ABSTRACT

Title of Dissertation: SOLVING THE INTEGRATED SCHOOL BELL TIME, AND BUS ROUTING AND SCHEDULING OPTIMIZATION PROBLEM UNDER THE DETERMINISTIC AND STOCHASTIC CONDITIONS

Zhongxiang Wang
Doctor of Philosophy
2019

Dissertation directed by: Professor
Ali Haghani
Department of Civil and Engineering

The school bus planning problem (SBPP) has drawn significant attention in research and practice because of its importance in pupil transportation. The major task of the SBPP is to simultaneously optimize the school bell times, the routing plan (a set of trips) and the scheduling plan (the assignment of buses to serve these trips) while maintaining the minimum level-of-service requirements with the objective that the total number of buses and the total vehicle time are both minimized. Many subproblems of

the SBPP have been well studied, but the integrated problem lacks much research due to its complexity.

A Mixed Integer Programming (MIP) model is proposed for the integrated SBPP. A novel decomposition method is developed to solve the model. It distinguishes itself from the literature with the consideration of trip compatibility in the routing stage, which is a piece of essential information in the following scheduling stage. This 'look ahead' strategy finds a new balance between the model integration and decomposition, which solves the problem efficiently as a decomposed problem but with the high solution quality as the integrated model. Three heuristic algorithms are proposed to solve the deterministic SBPP with the trip compatibility. Then, two mathematical programming models and a Column Generation-based algorithm are proposed for the SBPP under traffic congestion and stochastic travel time in a real uncertain world. These innovative algorithms incorporate the merits of the Simulated Annealing, Tabu Search, Insertion Algorithm, and Greedy Randomized Adaptive Search Procedure and gain the computational power that the existing methods do not have.

The experiments are conducted on randomly generated datasets, benchmark problems, and real-world cases. The results show that the proposed models and algorithms outperform the state-of-the-art method in all test problems by up to 25%. In a real-world case study, after the bell time adjustment, up to 41% of current buses can be saved with even better service with respect to the higher punctuality and shorter student ride time.

SOLVING THE INTEGRATED SCHOOL BELL TIME, AND BUS ROUTING AND SCHEDULING OPTIMIZATION PROBLEM UNDER THE DETERMINISTIC AND STOCHASTIC CONDITIONS

by

Zhongxiang Wang

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2019

Advisory Committee:
Professor Ali Haghani, Chair
Professor Martin Dresner
Professor Gang-Len Chang
Professor Paul Schonfeld
Associate professor Barton Forman

© Copyright by
Zhongxiang Wang
2019

Acknowledgments

This dissertation is dedicated to everyone I gratefully met at the University of Maryland. The past four years has been a wonderful journey and a life changer for me, rising from an immature undergraduate to an experienced researcher. Aware of how little I know, I am confident about my research in the field of school bus planning and vehicle routing. I am grateful to contribute in this field under the guidance and help of my advisor Dr. Ali Haghani.

Four years of study was not easy. The stress, the hesitation, and the depression are almost inevitable. To me, the most discouraging moment is when I realized, the more I know, the more I do not know. It is a valley I must go through before having a deeper self-cognition and regaining my self-confidence. I feel blessed to have that experience so that I can cherish the joy of this accomplishment, not just the degree, but the bravery, the honesty and all priceless merits I gained in facing the difficulties.

I am gratefully to have my forever supportive family, my father Zhiqin Wang, my mother Peijun Guo and my sister An Wang. Our bond is always my ultimate strength in the bottom of my heart. I am also truly thankful to every distinguished and prestigious professor I worked with, Dr. Paul Schonfeld, Dr. Martin Dresner, Dr. Gang-Len Chang, Dr. Barton Forman, Dr. Lei Zhang, Dr. Cinzia Cirillo, Dr. Michael Fu, Dr. Michael Ball, Dr. Eric Slud. I appreciate the accompaniment and the encouragement of my beloved friends, Liao Zhang, Yeming Hao, Liang Liang, Ali Shafahi, Huanjiao Hu, Ruiche Liu, and so many more. The past four years would not be so enjoyable without them. I wish a happy life to everyone I know, especially the ones I do not name here.

Table of Contents

Acknowledgments ii

Table of Contents iii

List of Tables vii

List of Figures x

List of Abbreviations xii

Chapter 1 Introduction 1

1.1 Problem decomposition 1

1.1.1 Bus stop selection (BSS) 2

1.1.2 School bus routing (SBR) 2

1.1.3 School bell time adjustment (BTA) 3

1.1.4 School bus scheduling (SBS) 3

1.2 Interrelationship between subproblems 4

1.2.1 Example 5

1.3 Motivation 8

1.4 Dissertation structure 9

Chapter 2 Literature Review 11

2.1 School bus routing 12

2.1.1 Classifications 12

2.1.2 Objective 16

2.1.3 Constraints 17

2.1.4 Algorithms 21

2.2 Bell time adjustment 31

2.3 School bus scheduling 32

2.3.1 Problem transformation 32

2.3.2 Objective 34

2.3.3 Constraints 34

2.3.4 Algorithms 36

2.4 Stochasticity 38

2.5 Research gap 40
2.6 Chapter summary 42
Chapter 3 Deterministic School Bell Time, Bus Routing and Scheduling Optimization Models 43
3.1 Assumptions 43
3.2 Notations for model formulation 44
3.3 The MBRSV (Model 1) 46
3.3.1 Model formulation 46
3.3.2 Formulation description 48
3.4 Decomposition scheme 52
3.5 The MBRS (Model 2) 55
3.5.1 The MVT relaxation 55
3.5.2 Model formulation 55
3.6 The MRS (Model 3) 57
3.6.1 Compatibility transformation 57
3.6.2 Model formulation 58
3.6.3 Compatibility overestimation 59
3.7 Decomposed single school routing problems 60
3.7.1 The SRC (Model 4) 60
3.7.2 The SRCA (Model 4') 61
3.7.3 The SRCW (Model 4") 64
3.8 The MBSV (Model 5) 66
3.9 Chapter summary 68
Chapter 4 Deterministic School Bell Time, Bus Routing and Scheduling Optimization Algorithms 69
4.1 Algorithm 1 to solve the MBRSV (Model 1) 69
4.2 Algorithm 2 to solve the SRC (Model 4) 72
4.2.1 Notations for Algorithm 2 72
4.2.2 Algorithm 2-1: minimum cost matching algorithm 74
4.2.3 Algorithm 2-2: post improvement algorithm 80
4.3 Algorithm 3 to solve the MBSV (Model 5) 87

