m+1}^l}$, $0 \leq m \leq |N_l| - 1$ are defined as follows.

$$\mathcal{N}_{s_m^l, s_{m+1}^l} = \{i | \varphi(s_m^l, i) + \varphi(i, s_{m+1}^l) > T^+_{s_{m+1}^l, l} - T^-_{s_m^l, l}, \forall i \in N\}, \\ \forall m \in \{1, ..., |N_l| - 1\}, \forall l \in L \tag{6.20}$$

$$\mathcal{A}_{s_m^l, s_{m+1}^l} = \{(i,j) | \varphi(s_m^l, i) + \tau(i,j) + \varphi(j, s_{m+1}^l) > T^+_{s_{m+1}^l, l} - T^-_{s_m^l, l}, (i,j) \in A\}, \\ \forall m \in \{1, ..., |N_l| - 1\}, \forall l \in L \tag{6.21}$$

where s_m^l and s_{m+1}^l represent two adjacent stops on bus line l. Note that s_1^l and $s_{|N_l|}^l$ are equivalent to the start s_l and the terminal d_l, respectively. With (6.20) and (6.21), we can obtain two new sets shown as follows:

$$\mathcal{N}_l = \{i | i \in \bigcap_{m\in\{1,\ldots,|N_l|-1\}} \mathcal{N}_{s_m^l, s_{m+1}^l}\}, \forall l \in L \tag{6.22}$$

$$\mathcal{A}_l = \{(i,j) | (i,j) \in \bigcap_{m\in\{1,\ldots,|N_l|-1\}} \mathcal{A}_{s_m^l, s_{m+1}^l}\}, \forall l \in L \tag{6.23}$$

It can be easily found that the nodes in $\mathcal{N}_l$ would not be passed by bus line l because if they are used the arrival time constraint at bus stop will be violated. Similarly, the arcs in $\mathcal{A}_l$ would not be passed by bus line l as well. Based on the above analysis, the corresponding variables t_i^l, x_{ij}^l and y_{ij}^l must be equal to 0, i.e., their values can be fixed as 0. Obviously, the search space for optimal solutions of the problem can be reduced. The corresponding constraints are represented as follows:

$$t_i^l = 0, \forall i \in \mathcal{N}_l, \forall l \in L, \tag{6.24}$$

$$\sum_{(i,j)\in A} (x_{ij}^l + y_{ij}^l) = 0, \forall (i,j) \in \mathcal{A}_l, \forall l \in L \tag{6.25}$$

b) Additional constraints

For each bus line $l \in L$, if it passes node $i \in N \backslash \{s_l\}$, the earliest and latest arrival time at i, t_i^l, are $T_0 + \varphi(s_l, i)$ and $T_{d_l,l}^+ - \varphi(i, d_l)$, respectively; and otherwise $t_i^l = 0$ according to its definition. Thus, two valid inequalities shown as follows hold without excluding feasible solutions.

$$t_i^l \leq (T_{d_l,l}^+ - \varphi(i, d_l)) \sum_{j:(j,i)\in A} (x_{ji}^l + y_{ji}^l), \forall i \in N \backslash \{s_l\}, \forall l \in L \tag{6.26}$$

$$t_i^l \geq (T_0 + \varphi(s_l, i)) \sum_{j:(j,i)\in A} (x_{ji}^l + y_{ji}^l), \forall i \in N \backslash \{s_l\}, \forall l \in L \tag{6.27}$$

Besides, we have the following three additional constraints.

$$\sum_{i:(i,j)\in A} (x_{ij}^l + y_{ij}^l) = 0, j = s_l, \forall l \in L \tag{6.28}$$

$$\sum_{j:(i,j)\in A} (x_{ij}^l + y_{ij}^l) = 0, i = d_L, \forall l \in L \tag{6.29}$$

$$\sum_{(i,j)\in A} (\tau_{ij} x_{ij}^l + \tau_{ij}' y_{ij}^l) \leq T_{d_l,l}^+ - T_0, \forall l \in L \tag{6.30}$$

Constraints (6.28) (resp. (6.29)) implies that there exist no entering arcs (resp. outgoing arcs) for start stop s_l (resp. terminal stop d_l), $\forall l \in L$. Constraint (6.30) means that the arrival time at terminal bus stop for line l should not be greater than the largest possible transit duration $T^+_{d_l,l} - T_0$. Note that constraints (6.26)-(6.30) are all additional, but they may tighten the model $\mathcal{P}'_1$ and generally reduce the tree size in its resolution via a branch-and-bound based method which is the main technique used in commercial optimization software like CPLEX. Consequently, they help reduce computational time via an MIP solver, as demonstrated in our computational results in the next subsection. A new model with reduced search space and additional constraints, denoted by $\mathcal{P}'_1$, is given as follows.

$$\mathcal{P}'_1: \quad \min \sum_{(i,j)\in A} C_{ij}z_{ij}$$
$$\text{s.t. Constraints } (6.2)-(6.5),(6.7)-(6.12),(6.15)-(6.19),(6.26)-(6.30)$$

6.2.2 ILP formulation

The following parameters and variables are firstly defined as follows.

Sets and parameters

s_{lq}: the q-th bus stop on bus line $l, s_{lq} \in N_l, l \in L, q \in \{1, ..., |N_l|\}$, s_{l1} and $s_{l|N_l|}$ are the origin and terminal station, respectively

Decision variables

x^l_{qij}: $x^l_{qij} = 1$, if a lane on arc (i, j) is reserved and located on the path between the q-th and the $(q + 1)$-th bus station of bus transit line $l \in L$ passes this arc; and 0 otherwise,$\forall q \in \{1, ..., |N_l| - 1\}$,$\forall (i, j) \in A$;

y^l_{qij}: $y^l_{qij} = 1$, if a lane on arc (i, j) is not reserved and located on the path between the q-th and the $(q + 1)$-th bus station of bus transit line $l \in L$ passes this arc; and 0 otherwise $\forall q \in \{1, ..., |N_l| - 1\}, \forall (i, j) \in A$.

With the notations and variables, the BLRP-PD can also be formulated as the following inter linear program $\mathcal{P}_2$.

$$\mathcal{P}_2: \quad \min \sum_{(i,j)\in A} C_{ij}z_{ij} \tag{6.31}$$

$$\text{s.t.} \sum_{(i,j)\in A} (x^l_{qij} + y^l_{qij}) = 1, i = s_{lq}, \forall q \in \{1, ..., |N_l| - 1\}, \forall l \in L \tag{6.32}$$

$$\sum_{(i,j)\in A} (x^l_{qij} + y^l_{qij}) = 1, j = s_{l,q+1}, \forall q \in \{1, ..., |N_l| - 1\}, \forall l \in L \tag{6.33}$$

$$\sum_{(i,j)\in A} (x^l_{qij} + y^l_{qij}) = 0, j = s_{l1}, \forall l \in L, \tag{6.34}$$

$$\sum_{(i,j)\in A} (x^l_{qij} + y^l_{qij}) = 0, i = s_{l,|N_l|}, \forall l \in L \tag{6.35}$$

$$\sum_{(i,j)\in A}\left(x^l_{qij}+y^l_{qij}\right)=\sum_{(ji)\in A}\left(x^l_{qij}+y^l_{qij}\right),\forall i\in N\backslash N_l,\forall l\in L \tag{6.36}$$

$$\sum_{q=1}^{|N_l|-1}\sum_{(i,j)\in A}(x^l_{qij}+y^l_{qij})\le 1,\forall j\in N,\forall l\in L \tag{6.37}$$

$$\sum_{q=1}^{|N_l|-1}\sum_{(j,i)\in A}(x^l_{qji}+y^l_{qji})\le 1,\forall j\in N,\forall l\in L \tag{6.38}$$

$$T^-_{q',l}\le\sum_{q=1}^{q'-1}\sum_{(i,j)\in A}(\tau_{ij}x^l_{qij}+\tau'_{ij}y^l_{qij})\le T^+_{q',l},q'\in\{2,...,|N_l|\},\forall l\in L \tag{6.39}$$

$$x^l_{qij}\le z_{ij},\forall(i,j)\in A,q\in\{1,...,|N_l|-1\},\forall l\in L, \tag{6.40}$$

$$y^l_{qij}\le 1-z_{ij},\forall(i,j)\in A,q\in\{1,...,|N_l|-1\},\forall l\in L \tag{6.41}$$

$$z_{ij}\in\{0,1\},\forall(i,j)\in A \tag{6.42}$$

$$x^l_{qij},y^l_{qij}\in\{0,1\},\forall(i,j)\in A,\forall q\in\{1,...,|N_l|-1\},\forall l\in L \tag{6.43}$$

Objective function (6.31) is to minimize the total negative impact of all reserved lanes. Constraints (6.32)-(6.38) ensure that there exists a bus path for bus line l, $\forall l\in L$. To be more specific, constraint (6.33) (resp. (6.34)) represents that there is only one outgoing arc from (resp. entering arc into) the bus stations of bus transit line l except terminal stop (resp. origin stop), $\forall l\in L$. Constraint 6.34 (resp. (6.35)) ensures that there are no entering arcs into the origin station (resp. outgoing arcs from the terminal station) for bus line l, $\forall l\in L$. Constraint (6.36) guarantees the flow conservation for intermediate nodes between bus stations of bus transit line l, $\forall l\in L$. Constraints (6.37) and (6.38) ensure that the bus path for bus line l, $\forall l\in L$ pass any node in the network at most once.

Instead of defining an additional arrival time variable t^l_i like the MIP in the above section to represent the bus arrival time window constraint, the arrival time at each bus stop on each bus transit line is guaranteed by a cumulative duration constraint, as represented by constraint (6.39). We note that in constraint (6.38), $\sum_{(i,j)\in A}(\tau_{ij}x^l_{qij}+\tau'_{ij}y^l_{qij})$ computes the travel duration from station s_{lq} to $s_{l,q+1}$, and thus $\sum_{q=1}^{q'-1}\sum_{(i,j)\in A}(\tau_{ij}x^l_{qij}+\tau'_{ij}y^l_{qij})$ calculates the total travel duration from the origin to $s_{lq'}$, $\forall q'\in\{2,...,|N_l|\}$, $\forall l\in L$. Without loss of generality, the start time at origin station for all bus transit lines is set to be 0. Thus, to guarantee the arrival times at bus stations is equivalent to restricting the bus travel duration from the origin to any other bus station for each bus transit line. For instance, for a bus transit line l, the arrival times at the second and third stations are equal to $\sum_{(i,j)\in A}(\tau_{ij}x^l_{1ij}+\tau'_{ij}y^l_{1ij})$ and $\sum_{(i,j)\in A}(\tau_{ij}x^l_{1ij}+\tau'_{ij}y^l_{1ij})+\sum_{(i,j)\in A}(\tau_{ij}x^l_{2ij}+\tau'_{ij}y^l_{2ij})$, respectively. As we will show in our computational results, such a formulation is much more efficient than that by defining arrival time variable.

Constraint (6.40) ensures that the bus transit path from station s_{lq} to $s_{l,q+1}$ can pass a reserved lane on arc (i, j), $(i, j) \in A$, only if one lane of this arc is reserved. Constraint (6.41) implies that z_{ij} and y^l_{qij} cannot both take the value of 1. According to their definitions, if $y^l_{qij} = 1$, then the bus transit path from station s_{lq} to $s_{l,q+1}$ passes arc (i, j) and this arc is not reserved, and otherwise it can pass a reserved one; consequently $z_{ij} = 0$. Constraints (6.42)-(6.43) impose the bounds of decision variables.

6.2.2.1 A tighter model with a valid inequality

Note that if the travel time on each road segment in transportation network is known, the travel time from $\forall i$ to $\forall j, i, j \in N$ can be easily obtained by the existing shortest path algorithm such as Floyd-Warshall shortest path algorithm. Let $\varphi(i, j)$ and T^l_q denote the shortest travel time from node i to j, $i, j \in N$ and the earliest arrival time at station s_{lq} on bus transit line $l, l \in L, q \in \{1, ..., |N_l|\}$ in an entirely reserved network, respectively. Then, for $\forall l \in L, \forall q \in \{1, ..., |N_l| - 1\}$, a set A^l_q can be defined as follows.

$$A^l_q = \{(i, j)|T^l_q + \varphi(s_{lq}, i) + \tau_{ij} + \varphi(j, s_{l,q+1}) > T^+_{q,l}, (i, j) \in A\}, \forall q \in \{1, ..., |N_l| - 1\} \tag{6.44}$$

According to the definition of A^l_q, we can easily find that $\forall(i, j) \in A^l_q$ will not be located on the path between station s_{lq} and $s_{l,q+1}$, $l \in L$, since the sum of the travel time from station s_{l1} to $s_{l,q+1}$ in an entirely reserved network is greater than $T^+_{q,l}$, i.e., the arrival time constraint (6.39) at stop $s_{l,q+1}$ is violated. Based on above analysis, the corresponding variables x^l_{qij} and y^l_{qij} must be equal to be 0, i.e., the values of these variables can be fixed as 0 without excluding any feasible solution. Then, a tighter integer linear program $\mathcal{P}'_2$ is defined as follows.

$$\begin{aligned}
\mathcal{P}'_2 : \quad & \min \sum_{(i,j)\in A} C_{ij} z_{ij} \\
\text{s.t.} \quad & \text{Constraints } (6.32) - (6.43) \\
& x^l_{qij} + y^l_{qij} = 0, \forall(i, j) \in A^l_q, q \in \{1, ..., |N_l| - 1\}, l \in L \qquad (6.45)
\end{aligned}$$

Note that constraint (6.45) is a valid inequality since model $\mathcal{P}'_2$ is adequate to define the considered problem but it greatly reduces the problem search space, as also shown in our experimental results. The complexity of the BLRP-PD is shown by the following theorem.

6.3 Solution approach

In this section, to efficiently solve the BLRP-PD, we first propose an enhanced cut-and-solve algorithm and an improved kernel search based heuristic is then developed for large-size problem instances.

6.3.1 Enhanced cut-and-solve algorithm

To exactly solve the BLRP-PD, an enhanced cut-and-solve based optimal algorithm is proposed in this section. As stated in Chpater 2, cut-and-solve method is a special branch-and-bound (B&B) iterative search strategy, which was firstly introduced by [30] to exactly solve the classical ATSP. In the following, we first present several improvements for the cut-and-solve method, and then the enhanced CS method is adapted to exactly solve the BLRP-PD.

The principle of the basic CS method has been described in Chapter 2. As pointed out by [30], to apply the CS method to solve an ILP (without loss of generality, a minimization problem), piercing cut plays a crucial role since it drives the branching of CS at each iteration and it should be specially designed for different optimization problems. In [30], a generic procedure was proposed to generate piercing cut based on variables' reduced costs by solving the corresponding linear relaxation problem. More precisely, the authors aimed to define variables set with large probability to take the value of 0 in an optimal solution of the ILP. Its core idea is to define a set of variables, denoted by U^n, that have large reduced cost. Then, the current problem (CP_n) is partitioned into SP^n (formed by adding the constraint that the sum of the variables in U^n is equal to zero) and RP^n (formed by adding the constraint that the sum of the variables in U^n is greater than or equal to one). This idea is subsequently followed by [45], [42]. We refer to the CS iteration with the above generic piercing cut as basic CS algorithm. To further improve its performance, an enhanced version of the CS algorithm is proposed by considering the following improvements.

Enhanced piercing cut: Defining the piercing cut only using decision variables' reduced costs may result in low solution efficiency for an ILP, as shown in our preliminary experiments for the BLRP-PD, we propose to enhance the piercing cut by additionally taking into account the parameters influencing the objective function value. Variables selected by combining the variables' reduced cost and the related parameters may have larger probability to take the value of 0 in an optimal solution.

Acceleration of SP^n resolution: In the basic CS algorithm, each sparse problem should be exactly solved. Although a sparse problem is relatively easy to solve, a sequence of sparse problems may be time-consuming, especially for large-size problems. To save the computational effort, for $n \geq 2$ we propose to set the current best upper bound UB^b as an upper bound to the objective function of SP^n. Indeed, we are only interested in finding solution with better upper bound than UB^b.

6.3.1.1 Enhanced CS algorithm for the BLRP-PD

The BLRP-PD contains multiple sets of binary decision variables, i.e., $z_{ij}, x^l_{qij}, y^l_{qij}$. It should be very careful to select which variables to form U^n for designing CS algorithm. In the BLRP-PD, we can find that the value of z_{ij} greatly influences the values of x^l_{qij} and y^l_{qij} by (6.40) and (6.41). For example, if $z_{ij} = 0$, then $x^l_{qij} = 0$ by (6.40) and if $z_{ij} = 1$, then $y^l_{qij} = 0$ by (6.41). In addition, the objective function value is only related with z_{ij}. For above reasons, only z_{ij} is used to define U^n. Furthermore, C_{ij} directly influencing the objective function value is taken into account in defining U^n. It is defined as:

$$U^n = \{z_{ij} | C_{ij}\hat{z}_{ij} > \alpha^n\}, \forall (i,j) \in A \tag{6.46}$$

where α^n is a given positive value, $\hat{z}_{ij}$ is the reduced cost of z_{ij} obtained by solving the corresponding linearly relaxed problem. Parameter α^n is decided by the distribution of values of $C_{ij}\hat{z}_{ij}$, $\forall (i,j) \in A$. Based on preliminary tests, α^n is set as $\text{median}\{C_{ij}\hat{z}_{ij} | (i,j) \in A\}$ in this chapter. The idea is to select variables z_{ij} with large $C_{ij}\hat{z}_{ij}$ to form set U^n since they have larger probability to take the value of 0 in an optimal solution of the BLRP-PD.