4.3.1 The MBS (Model 7) 88
4.3.2 Illegal route update (IRU) 88
4.3.3 Discussion 90
4.4 Model comparison 93
4.5 Chapter summary 95
Chapter 5 Stochastic School Bell Time and Bus Scheduling Optimization Models and Algorithms 96
5.1 Time-dependent multi-school bell time and scheduling 97
5.1.1 Notations 97
5.1.2 The TD_MBSV (Model 8) 98
5.2 Chance-constrained multi-school bell time and scheduling 100
5.2.1 Notations 100
5.2.2 The TS_MBSV (Model 9) 101
5.3 Stochastic multi-school bell time and scheduling (Algorithm 4) 105
5.3.1 Column generation-based stochastic bus scheduling (Algorithm 4-1) 107
5.3.2 Stochastic school bell time adjustment (Algorithm 4-2) 124
5.4 Chapter summary 127
Chapter 6 Computational Experiment 129
6.1 Experiment 1: test on Algorithm 1 129
6.1.1 Experiment target 129
6.1.2 Test data generation 130
6.1.3 Experiment 1-1: model relaxation and transformation 131
6.1.4 Experiment 1-2: model decomposition 133
6.1.5 Experiment 1-3: sensitivity of the SRC (Model 4) 139
6.1.6 Experiment 1-4: the marginal cost of a school bus 140
6.2 Experiments 2: test on Algorithm 2 142
6.2.1 Experiment target 142
6.2.2 Experiment 2-1: test on exact and heuristic algorithm 143
6.2.3 Experiment 2-2: preliminary benchmark problems 145
6.2.4 Experiment 2-3: sensitivity of the PMCMPI 147
6.2.5 Experiment 2-4: full benchmark problem 148

6.3 Experiments 3: test on Algorithm 3 153
6.3.1 Experiment setup 153
6.3.2 Comparison of the results 154
6.4 Experiments 4: test on stochastic models 156
6.4.1 Test problem generation 156
6.4.2 Experiment setup 158
6.4.3 Comparison of the solutions 160
6.4.4 Sensitivity analysis 164
6.5 Chapter summary 168
Chapter 7 Real-World Case Study 170
7.1 Case 1: general-purpose bus transportation 170
7.1.1 Experiment setup 170
7.1.2 Results 172
7.2 Case 2: bus transportation with special needs 173
7.2.1 Experiment setup 173
7.2.2 Deterministic school bell time, bus routing, and scheduling 175
7.2.3 Stochastic school bell times and bus scheduling 190
7.3 Chapter summary 198
Chapter 8 Conclusions and Future Research 199
8.1 Summary 199
8.2 Conclusion 201
8.3 Future research 204
Bibliography 207

List of Tables

Table 1 Summary of three plans for the example in Figure 1 8

Table 2 Literature review of school bus planning problem 13

Table 3 Notation summary for MBRSV (Model 1) 45

Table 4 Parameters for the MBSV (Model 5) 66

Table 5 Notation summary for Algorithm 2 73

Table 6 Model comparison in two test problems 93

Table 7 Analytical model comparison 94

Table 8 Notation summary for the TD_MBSV (Model 8) 98

Table 9 Notation summary for the TS_MBSV (Model 9) 101

Table 10 Notation summary for Algorithm 4 106

Table 11 Notation summary for Algorithm 4-1 108

Table 12 Result summary for experiment 1-1 (Models 1, 2, and 3) 132

Table 13 Result summary for experiment 1-2 (decomposition in Algorithm 1) 137

Table 14 Result summary for experiment 1-3 (sensitivity of the SRC, Model 4) 140

Table 15 Result summary for experiment 1-4 (the marginal cost of a school bus) 141

Table 16 Result summary for the experiment 2-1 (the SRC, the PMCM, and the SMCM) 144

Table 17 Result summary for experiment 2-2 (preliminary benchmark problem) 146

Table 18 Sensitivity analysis for experiment 2-3 (the MCMPI) 148

Table 19 Result summary for experiment 2-4 (benchmark with the MRT=2700) 150

Table 20 Result summary for experiment 2-4 (benchmark with the MRT=5400) 151

Table 21 Result summary for experiment 3 155

Table 22 Result summary for Model 5, 8, 9 and Algorithm 4 161
Table 23 Solutions under different level of uncertainty 165
Table 24 Solutions of Alg4G with different sample sizes 167
Table 25 Proposed school bell time range for case study 1 171
Table 26 Illustration of trip information of case study 2 174
Table 27 Summary statistics for case study 2 175
Table 28 Sensitivity analysis for the coefficients in Algorithm 2 for case study 2 ... 177
Table 29 Summary of deterministic school bus routing and scheduling plans of case study 2 178
Table 30 Summary of deterministic school bell time, bus routing and scheduling plans of case study 2 181
Table 31 Comparison of the deterministic solutions with and without the bell time adjustment of case study 2 182
Table 32 Sensitivity analysis for the buffer coefficients of travel time from the Google Map API 185
Table 33 Stochastic on-time test for deterministic bus routing and scheduling plans of case study 2 189
Table 34 Summary of stochastic bus scheduling plans of case study 2 191
Table 35 Summary of stochastic school bell time and bus scheduling plans of case study 2 193
Table 36 Comparison of the stochastic solutions with and without the bell time adjustment of case study 2 194

Table 37 Summary of stochastic bus scheduling plans of case study 2 (Amplitude=20%) 196

List of Figures

Figure 1 Different approaches to solve the school bus planning problem 5

Figure 2 Example of the school bus routing problem 6

Figure 3 Graphical representation of scheduling of the example in Figure 2 7

Figure 4 Coverage of the School bus planning problem in recent papers 11

Figure 5 Illustration of the 2-opt 28

Figure 6 Illustration of the time-dependent directed scheduling graph 33

Figure 7 Illustration of the MBRSV decomposition 54

Figure 8 Illustration of compatibility overestimation 60

Figure 9 Illustration of the SRCA (Model 4') 64

Figure 10 Illustration of the bipartite graph for matching problem 76

Figure 11 Graphical example of the minimum cost matching trip generation algorithm 80

Figure 12 Illustration of the Post Improvement Algorithm 86

Figure 13 Illustration of Algorithm 3 90

Figure 14 Illustration of the Column Generation 113

Figure 15 Illustration of an unbalanced matrix 115

Figure 16 Comparison of the running times for the SRCW, the PMCM, and the SMCM 145

Figure 17 Savings in the number of buses compared to Park (MRT=5400 seconds) 152

Figure 18 Illustration of the generation of stochastic travel times and deadheads ... 158

Figure 19 Solution summary of case study 1 172

Figure 20 Comparison of the solutions for case study 2 under deterministic condition 183
Figure 21 Comparison of the solutions for case study 2 under different buffer coefficients 186
Figure 22 Comparison of all stochastic solutions for case study 2 195
Figure 23 Comparison of the stochastic bus scheduling plans of case study 2 (Amplitude=10% and 20%) 197

List of Abbreviations

Problems

SBPP: School Bus Planning Problem

BSS: Bus Stop Selection Problem

SBR: School Bus Routing Problem

BTA: Bell Time Adjustment Problem

SBS: School Bus Scheduling Problem

VRP: Vehicle Routing Problem

TSP: Traveling Salesman Problem

m-TSP: Traveling Salesman Problem with m-Trips

Models

MBRSV: Multi-School Bell time, Bus Routing and Scheduling with the Maximum Vehicle Time Constraints Problem (Model 1)

MNT: Minimum Number of Trips (in Model 1)

AAT: Additional Allowed Trips (in Model 1)

MBRS: Multi-School Bell time, Bus Routing and Scheduling Problem (Model 2)

MRS: Multi-School Bus Routing and Scheduling Problems (Model 3)

CT: Compatibility Transformation (in Model 3)

SRC: Single-School Bus Routing Problems with the Compatibility (Model 4)

SRCA: Single-school Bus Routing Problem with the Compatibility Assignment (Model 4')

SRCW: Single-School Bus Routing Problem with the Compatibility Assignment and the Weight Adjustment as (Model 4")

OA: Objective Adjustment (in Model 4)

CA: Compatibility Assignment (in Model 4’ and Model 4”)

WA: Weight Adjustment (in Model 4”)

UTC: Unassigned Trip Capacity (in Model 4’ and Model 4”)

MBSV: Multi-School Bell time and Bus Scheduling with the Maximum Vehicle Time Constraints Problem (Model 5)

MCPM: Minimum Cost Perfect Matching Problem (Model 6)

MBS: Multi-School Bell Time and Bus Scheduling Problem (Model 7)

TD_MBSV: Time-Dependent Multi-School Bell Time and Bus Scheduling with the Maximum Vehicle Time Constraints Problem (Model 8)

TS_MBSV: Trip-based Stochastic On-Time Constrained Multi-School Bell Time and Bus Scheduling with the Maximum Vehicle Time Constraints Problem (Model 9)

PMS: Set Partitioned Multi-School Scheduling Problem (Model 10)

CMS: Column Generation-based Multi-School Bus Scheduling Problem (Model 11)

SR: Single-School Bus Routing Problem (without compatibility) (Model 12)

MS: Multi-School Bus Scheduling Problem (Model 13)

MinN: Traditional Routing and Scheduling Decomposition Method, Use Minimizing the Number of Trips as the Routing Objective

MinTT: Traditional Routing and Scheduling Decomposition Method, Use Minimizing the Total Ride Time as the Routing Objective

MinNT: Traditional Routing and Scheduling Decomposition Method, Use Minimizing the Number of the Trips and Total Ride Time as the Routing Objective

Algorithms

DMBRS: Decomposition-Based Multi-School Bell Time, Bus Routing and Scheduling Algorithm (Algorithm 1)

MCMPI: Minimum Cost Matching with the Post Improvement (Algorithm 2)

MCM: Minimum Cost Matching Algorithm (Algorithm 2-1)

PI: Post Improvement Algorithm (Algorithm 2-2)

SMCM: Sequential Minimum Cost Matching Algorithm (in Algorithm 2-1)

PMCM: Parallel Minimum Cost Matching Algorithm (in Algorithm 2-1)

PNT: Predefined Number of Trips (in Algorithm 2)

SA: Simulated Annealing Algorithm (in Algorithm 2-2)

TS: Tabu Search Algorithm (in Algorithm 2-2)

IRU: Illegal Route Update (in Algorithm 3)

IBR: Illegal Bus Routes (in Algorithm 3)

SMBS: Stochastic Multi-School Bell Time and Bus Scheduling Algorithm (Algorithm 4)

CG_MS: Column Generation-Based Stochastic Multi-School Bus Scheduling Algorithm (Algorithm 4-1)

SMB: Stochastic Multi-School Bell Time Adjustment Algorithm (Algorithm 4-2)

IBA: Insertion-Based Algorithm

SIBA: Sequential Insertion-Based Algorithm

PIBA: Parallel Insertion-Based Algorithm

GENI: Generalized Insertion Procedure

CWSM: Clarke and Wright Saving Method

GRASP: Greedy Randomized Adaptive Search Procedure

GA: Genetic Algorithm

CG: Column Generation

LS: Local Search Algorithm

RFCS: Route First, Cluster Second Algorithm

CFRS: Cluster First, Route Second Algorithm

CFIS: Cluster First, Initial Second Algorithm

DSHH: Dynamic Stochastic Hedging Heuristic

Constraints

ACF: Artificial Commodity Flow (Sub-Tour Elimination Constraint)

MRT: Maximum Ride Time Constraint (Per Trip)

MVT: Maximum Vehicle Time Constraint (Per Bus Route)

SOT: Stochastic On-Time Constraint (Per Bus Route)

TSOT: Trip-based Stochastic On-Time Constraint (Per Trip)

Measures of Effectiveness

NOV: Number of Variables

NOB: Number of Buses

NOT: Number of Trips

RT: Running Time

Gap: Optimality Gap

TVT: Total Vehicle Time

TDD: Total Deadhead Duration

ParaMCM: Parameter of the Minimum Cost Matching (MCM) Algorithm

ParaPI: Parameters for the Post Improvement (PI) Algorithm

Prob: Probability of Being on Time

MinProb: Minimum Probability of Being on Time

MeanProb: Mean Probability of Being on Time

$N<\rho$: Number of Bus Routes with the Probability of Being on Time Less Than or Equal to ρ

$R<\rho$: Ratio of Bus Routes with the Probability of Being on Time Less Than or Equal to ρ

Others

LP: Linear Programming

MIP: Mixed Integer Programming

TDSG: Time-Dependent Directed Scheduling Graph

API: Application Programming Interface

ES: Elementary School

MS: Middle School

HS: High School

stdTT: Standard Deviation of the Trip Travel Time (in Stochastic Data Generation)

stdDD: Stand Deviation of the Deadhead Duration (in Stochastic Data Generation)

samSize: Sample Size (in Stochastic Data Generation)

testSize: Test (Data) Size (in Stochastic Data Generation)