In order to further improve the performance of the CS method for the BLRP-PD, a new variable set U'_n ($n \geq 2$) based on (6.46) is defined as follows.

$$U'_n = \{z_{ij} | C_{ij}\hat{z}_{ij} > \alpha^n, z_{ij} \in U'_{n-1}\}, \forall (i,j) \in A \tag{6.47}$$

For $n = 1$, $U'_1 = U^1$. Once U'_n is obtained, then piercing cut PC^n for the BLRP-PD is defined as follows.

$$PC^n : \sum_{z_{ij} \in U'_n} z_{ij} \geq 1 \tag{6.48}$$

With PC^n, according to the principle of the CS method at n ($n \geq 1$) iteration, CP^n (CP^1 is defined as $\mathcal{P}'_2$) is divided into SP^n and RP^n, which can be defined, respectively, as

$$SP^n : \qquad \min \sum_{(i,j) \in A} C_{ij} z_{ij}$$

$$
\begin{aligned}
s.t. \quad & \text{Constraints (6.32)} - \text{(6.43) and (6.45)} \\
& \sum_{z_{ij} \in U'_t} z_{ij} \geq 1, \forall t \in \{1, ..., n-1\} && (6.49) \\
& \sum_{z_{ij} \in U'_n} z_{ij} = 0 && (6.50) \\
RP^n: \quad & \min \sum_{(i,j)\in A} C_{ij} z_{ij} \\
s.t. \quad & \text{Constraints (6.32)} - \text{(6.43) and (6.45)} \\
& \sum_{z_{ij} \in U'_t} z_{ij} \geq 1, \forall t \in \{1, ..., n\} && (6.51)
\end{aligned}
$$

Note that the search space of SP^n is relatively small since part of its variables are fixed to be 0 by (6.50). To accelerate the solution of SP^n and RP^n, the number of constraints of them are reduced with the following theorem.

Theorem 10 *For $n \geq 2$, $U'_1 \supseteq U'_2 \supseteq ... \supseteq U'_n$ and constraint (6.49)(resp.(6.51)) in SP'_n(resp. RP'_n) is equal to the following (6.52)(resp.(6.53)).*

$$\sum_{z_{ij} \in U'_{n-1} \backslash U'_n} z_{ij} \geq 1 \tag{6.52}$$

$$\sum_{z_{ij} \in U'_n} z_{ij} \geq 1 \tag{6.53}$$

Proof: The correctness can be proved similar to the proof of Theorem 2 in [45]. For details, please refer to [45]. □

According to Theorem 2, for $n \geq 2$, SP^n and RP^n are reduced as SP'_n and RP'_n, respectively.

$$
\begin{aligned}
SP'_n: \quad & \min \sum_{(i,j)\in A} C_{ij} z_{ij} \\
s.t. \quad & \text{Constraints (6.32)} - \text{(6.43), (6.45), (6.50) and (6.52)} \\
RP'_n: \quad & \min \sum_{(i,j)\in A} C_{ij} z_{ij} \\
s.t. \quad & \text{Constraints (6.32)} - \text{(6.43), (6.45) and (6.53)}
\end{aligned}
$$

It can be seen that $n-1$ inequalities in (6.49) for SP^n is reduced to only one inequality in (6.52) for SP'_n and $n-1$ inequalities in (6.51) for SP^n is totally removed in (6.53) for RP'_n.

Furthermore, according to the proposed improvement in the enhanced CS algorithm, for $n \geq 2$, the current best upper bound UB^b is set as an upper bound to the

Enhanced cut-and-solve method for the BLRP-PD

```
1: Initialize $n = 1$, $UB^b = +\infty, CP^1 = \mathcal{P}_2'$.
2: Solve the linear relaxation problem $CP^n$ and obtain reduced cost of $z_{ij}$, i.e., $\hat{z}_{ij}$,
   $(i, j) \in A$.
3: Define set $U_n'$, $PC^n$, $SP_n''$ and $RP_n'$.
4: Solve $SP_n''$ and (if feasible) obtain its optimal objective function value $UB^n$ and
   $UB^b$ is updated as $UB^n$;
5: Solve the linearly relaxed $RP_n'$ to obtain $LB^n$.
6: if $UB^b \leq LB^n$, output $UB^b$ and the corresponding solution as the global objective
   function value and solution, respectively, and end;
7: else set $n = n + 1$ and go back to Step 3.
```

Fig. 6.4: Algorithm BLRP-PD: algorithm for the BLRP-PD

objective function of SP_n', which is represented as follows.

$$\sum_{(i,j)\in A} C_{ij} z_{ij} < UB^b \tag{6.54}$$

Then, for $n \geq 2$, SP_n' is redefined as the following SP_n''.

$$\begin{aligned} SP_n'': \qquad & \min \sum_{(i,j)\in A} C_{ij} z_{ij} \\ s.t. \qquad & \text{Constraints}(6.32) - (6.43), (6.45), (6.50), (6.52), \text{ and } (6.54) \end{aligned}$$

SP_1'' and RP_1' are still defined as SP^1 and RP^1, respectively. Via extensive preliminary experiments, it was shown that the computational time of the CS algorithm were significantly reduced after introducing (6.54).

The enhanced cut-and-solve method to find the optimal solution of the BLRP-PD is depicted in Fig. 6.4.

6.3.2 Improved kernel search method for the BLRP-PD

In order to solve large-size problem instances within acceptable computational time, an improved KS based heuristic is proposed to solve the BLRP-PD in this section. The KS method is an iterative heuristic recently introduced by [5] for solving the multidimensional knapsack problem. Its core idea is to identify subsets of variables and exactly solve a sequence of subproblems restricted to these subsets. Its exciting results motivate us to apply the framework of KS to solve the BLRP-PD. The principle of kernel search method has been described in Chapter 2. In the following, an improved KS based heuristic is specially designed for the BLRP-PD.

As discussed previously, one core part of the KS method is to construct appropriate kernels and buckets. In the literature, kernels and buckets are usually composed of all sets of variables. However, for complex models with multiple sets of variables like the BLRP-PD including up to three sets of integer variables, the designed KS method taking into account all sets of variables may be time-consuming due to excessive iterations or large-size restricted problems.

The BLRP-PD includes three sets of binary variables and the two sets x^l_{qij} and y^l_{qij} (a reserved lane is passed or not) are greatly affected by z_{ij} (an arc is reserved or not). Besides, the objective function value is only related with z_{ij}. These characteristics incite us to form the kernels and buckets only using variables z_{ij}, instead of all sets of variables done in the previous KS methods, which means z_{ij} is considered promising if arc (i, j) is likely to be reserved in an optimal solution of the BLRP-PD.

Let ILP(z, x, y) and LP(z, x, y) denote the original problem, i.e., $\mathcal{P}'_2$, and its linear relaxation, respectively, and we then present the details of the KS method for the BLRP-PD in the following.

At the first iteration of the KS method, LP(z, x, y) is optimally solved. If its optimal solution is integer, then it is an optimal solution of ILP(z, x, y) and the KS terminates. Otherwise, the initial kernel and the buckets, denoted by $K_1(z)$ and $\{B_l(z)\}, l = 1, ..., m$, respectively, are constructed by the following criteria.

Sorting criterion: For the variables $z_{ij} > 0$ in the optimal solution of LP(z, x, y), they are firstly sorted in non-increasing order of their values, and the variables $z_{ij} = 0$ are subsequently sorted in non-decreasing order of the value of $C_{ij}\hat{z}_{ij}$ instead of only the reduced cost in the literature. Such sorting aims to make the variables z_{ij} that are most likely to take the value of 1 in an optimal solution of ILP(z, x, y) in the first positions and those with the least probability in the last positions.

Kernel initialization: Initial kernel $K_1(z)$ contains the first C variables in the above sorted list, and C is set as the number of variables z_{ij} with $z_{ij} > 0$ in the optimal solution of LP(x, y, z).

Buckets construction: Overlapping and disjoint buckets have been proposed in the literature. Based on preliminary experiments, the remaining variables are partitioned into $m := \lceil \frac{|A|-C}{L} \rceil$ disjoint subsets, in which the first $m - 1$ buckets have the same length L and the last one may have a smaller number and $|A|$ is the number of variables z_{ij}, i.e., the number of arcs in the network. The value of L is set equal to C.

The first restricted problem, denoted by ILP$(K_1(z))$ (ILP$_1$ in short), can be defined as follows.

$$\begin{aligned} \text{ILP}_1: \qquad & \min \sum_{(i,j)\in A} C_{ij} z_{ij} \\ s.t. \qquad & \text{Constraints (6.32)} - \text{(6.43)},\ \text{and (6.45)} \\ & z_{ij} = 0, \forall z_{ij} \notin K_1(z) \qquad (6.55) \end{aligned}$$

where constraint (6.55) restricts ILP(z, x, y) to $K_1(z)$ by fixing $z_{ij} \notin K_1(z)$ as 0. The optimal solution (if exists) of ILP$_1$ provides an upper bound of ILP(Z, X, Y) and the current best upper bound obtained by the KS method, denoted by U^b, is set as the optimal function value of ILP$_1$.

At the l-th iteration, where $2 \leq l \leq \overline{m} + 1$, the restricted problem, denote by ILP$(K_{l-1}(z) \bigcup B_{l-1}(z))$ (ILP$_l$ in short), is formed as follows:

$$\begin{aligned} \text{ILP}_l: \qquad & \min \sum_{(i,j)\in A} C_{ij} z_{ij} \\ s.t. \qquad & \text{Constraints (6.32)} - \text{(6.43)},\ \text{and (6.45)} \\ & z_{ij} = 0, \forall z_{ij} \notin K_{l-1}(z) \bigcup B_{l-1}(z) \qquad (6.56) \\ & C_{ij} z_{ij} \leq U^b \qquad (6.57) \\ & \sum_{z_{ij}\in B_{l-1}} z_{ij} \geq 1 \qquad (6.58) \end{aligned}$$

where constraint (6.56) aims to restrict ILP(z, x, y) to $K_{l-1}(z) \bigcup B_{l-1}(z)$. Two additional constraints (6.57) and (6.58) are added to reduce the computational effort, which aims to guarantee that the corresponding upper bound is not worse than the current best upper bound and at least one variable in $B_{l-1}(z)$ is selected in the optimal solution, respectively. Indeed, we are only interested in those solutions that improve the current beset upper bound and involve at least one new variable from the current bucket. In addition, at the second iteration, the kernel for ILP$_2$ is the initial kernel $K_1(z)$, while for the $l(l \geq 3)$-th iteration, the kernel K_{l-1} is updated with respect to the previous iteration. In addition, we decide to analyze all the buckets to find better solution as far as possible, i.e., $\overline{m} := m$ and each restricted problem is exactly solved.

Kernel updating: If ILP$(K_{l-1}(z) \bigcup B_{l-1}(z))$ is infeasible, $K_l(z)$ for the next iteration is set equal to $K_{l-1}(z)$, and otherwise $K_l(z)$ is updated as $K_{l-1}(z) \bigcup B^+_{l-1}(z) \backslash K^-_{l-1}(z)$, where $B^+_{l-1}(z)$ consists of $z_{ij} \in B_{l-1}(z)$ taking the value of 1 in the optimal solution of the current problem and $K^-_{l-1}(z) \subseteq K_{l-1}(z)$ contains $z_{ij} \in K_{l-1}(z)$ taking the value of 0 in the optimal solution as well as in h of previous iterations since they have been added to the kernel. Parameter h is set as 2.

Via preliminary experiments, it was found that for some large-size instances, their restricted problems are still not easy to solve due to their large search space. By analyzing the preliminary experimental results, we found that a large part of variables z_{ij} taking the value of 0 (resp. 1) in the optimal solution of ILP(z, x, y) also take the value of 0 (resp. 1) in the optimal solution of LP(z, x, y). Based on the above observation, the following variable fixing strategy is designed to further improve the performance of the proposed KS method.

Once LP(z, x, y) is solved, variables z_{ij} can be partitioned into three subsets: $Z(0), Z(1)$ and $Z(0-1)$ with $z_{ij} = 0, z_{ij} = 1, 0 < z_{ij} < 1$ respectively. Subsequently, a part of variables that have larger probability to take the same value in an optimal solution are selected from $Z(0)$ (resp. $Z(1)$) are fixed as 0 (resp. 1). The fixed variables sets are defined as the following $Z^f(0)$ and $Z^f(1)$, respectively.

$$Z^f(0) = \{z_{ij} | C_{ij}\hat{z}_{ij} > \lambda_0, \forall z_{ij} \in Z(0)\}, \tag{6.59}$$

$$Z^f(1) = \{z_{ij} | \hat{w}_{ij}(\tau'_{ij} - \tau_{ij})/C_{ij}) > \lambda_1, \forall z_{ij} \in Z(1)\}, \tag{6.60}$$

where λ_0 and λ_1 are two given parameters, which are set as median$\{C_{ij}\hat{z}_{ij} | (i, j) \in Z(0)\}$ and median$\{\hat{w}_{ij}(\tau'_{ij} - \tau_{ij})/C_{ij} | (i, j) \in Z(1)\}$, respectively. $\hat{w}_{ij}$ calculates the total time of reserved arc a passed by all bus transit lines with the optimal solution of LP(z, x, y) and $(\tau'_{ij} - \tau_{ij})/C_{ij})$ represents the reduced time per unit impact. (6.59) aims to select variables z_{ij} with larger $C_{ij}\hat{z}_{ij}$ from $Z(0)$ since they have larger probability to be not reserved, i.e., $z_{ij} = 0$, whereas (6.60) aims to select variables z_{ij} with larger $\hat{w}_{ij}(\tau'_{ij} - \tau_{ij})/C_{ij})$ due to lane reservation from $Z(1)$ since they have larger probability to be reserved, i.e., $z_{ij} = 1$.

With the above strategy, new restricted problem ILP$'_1$ and ILP$'_l(2 \leq l \leq \overline{m} + 1)$ can be defined as follows.

$$\begin{aligned}
\text{ILP}'_1: \quad & \min \sum_{(i,j) \in A} C_{ij} z_{ij} \\
s.t. \quad & \text{Constraints } (6.32) - (6.43),\ (6.45) \text{ and } (6.55) \\
& z_{ij} = 0, \forall z_{ij} \in Z^f(0) \qquad (6.61) \\
& z_{ij} = 1, \forall z_{ij} \in Z^f(1) \qquad (6.62) \\
\text{ILP}'_l: \quad & \min \sum_{(i,j) \in A} C_{ij} z_{ij} \\
s.t. \quad & \text{Constraints } (6.32) - (6.43),\ (6.45),\ (6.56) - (6.58), (6.61) \text{ and } (6.62)
\end{aligned}$$

Note that with the variable fixing strategy the previous variable sorting, kernel initialization and bucket construction are performed after excluding variables in $Z^f(0)$

Improved KS based heuristic for the BLRP-PD

```
Initialize $l := 1$ and $U^b := +\infty$
Solve LP$(Z, X, Y)$ exactly and **if** the optimal solution is integer, it is outputted as an optimal solution of ILP$(Z, X, Y)$ and **stop**; and **otherwise** obtain variables' values and reduced costs
Define $Z^f(0)$ and $Z^f(1)$ by (6.59) and (6.60) and sort the remaining $z_{ij}$ with *Sorting criterion*
Initialize $K_1(z)$ and construct $\{B_l(z)\}, l = 1, ..., m$ with *Kernel initialization* and *Bucket construction*, respectively. Initialize $\overline{m} = m$
Solve ILP$'_1$ exactly and **if** it is feasible, $U^b$ is updated as its objective function value. $l = l + 1$
**while** $l \leq \overline{m} + 1$ **do**
    Construct $K_{l-1}(z) \bigcup B_{l-1}(z)$
    Solve ILP$'_l$ exactly and **if** it is feasible, **then** update $U^b$
    Define $K_l(z)$ for ILP$'_{l+1}$ with *Kernel updating*
    Let $l = l + 1$
**end while**
Output $U^b$ and its solution as the obtained objective function value and solution, respectively.
```

Fig. 6.5: Heuristic BLRP-PD: kernel search for the BLRP-PD

and $Z^f(1)$, and $m := \lceil \frac{|A| - |Z^f(0)| - |Z^f(1)| - C}{L} \rceil$ buckets are created, where $|Z^f(0)|$ (resp. $|Z^f(1)|$) denotes the number of variables in $Z^f(0)$ (resp. $Z^f(1)$). U^b is updated when any ILP$'_l$ is feasible. The KS heuristic for the BLRP-PD is sketched in Fig. 6.5.