Chapter 1 Introduction

According to the National Center for Education Statistics (2016), the U.S. spent over $23 billion on public student transportation in 2012-2013, which is about $914 per student. Due to the vast amount of funds being invested in school transportation, a good school bus transportation plan is of huge importance to the school transportation service providers. The School Bus Planning Problem (SBPP) is to find the least-cost school bus transportation plan, which transports the students between their home and the schools while addressing the safety concern, the time window constraints, the level-of-service requirements, etc. The cost of school bus transportation plan consists of the bus acquisition cost, drivers' wage cost, bus insurance, fuel cost, bus maintenance cost, etc. Usually, one driver is assigned to one bus. Thus, the bus acquisition cost and the drivers' wage are fully determined by the number of buses. The rest of the cost highly depends on the total bus travel distance or travel time. Therefore, the objective of the SBPP is to minimize the weighted sum of the number of buses used and the total vehicle time/distance.

1.1 Problem decomposition

Due to the complexity of the School Bus Planning Problem (SBPP), it is usually decomposed into smaller subproblems. One widely adopted decomposition method was developed by Desrosiers et al. (1981), which decomposed the SBPP into four subproblems: bus stop selection, bus trip generation, school bell time adjustment, and route scheduling. The input data, the main task, and output for each subproblem is

presented in the next four subsections. The detailed objective and constraints will be presented in Chapter 2.

1.1.1 Bus stop selection (BSS)

Input: students' home locations, geographic road network

Output: bus stop locations, number of students at each stop

The main task of bus stop selection is to find a set of bus stop locations such that the students can walk to these stops and take the school buses. Usually, the objective is to minimize the weighted sum of the number of bus stops and the total walking distance of the students. The constraints include the maximum walking distance for students, the maximum number of students at each stop, and the geographic limitations. Some specific constraints can vary from case to case. For instance, some school districts require that the stop must be on the same side of the pupils' home such that the pupils do not need to cross the street to take the bus.

1.1.2 School bus routing (SBR)

Input: bus stop locations, school locations, number of students at each stop, travel time/distance between nodes (including the stops and the schools)

Output: school bus routing plan (a set of trips)

The School Bus Routing (SBR) problem, which is also known as the trip generation problem, is a variant of the Vehicle Routing Problem (VRP). The main target of the school bus routing problem is to generate a set of trips to visit all the stops. An (afternoon) ***trip*** starts from a school, sequentially goes through a set of school bus

stops associated with this school while satisfying the bus capacity constraint, and the Maximum Ride Time (MRT) constraints if they exist.

1.1.3 School bell time adjustment (BTA)

Input: school bus routing plan, school bell time window

Output: exact school bell time

The bell time adjustment is to find the best school bell times within a certain time window such that a good bus scheduling plan (revealed after solving the school bus scheduling problem) can be obtained. It is usually incorporated into the school bus scheduling problem (Fügenschuh, 2009; Fügenschuh, 2011).

1.1.4 School bus scheduling (SBS)

Input: school bus routing plan, exact school bell time

Output: school bus scheduling plan

The school bus scheduling (SBS) problem, which is also known as the route scheduling, is to find the best scheduling plan to accommodate the school bus routing plan given the school bell time. The scheduling plan is made of a set of bus routes. A ***bus route*** starts from the depot, sequentially serves a set of compatible trips and goes back to the depot. A trip pair (*a*, *b*) is ***compatible*** if after serving trip *a*; the bus has enough time to travel from the last stop of trip *a* to the first stop of trip *b* before trip *b*'s start time. The ***deadhead*** of trip pair (*a*, *b*) is defined as the travel time between the last stop of trip *a* to the first stop of trip *b*. The ***idle time*** at trip *b* is the time that a bus needs to wait at the first stop of trip *b* until its scheduled service start time. This is the final

step of the SBPP, and the final school bus plan will be revealed after solving the school bus scheduling problem.

1.2 Interrelationship between subproblems

Although the SBPP is usually divided into four subproblems, the interrelationship between these subproblems cannot be ignored. The locations of bus stops have a vital impact on the school bus routing plan because the travel time/distance between stops depend on the selection of the stops. Similarly, the school bus schedule is influenced by the routing plan and school bell time. As a result, any change in the preceding process will affect the school bus scheduling in the end. Because of the interrelationship, many studies formulate integrated mathematical models and develop algorithms to solve these joint problems. There are many different combinations of joint problems, and some of them are shown in Figure 1.

In Figure 1, Approach 1 solves each subproblem as an individual problem. In this way, each problem is easy to solve, but the interrelationship between these subproblems are severely overlooked. Approach 2 considers the bell time and the scheduling problem together, which is easy to implement and has a better result than approach 1 (Fügenschuh, 2009; Fügenschuh, 2011). Approach 3 combines the stop selection and routing problem, which considered the impact of the stop selection on the routing problem (Schittekat et al., 2013; Faraj et al., 2014; Silva et al., 2015). Approach 4 is the joint problem of routing and scheduling after fixing the school bell times (Park et al., 2012; Kumar and Jain, 2015; Yan et al., 2015).