6.4 Computational results

In this section, numerical experimental results are presented and discussed to demonstrate the performance of the proposed model and algorithms. They have been implemented in Visual C++ on a PC with 2.5 GHz CPU and 2.95 GB RAM under Windows 7. The SP''_n and linearly relaxed RP'_n in Algorithm 1, and LP(z, x, y) and ILP$'_l$ in Algorithm 2 were solved using commercial optimization software CPLEX (version 12.6). Via preliminary experiments, the parameters of CPLEX are set as follows. The shifting algorithm is chosen as the LP optimizer (parameter RootAlg). The dynamic search is used as the ILP optimizer, and furthermore, we choose to branch based on pseudo costs (parameter Varsel), moderately generate flow cover (parameter FlowCovers) and mixed integer rounding cuts (parameter MIRCuts), but not perform probing (parameter Probe) to save computational efforts. All the other parameters are in default settings.

The performance of the proposed models and algorithms was evaluated on 82 randomly generated problem sets with five instances for each set (i.e., 410 instances in total). For each instance, the results of all the proposed model and algorithms will include both a lane reservation scheme and a designed station arrival-time guaranteed path for each bus transit line. Since the proposed exact enhanced CS algorithm can obtain optimal solutions, it is evaluated only in terms of the computational efficiency. The computational efficiency of the proposed exact CS algorithm is compared with the well-known commercial ILP solver CPLEX. The obtained solution may be not unique since there may be multiple optimal ones for a same instance.

For the KS heuristic, since its solution is not guaranteed to be optimal, it is evaluated in terms of computational time and solution quality. In the literature, to evaluate the solution quality of a heuristic, the gap between its obtained objective function value and the optimal one has been widely used. Consequently, to evaluate the KS heuristic, the gap is calculated by the formula $(U^b - U^*)/U^*$, where U^b and U^* are the objective function values found by the KS and CS algorithms, respectively. For the evaluation of computational time, we compare it with the CS method because it is faster than CPLEX for most and large-size instances. Note that as the KS algorithm is a heuristic, theoretically the obtained solution and its objective function value cannot be guaranteed to be unique. Therefore, we run the KS heuristic five runs for each instance and report its average computational results. We observe that the objective function value for each instance varies slightly for five runs. Note that the computational time of each method is limited to 18000s (CPU seconds) for each instance. For simplicity, let $CT_{\mathcal{P}_1'}, CT_{\mathcal{P}_2}, CT_{\mathcal{P}_2'}$ denote the computational time spent by CPLEX for solving $\mathcal{P}_1', \mathcal{P}_2$, and $\mathcal{P}_2'$, respectively; CT_{CS} and CT_{KS} denote the computational time spent by the enhanced CS algorithm and improved KS heuristic for solving $\mathcal{P}_2'$, respectively. With the notations above, the computational results are summarized in Tables 6.1-6.6 and Fig.'s 6.6 and 6.7. Note that each value for each set in the result tables is its average value of the five instances.

In this chapter, to thoroughly evaluate the proposed algorithms, extensive numerical experiments are conducted. For these test instances, the transportation networks are generated based on the Waxman's network model [121] and the input parameters are estimated according to the previous studies due to the lack of benchmark instances in the literature. The graph $G\{N, A\}$ generated based on the well-known Waxman's network generation method [121] is shown as follows. Given the required network size (i.e., the number of nodes $|N|$ and arcs $|A|$), the nodes are randomly distributed in a Euclidean plane $[0, 100] \times [0, 100]$, and the existence probability of an arc a between any pair of nodes is decided by a probability function $\alpha \exp(-L_{ij}/\beta L)$, $0 \leq \alpha, \beta \leq 1$, where L_{ij} and L denote the Euclidean distance of arc a and the maximum Euclidean distance between any pair of nodes in the network, respectively. The parameter α is proportional to the number of arcs and an increase in parameter β gives a higher ratio of long arcs to short ones in the graph. To simulate the practical network, the ratio $|N|/|A|$ falls in [3, 4] as in [43].

Then, with the network and the number of bus transit lines, i.e., $|L|$, and the number of stations among each bus transit line, i.e., $|N_l|$, bus transit lines are generated as follows. For bus transit line $k, \forall l \in L$, its $|N_l|$ stations are generated from set N. Subsequently, one station is firstly selected from them as the origin station. Finally, the orders of the rest stations are sequentially determined based on the minimum Euclidean distance away from the previous determined station. For example, the second station is considered as the one in the remaining $|N_L| - 1$ stations with the minimum Euclidean distance from the origin station. Travel time parameters τ'_{ij} and τ''_{ij} are both appropriately estimated using the BPR function, which are considered as the corresponding average value, respectively, as in Chapter 4. The bus travel time on the reserved lane on arc a, i.e., τ_{ij}, is approximately estimated as the free-flow travel time that is calculated by $L_{ij}/\mathcal{V}_{ij}$, where $\mathcal{V}_{ij}$ denotes the free-flow travel speed on arc a, respectively. The negative impact is estimated as $C_{ij} = P_{ij}(\tau''_{ij} - \tau'_{ij})$. Parameter P_{ij} is also calculated as $b_{ij}v_{ij}$, where v_{ij} and b_{ij} denote the average number of non-bus vehicles per unit time on arc a and the average passenger count inside each one, respectively. The arrival time parameter $T^+_{i,l}$ is defined as $T^+_{i,l} = \hat{T}^l_i - r^t_l(\hat{T}^l_i - T^l_i)$, where $\hat{T}^l_i$ (resp. T^l_i) represents the earliest arrival time at the i-th station in a non-reserved (resp. entirely reserved) network. $T^-_{i,l}$ is set to be $\hat{T}^l_i$. In default case, b_{ij} is generated in the interval $[1, 3]$, and r^t_l is set as 0.2. Computational experiments of sensitive analysis for b_{ij} and r^t_l have been conducted.

To evaluate the performance of the proposed models, we first solve some instances through the them using CPLEX. Table 6.1 reports their comparison results.

From Table 6.1, we can see that $CT_{\mathcal{P}_2}$ is much smaller than $CT_{\mathcal{P}'_1}$ over all sets 1-14. The average $CT_{\mathcal{P}_2}$ is only 1.13 % and 0.49% of the average $CT_{\mathcal{P}'_1}$ for $|N_l| = 5$ and 7, respectively. Moreover, $CT_{\mathcal{P}_2}$ gradually increases while $CT_{\mathcal{P}'_1}$ rapidly increases with the problem size for both $|N_l| = 5$ and 7. In addition, $CT_{P_2}/CT_{P'_1}$ ranges between 0.48% and 16.34% for $|N_l| = 5$, while $CT_{P_2}/CT_{P'_1}$ ranges between 0.27% and 9.6% for $|N_l| = 7$. This implies that $\mathcal{P}_2$ is more efficient than the $\mathcal{P}'_1$ for larger $|N_l|$. It is worth

Table 6.1: Comparison results for the proposed formulations

Set	$\|N\|$	$\|A\|$	$\|L\|$	$\|N_l\|$	$CT_{\mathcal{P}_1'}$	$CT_{\mathcal{P}_2}$	$CT_{\mathcal{P}_2}/CT_{\mathcal{P}_1'}\%$
1	40	128	5	5	20.24	3.31	16.34
2	45	152	5	5	27.75	4.34	15.34
3	50	174	6	5	45.96	4.85	10.56
4	55	192	6	5	65.20	7.41	10.94
5	60	210	7	5	219.24	11.53	5.26
6	65	226	7	5	1149.21	11.91	1.04
7	70	252	8	5	4000.11	19.40	0.48
Average					789.67	8.92	1.13
8	40	128	5	7	34.90	3.35	9.60
9	45	152	5	7	156.88	4.95	3.16
10	50	174	6	7	261.72	7.39	2.82
11	55	192	6	7	552.70	9.27	1.68
12	60	210	7	7	813.53	16.18	1.99
13	65	226	7	7	2596.57	20.28	0.78
14	70	252	8	7	-	48.10	0.27
Average					>3200.33	15.65	0.49

noting that CPLEX cannot obtain an optimal solution using the $\mathcal{P}_1'$ within 18000s for set 14 while it takes only 48.10s with P_1'. These results show that the proposed ILP significantly outperforms the proposed MIP, which may be because the former model is much tighter and more compact than the latter one. Because the proposed ILP achieves much better performance, only the ILP is considered in the following.

Table 6.2 presents the computational results for small-size instances with $|N|$ increasing from 40 to 70, $|L|$ increasing from five to eight, and $|N_l|$ increasing from five to seven. We first analyze the performance of the valid inequality (6.45). It can be seen from Table 6.2 that $CT_{\mathcal{P}_2}$ is much smaller than $CT_{\mathcal{P}_2'}$ over all sets 15-28, which means that the valid inequality (6.45) is effective in reducing the search space for solving the considered BLRP-PD. $CT_{P_2'}$ varies from 1.72s to 51.52s, CT_{CS} varies from 2.24s to 39.15s, and CT_{KS} varies from 2.86s to 32.67s, which indicate that all the methods (i.e., CPLEX, CS algorithm and KS heuristic) can solve all the small-size instances within relatively short computation time. We can see that for the sets 15-27 CPLEX runs faster than the proposed algorithms which indicates that the direct use of CPLEX is more efficient for these small-size instances. This may be because the search space of these small-size instances is relatively small, CPLEX can easily solve them but both the proposed CS and KS algorithms are based on

Table 6.2: Comparison results for the instances with $|N|$ = 40-70

Set	$\|N\|$	$\|A\|$	$\|L\|$	$\|N_l\|$	$CT_{\mathcal{P}_2}$	$CT_{\mathcal{P}'_2}$	CT_{CS}	CT_{KS}	Gap %
15	40	128	5	5	5.52	1.75	2.24	2.86	0.05
16	45	152	5	5	5.95	1.72	2.48	3.38	0.31
17	50	174	6	5	8.02	1.85	3.27	4.14	0.82
18	55	192	6	5	9.91	2.14	2.92	7.14	0.07
19	60	210	7	5	23.01	2.91	3.45	8.63	0.02
20	65	226	7	5	36.69	5.64	5.24	7.71	0.73
21	70	252	8	5	42.66	6.40	8.31	9.14	0.56
Average					18.82	3.20	3.99	6.14	0.37
22	40	128	5	7	7.64	2.23	3.26	4.32	0.24
23	45	152	5	7	11.35	2.43	3.52	4.28	0.30
24	50	174	6	7	27.50	5.44	7.06	10.25	0.15
25	55	192	6	7	29.16	6.24	8.78	16.18	0.69
26	60	210	7	7	51.26	7.22	9.10	23.63	0.86
27	65	226	7	7	67.22	13.69	14.95	25.10	0.51
28	70	252	8	7	289.51	51.52	39.15	32.67	0.25
Average					54.81	12.70	12.26	16.62	0.43

the iterative solution of a sequence of the original problem's sub-problems. For these small-size instances, the difference in computational time between the subproblem and the original problem may be very small, which results in their more computational time, as compared to the direct use of CPLEX. However, it can be observed that for $|N_l|$ = 5 or 7, the computational time of the three methods increase with $|N|$ and $|L|$, while the increasing trend of the CS and KS is more slowly than that of CPLEX. Take sets 22 and 28 as an example, the computational time of CPLEX increases 22.1 (51.52/2.23-1) times while those of the proposed CS and KS increase 11 (39.15/3.26-1) and 6.56 (32.67/4.32-1) times, respectively. Moreover, for given $|N|$ and $|L|$, the computational time of the three methods increase with $|N_l|$. Take sets 21 and 28 for example, $CT_{P'_2}$, CT_{CS} and CT_{KS} are 6.40s, 8.31s and 9.14s, respectively for $|N_l|$=5, while they are 51.52s, 39.15s and 32.67s, respectively for $|N_l|$ = 7. This implies that the increase of $|N_l|$ increases the computational time of solving the BLRP-PD. Besides, the gap varies from 0.02% to 0.86% and the average gap for $|N_l|$=5 and 7 are 0.37% and 0.43%, respectively, which shows that the proposed KS heuristic can obtain high-quality near-optimal solutions.

Table 6.3 and Fig. 6.6 report the results for medium-size instances with fixed $|N_l|$ = 8 and $|N|$ and $|L|$ increasing from 75 to 100 and 7 to 12, respectively, from which

Table 6.3: Comparison results for the instances with $|N|$ = 75-100

Set	$\|N\|$	$\|A\|$	$\|L\|$	$CT_{\mathcal{P}_2}$	$CT_{\mathcal{P}'_2}$	CT_{CS}	CT_{KS}	Gap %
29	75	274	7	410.21	155.86	72.63	47.70	0.13
30	80	292	8	379.00	163.46	79.16	60.29	0.21
31	80	292	9	526.14	217.89	129.77	66.65	0.04
32	85	312	9	632.15	183.02	153.27	107.59	0.19
33	85	312	10	796.73	259.60	214.94	131.27	1.23
34	90	330	10	1832.22	670.30	459.63	151.04	0.50
35	90	330	11	1819.95	743.57	534.71	195.44	1.12
36	95	350	11	4569.43	930.12	777.12	227.73	1.30
37	95	350	12	7134.53	1143.94	965.93	392.69	0.47
38	100	366	12	9295.05	2273.00	1618.90	441.56	0.63
Average				2739.54	674.08	500.61	182.20	0.58

we can see: 1) $CT_{\mathcal{P}'_2}$ is smaller than $CT_{\mathcal{P}_2}$ on all sets 29-38, which means that the valid inequality (6.45) is effective in reducing the search space of the studied BLRP-PD; 2) $CT_{\mathcal{P}'_2}$ (resp. CT_{CS}) varies from 155.86s (resp. 72.63s) to 2273.00s (resp. 1618.90s) with the average value 674.08s (resp. 500.61s). CT_{CS} is smaller than $CT_{\mathcal{P}'_2}$ on each problem set and the average $CT_{CS}/CT_{\mathcal{P}'_2}$ is 74.27% (500.61/674.08). As shown in Fig. 6.6, $CT_{\mathcal{P}'_2}$ and CT_{CS} both increase with the problem size but the latter increases more slowly than the former. These results indicate that the CS algorithm is more efficient than CPLEX in finding optimal solutions; 3) CT_{KS} varies from 47.07s to 441.56s with the average value 182.20s. CT_{KS} is smaller than $CT_{\mathcal{P}'_2}$ and CT_{CS} on all the problem sets, and the proposed KS heuristic only spends an average 27.03% (182.20/674.08) (resp. 36.48% (182.20/500.61)) time of that spent by CPLEX (resp. the CS algorithm). From Fig. 6.6, we can see that CT_{KS} increases much more slowly with the problem size than $CT_{P'_2}$ and CT_{CS}. These results show that the proposed KS heuristic outperforms than CPLEX and the proposed CS algorithm in terms of computational time. On the other hand, the gap varies between 0.04% to 1.3% with the average value 0.58%, which indicates that the proposed KS heuristic not only can obtain high-quality near-optimal solutions, but also is stable in the sense that the gap varies slightly with the problem size; and 4) for given $|N|$ and $|N_l|$, $CT_{\mathcal{P}'_2}$, CT_{CS} and CT_{KS} increase with $|L|$. Take sets 36 and 37 for example, the instances in both sets have same $|N|$ and different $|L|$, $CT_{P'_2}$, CT_{CS} and CT_{KS} on set 37 are greater than that on set 36, respectively. We can also find that the computational time of all the methods have an increasing trend with $|N|$ for fixed $|L|$ and $|N_l|$. Take sets 35 and 36 for example, $CT_{\mathcal{P}'_2}$, CT_{CS} and CT_{KS} increase with $|N|$. These results imply that the increases of $|N|$ and $|L|$ also increase the computational time of the BLRP-PD.