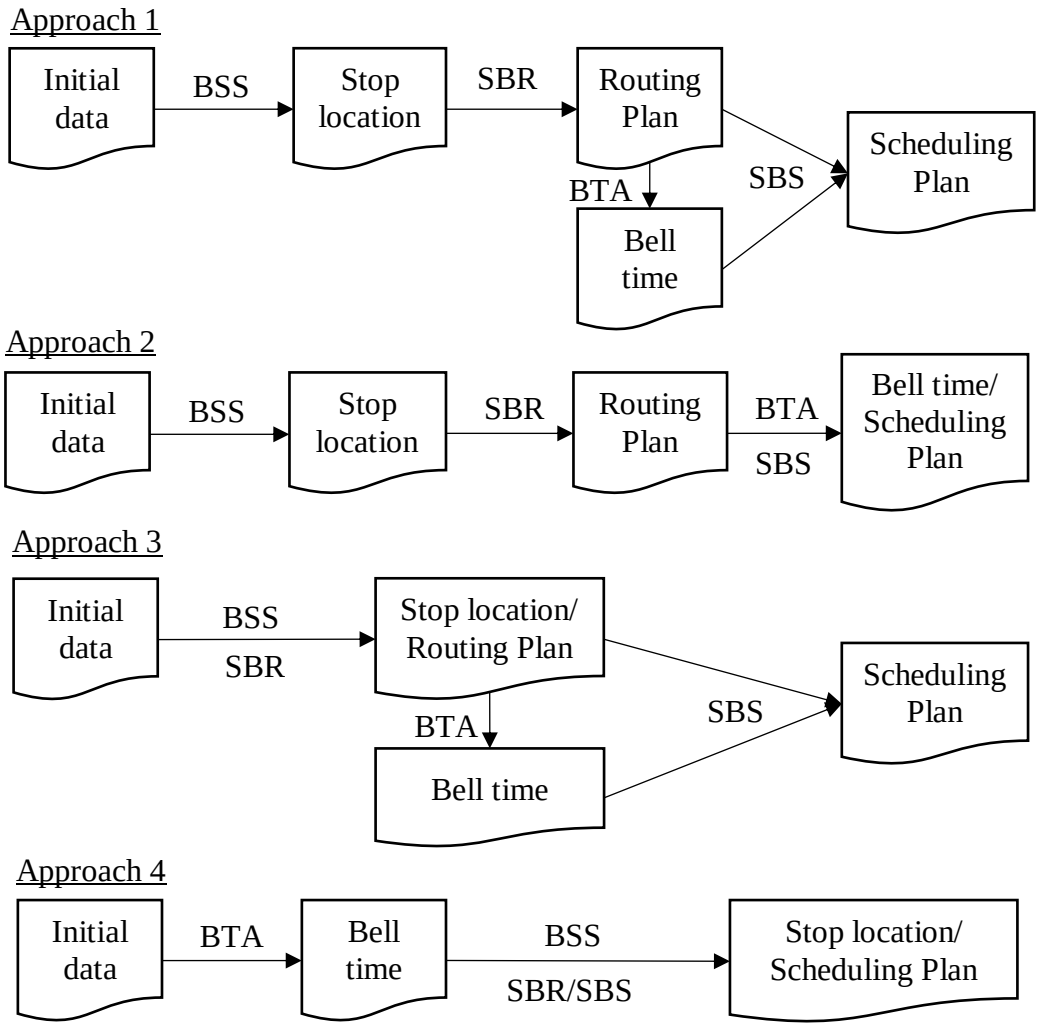


Figure 1 Different approaches to solve the school bus planning problem

1.2.1 Example

To explicitly explain the interrelationship between the subproblems, one example for the school bus routing, bell time adjustment, and schedule problem is shown in Figure 2, Figure 3 and Table 1. The stop location is assumed to be predetermined. In this example, there are six stops: S1, S2, and S3 are stops for school 1 and S4, S5, and S6 are for school 2. School 1 dismisses at 2:00 PM and school 2 dismisses at 2:30 PM. The travel time (in minutes) for each node pair is marked next

to the edges. With various objectives, the routing plans could be different. Let's consider two routing plans. Plan 1 has three trips: trip 1 is School 1→S3→S2→S1 (30 minutes), trip 4 is School 2→S5→S4 (40 minutes), and trip 5 is School2→S6→S7 (40 minutes) with a total travel time (with students on board) of 110 minutes. Plan 2 has four trips: trip 2 is school 1→S1 (15 minutes), and trip 3 is school 1→S3→S2 (20 minutes), trips 4 and 5 are the same as in the Plan 1. In this case, the number of trips (4 trips) and the total travel time (115 minutes) for Plan 2 are both higher than that from Plan 1 (3 trips with 110 travel minutes).

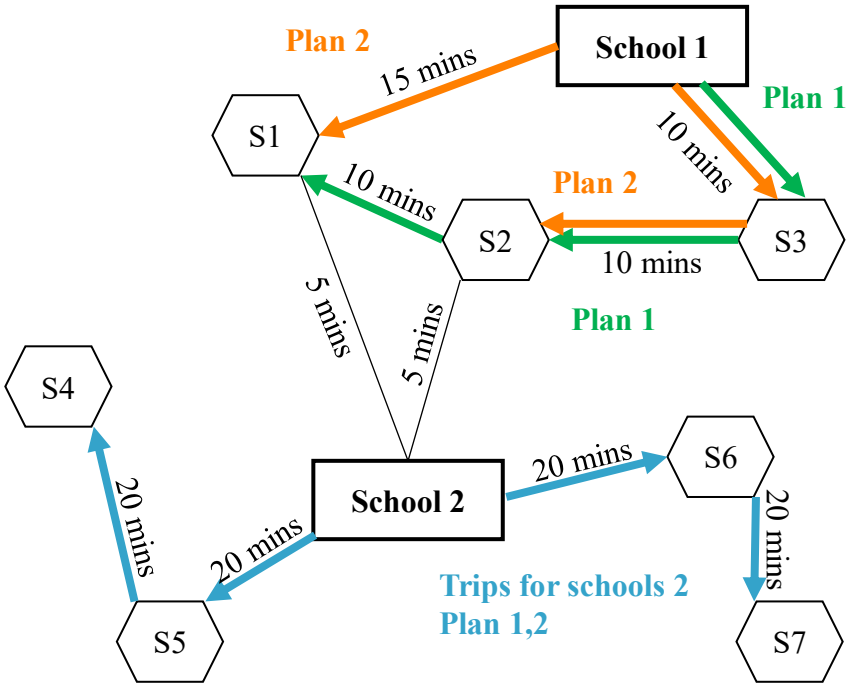


Figure 2 Example of the school bus routing problem

Then, the school bus scheduling problems are solved using these routing plans as input. The best scheduling plans are shown in Figure 3. For Plan 1, the earliest time that a bus can reach school 2 after serving trip 1 is 2:35 PM, which is later than the dismissal time of school 2 (2:30 PM). Hence, trip 1 is not compatible with any trip from

school 2. Three buses are needed – one bus for each trip. But for Plan 2, trip 2 and trip 3 are compatible with trip 4 and trip 5 from school 2. Only two buses are needed – each bus serves two trips (Bus 1: trip 2 → trip 4; Bus 2: trip 3 → trip 5). In this example, although Plan 1 has a fewer number of trips and shorter travel time in the routing stage, it requires more buses after solving the school bus scheduling problem.