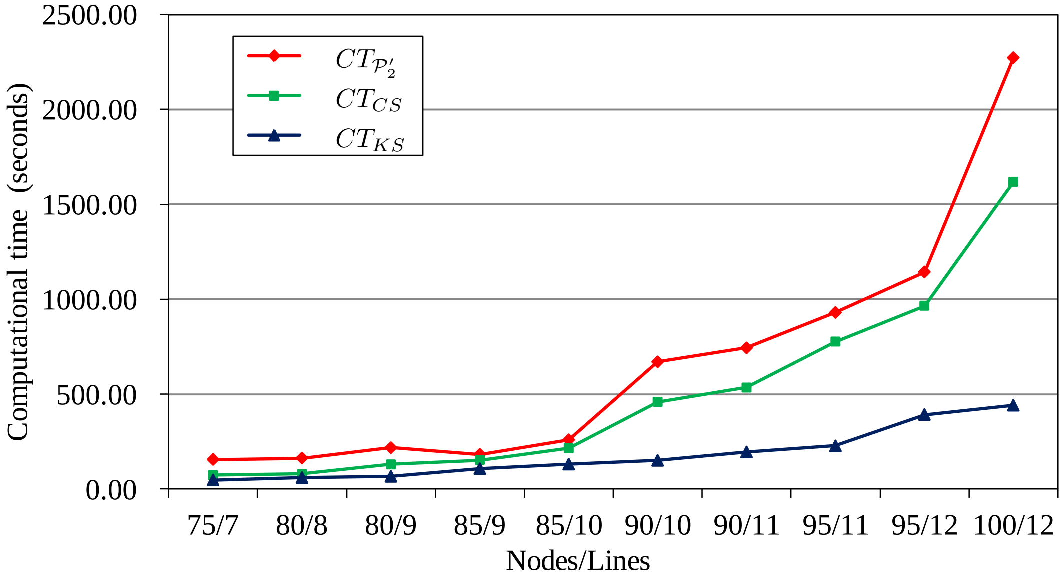


Fig. 6.6: Comparison results for the instances with $|N| = 75$-100

Table 6.4: Comparison results for the instances with $|N| = 110$-150

Set	$\|N\|$	$\|A\|$	$\|L\|$	$CT_{\mathcal{P}_2}$	$CT_{\mathcal{P}_2'}$	CT_{CS}	CT_{KS}	Gap %
39	110	398	12	8290.09	1494.87	926.52	397.23	0.38
40	115	436	13	-	4342.55	1882.47	632.40	1.99
41	120	452	13	-	3744.84	1897.17	664.82	0.82
42	125	454	14	-	4033.90	1531.12	438.55	0.69
43	130	480	14	-	4318.56	2505.78	952.98	1.11
44	135	512	15	-	8819.48	4958.71	1199.72	0.89
45	140	516	15	-	10677.81	6774.82	1397.03	1.14
46	145	518	16	-	-	7032.51	1082.63	1.49
Average					>6438.7	3438.64	845.6	1.06
47	150	568	16	-	-	-	1325.14	1.14

In Table 6.4 and Fig. 6.7, we present the results for larger-size instances with $|N_l| = 8$. Some observations can be obtained from the results: 1) CPLEX can only exactly solve the problem set 39 with $\mathcal{P}_2$, while seven sets with $\mathcal{P}_2'$, which further implies the effectiveness of the valid inequality (6.45); 2) CT_{CS} is smaller than $CT_{P_2'}$ over sets 39-46 and average $CT_{CS}/CT_{\mathcal{P}_2'}$ is less than 53.41% (3438.64/6438.79). Moreover, CT_{CS} increases more gradually than $CT_{\mathcal{P}_2'}$ with the problem size as shown in Fig. 6.7. These results show that the proposed CS algorithm is more efficient than

CPLEX for the larger-size instances in terms of computational time. In addition, it is worth noting that CPLEX cannot exactly solve all the instances in set 46 within 18000s, whereas the proposed CS algorithm only spends an average time of 6438.79s to find optimal solutions for all instances; and 3) for the KS heuristic, the gap varies between 0.38% and 1.99% and its average value is 1.06% on sets 39-46, which demonstrates that the proposed KS heuristic is stable and can find high-quality near-optimal solutions for the larger-size instances. On the other hand, CT_{KS} is far smaller than $CT_{\mathcal{P}_2'}$ and CT_{CS} on all the sets 39-46. On average, the proposed KS heuristic only spends less than 13.13% (845.67/6438.79) (resp. 24.59% (845.67/3438.64)) time of CPLEX (resp. the CS algorithm). Moreover, CT_{KS} increases much more slowly than $CT_{\mathcal{P}_2'}$ and CT_{CS}. Especially, both CPLEX and the proposed CS algorithm cannot exactly solve all the instances within 18000s due to the NP-hardness of the considered BLRP-PD, while the proposed KS heuristic obtain the solutions of a small average gap of 1.14% within 1325.14s. These results demonstrate that the KS heuristic significantly outperforms CPLEX and the CS algorithm in terms of computational time for the large-size instances.

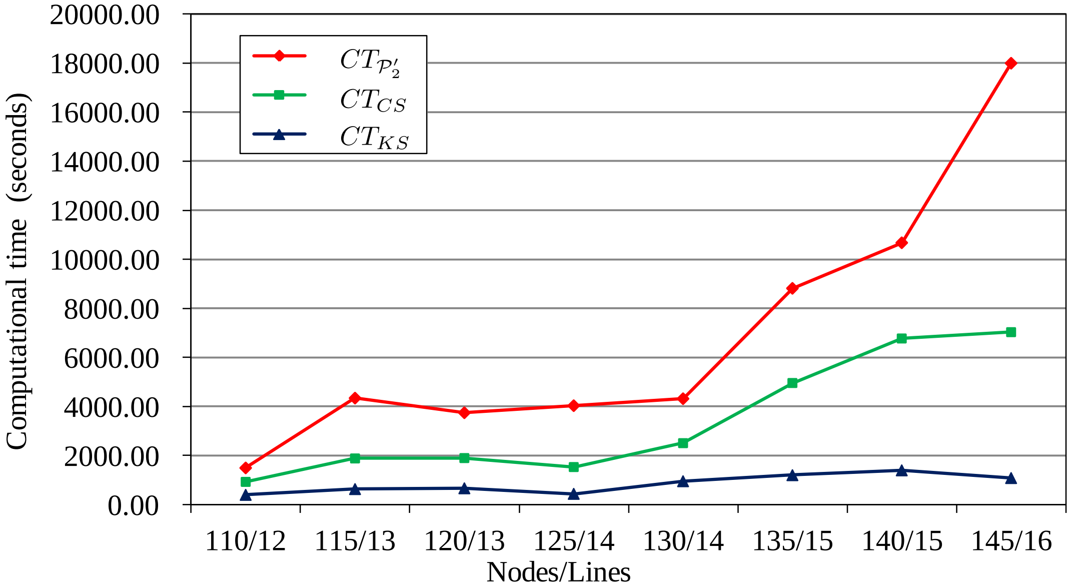


Fig. 6.7: Comparison results for the instances with $|N| = 110\text{-}145$

To further evaluate the performance of the proposed CS algorithm and KS heuristic, sensitive analysis for parameters b_{ij} and r_l^t are conducted to evaluate the stability of the proposed algorithms to different input parameters. That is to say, the sensitive analysis experiments are performed to test whether the CS algorithm can efficiently obtain the optimal solutions stably and whether the KS heuristic can efficiently find high-quality solutions stably for the studied problem. We test three scenarios for each parameter and the computational results are reported in Tables 6.5 and 6.6.

Table 6.5: Comparison for sensitive analysis of various impact

Set	b_{ij}	$\|N\|$	$\|L\|$	$CT_{\mathcal{P}_2'}$	$CT_{CS}/CT_{\mathcal{P}_2'}$ %	$CT_{KS}/CT_{\mathcal{P}_2'}$ %	Gap %
48		80	8	39.63	77.26	77.73	0.25
49		90	9	52.23	72.93	66.91	0.28
50	[1, 3]	100	10	155.76	79.02	53.23	0.69
51		110	11	210.50	63.99	50.59	1.04
52		120	12	525.88	51.29	23.27	1.35
Average				196.80	60.59	38.37	0.72
53		80	8	30.85	83.27	69.29	0.28
54		90	9	64.23	55.33	50.02	0.12
55	[3, 5]	100	10	132.13	62.98	55.43	1.15
56		110	11	207.72	55.63	44.84	1.07
57		120	12	560.81	48.66	29.72	1.42
Average				199.15	53.52	38.82	0.81
58		80	8	39.22	78.24	74.46	0.48
59		90	9	63.10	98.63	79.12	1.01
60	[1, 5]	100	10	129.64	45.36	41.69	0.79
61		110	11	235.48	54.88	46.62	1.06
62		120	12	520.12	62.53	23.35	0.82
Average				197.51	61.38	36.90	0.83

It can be observed from Table 6.5 that the ranges of $CT_{CS}/CT_{\mathcal{P}_2'}$ for $b_{ij} \in [1,3], [3,5]$ and $[1,5]$ are 51.29%-79.02%, 48.66%-83.27% and 45.36%-98.63%, respectively. The average $CT_{CS}/CT_{\mathcal{P}_2'}$'s for the three scenarios are 60.59%, 53.52% and 61.38%, respectively. These results show that the performance of the proposed CS algorithm is insensitive to parameter b_{ij}. For the proposed KS heuristic, we can see that $CT_{KS}/CT_{\mathcal{P}_2'}$ for the three scenarios ranges between 23.27% and 77.73%, 29.72% and 69.29%, 23.25% and 79.12%, respectively and the average values are 38.37%, 38.82% and 36.90%, respectively. Moreover, it can be seen that the gap varies from 0.25% to 1.35% with the average value 0.72% for $b_{ij} \in [1,3]$, from 0.12% to 1.42% with the average value 0.81% for $b_{ij} \in [3,5]$, and from 0.48% to 1.06% with the average value 0.83% for $b_{ij} \in [1,5]$, respectively. These results indicate that the proposed KS heuristic is also stable to the changes of parameter b_{ij}.

Table 6.6 presents the results of sensitive analysis of r_l^t. From Table 6.6, we can find similar results to those in Table 6.6. This indicates that the performance of our

Table 6.6: Comparison for sensitive analysis of different time windows

Set	r_l^t	$\|N\|$	$\|L\|$	$CT_{\mathcal{P}_2'}$	$CT_{CS}/CT_{\mathcal{P}_2'}$ %	$CT_{KS}/CT_{\mathcal{P}_2'}$ %	Gap %
48		80	8	39.63	77.26	77.73	0.25
49		90	9	52.23	72.93	66.91	0.28
50	0.2	100	10	155.76	79.02	53.23	0.69
51		110	11	210.50	63.99	50.59	1.04
52		120	12	525.88	51.29	23.27	1.35
Average				196.80	60.59	38.37	0.72
63		80	8	30.40	82.34	79.46	0.52
64		90	9	79.80	61.96	57.62	1.07
65	[0.2, 0.3]	100	10	152.27	66.53	56.22	0.89
66		110	11	206.10	72.31	49.28	0.56
67		120	12	464.14	59.53	40.70	0.73
Average				186.54	64.44	47.84	0.75
68		80	8	32.57	79.40	78.24	0.33
69		90	9	82.63	82.31	63.04	0.79
70	[0.3, 0.4]	100	10	141.26	83.93	62.19	0.60
71		110	11	223.54	78.38	45.62	0.93
72		120	12	604.25	57.74	32.78	0.50
Average				216.85	67.93	42.93	0.63

CS algorithm and KS heuristic is insensitive to the changes of parameters r_l^t.

6.5 Conclusions

This chapter has investigated a new BLRP motivated by achieving the rapid and reliable bus transportation with bus lane reservation strategy. For the considered problem, we first developed a mixed-integer non-linear program, which was subsequently transformed to be an equivalent linear one. Valid inequalities were added to tighten the proposed MIP. Then, an integer linear program was formulated and valid inequalities were also added to reduce the solution space. Finally, an enhanced cut-and-solve algorithm with new piecing cut was proposed to exactly solve the considered problem. To be able to more efficiently tackle large-size instances, an improved kernel search based heuristic was developed. Computational results demonstrate that: 1) the proposed ILP is much more efficient than that proposed MIP; 2) the enhanced CS algorithm outperforms CPLEX in finding optimal solutions; and 3) the improved KS based heuristic can yield high-quality solutions for large-size instances with up to 150 nodes and 568 arcs in the network and 16 bus transit lines within acceptable computational time.

Chapter 7

Bus lane reservation problem with line design

7.1 Introduction

The bus transit system design is highly complex and consists of five main stages: line planning, frequency setting, timetable development, vehicle scheduling and driver scheduling [21]. It has been pointed out by many researchers that bus line planning is a strategic (long-term) decision problem and has been identified as the most important stage in bus transit system design [104], since it would directly determine the total length of bus lines, thereby influencing the required vehicle fleet size and it influences the total passenger travel time and transfer times. Moreover, the bus line planning also influences the decisions of the remaining stages. As reported by [69], efficient bus lines would increase bus transit service level and reduce the bus company's expenditure. The classical bus line planning problem consists of finding an efficient set of bus lines in an already existing urban transport network, usually with previously defined bus stops, to cover the travel demand of passengers while respecting bus operating budget [40]. In the literature, this problem has been well addressed, see [10], [20], [21], [40], [63], [69], [91], [136]. However, as already stated in Chapter 1, bus transit service is becoming less and less attractive due to its inefficient service, such as long travel time and unreliable service, caused by traffic congestion. Bus lane reservation, as a traffic management strategy for promoting bus priority, has been widely applied in real life. Thus, it would be of great significance to consider bus lane reservation into the classic bus line planning. To the best of our knowledge, no work has addressed the bus line planning considering bus lane reservation, which is the focus of this chapter.

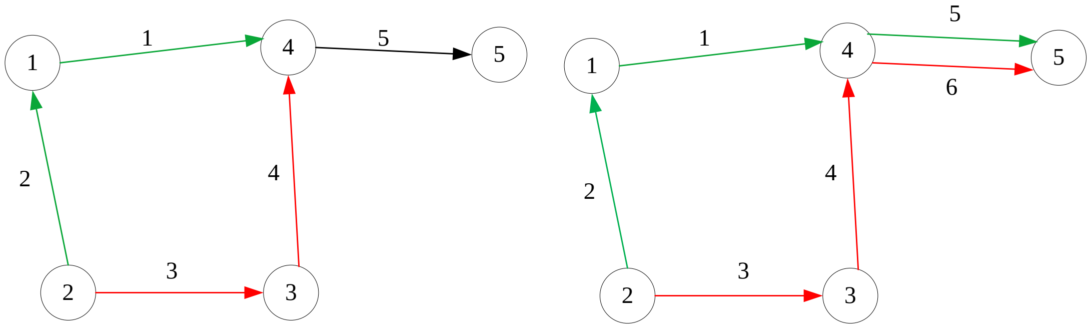
In this chapter, we investigate a new bus lane reservation and bus line design integrated optimization problem (called BLRP-LD in short). The problem consists of optimally designing bus lines and reserving exclusive bus lanes from the transit network simultaneously, thereby providing rapid and reliable bus transit system. Compared with the BLRPs studied in Chapters 5 and 6, the significant difference is that the bus lines with their stations need to be optimally determined along with bus lane reservation. Moreover, unlike the previous BLRPs that are single-objective optimization problems, the objective of the BLRP-LD is to simultaneously minimize the total negative impact of reserved lanes and the total travel time of all passengers. Because of the introduction of bus line design, the studied BLRP-LD becomes more complex than those in the previous chapters and the proposed approaches in previous BLRPs cannot be directly applied to the problem. In this chapter, to address the BLRP-LD, we first develop a new bi-objective integer nonlinear programming model. Then, the nonlinear one is equivalently transformed to be a linear one. Some valid inequalities are explored to reduce the solution space. Finally, exact ε-constraint method is then adapted to find its Pareto front.

The remainder of this chapter is organized as follows. In Section 7.2, the formulation of the BLRP-LD is presented and its complexity is analyzed. Exact ε-constraint method is adapted to solve the problem in Section 7.3. Computational results on an instance based on a benchmark for the classical bus line planning problem and randomly generated instances are reported in Section 7.4. Finally, Section 7.5 concludes this chapter.

7.2 Problem formulation

The BLRP-LD can be defined on a directed graph $G = \{N, A\}$ with a node set N and an arc set A. A node and an arc represent a road intersection or a bus station and a road link connecting two nodes, respectively. Let K denote the set of origin-destination (OD) pairs and $(o_k,\ d_k)$, and D_k be the origin, destination, and the number of passengers of OD pair $k \in K$, respectively. Naturally, o_k and d_k are stations. Given a set of candidate bus lines L, each line $l \in L$ is specified by a sequence of arcs. Let A_l denote the set of links passed by line l.

To meet the travel demand of a OD pair means finding a feasible travel path through the bus transit system. More specifically, the travel demand will be satisfied by one or several bus lines as well as links that are some segments of the bus lines. All these passed links constitute the travel path. However, different bus lines may contain



(a) $G = \{N, A\}$ (b) $G' = \{N, A'\}$

Fig. 7.1: Illustration of network G'

same road links. Thus, to distinguish the same links passed by different lines $l \in L$, we introduce a new network, $G' = \{N, A'\}$, where A' is the set of arcs converted by all lines in L, in which each arc only belongs to a specific bus line. Let A'_l denote the set of arcs passed by line l in G'. A small example (see Fig. 7.1) is given to illustrate the network G'.

In this example, there exists only one OD pair whose origin and destination are 2 and 5, respectively. According to this OD pair, the line set L is computed, which contains two candidate lines. Bus line 1 (resp. 2) is $2-1-4-5$ (resp. $2-3-4-5$), $A_1 = \{2, 1, 5\}$ and $A_2 = \{3, 4, 5\}$, as shown in 7.1 (a). Both A_1 and A_2 contain arc 5 in G. The corresponding network G' is shown in Fig. 7.1 (b), with which $A'_1 = \{2, 1, 5\}$ and $A'_2 = \{3, 4, 6\}$. We note that the network G is still necessary in order to define the negative impact of lane reservation, i.e., objective function f_2 described later, which cannot be defined by using G' since multiple selected lines may pass the same reserved lane.

Given a passenger OD matrix containing the information of OD pairs and a set of candidate bus lines L, the considered BLRP-LD consists of optimally choosing a subset of lines from set L to cover all travel demands and reserving bus lanes in the transit network to provide rapid and reliable bus system. The objectives are to minimize the overall passenger travel time and to minimize the total negative impact of bus lanes.

To well investigate the problem, several assumptions for the considered problem are made as follows: 1) passengers behave selfish and their choice of lines is based on shortest travel time including the penalty time of changing lines. Like most previous studies, e.g., [22], [39], a penalty time of five minutes is set for each transfer; 2) there will always be sufficient buses on each line to ensure that all passengers can complete their trips in the fastest way; 3) the bus operating cost is in proportional to the total transit time and any bus line is operated in a bidirectional way, as in [100],[101]. It is understandable that the larger the total travel time on a bus line, the more buses and drivers to be needed for guaranteeing a given bus service frequency; and 4) at least two lanes exist on each road link such that one lane can be reserved and a bus lane is allowed to be shared by multiple bus lines so as to increase its effectiveness. To formulate the problem, the notations and decision variables are first listed as follows.

Sets and parameters

N	set of nodes in G and G'
A	set of arcs in G
L	set of candidate bus lines, line $l \in L$
A_l	set of arcs in G on line l with a given passing order, $A_l \subseteq A$
$L(a)$	set of bus lines passing arc $a \in A$, $L(a) \subseteq L$
A'	set of arcs in G'
A'_l	set of arcs on line l, $A'_l \subseteq A'$
A'^{+}_i	set of arcs coming into node $i \in N$, $A'^{+}_i \subseteq A'$
A'^{-}_i	set of arcs outgoing from node $i \in N$, $A'^{-}_i \subseteq A'$
$A'(a)$	set of arcs in A' corresponding to arc $a \in A$, $A'(a) \subseteq A'$
B	available bus operating budget expressed by the total transit time
K	set of passenger OD pairs, $k \in K$
o_k	origin node of OD pair $k \in K$
d_k	destination node of OD pair $k \in K$
D_k	amount of passengers of OD pair $k \in K$
P_T	penalty time per transfer (i.e., changing a line)
τ_a	travel time on a bus lane on arc $a \in A, A'$
τ'_a	travel time on arc $a \in A, A'$ without bus lanes
C_a	negative impact of implementing a bus lane on arc $a \in A$
M	a large positive number

Decision variables

z_a	$z_a = 1$ if arc a is reserved; and 0 otherwise, $a \in A$;
u_a^k	$u_a^k = 1$ if the path of OD pair $k \in K$ pass arc $a \in A'$ and a bus lane is reserved on it; and 0 otherwise;
v_a^k	$v_a^k = 1$ if the path of OD pair k pass arc a and no arcs are reserved on it; and 0 otherwise; $a \in A', k \in K$;
y_l	$y_l = 1$ if line l is selected from the candidate line pool; and 0 otherwise, $l \in L$;
x_l^k	$x_l^k = 1$ if the path of OD pair k uses line l; and 0 otherwise, $l \in L, k \in K$.

With the notations defined above, the BLRP-LD can be formulated as the following bi-objective integer nonlinear program.

$$\mathcal{P}_{bl} : f_1 : \min \sum_{k \in K} \sum_{a \in A'} D_k(\tau_a u_a^k + \tau'_a v_a^k) + \sum_{k \in K} D_k P_T (\sum_{l \in L} x_l^k - 1) \tag{7.1}$$

$$f_2 : \min \sum_{a \in A} C_a z_a \tag{7.2}$$

$$\text{s.t.} \sum_{l \in L} \sum_{a \in A_l} (\tau_a z_a + \tau'_a (1 - z_a)) y_l \leq B \tag{7.3}$$

$$\sum_{a \in A'^{-}_{o_k}} (u^k_a + v^k_a) = 1, \forall k \in K \tag{7.4}$$

$$\sum_{a \in A'^{+}_{d_k}} (u^k_a + v^k_a) = 1, \forall k \in K \tag{7.5}$$

$$\sum_{a \in A'^{-}_{d_k}} (u^k_a + v^k_a) = 0, \forall k \in K \tag{7.6}$$

$$\sum_{a \in A'^{+}_{o_k}} (u^k_a + v^k_a) = 0, \forall k \in K \tag{7.7}$$

$$\sum_{a \in A'^{+}_{i}} (u^k_a + v^k_a) = \sum_{a \in A'^{-}_{i}} (u^k_a + v^k_a), \forall i \in N \backslash \{o_k, d_k\}, \forall k \in K \tag{7.8}$$

$$\sum_{a \in A'_l} (u^k_a + v^k_a) \leq |A'_l| y_l, \forall k \in K, \forall l \in L \tag{7.9}$$

$$\sum_{a \in A'_l} (u^k_a + v^k_a) \leq |A'_l| x^k_l, \forall k \in K, \forall l \in L \tag{7.10}$$

$$\sum_{k \in K} u^k_{a'} \leq |K| z_a, \forall a' \in A'(a), \forall a \in A \tag{7.11}$$

$$z_a, u^k_{a'}, v^k_{a'}, y_l, x^k_l \in \{0, 1\}, \forall a \in A, \forall a' \in A', \forall k \in K, \forall l \in L \tag{7.12}$$

Objective (7.1) is to minimize the travel time of all passengers including penalty times of changing lines. Note that $\sum_{l \in L} x^k_l$ computes the number of lines taken by OD pair k, which must be greater than or equal to 1 so as to complete its trip, and thus $\sum_{l \in L} x^k_l - 1$ computes its required number of transfers. Objective (7.2) is to minimize the negative impact caused by all bus lanes. Constraint (7.3) ensures that the total operating cost should not exceed the available operating budget B. Constraints (7.4)-(7.8) guarantee that there exist a path for passengers of each OD pair and this path is a shortest path due to objective (7.1). To be more specific, constraint (7.4) (resp. (7.5)) implies that there exists only one arc outgoing from (resp. coming into) origin station o_k (resp. destination station d_k). Constraint (7.6) (resp. (7.7)) ensures that there are no arcs coming into (resp. outgoing from) origin station d_k (resp. destination station o_k). Constraint (7.8) ensures the flow conservation for intermediate notes between origin and destination stations for each OD pair $k \in K$. Constraint (7.9) guarantees that bus line l must be selected if it is used by any OD pair $k \in K$. Constraint (7.10) determines the bus line taken by OD pair k, $\forall k \in K$. Constraint (7.11) makes sure that any OD pair k can pass bus lane on arc $a \in A'$ only if the corresponding arc in the road network G is reserved. Constraint (7.12) enforces the bounds of all decision variables.

The complexity of the BLRP-LD is shown by the following theorem.

Theorem 11 *The BLRP-LD is NP-hard.*

Proof: If the negative impact of reserving a bus lane on each arc is small enough (i.e., objective f_2 can be removed), then the BLRP-LD corresponds to the particular case, a bus line planning problem, which is well-known to be NP-hard [39]. Therefore, the BLRP-LD in general case is NP-hard as well. □

7.2.1 Model linearisation

It can be observed that the model $\mathcal{P}_{bl}$ is non-linear due to the existence of non-linear constraint (7.3). Hence, in this subsection, we will transform it into an equivalent linear one by reformulating constraint (7.3). Before proceeding, a new variable is defined as follows.

w_l: operating cost on line l; w_l is set as 0 if line l is not selected from the candidate line pool, $\forall l \in L$.

For $\forall l \in L$, there exist two cases, which are a) the bus line l is selected and b) it is not selected. Therefore, the corresponding operating cost also exist two cases, i.e., 0 or $\sum_{a \in A_l}(\tau_a z_a + \tau'_a(1 - z_a))$. Then, w_l can be easily formulated as follows. a) *Bus line l is not selected.* Thus, $w_l = 0$, then

$$w_l \leq y_l M, \forall l \in L \tag{7.13}$$

b) *Bus line l is selected.* Thus, $w_l = \sum_{a \in A_l}(\tau_a z_a + \tau'_a(1 - z_a))$, then we have

$$w_l - \sum_{a \in A_l}(\tau_a z_a + \tau'_a(1 - z_a)) \geq M(y_l - 1), \forall l \in L \tag{7.14}$$

$$w_l - \sum_{a \in A_l}(\tau_a z_a + \tau'_a(1 - z_a)) \leq M(1 - y_l), \forall l \in L \tag{7.15}$$

Based on the above analysis, constraint (7.3) can be reformulated as follows.

$$\sum_{l \in L} w_l \leq B \tag{7.16}$$

$$w_l \geq 0, \forall l \in L \tag{7.17}$$

and constraints (7.13) – (7.15)

Then, the non-linear model $\mathcal{P}_{bl}$ can be transformed into the following mixed-integer linear program $\mathcal{P}'_{bl}$.

$$\begin{aligned}
\mathcal{P}'_{bl} :& f_1 : \min \sum_{k\in K}\sum_{a\in A'} D_k(\tau_a u_a^k + \tau'_a v_a^k) + \sum_{k\in K} D_k P_T(\sum_{l\in L} x_l^k - 1)\\
& f_1 : \min \sum_{a\in A} C_a z_a\\
\text{s.t.}\quad & \text{Constraints (7.4)} - \text{(7.17)}
\end{aligned}$$

7.2.2 Valid inequalities

To further tighten the formulated model and reduce its solution space, the following valid inequalities are introduced.

$$\sum_{a\in A_i'^{+}} (u_a^k + v_a^k) \le 1, \forall i \in N\backslash\{o_k, d_k\}, \forall k \in K \tag{7.18}$$

$$\sum_{a\in A_i'^{-}} (u_a^k + v_a^k) \le 1, \forall i \in N\backslash\{o_k, d_k\}, \forall k \in K \tag{7.19}$$

$$z_a \le \sum_{l\in L(a)} y_l, \forall a \in A \tag{7.20}$$

$$\sum_{k\in K} x_l^k \le |K| y_l, \forall l \in L \tag{7.21}$$

$$y_l \le \sum_{k\in K} x_l^k, \forall l \in L \tag{7.22}$$

where constraints (7.18) and (7.19) ensures that the passengers of OD pair k pass intermediate nodes in the network at most once. Constraint (7.20) states that arc $a \in A$ can have a reserved lane only if it is included in the paths of bus lines. Constraint (7.21) states that bus line l is selected when any OD pair uses it. Constraint (7.22) bus line $l \in L$ is selected only if there exists at least one OD pair uses this line. With these valid inequalities, we derive the following program.

$$\begin{aligned}
\mathcal{P}''_{bl} :& f_1 : \min \sum_{k\in K}\sum_{a\in A'} D_k(\tau_a u_a^k + \tau'_a v_a^k) + \sum_{k\in K} D_k P_T(\sum_{l\in L} x_l^k - 1)\\
& f_2 : \min \sum_{a\in A} C_a z_a\\
\text{s.t.}\quad & \text{Constraints (7.4)} - \text{(7.22)}
\end{aligned}$$

7.3 Solution approach

In Chapter 2, we have summarized the techniques for solving a multi-objective optimization problem, such as weighted sum method, ε-constraint method and Pareto-based evolutionary algorithm. In this section, the exact ε-constraint method introduced in Chapter 4 is adapted to obtain the Pareto front for the BLRP-LD. The advantages, principle and resolution procedure have been described in Chapter 4. With the ε-constraint method, the bi-objective model $\mathcal{P}'_{bl}$ can be transformed into a series of single-objective ones.

As mentioned above, the BLRP-LD has two conflicting objectives. The first objective is selected as the preferred one in this thesis. With the ε-constraint method, the $\mathcal{P}'_{bl}$ can be transformed into the following single-objective problem:

$$\begin{aligned}
\mathcal{P}'_{bl}(\varepsilon) : \min & \sum_{k\in K}\sum_{a\in A'} D_k(\tau_a u_a^k + \tau'_a v_a^k) + \sum_{k\in K} D_k P_T(\sum_{l\in L} x_l^k - 1) \\
\text{s.t.} & \sum_{a\in A} C_a z_a \leq \varepsilon \qquad\qquad (7.23) \\
& \text{and constraints } (7.4) - (7.17)
\end{aligned}$$

where ε denotes an upper bound of f_2.

Complexity analysis: If the value of ε is large enough, then the special case of the single-objective problem $\mathcal{P}'_{bl}(\varepsilon)$ can be reduced to a bus line planning problem, which is known to be NP-hard [39]. Thus, the single-objective problem $\mathcal{P}'_{bl}(\varepsilon)$ in general case is also NP-hard. Note that the model $\mathcal{P}'_{bl}(\varepsilon)$ is linear such that it can be solved by the commercial solvers like CPLEX.

7.3.1 Computation of Ideal and Nadir points

According the principle of the exact ε-constraint method, we first need to compute the ideal and nadir points to determine the range of ε. By Definitions 3 and 4, they are obtained by exactly solving the following four single-objective optimization problems.

$$\begin{aligned}
\mathcal{P}(f_1^I) : f_1^I = \min & \sum_{k\in K}\sum_{a\in A'} D_k(\tau_a u_a^k + \tau'_a v_a^k) + \sum_{k\in K} D_k P_T(\sum_{l\in L} x_l^k - 1) \\
\text{s.t.} \quad & \text{Constraints } (7.4) - (7.17) \\
\mathcal{P}(f_2^I) : f_2^I = \min & \sum_{a\in A} C_a z_a \\
\text{s.t.} \quad & \text{Constraints } (7.4) - (7.17)
\end{aligned}$$

The problem $\mathcal{P}(f_1^N)$(resp. $\mathcal{P}(f_2^N)$) is formed by adding to $\mathcal{P}(f_1^I)$ (resp. $\mathcal{P}(f_2^I)$) constraint (7.24) (resp. (7.25)) that fixes the optimal value of f_2 (resp. f_1).

$$\mathcal{P}(f_1^N) : f_1^N = \min \sum_{k\in K}\sum_{a\in A'} D_k(\tau_a u_a^k + \tau'_a v_a^k) + \sum_{k\in K} D_k P_T(\sum_{l\in L} x_l^k - 1)$$

Algorithm BLRP-LD

```
Transform model $\mathcal{P}'_{bl}$ into $\mathcal{P}'_{bl}(\varepsilon)$.
Set $\delta$ as the minimal unit value of $C_a, a \in A$ and $Y'_N = \varnothing$
Obtain $f^I = \{f_1^I, f_2^N\}$ and $f^N = \{f_1^N, f_2^I\}$ by exactly solving $\mathcal{P}(f_1^I)$, $\mathcal{P}(f_2^I)$, $\mathcal{P}(f_1^N)$
  and $\mathcal{P}(f_2^N)$.
Set $Y'_N = \{(f_1^I, f_2^N)\}$. Let $j = 2$ and $\varepsilon = f_2^N - \delta$, respectively.
while ($\varepsilon \leq f_1^N$) do
    Solve $\mathcal{P}'_{bl}(\varepsilon)$ exactly, and obtain the optimal solution and the corresponding ob-
      jective vector $(f_1(\varepsilon), f_2(\varepsilon))$;
    $\varepsilon = f_2(\varepsilon) - \delta$;
end while
Remove dominated points from $Y'_N$ (if existing) to obtain the Pareto front $Y_N$.
```

Fig. 7.2: Algorithm BLRP-LD: algorithm for the BLRP-LD.

$$\text{s.t.} \sum_{a \in A} C_a z_a = f_2^I \tag{7.24}$$

$$\text{and constraints } (7.4) - (7.17)$$

$$\mathcal{P}(f_2^I) : f_2^N = \min \sum_{a \in A} C_a z_a$$

$$\text{s.t.} \sum_{k \in K} \sum_{a \in A'} D_k(\tau_a u_a^k + \tau'_a v_a^k) + \sum_{k \in K} D_k P_T(\sum_{l \in L} x_l^k - 1) = f_1^I \tag{7.25}$$

$$\text{and constraints } (7.4) - (7.17)$$

Thus, the value of ε is bounded by $[f_2^I, f_2^N]$.

7.3.2 Defition of paramter δ

Parameter δ is defined as the minimum unit value of f_2 according to its definition in Chapter 4. Hence, the value of parameter δ is set to be the minimal unit value of $C_a, \forall a \in A$. In what follows, the exact ε-constraint method to find the Pareto front is outlined as Algorithm BLRP-LD.

7.4 Computational results

In this section, we report numerical experiments on a benchmark instance in [20] and randomly generated instances to evaluate the performance of the proposed algorithm. The proposed algorithm (i.e., Algorithm BLRP-LD) is coded in Visual C++ embedded with commercial software CPLEX (Version 12.6). All single-objective problems in Algorithm BLRP-LD are exactly solved by CPLEX. All the experiments are conducted in a PC with 2.5 GHz CPU and 2.95 GB RAM under Windows 7.