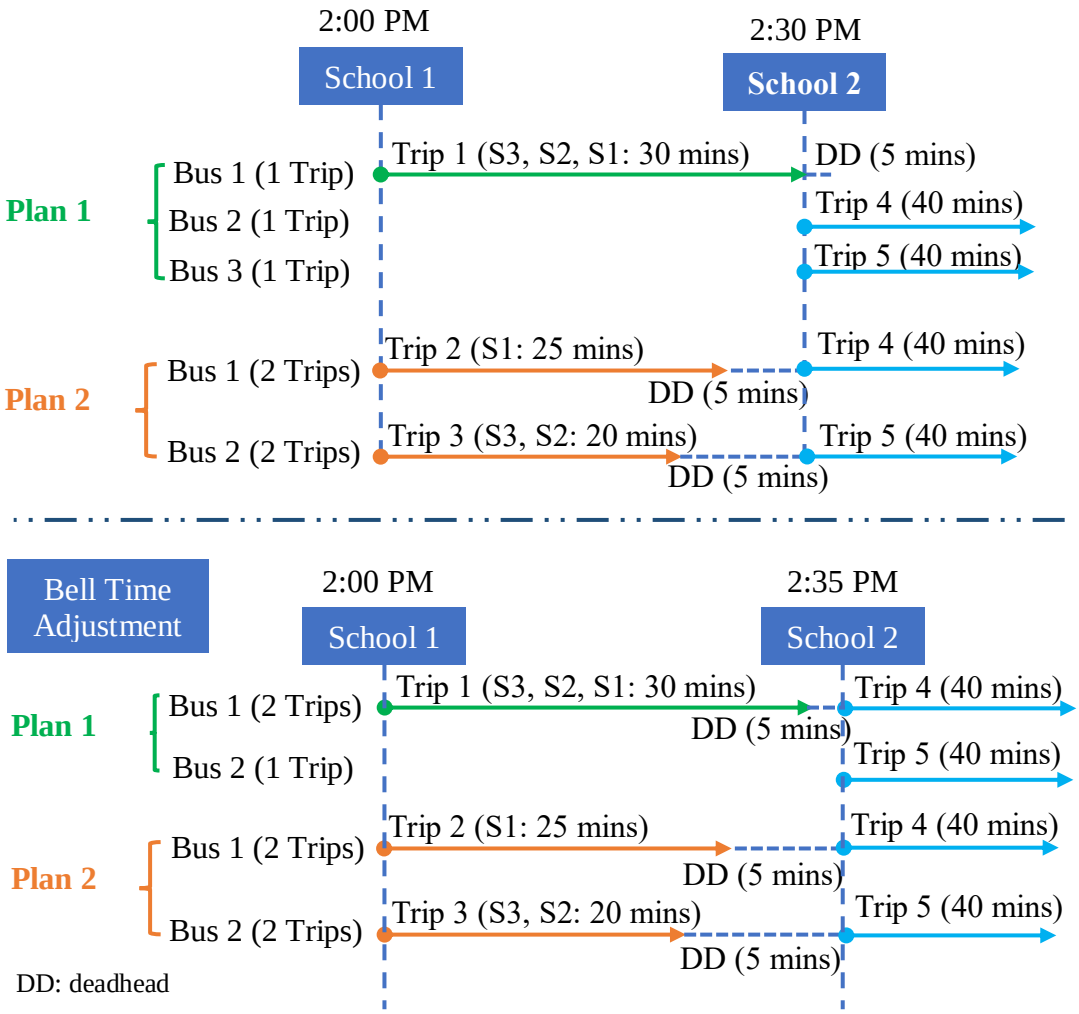


Figure 3 Graphical representation of scheduling of the example in Figure 2

Alternatively, if schools can adjust their bell time within a certain range, the final school bus scheduling plan could be totally different (shown in Figure 3). If school 2 can dismiss 5 minutes later, trip 1 would be compatible with trips from school 2.

Then, only two buses are needed for Plan 1. The final scheduling Plan 1 is better than Plan 2 in terms of the same number of buses but shorter travel time. The routing plans and scheduling plans are summarized in Table 1. This example shows the interrelationship among these subproblems in school bus planning problems. A good routing plan (in terms of the number of trips and total travel time/distance) does not guarantee to lead to a good scheduling plan. And the school bus bell time could also change the school bus scheduling plan completely.

Table 1 Summary of three plans for the example in Figure 1

	Plan 1	Plan 2	Plan 1 (BTA)
Number of trips	3	4	3
Total travel time	110 mins	115 mins	110 mins
Number of buses	**3**	**2**	**2**

As shown in the above example, the routing, bell time adjustment, and scheduling can all play an essential role in the school bus planning problem. Thus, they are all considered in this dissertation.

1.3 Motivation

The most difficult and interesting properties in the school bus planning problem are the separation and interrelationship between the subproblems. On the one hand, these subproblems are highly related. If they are solved as independent problems, the optimality at the early subproblems does not guarantee a good solution at the final stage. On the other hand, incorporating several subproblems into one is more likely to generate a good final solution, but the integrated problem would be extremely difficult to solve. With the increase in the problem size, these integrated problems can easily

become unsolvable under the current computing power. Therefore, finding the critical balance between model integration and solvability is an important task.

In addition, the uncertainty of the travel time is an unignorable challenge we must face in real-world. Several stochastic school bus routing models and algorithms have been proposed. However, to the best of our knowledge, no stochastic school bell time, and bus scheduling model or algorithm exists, and its importance is without question. Formulating and solving the stochastic school bell time and bus scheduling optimization problem is another task we want to complete in this dissertation.

1.4 Dissertation structure

The rest of the dissertation is arranged as follows. Chapter 2 summarizes the current research in the field of school bus planning. The research gap is discussed at the end of Chapter 2.

In Chapter 3, the deterministic school bell time, bus routing and scheduling models are proposed. The assumptions of the models are first specified. An integrated model is then presented. Due to its complexity, it is decomposed into many subproblems to solve.

In Chapter 4, three heuristic algorithms are proposed to solve the deterministic models presented in Chapter 3. In the end, the models used in Chapter 3 and 4 are compared with respect to their scope, the number of variables and the number of constraints.

In Chapter 5, the stochastic multi-school bell time and scheduling problem is solved. A time-dependent model, a chance-constrained model, and a column

generation-based algorithm are presented. Each model and algorithm can solve the problem with different assumptions of the uncertainty.

Chapter 6 conducts several experiments to test the correctness of the models and the effectiveness of the algorithms on randomly generated problems and the benchmark problems. A real-world case study is conducted in Chapter 7.

In the end, Chapter 8 concludes this research. The future research directions are discussed in this chapter.