To check whether it is beneficial to reserve bus lanes for bus transit system, we will compare the travel time with bus lanes with that without bus lanes. To this aim, we first calculate the travel time of each OD pair k through bus transit system with bus lanes τ_k and that without bus lanes τ'_k, then the decreased rate of the travel time DR_k is computed as $DR_k = (\tau'_k - \tau_k)/\tau'_k$, which indicates how much the travel time reduces for OD pair k after using bus lanes for the bus transit system. In order to reflect the benefits of bus lanes for all OD pairs, the average decreased rate of all OD pairs is computed as:

$$DR = \frac{1}{|K|}\sum_{k\in K}\frac{\tau'_k - \tau_k}{\tau'_k} \tag{7.26}$$

Similarly, it can also examine how much the average travel time increases due to bus lane reservation for the private car travelers belonging to the same OD pairs. Let IR be the average increased rate of the travel time for private car travelers, which is computed as:

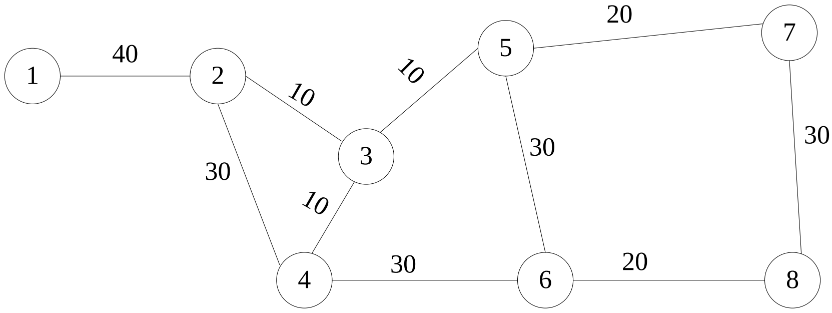
$$IR = \frac{1}{|K|}\sum_{k\in K}\frac{T_k - T'_k}{T'_k} \tag{7.27}$$

where T_k (resp. T'_k) denotes the travel time of OD pair k through private cars after (resp. without) reserving bus lanes. Note that T'_k is assumed to be the travel time via the corresponding shortest path and T_k is considered to be the travel time on the same path after reserving bus lanes. For the sake of convenience, let $|F|$ and No. denote the number of nondominated points on the Pareto front and the label of each solution, respectively, and let $CT_{\mathcal{P}'_{bl}}$ (resp. $CT_{\mathcal{P}''_{bl}}$) denote the computational time (CPU seconds) by Algorithm BLRP-LD in which all single-objective models are without (resp. with) valid inequalities.

7.4.1 A benchmark instance

We first test a small instance generated based on a benchmark instance for the bus line planning problem [20]. It has eight nodes, 20 arcs and four OD pairs. Fig. 7.3 shows the corresponding network graph G and OD pairs K where the link travel time τ'_a (resp. demand quantity D_k) is expressed in minutes (resp. trips per minute).

Like most previous studies concerning bus line planning problem [20], we generate the pool of all candidate lines by computing $\mathcal{K}$-shortest paths between the pair of nodes corresponding to each OD pair. To this aim, we use Yen's $\mathcal{K}$-shortest path algorithm [132]. The resulting bus line set L for this instance has 22 elements. Since



O	D	Demand
3	1	1.0
3	8	2.0
4	7	0.5
6	1	0.5

Fig. 7.3: Network of the benchmark instance

the work [20] has not considered the bus lane reservation. The remaining parameters related to lane reservation for the BLRP-LD are generated based on [42] as follows: τ_a is defined as $\phi_a \tau'_a$, where ϕ_a is randomly generated in the interval [0.5, 0.8] and the impact of a reserved lane on arc a is defined as $C_a = r_a \tau'_a$, where r_a is is randomly generated in the interval [0.2, 0.3]. The travel time of non-bus vehicles on arc a with a bus lane, denoted by τ''_a, is calculated as $\tau''_a = \tau'_a + C_a$. Three sets of parameter B: 200, 300 and 400 are respectively considered to show its impact on the output for this instance.

Table 7.1: Computational results for the benchmark instance

B	$\lvert F\rvert$	$CT_{\mathcal{P}'_{bl}}$	$CT_{\mathcal{P}''_{bl}}$	$CT_{\mathcal{P}''_{bl}}/CT_{\mathcal{P}'_{bl}}(\%)$
200	4	3.79	2.90	76.54
300	32	49.62	25.52	51.43
400	35	21.67	17.55	80.99

From Table 7.1, we can observe that the proposed algorithm can find the Pareto front of the benchmark instance under different scenarios within one minute. Moreover, it can be observed that $CT_{\mathcal{P}''_{bl}}$ is less than $CT_{\mathcal{P}'_{bl}}$ for each B. This indicates that the proposed valid inequalities are effective in accelerating the proposed algorithm. Besides, it can be also seen that the more the bus operating budget, i.e., the bigger the value of B, the more the number of nondominated solutions. This may be because that the solution space becomes lager as B increases. In the following, we will present the obtained nondiminated solutions under each B.

We can see in Table 7.2 that there exist four nondominated solutions for the benchmark instance with setting $B = 200$, and the minimal impact of reserved lanes, i.e., f_2^I, is 60.8, which implies that if without considering lane reservation, then no feasible solution exists due to the violation of constraint (7.3). Note that because

Table 7.2: Nondominated solutions of the benchmark instance with $B = 200$

No.	f_1	f_2	DR(%)	IR(%)
1	146.75	73.40	35.01	23.44
2	151.85	63.50	36.41	21.44
3	157.85	61.00	32.32	17.68
4	160.65	60.80	34.58	20.74

τ'_k is not available for bus transit system without bus lanes, here DR is computed according to (7.26) by replacing τ'_k with T'_k, since in this case the passengers can only travel by private car. It can be found Table 7.2 that DR ranges from 32.32% to 35.01%, while IR ranges from 17.68% to 23.44%.

Table 7.3: Nondominated solutions of the benchmark instance with $B = 300$

No.	f_1	f_2	DR(%)	IR(%)
1	142.15	40.00	41.12	25.19
2	143.50	37.10	39.62	23.38
3	148.90	35.60	36.62	21.63
4	150.20	31.00	35.26	22.69
5	152.90	28.10	33.01	21.63
6	157.55	25.60	30.60	18.93
7	163.65	25.10	28.83	17.75
8	168.15	23.50	26.71	9.35
9	168.30	22.60	26.42	15.81
10	169.90	22.50	26.40	15.04
11	172.30	20.40	27.29	11.29
12	174.55	20.00	23.99	13.10
13	176.30	19.10	28.14	17.13
14	176.95	17.90	24.88	9.35
15	180.95	16.60	25.73	15.18
16	187.05	16.10	23.96	14.00
17	187.70	14.90	20.70	6.22
18	191.70	13.60	21.54	12.06
19	193.30	13.50	22.91	11.29
20	197.45	13.40	20.11	8.17
21	197.95	11.00	19.11	9.35
22	202.10	10.90	17.70	6.22
23	204.05	10.50	18.73	8.17
24	205.75	9.90	9.94	3.13
25	208.70	8.00	16.32	6.22
26	216.50	6.90	5.76	0.00
27	220.50	5.60	6.61	5.83
28	222.10	5.50	6.59	5.07
29	226.25	5.40	5.18	1.94
30	226.75	3.00	5.57	3.13
31	230.90	2.90	2.76	0.00
32	232.85	2.50	3.80	1.94
33	237.50	0.00	0.00	0.00

In Tables 7.3 and 7.4, we can see that if without bus lanes, there still exist feasible solutions for the bus transit system, as the minimal impact of reserved lanes (f_2^I) is 0. It can be observed in Table 7.3 (resp. 7.4) that DR ranges from 0 to 41.12% (resp. 0 to 39.30%), while DR's in both tables range from 0 to 25.19%. Besides, for each nondominated solution in both tables, the value of DR is greater than that of IR. These results indicate that bus lane reservation can provide great benefits at a relatively low cost. To visually show the solutions, Fig. 7.4 gives the Pareto front of the benchmark instance with $B = 400$.

Table 7.4: Nondominated solutions of the benchmark instance with $B = 400$

No.	f_1	f_2	$DR(\%)$	$IR(\%)$	No.	f_1	f_2	$DR(\%)$	$IR(\%)$
1	142.15	40.00	39.30	25.19	19	185.20	14.90	19.31	6.22
2	143.50	37.10	37.62	23.38	20	189.20	13.60	22.02	12.06
3	145.00	34.40	36.78	22.16	21	190.80	13.50	21.82	11.29
4	147.55	33.70	36.30	23.44	22	194.95	13.40	18.72	8.17
5	147.70	31.00	36.22	22.69	23	195.45	11.00	19.41	9.35
6	148.90	30.80	34.62	21.63	24	199.60	10.90	16.31	6.22
7	150.40	28.10	33.78	20.88	25	201.55	10.50	17.34	8.17
8	155.05	25.60	31.37	18.93	26	203.25	9.90	8.85	3.13
9	161.15	25.10	29.30	17.75	27	206.20	8.00	14.93	6.22
10	165.65	23.50	25.62	9.35	28	214.00	6.90	4.38	0.00
11	165.80	22.60	26.89	15.81	29	218.00	5.60	7.08	5.83
12	167.40	22.50	21.16	15.04	30	219.60	5.50	6.89	5.07
13	169.80	20.40	26.20	11.29	31	223.75	5.40	3.79	1.94
14	172.05	20.00	24.29	13.10	32	224.25	3.00	4.48	3.13
15	173.80	19.10	28.91	17.13	33	228.40	2.90	1.38	0.00
16	174.45	17.90	23.79	9.35	34	230.35	2.50	2.41	1.94
17	178.45	16.60	26.50	15.18	35	235.00	0.00	0.00	0.00
18	184.55	16.10	24.43	14.00					

7.4.2 Randomly generated instances

To further evaluate the proposed algorithm, we also test larger-size randomly generated instance sets. Each set includes five instances, which are generated in the following way. The network graph $G(N, A)$ is generated based on Waxman's network model [121]. OD pairs are randomly selected from node set N and the demand quantity of each OD pair is generated in the interval [1, 5] trips/minute [20]. The travel time on each arc, τ_a, is randomly generated in the interval [2, 10] minutes. The pool of all candidate lines is also obtained by computing $\mathcal{K}$-shortest paths between the pair of node corresponding to each OD pair with Yen's $\mathcal{K}$-shortest path algorithm [132]. The network G' is constructed according to the candidate lines. Parameters τ_a and C_a are generated in the same way as in the previous subsection. B is defined as $2r_b \sum_{k\in K} l(o_k, d_k)$, where $l(o_k, d_k)$ denotes the travel duration from o_k to d_k on a non-reserved shortest path and r_b is generated from [0.5, 0.8]. The computational results are reported in Table 7.5. Due to space limitations, the information of the nondominated solutions for each instance are not presented here, but

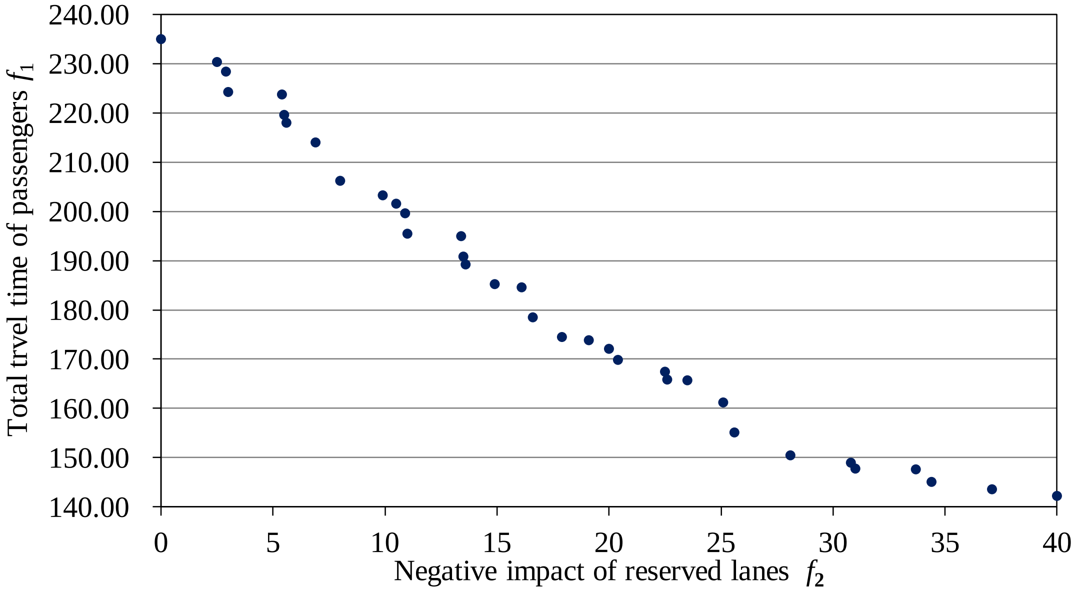


Fig. 7.4: Pareto front of the benchmark instance with $B = 400$

they can be similarly analyzed, as in the previous subsection. In fact, this subsection mainly focuses on evaluating the computational efficiency and ability of the proposed algorithm for larger-size instances. Each value for each set in the result table is its average value of five instances and the computational time of the proposed algorithm for each instance is limited to 18000s.

Table 7.5: Computational results for different sized instances

Set	$\|N\|$	$\|A\|$	$\|K\|$	$\|F\|$	$CT_{\mathcal{P}'_{bl}}$	$CT_{\mathcal{P}''_{bl}}$	$CT_{\mathcal{P}''_{bl}}/CT_{\mathcal{P}'_{bl}}(\%)$
1	9	26	4	10.60	11.69	11.50	98.36
2	9	26	5	43.20	538.16	410.13	76.21
3	10	30	5	56.20	584.25	381.74	65.34
4	10	30	6	27.60	315.14	266.46	84.55
5	11	30	5	23.60	507.87	381.60	75.14
6	11	30	6	45.00	4763.95	2935.25	61.61
7	12	34	6	37.80	14773.87	5428.24	36.74
Average					3070.70	1402.13	71.14
8	12	34	7	-	-	-	-

From Table 7.5, it can be seen that the proposed algorithm can find the Pareto fronts for problem sets 1-7 within given computational time. We can also observe that $CT_{\mathcal{P}''_{bl}}$ is less than $CT_{\mathcal{P}'_{bl}}$ over all sets 1-7 and the former is only 71.14% of the latter on average, which indicates that the proposed valid inequalities are effective in reducing the computational time of the proposed algorithm. Furthermore, it can be found from Fig. 7.5 that the computational time spent by the proposed algorithm rapidly increases with the problem size due to its NP-hardness. We note that the proposed algorithm fails to find the Pareto front for problem set 8 within 18000s.

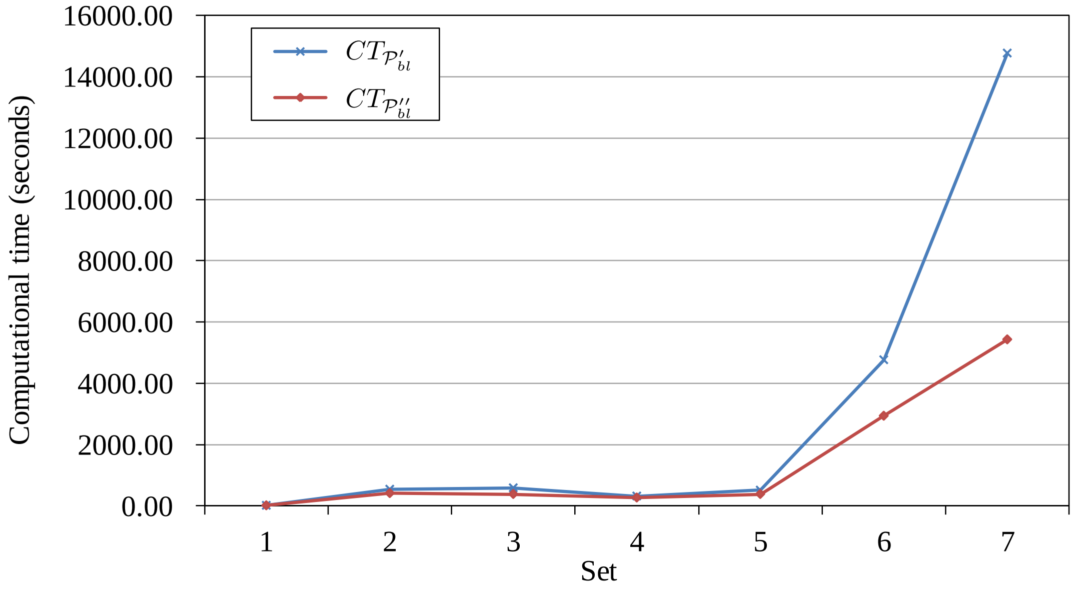


Fig. 7.5: Computational results for different sized instances

Due to a restriction of limited research time, the study on the BLRP-LD is in its beginning step. The preliminary computational results shows that the proposed algorithm can obtain Pareto fronts for small-size problems. However, as the size of problem increases, it becomes very difficult to solve. Therefore, it is necessary to explore more properties for the problem to speed up its resolution and develop more efficient approaches for larger-size problems in the future.

7.5 Conclusions

In this chapter, we have studied a new bus lane reservation and bus line integrated optimization problem, whose optimization objectives are to minimize the total travel time of all passengers and to minimize the total negative impact of all reserved lanes. For the considered problem, we first developed a bi-objective mixed-integer nonlinear program and then transformed it to be an equivalent linear one by reformulating its nonlinear constraints. Some valid inequalities were proposed to reduce the search space. The problem was shown to be NP-hard. Then, an exact ε-constraint method was proposed to derive the Pareto front. Preliminary computational result showed that the proposed method can efficiently find the Pareto front of small-size problems and the derived valid inequalities are useful in accelerating its resolution. However, it becomes quite difficult to solve large-size instances. More efficient methods should be developed in the future.

Chapter 8

Conclusions and perspectives

In this thesis, we have investigated two categories of lane reservation problems, which aim to meet special transportation requirements and improve the performance of bus transit systems via optimally reserving lanes at a macroscopic network level, respectively. For all the studied problems, appropriate mathematical models are formulated and all studied problems are proved to be NP-hard. Moreover, according to their characteristics and the derived problem properties, appropriate resolution methods, including two-phase algorithm, improved ε-constraint method, cut-and-solve method, and kernel search, are devised. Computational results on benchmarks and randomly generated instances demonstrate that the proposed algorithms outperform the state-of-the-art algorithms and commercial software CPLEX.

First of all, we investigated two lane reservation problems (LRPs) for meeting special transportation requirements. The first one is large-size LRP for future automated truck freight transportation. For the problem, an improved integer linear program (ILP) was provided. Its several special cases was identified to be classical combinatorial optimization problems. Based on the derived properties, a two-phase exact algorithm was developed. Computational results on benchmark and newly generated larger-size instances indicated that the proposed algorithm significantly outperforms the state-of-the-art method, since the computational time by our algorithm is only 11.01% and 37.54% of that by the state-of-the-art algorithm for benchmark instances under two scenarios, respectively, and our algorithm can solve instances with up to 700 nodes and 55 tasks, while the state-of-the-art method failed to solve instances with 180 nodes and 40 tasks. The second one is a robust lane reservation problem (RLRP) for large-scale special events, which extends the existing LRPs by considering the uncertain traffic features. For the problem, a bi-objective ILP was first presented. To derive its Pareto front, an improved exact ε-constraint and cut-and-solve combined method was devised. Computational results on an instance based on a real network topology and randomly generated instances showed that the proposed algorithm is more efficient than CPLEX, since it spends 37% computational time of CPLEX for different sized problems.

Then, we investigated three bus lane reservation problems (BLRPs) in different application contexts for improving the performance of bus transit systems. We first studied a BLRP where bus lines and paths are assumed to be predetermined. An ILP was formulated and an exact cut-and-solve method with an improved piercing cut strategy was devised. Computational results indicated that the proposed algorithm is more efficient than CLPEX, since the proposed method spends an average of 73% computational time of CPLEX for different sized problems. The BLRP is extended to the BLRP with path design (BLRP-PD) where bus lines with their stations are known but the bus paths need to be determined. For the BLRP-PD, we formulated an ILP and valid inequalities were explored for it to reduce the search space. Finally, an enhanced exact cut-and-solve method with new piecing cut generation strategy and an improved kernel search method were devised. Computational results on randomly generated instances confirmed the effectiveness of the proposed approaches. Finally, we studied a BLRP with line design (BLRP-LD) where both bus lines need to be determined along with bus lane reservation. For the BLRP-LD, a bi-objective mixed-integer nonlinear program was first formulated, which was then equivalently transformed to be an linear one. Several valid inequalities were added to reduce its search space for Pareto optimal solutions. Its complexity was analyzed. An exact ε-constraint method was proposed to obtain its Pareto front. Preliminary computational results indicated that the proposed algorithm can exactly solve small-size problems. However, the computational time increases rapidly as the size of the problem increases.

Although models and methods are developed in this thesis for several lane reservation problems, there is still much work to do in our future research.

First of all, one of the main concerns for the investigated LRPs is to optimally reserve lanes in transportation networks to minimize their negative impact on non-reserved traffic. In this thesis, the negative impact of a reserved lane is assumed as a parameter. In practical situations, the estimation for this parameter is quite challenging because it is related with multiple factors, such as road traffic condition, type of the reserved lanes, influence among road links and periods of lane reservation. Due to its complexity, up to now, researchers have not come to a consensus on how to evaluate the negative impact caused by reserved lanes. Such issue deserves further and systemic studies in the future and the research results will provide valuable information for our future research.

Although the proposed algorithm for each problem has achieved relatively good performance, the computational time increases with the size of the problem because of its NP-hardness. Therefore, an important future direction is to further exploit the properties of the studied problem and develop more efficient and effective problem-specific algorithms. For example, we can observe for the LRP in Chapter 3 that it becomes very difficult to solve the problem $\mathcal{P}_l''$ in the second phase with a direct use of CPLEX for large-size problems, one direction may be to develop more efficient methods for $\mathcal{P}_l''$ to further accelerate the proposed two-phase exact algorithm and for the RLRP in Chapter 4, more efficient strengthening techniques need to be studied to increase the efficiency of the proposed ε-constraint and cut-and-solve combined method. Especially, it becomes much more difficult to solve large-size bus lane reservation and bus line planning integrated optimization problem (i.e., BLRP-LD in Chapter 7), efficient and effective multi-objective evolutionary algorithms may be developed by exploring the properties of the BLRP-LD to yield well-distributed nondominated solutions for larger-size problems within a short time.

Moreover, we may extend our studied problems from the following aspects:

1) In reality, the link traffic condition may vary with time period of a day. The concept of robustness was introduced for the RLRP in Chapter 4 to cope with the uncertain traffic conditions from the “robustness” perspective. In our future research, we may introduce the concept of lane reservation robustness to the other LRPs. On the other hand, we may directly formulate such issue to expand the proposed formulations, giving lane reservation problems with time-varying traffic condition.

2) In this thesis, we considered optimally deciding which lanes in transportation networks to be reserved. However, deciding appropriate time intervals of implementing lane reservation was not considered. It has been suggested by many researches that intermittent bus lanes would greatly reduce their negative impact on normal traffic. Therefore, we may extend the investigated problems by considering the temporal decision of lane reservation.

3) Lane reservation may generate other impact, such as traffic diversion and mode shift and considering them into the studied BLRPs will be another research future direction. Besides, traffic congestion results in not only inefficient transportation but also environment related issues, e.g., the increase in CO_2 emissions. The final goal of bus lane reservation is to alleviate traffic congestion by providing reliable and rapid bus transit, thereby attracting more people to travel by buses instead of private cars.

Thus, the total CO_2 emissions may be reduced. In our future work, we can add the CO_2 emission evaluation to the studied LRPs.

4) Because of the complexity of the integrated optimization of bus lane reservation and bus line planning, it is assumed for the BLRP-LD in Chapter 7 that: 1) the passengers belonging to a same OD pair choose the same shortest paths to arrive at their destination; and 2) the buses on each line are sufficient enough to support passengers to complete their trips. We can relax the above assumptions by introducing passenger transit assignment and considering bus frequency and capacity in the future research.

Bibliography

[1] High-occupancy vehicle lane. `https://en.wikipedia.org/wiki/High-occupancy_vehicle_lane`.

[2] H. A Abbass, R. Sarker, and C. Newton. Pde: a pareto-frontier differential evolution approach for multi-objective optimization problems. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, volume 2, pages 971–978. IEEE, 2001.

[3] U. Akinc and Basheer M Khumawala. An efficient branch and bound algorithm for the capacitated warehouse location problem. *Management Science*, 23(6):585–594, 1977.

[4] H. Allaoui and A. Artiba. Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints. *Computers & Industrial Engineering*, 47(4):431–450, 2004.

[5] E. Angelelli, R. Mansini, and M. Speranza. Kernel search: A general heuristic for the multi-dimensional knapsack problem. *Computers & Operations Research*, 37(11):2017–2026, 2010.

[6] E. Angelelli, R. Mansini, and M. Speranza. Kernel search: A new heuristic framework for portfolio selection. *Computational Optimization and Applications*, 51(1):345–361, 2012.

[7] V. Arasan and P Vedagiri. Microsimulation study of the effect of exclusive bus lanes on heterogeneous traffic flow. *Journal of Urban Planning and Development*, 136(1):50–58, 2010.

[8] V. T. Arasan and P. Vedagiri. Bus priority on roads carrying heterogeneous traffic: a study using computer simulation. *European Journal of Transport and Infrastructure Research*, 8(1):45–64, 2008.

[9] V. T. Arasan and P. Vedagiri. Study of the impact of exclusive bus lane under highly heterogeneous traffic condition. *Public Transport*, 2(1–2):135–155, 2010.

[10] M. H. Baaj and H. S Mahmassani. An ai-based approach for transit route system planning and design. *Journal of advanced transportation*, 25(2):187–209, 1991.

[11] E. Balas. A note on the branch-and-bound principle. *Operations Research*, 16(2):442–445, 1968.

[12] P. Baptiste, J. Carlier, and A. Jouglet. A branch-and-bound procedure to minimize total tardiness on one machine with arbitrary release dates. *European Journal of Operational Research*, 158(3):595–608, 2004.

[13] P. Baptiste and A. Jouglet. On minimizing total tardiness in a serial batching problem. *RAIRO-Operations Research-Recherche Opérationnelle*, 35(1):107–115, 2001.

[14] R. Bellman. Dynamic programming and lagrange multipliers. *Proceedings of the National Academy of Sciences*, 42(10):767–769, 1956.

[15] J. Bérubé, M. Gendreau, and J. Potvin. An exact ε-constraint method for bi-objective combinatorial optimization problems: Application to the traveling salesman problem with profits. *European Journal of Operational Research*, 194(1):39–50, 2009.

[16] J. Black. Urban arterial road demand management-environment and energy, with particular reference to public transport priority. In *Road Demand Management Seminar, 1991, Melbourne, Australia*, number AP 10/91, 1991.

[17] J. Black. Strategic transport planning, demand analysis of transport infrastructure and transport services for the 27th summer, olympiad held in sydney, australia, 2000. *Journal of Transportation Engineering and Information*, 2(2):14–30, 2004.

[18] E. Bouillet. *Path routing in mesh optical networks*. John Wiley & Sons, 2007.

[19] D. Branston. Link capacity functions: A review. *Transportation Research*, 10(4):223–236, 1976.

[20] H. Cancela, A. Mauttone, and M. Urquhart. Mathematical programming formulations for transit network design. *Transportation Research Part B: Methodological*, 77:17–37, 2015.

[21] A. Ceder and N. Wilson. Bus network design. *Transportation Research Part B: Methodological*, 20(4):331–344, 1986.

[22] P. Chakroborty and T. Wivedi. Optimal route network design for transit systems using genetic algorithms. *Engineering Optimization*, 34(1):83–100, 2002.

[23] A. Che and C. Chu. Cyclic hoist scheduling in large real-life electroplating lines. *OR Spectrum*, 29(3):445–470, 2007.

[24] A. Che, P. Wu, F. Chu, and M. Zhou. Improved quantum-inspired evolutionary algorithm for large-size lane reservation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(12):1535–1548, 2015.

[25] K. Chebil and M. Khemakhem. A dynamic programming algorithm for the knapsack problem with setup. *Computers & Operations Research*, 64:40–50, 2015.

[26] X. Chen, L. Yu, L. Zhu, J. Guo, and M. Sun. Microscopic traffic simulation approach to the capacity impact analysis of weaving sections for the exclusive bus lanes on an urban expressway. *Journal of Transportation Engineering*, 136(10):895–902, 2010.

[27] D. Choi and W. Choi. Effects of an exclusive bus lane for the oversaturated freeway in Korea. In *Institute of Transportation Engineers 65th Annual Meeting*, 1995.

[28] C. Chu, F. Chu, M. Zhou, H. Chen, and Q. Shen. A polynomial dynamic programming algorithm for crude oil transportation planning. *IEEE Transactions on Automation Science and Engineering*, 9(1):42–55, 2012.

[29] J. Climaco and E. Martins. A bicriterion shortest path algorithm. *European Journal of Operational Research*, 11(4):399–404, 1982.

[30] S. Climer and W. Zhang. Cut-and-solve: An iterative search strategy for combinatorial optimization problems. *Artificial Intelligence*, 170(8-9):714–738, 2006.

[31] T. Crainic. *Long-haul freight transportation.* Springer, 2003.

[32] T. Crainic and G. Laporte. Planning models for freight transportation. *European journal of operational research*, 97(3):409–438, 1997.

[33] J. Dahlgren. High occupancy vehicle lanes: Not always more effective than general purpose lanes. *Transportation Research Part A: Policy and Practice*, 32(2):99–114, 1998.

[34] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[35] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[36] M. Dorigo, M. Birattari, and T. Stützle. Ant colony optimization. *Computational Intelligence Magazine, IEEE*, 1(4):28–39, 2006.

[37] M. Ehrgott. *Multicriteria optimization*. Springer Science & Business Media, 2006.

[38] M. Eichler and C. Daganzo. Bus lanes with intermittent priority: Strategy formulae and an evaluation. *Transportation Research Part B: Methodological*, 40(9):731–744, 2006.

[39] L. Fan and C. Mumford. A metaheuristic approach to the urban transit routing problem. *Journal of Heuristics*, 16(3):353–372, 2010.

[40] L. Fan and C. L Mumford. A metaheuristic approach to the urban transit routing problem. *Journal of Heuristics*, 16(3):353–372, 2010.

[41] Y. Fang. *Study of lane reservation problems in a transportation network*. PhD thesis, Ph.D. dissertation, Université de Technologie de Troyes, France, 2013.

[42] Y. Fang, F. Chu, S. Mammar, and A. Che. An optimal algorithm for automated truck freight transportation via lane reservation strategy. *Transportation Research Part C: Emerging Technologies*, 26:170–183, 2013.

[43] Y. Fang, F. Chu, S. Mammar, and A. Che. A cut-and-solve-based algorithm for optimal lane reservation with dynamic link travel times. *International Journal of Production Research*, 52(4):1003–1015, 2014.

[44] Y. Fang, F. Chu, S. Mammar, and Q. Shi. A new cut-and-solve and cutting plane combined approach for the capacitated lane reservation problem. *Computers & Industrial Engineering*, 80:212–221, 2015.

[45] Y. Fang, F. Chu, S. Mammar, and M. Zhou. Optimal lane reservation in transportation network. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):482–491, 2012.

[46] J. Feng, A. Che, and N. Wang. Bi-objective cyclic scheduling in a robotic cell with processing time windows and non-euclidean travel times. *International Journal of Production Research*, 52(9):2505–2518, 2014.

[47] T. Feo and M. Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8(2):67–71, 1989.

[48] K. Fleszar, I. Osman, and K. Hindi. A variable neighbourhood search algorithm for the open vehicle routing problem. *European Journal of Operational Research*, 195(3):803–809, 2009.

[49] M. Frutos, A. C. Olivera, and F. Tohmé. A memetic algorithm based on a NSGAII scheme for the flexible job-shop scheduling problem. *Annals of Operations Research*, 181(1):745–765, 2010.

[50] C. Fuhs and J. Obenberger. Development of high-occupancy vehicle facilities: Review of national trends. *Transportation Research Record*, 1781:1–9, 2002.

[51] A. Gan, H. Yue, I. Ubaka, and F. Zhao. Development of operational performance and decision models for arterial bus lanes. *Transportation Research Record: Journal of the Transportation Research Board*, (1858):18–30, 2003.

[52] M. Gendreau, M. Iori, G. Laporte, and S. Martello. A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks*, 51(1):4–18, 2008.

[53] F. Glover. Tabu search-part I. *ORSA Journal on computing*, 1(3):190–206, 1989.

[54] G. Guastaroba and M. Speranza. Kernel search: An application to the index tracking problem. *European Journal of Operational Research*, 217(1):54–68, 2012.

[55] G. Guastaroba and M. Speranza. Kernel search for the capacitated facility location problem. *Journal of Heuristics*, 18(6):877–917, 2012.

[56] G. Guastaroba and M. Speranza. A heuristic for bilp problems: the single source capacitated facility location problem. *European Journal of Operational Research*, 238(2):438–450, 2014.

[57] Y. Haimes, L. Ladson, and D. Wismer. Bicriterion formulation of problems of integrated system identification and system optimization, 1971.

[58] D. Hidalgo and L. Gutiérrez. Brt and bhls around the world: Explosive growth, large positive impacts and many issues outstanding. *Research in Transportation Economics*, 39(1):8–13, 2013.

[59] J.H. Holland. *Adaptation in natural and artificial systems.* Ann Arbor, MI: University of Michigan Press, 1975.

[60] E. Ignall and L. Schrage. Application of the branch and bound technique to some flow-shop scheduling problems. *Operations Research*, 13(3):400–412, 1965.

[61] D. Jepson and L. Ferreira. Assessing travel time impacts of measures to enhance bus operations. part ii: Assessment criteria and main findings. *Road and Transport Research: a journal of Australian and New Zealand research and practice*, 9(1):3–18, 2000.

[62] L. Jian. The study of install condi tions of bus lane. *Communications Standardization*, 1:021, 2003.

[63] M. P John, C. L Mumford, and R. Lewis. An improved multi-objective algorithm for the urban transit routing problem. In *Evolutionary Computation in Combinatorial Optimisation*, pages 49–60. Springer, 2014.

[64] N. Jozefowiez, F. Semet, and E. Talbi. The bi-objective covering tour problem. *Computers & operations research*, 34(7):1929–1942, 2007.

[65] S. Karim. The effect of bus lane on the travel time of other modes using floating car method. In *Proceedings of the Eastern Asia Society for Transportation Studies*, pages 135–149, Fukuoka, Japan, Oct. 2003.

[66] R.M. Karp. *Reducibility among combinatorial problems.* Complexity of Computer Computations. Plenum Press, New York, 1972.

[67] J. Kennedy. Particle swarm optimization. In *Encyclopedia of machine learning*, pages 760–766. Springer, 2011.

[68] H. Khoo, L. Teoh, and Q. Meng. A bi-objective optimization approach for exclusive bus lane selection and scheduling design. *Engineering Optimization*, 46(7):987–1007, 2014.

[69] F. Kılıç and M. Gök. A demand based route generation algorithm for public transit network design. *Computers & Operations Research*, 51:21–29, 2014.