Chapter 2 Literature Review

Most of the studies in the field of the school bus planning problem (SBPP) only solve one or several subproblems using Operations Research techniques. The coverages of the problem in recent papers on the school bus planning problem (since 2005) are shown in Figure 4. Park and Kim (2010) had a nice review of the SBPP. The majority focused on the school bus routing problem, bus scheduling, and bus stop selection. The bell time adjustment problem drew the least attention. And in the limited number of studies, the bell time adjustment was incorporated into the school bus scheduling problem.

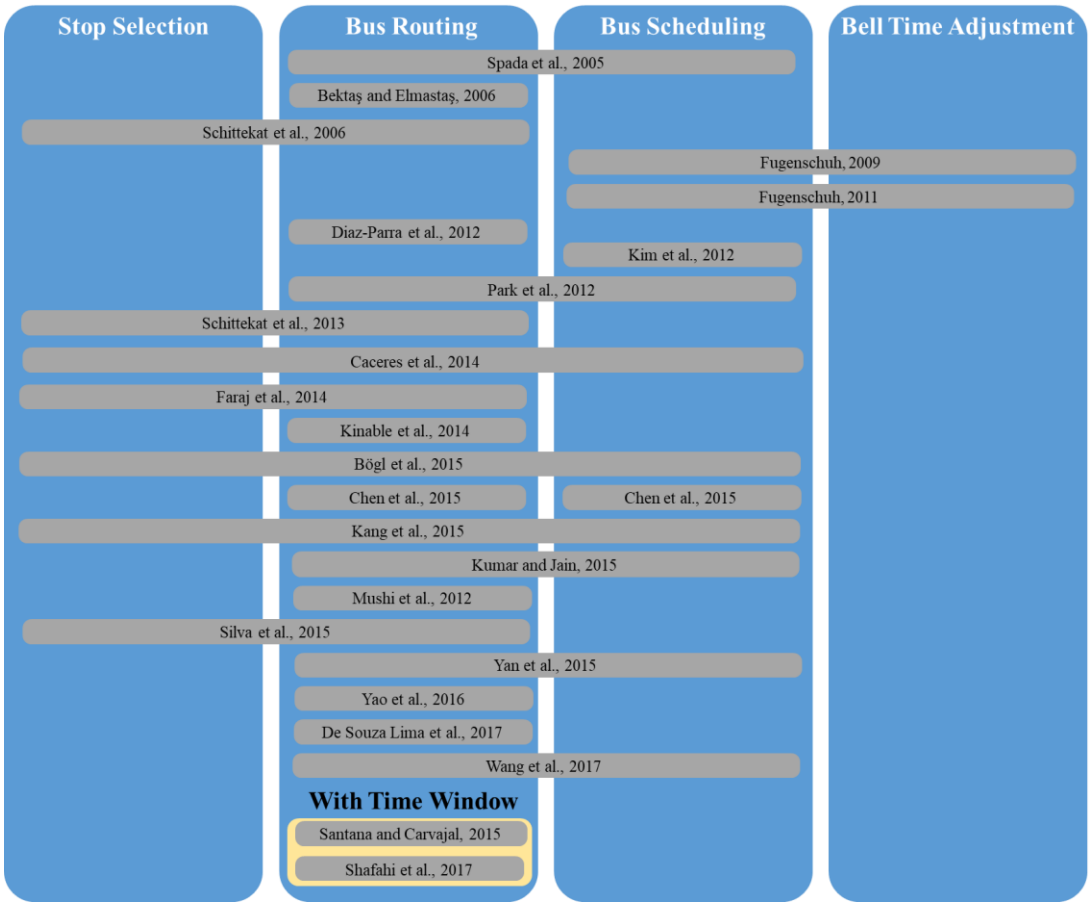


Figure 4 Coverage of the School bus planning problem in recent papers

From our empirical experience, the locations of the bus stops are usually predetermined. At least, it is unlikely to be changed often in comparison to the routing plan and the scheduling plan. Thus, this dissertation only focuses on the school bus routing, bell time adjustment, and school bus scheduling problem. The literature review is summarized in the scope of school bus routing, bell time adjustment and school bus scheduling problem to present the contribution of the existing studies. The detailed objectives, constraints and other characteristics of the studies on the school bus planning problems are summarized in Table 2.

2.1 School bus routing

The School Bus Routing (SBR) is a capacitated Traveling Salesman Problem with m trips (m-TSP; Desrosiers et al., 1995), which falls into the category of the general Vehicle Routing Problem (VRP).

2.1.1 Classifications

The first characteristic that separates the SBR problems is the number of schools. About half of the papers (in Table 2) solve the single-school routing problems. The papers working on the multi-school problem tackle the complexity of the system in two ways. They either: 1) decompose the multi-school into many single-school subproblems (Corberán et al., 2002); or: 2) use meta-heuristic algorithms to solve the problem (Braca et al., 1997 and Park et al., 2012).

Table 2 Literature review of school bus planning problem

Author	Year	Objective	Constraints	# of School	Fleet Mix	Data
Newton and Thomas	1969	Not specified	LOG, C, MRT	S	HO	Artificial, up to80 stops
Angel et al.	1972	TTD, NOT	LOG, C, MRT	M	HO	1,500 students, Tippecanoe, Indiana
Bennett and Gazis	1972	TTD	LOG, C	S	HO	256 students, Toms River, New Jersey
Newton and Thomas	1974	TTD, NOT	LOG, C, MRT	M	HO	1,097 students, 76 stops
Verderber	1974	TTD, NOT	LOG, C, MRT	M	HO	11,000 students, New York
Bodin and Berman	1979	NOT	LOG, C, MRT, TW	M	HO	13,000 students, Brentwood, New York
Gavish and Shlifer	1979	TTD, NOT	LOG, C, MRT	S	HO	Artificial, 21 stops
Dulac et al.	1980	TTD, NOT	LOG, C, MRT, MWT	S	HO	585 students, 99 stops, Drummondville, Canada
Hargroves and Demetsky	1981	TTD, NOT	LOG, C, MRT, MNS	M	HT	8,537 students, Albemarle, Virginia
Chapleau et al.	1985	NOB	LOG, TW	M	HO	37 schools, New Haven, Connecticut
Russell and Morrel	1986	TTD	LOG, C, MRT	M	HO	140 disabled student on nine bus routes
Desrosiers et al.	1986b	TTD, NOT	LOG, C, MRT, MWT	M	HO	About 16,000 students, Drummondville, Canada
Chen et al.	1990	TTD, NOT	LOG, C, MRT, TW	M	HO	2,413 students, 40 buses, 48 routes, Choctaw, Alabama
Thangiah and Nygard	1992	TTD, NOT	LOG, C, MRT	S	HT	Nine buses, 315 students, 180 stops; Eight buses, 210 students, 116 stops
Bowerman et al.	1995	TTD, NOT, SWD, BAL	LOG, C, MRT	S	HO	138 students, Ontario, Canada
Braca et al.	1997	TTD, NOB	LOG, C, MRT, TW, MNS	M	HO	838 stops, 73 schools, Manhattan, New York
Corberán et al.	2002	NOT, MRT	LOG, C, MRT	S	HO	19 schools, from 2 to 14 buses, Burgos, Spain
Li and Fu	2002	TTD, NOT, BAL	LOG, C	S	HT	86 students, 54 stops, Hong Kong
Ripplinger	2005	TTD	LOG, C, MRT	S	HT	131 student, artificial
Spada et al.	2005	TL	LOG, TW C, SCH	M	HT	274 students, Switzerland
Schittekat et al.	2006	TTD	LOG, C, SWD, SBL	S	HO	Artificial, 50 students, 10 stops
Pacheco and Martí	2006	NOT, MRT	LOG, C	S	HO	Benchmark problem from Corberán et al., 2002
Bektaş and Elmastaş	2007	TTD, NOT	LOG, C, MRT, TW, SBL	S	HO	519 students, central Ankara, Turkey
Fügenschuh	2009	NOB, DD	SCH, BT	M	HO	5 counties data from German, up to 490 trips, 102 schools