[70] S. Kirkpatrick, M. Vecchi, et al. Optimization by simmulated annealing. *science*, 220(4598):671–680, 1983.

[71] J. Kwon and P. Varaiya. Effectiveness of California's high occupancy vehicle (HOV) system. *Transportation Research Part C: Emerging Technologies*, 16(1):98–115, 2008.

[72] A. Land and A. Doig. An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society*, pages 497–520, 1960.

[73] E. L. Lawler and D. E. Wood. Branch-and-bound methods: A survey. *Operations Research*, 14(4):699–719, 1966.

[74] D. Lee, J. Chen, and J. Cao. The continuous berth allocation problem: A greedy randomized adaptive search solution. *Transportation Research Part E: Logistics and Transportation Review*, 46(6):1017–1029, 2010.

[75] M. Leitner, I. Ljubić, and M. Sinnl. Solving the bi-objective prize-collecting steiner tree problem with the ε-constraint method. *Electronic Notes in Discrete Mathematics*, 41:181–188, 2013.

[76] S. Li and Y. Ju. Evaluation of bus-exclusive lanes. *IEEE Transactions on Intelligent Transportation Systems*, 10(2):236–245, 2009.

[77] T. Li, J. Ding, Y. Sun, and W. He. Traffic condition for bus lane setting on urban arteries. *Journal of Kunming University of Science and Technology: Science and Technology*, 35(1):56–60, 2010.

[78] J. Little, KG Murty, D. W Sweeney, and C. Karel. An algorithm for the traveling salesman problem. *Operations Research*, 11(6):972–989, 1963.

[79] M. Liu, F. Chu, Z. Zhang, and C. Chu. A polynomial-time heuristic for the quay crane double-cycling problem with internal-reshuffling operations. *Transportation Research Part E: Logistics and Transportation Review*, 81:52–74, 2015.

[80] M. Liu, S. Wang, C. Chu, and F. Chu. An improved exact algorithm for single-machine scheduling to minimise the number of tardy jobs with periodic maintenance. *International Journal of Production Research*, pages 1–12, 2015.

[81] R. B. Machemehl, T. W. Rioux, A. Tsyganov, and P. Poolman. Freeway operational flexibility concepts. Technical report, NO. 1844-1, Center for Transportation Research, University of Texas at Austin, Austin, Texas, 2001.

[82] C Maclennan. Priority for public transport and other high occupancy vehicles on urban roads. *Routes/Roads*, Special II(10.07 A):5–38, 1995.

[83] S. K. Mandal, D. Pacciarelli, A. Løkketangen, and G. Hasle. A memetic NSGA-II for the bi-objective mixed capacitated general routing problem. *Journal of Heuristics*, 21(3):359–390, 2015.

[84] P. Martin, J. Perrin, R. Lambert, and P. Wu. Evaluate effectiveness of high occupancy vehicle (HOV) lanes. Technical report, NO. UTL-1001-48, Civil & Environmental Engineering Department, University of Utah, Salt Lake City, Utah, 2002.

[85] G. Mavrotas. Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. *Applied mathematics and computation*, 213(2):455–465, 2009.

[86] M. Mesbah, M. Sarvi, and G. Currie. Optimization of transit priority in the transportation network using a genetic algorithm. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):908–919, 2011.

[87] M. Mesbah, M. Sarvi, I. Ouveysi, and G. Currie. Optimization of transit priority in the transportation network using a decomposition methodology. *Transportation Research Part C: Emerging Technologies*, 19(2):363–373, 2011.

[88] K. Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media, 2012.

[89] L. Mitten. Branch-and-bound methods: General formulation and properties. *Operations Research*, 18(1):24–34, 1970.

[90] N. Mladenović and P. Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100, 1997.

[91] M. Nikolić and D. Teodorović. Transit network design by bee colony optimization. *Expert Systems with Applications*, 40(15):5945–5955, 2013.

[92] I. Osman and G. Laporte. Metaheuristics: A bibliography. *Annals of Operations Research*, 63(5):511–623, 1996.

[93] Q. Pan, M. Tasgetiren, and Y. Liang. A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem. *Computers & Operations Research*, 35(9):2807–2839, 2008.

[94] F Pezzella, G Morganti, and G Ciaschetti. A genetic algorithm for the flexible job-shop scheduling problem. *Computers & Operations Research*, 35(10):3202–3212, 2008.

[95] D. Pisinger. A minimal algorithm for the 0-1 knapsack problem. *Operations Research*, 45(5):758–767, 1997.

[96] J. Princeton and S. Cohens. Impact of a dedicated lane on the capacity and the level of service of an urban motorway. *Procedia Social and Behavioral Sciences*, 16:196–206, 2011.

[97] P. Reiter and W. Gutjahr. Exact hybrid algorithms for solving a bi-objective vehicle routing problem. *Central European Journal of Operations Research*, 20(1):19–43, 2012.

[98] M. Reyes-Sierra and C. Coello. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International journal of computational intelligence research*, 2(3):287–308, 2006.

[99] S. Schijns and P. Eng. Brisbane, Australia–HOV metropolis? In *10th International Conference on High–Occupancy Vehicle Systems*, Dallas, Texas, Aug. 2000.

[100] A. Schöbel. Line planning in public transportation: models and methods. *OR spectrum*, 34(3):491–510, 2012.

[101] A. Schöbel and S. Scholl. Line planning with minimal traveling time. In *OASIcs-OpenAccess Series in Informatics*, volume 2. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2006.

[102] Y. Seo, J. Park, H. Jang, and Y. Lee. A study on setting-up a methodology and criterion of exclusive bus lane in urban area. In *Proceedings of the Eastern Asia Society for Transportation Studies*, volume 5, pages 339–351, 2005.

[103] A. S. Shalaby. Simulating performance impacts of bus lanes and supporting measures. *Journal of transportation engineering*, 125(5):390–397, 1999.

[104] H. Shih, M.and Mahmassani. A design methodology for bus transit networks with coordinated operations. Technical report, 1994.

[105] S. E. Shladover. Truck automation operational concept alternatives. In *2010 IEEE Intelligent Vehicles Symposium*, pages 1072–1077, San Diego, USA, Jun. 2010.

[106] N. Smith and D. Hensher. The future of exclusive busways: the Brazilian experience. *Transport Reviews*, 18(2):131–152, 1998.

[107] E. Sullivan and M. Burris. Benefit-cost analysis of variable pricing projects: SR-91 express lanes. *Journal of transportation engineering*, 132(3):191–198, 2006.

[108] X. Sun, H. Lu, and Y. Fan. Optimal bus lane infrastructure design. *Transportation Research Record: Journal of the Transportation Research Board*, (2467):1–11, 2014.

[109] J. Surprenant-Legault and A. El-Geneidy. Introduction of reserved bus lane: Impact on bus running time and on-time performance. *Transportation Research Record: Journal of the Transportation Research Board*, (2218):10–18, 2011.

[110] L. Tang and X. Wang. A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 17(1):20–45, 2013.

[111] K. F. Turnbull. HOV project case studies: history and institutional arrangements. Technical report, NO. DOT-T-92-13, Texas Transportation Institute, Texas A & M University System, College Station, Texas, 1990.

[112] K. F. Turnbull. HOV and HOT lanes in the United States. In *Proceedings of European Transport Conference*, Strasbourg, France, Sept. 2005.

[113] K. F. Turnbull and T. DeJohn. New Jersey I-80 and I-287 HOV lane case study. Technical report, NO. FHWAOP-00-018, Texas Transportation Institute, Texas A & M University System, College Station, Texas, 2000.

[114] J. Viegas and B. Lu. The intermittent bus lane signals setting within an area. *Transportation Research Part C: Emerging Technologies*, 12(6):453–469, 2004.

[115] J. Viegas, R. Roque, B. Lu, and J. Vieira. Intermittent bus lane system: demonstration in lisbon, portugal.

[116] F. Vincent, S. Lin, W. Lee, and C. Ting. A simulated annealing heuristic for the capacitated location routing problem. *Computers & Industrial Engineering*, 58(2):288–299, 2010.

[117] C. Vira and Y. Haimes. *Multiobjective decision making: theory and methodology.* Number 8. North-Holland, 1983.

[118] C. von Lücken, B. Barán, and C. Brizuela. A survey on multi-objective evolutionary algorithms for many-objective problems. *Computational Optimization and Applications*, 58(3):707–756, 2014.

[119] J. Walteros, A. Medaglia, and G. Riaño. Hybrid algorithm for route design on bus rapid transit systems. *Transportation Science*, 49(1):66–84, 2013.

[120] H. Wang. A two-phase ant colony algorithm for multi-echelon defective supply chain network design. *European Journal of Operational Research*, 192(1):243–252, 2009.

[121] B. M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, 1988.

[122] L. Wei and T. Chong. Theory and practice of bus lane operation in Kunming. *DISP*, 151:68–72, 2002.

[123] R. Wong. A dual ascent approach for steiner tree problems on a directed graph. *Mathematical programming*, 28(3):271–287, 1984.

[124] P. Wu, A. Che, and F. Chu. A quantum evolutionary algorithm for lane reservation problem. In *2013 10th IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pages 264–268. IEEE, 2013.

[125] Y. Wu, C. Chu, F. Chu, and N. Wu. Heuristic for lane reservation problem in time constrained transportation. In *Proceedings of Automation Science and Engineering*, pages 543–548, Bangalore, India, Aug. 2009.

[126] Y. Wu and N. Wu. An approximate algorithm for the lane reservation problem in time constrained transportation. In *2010 2nd International Conference on Advanced Computer Control (ICACC)*, pages 192–196, Shenyang, China, Mar. 2010.

[127] Y. Xiao, K. Thulasiraman, G. Xue, and A. Jüttner. The constrained shortest path problem: algorithmic approaches and an algebraic study with generalization. *AKCE International Journal of Graphs and Combinatorics*, 2(2):63–86, 2005.

[128] H. Yang and W. Wang. An innovative dynamic bus lane system and its simulation-based performance investigation. In *Proceedings of Intelligent Vehicles Symposium*, Xi'an, China, Jun. 2009.

[129] Z. Yang, F. Chu, and H. Chen. A cut-and-solve based algorithm for the single-source capacitated facility location problem. *European Journal of Operational Research*, 221(3):521–532, 2012.

[130] B. Yao, P. Hu, X. Lu, J. Gao, and M. Zhang. Transit network design based on travel time reliability. *Transportation Research Part C: Emerging Technologies*, 43:233–248, 2014.

[131] J. Yao, F. Shi, Z. Zhou, and J. Qin. Combinatorial optimization of exclusive bus lanes and bus frequencies in multi-modal transportation network. *Journal of Transportation Engineering*, 138(12):1422–1429, 2012.

[132] J. Yen. Finding the K shortest loopless paths in a network. *management Science*, 17(11):712–716, 1971.

[133] L. Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, 8(1):59–60, 1963.

[134] E. Zagorianakos. Athens 2004 olympic games' transportation plan: a missed opportunity for strategic environmental assessment (sea) integration? *Journal of Transport Geography*, 12(2):115–125, 2004.

[135] W. Zhang, Y. Huang, and G. Hu. Study on design standard for urban bus-lane. *Communications Standardization*, 7:33–36, 2003.

[136] H. Zhao, R. Jiang, et al. The memetic algorithm for the optimization of urban transit network. *Expert Systems with Applications*, 42(7):3760–3773, 2015.

[137] Z. Zhou. *Study on Optimization of Hazardous Material Transportation via Lane Reservation*. PhD thesis, Ph.D. dissertation, Université d'Évry Val d'Essonne, France, 2014.

[138] Z. Zhou, A. Che, F. Chu, and C. Chu. Model and method for multiobjective time-dependent hazardous material transportation. *Mathematical Problems in Engineering*, 2014, 2014.

[139] Z. Zhou, F. Chu, A. Che, and M. Zhou. ε-constraint and fuzzy logic-based optimization of hazardous material transportation via lane reservation. *IEEE Transactions on Intelligent Transportation Systems*, 14(2):847–857, 2013.

[140] H. Zhu. Numerical study of urban traffic flow with dedicated bus lane and intermittent bus lane. *Physica A: Statistical Mechanics and its Applications*, 389(16):3134–3139, 2010.

[141] E. Zitzler, M. Laumanns, L. Thiele, E. Zitzler, E. Zitzler, L. Thiele, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm, 2001.

My publications

Journal papers

P. Wu, A. Che, F. Chu, and M. Zhou. An improved exact ε-constraint and cut-and-solve combined method for bi-objective robust lane reservation. *IEEE Transactions on Intelligent Transportation Systems*, 16(3): 1479-1492, 2015.

A. Che, P. Wu, F. Chu, and M. Zhou. Improved quantum-inspired evolutionary algorithm for large-size lane reservation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(12): 1535-1548, 2015.

P. Wu, F. Chu, and A. Che. A fast two-phase exact algorithm for large-size lane reservation in automated truck freight transportation. *International Journal of Production Research, submitted to publish*, 2016.

P. Wu, F. Chu, and A. Che. Exact and heuristic algorithms for rapid and station arrival-time guaranteed bus transit via lane reservation. *IEEE Transactions on Intelligent Transportation Systems, submitted to publish*, 2015.

Conference papers

P. Wu, F. Chu, and A. Che. Mixed-integer programming for a new bus-lane reservation problem. In *Proceedings of 2015 18th IEEE International Conference on Intelligent Transportation Systems*, pages 2782-2787, Spain, Sep. 2015.

P. Wu, A. Che, and F. Chu. A heuristic for bus-lane reservation problem. In *Proceedings of 2015 Congrès Annuel de la Société Francaise de Recherche Opèrationnelle et Aide à la Décisions*, Marseille, France, Feb. 2015.

P. Wu, F. Chu, A. Che, and Q. Shi. A bus lane reservation problem in urban bus transit network. In *Proceedings of 2014 17th IEEE International Conference on Intelligent Transportation Systems*, pages 2864-2869, Qingdao, China, Oct. 2014.

P. Wu, A. Che, and F. Chu. Bi-objective optimization of lane reservation in transportation network. In *Proceedings of 2014 International Conference on Green Supply Chain*, Arras, France, Jun. 2014.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies
de l'information
et de la communication (STIC)

Titre : Études de problèmes de réservation de voie dans des réseaux de transport

Mots clés : Planification et gestion des transports, réservation de voie, transport de marchandises, réseaux de bus, optimisation combinatoire, algorithmes

Résumé : Aujourd'hui, le transport est devenu indispensable dans la vie quotidienne. Cependant, la congestion du trafic du fait de la forte urbanisation et de l'augmentation rapide du nombre de véhicules a réduit l'efficacité du système de transport et a causé d'énorme pollution urbaine. Dans ce contexte, pour répondre aux besoins spécifiques de transport et améliorer la performance des systèmes de transport, la réservation de voie, en tant que stratégie de gestion du trafic flexible, a été largement mise en oeuvre. La majorité des études existantes sur la réservation de voie se focalisent au niveau microscopique, par exemple, un segment de route principale. Dans cette thèse, nous nous concentrons sur la réservation optimale des voies dans un réseau de transport au niveau macroscopique en minimisant son impact négatif pour deux catégories de problèmes. Nous étudions d'abord des problèmes de réservation robuste de voie et de grande taille pour les futurs poids lourds intelligents et les grands événements spéciaux. Ensuite, nous étudions la réservation de voie dans le but d'améliorer la performance du transport public avec des hypothèses spécifiques. Pour chaque problème étudié dans cette thèse, des modèles appropriés sont construits et leurs complexités sont analysées. Différentes approches de résolution sont élaborées en fonction des caractéristiques des problèmes, à savoir : une méthode exacte à deux phases, une méthode de ε-contrainte, une méthode de « cut and solve », et une méthode de « kernel search ». La performance des algorithmes proposés est évaluée à l'aide de benchmarks et d'instances générées aléatoirement. Les expériences numériques montrent que les algorithmes proposés sont plus performants que les algorithmes existant dans la littérature et le progiciel commercial CPLEX.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

Title : A study on lane reservation problems in transportation networks

Keywords : Transportation planning and management, lane reservation, freight transportation, bus transit systems, combinatorial optimization, algorithms

Abstract: Nowadays, transportation has become an indispensable part in modern life. However, heavy traffic congestion due to high urbanization and rapid increase of vehicles has caused low transportation efficiency and huge amounts of urban pollution. In this context, to meet special transportation requirements and improve the performance of transportation systems, lane reservation, as a flexible and economic traffic management strategy, has been widely implemented in real life. The majority of studies about lane reservation in the literature focus on the impact at a microscope level, e.g., a single link or corridor. In this thesis, we focus on optimally reserving lanes at a macroscopic network level with the objective of minimizing negative impact for two categories of problems. We firstly investigate the large-size and robust lane reservation problems in the contexts of future automated truck freight transportation and large-scale special events. Then, we study lane reservation for improving the performance of bus transit system under different assumptions. For all problems studied in this thesis, appropriate models are provided and their complexities are analyzed. Different resolution approaches are developed according to the characteristics of problems, including exact two-phase method, exact ε-constraint based method, cut-and-solve method, and kernel search method. The performance of the proposed algorithms is evaluated by benchmark and randomly generated instances. Extensive numerical experiments show that the proposed algorithms outperform the state-of-the-art algorithms and the commercial software CPLEX.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l’Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France