Author	Year	Objective	Constraints	# of School	Fleet Mix	Data
Fügenschuh	2011	NOB	SCH, BT	M	HO	2 counties data from German, up to 191 trips
Díaz-Parra et al.	2012	TTD, NOT	LOG, C, MRT, SBL	S	HO	Artificial, 50 problems, each problem has 200 bus stop
Kim et al.	2012	NOB, DD	LOG, SCH	M	HO/HT	Artificial, up to 100 schools, 562 trips, and 28175 students
Park et al.	2012	NOB	LOG, C, MRT, TW	M	HT	Artificial, up to 100 schools, 2000 stops, 32048 students
Schittekat et al.	2013	TTD	LOG, C, SBL	S	HO	Artificial, up to 80 stops, 800 students
Faraj et al.	2014	TTD	LOG, C, MRT	M	HT	944 students, 23 schools, the Brazilian city
Kinable et al.	2014	TTD	LOG, C, MT	S	HO	Artificial, up to 40 stops and 800 students
Bögl et al.	2015	TTD, PLT	LOG,C,SBL, MWD, TSF	M	HO	Artificial, up to 8 schools, 500 students
Chen et al.	2015	TTD, NOB, DD	LOG, SCH	S	HO /HT	Benchmark problems from Park, Tae and Kim 2012; Kim, Kim and Park, 2012
Kang et al.	2015	TTD	LOG, C, MRT	M	HT	26 students, 6 schools, 3 buses
Kumar and Jain	2015	TTD	LOG, LOG	S	HO	Artificial, up to 40 schools, 235 trips, 11600 students
Mushi et al.	2015	TTD	LOG, C	S	HO	58 stops, 456 students, Dar es Salaam, Africa
Santana and Carvajal	2015	TTD	LOG, C, TW	S	HO	600 students, 440 nodes, Bogota, Colombia
Silva et al.	2015	TTD	LOG, C, SBL	M	HT	716 students, 23 schools, the Brazilian city
Yan et al.	2015	TTD, PLT	LOG, LOG	S	HO	Six universities (treated as one school), 400 students, Taiwan
Yao et al.	2016	TTD	LOG, C, MRT	M	HO	Artificial, up to 2 schools, 116 stops, and 1088 students
Caceres et al.	2017	NOB, ETD	POP, PAO, EMRT	M	HO	Williamsville Central School District, 13 schools, up to 177 stops and 1237 student per school
de Souza Lima et al.	2017	TTD, NOT, BAL	LOG, C	M	HT	Artificial, up to 20 schools, 150 stops; Benchmark from Park, Tae and Kim 2012
Shafahi et al.	2017	NOT, TTD, TC	LOG, C, SCH	M	HO	Artificial, up to 25 schools, 200 stops and 180 students per school
Wang et al.	2017	NOT, TTD, TC	LOG, C, SCH, MRT	M	HO	1. Artificial, up to 30 schools, 300 schools; 2. from Shafahi et al. 2017
Caceres et al.	2018	TTD, NOT	LOG, C, MRT, SBL	M	HO	1. Park et al. (2012); 2. 39 schools and 227 stops in Williamsville Central School District

Note: Objective: NOT: Number of trips; NOB: Number of buses; TTD: Total travel distance (time); TC: trip compatibility; MRT: Maximum ride time; PLT: Penalty; BAL: Balance between each trip; SWD: Student waiting time; TL: Time loss (delay + wait time); DD: deadhead; ETD: Expected total travel distance. **Constraints:** LOG: Logistic constraints; C: Capacity constraints; TW: Time window constraints; MWD: Maximum walk distance constraints; SBL: Sub-tour elimination constraint; STO: Stochastic constraints; TSF: Transfer constraints; SCH: Scheduling constraints; BT: bell time constraints; MNS: Minimal number of student to create a bus trip; POS: probability of overcrowding of the students on buses; PAO: probability of arriving on time of the buses; EMRT: expected maximum ride time. **# of schools:** S: Single-school; M: Multi-school. **Fleet Mix:** HO: Homogeneous fleet; HT: Heterogeneous fleet.

Based on whether a trip can serve stops from different schools, the SBR can be either single-load or mixed-load. Single-load is referred to as the condition where trips are dedicated to schools, and each trip only visits the stops of the same school. Still, routes can serve trips from various schools. This is the underlying reason why a single-load multi-school bus routing problem can be decomposed into many single-school bus routing problems. On the other hand, if a trip can visit stops from more than one school, this is a mixed-load problem. It is beneficial to pick up all students living in a remote area with one trip. Park et al. (2012) showed that the mix-load plan could save up to 22% of buses from the single-load plan.

But the mixed-load bus routing problem is extremely difficult to solve even using some heuristic algorithms. Take the insertion-based heuristic algorithm (which will be discussed more in section 2.1.4.5) for example. Un-routed stops are inserted into the bus trips at the minimum cost position to generate the mixed-load bus routes. At every iteration, the algorithm needs to examine whether the associated school of the inserting stop exists in the trip in which the stop is being inserted. And if the associated school does not, the algorithm must check whether it is feasible to insert this stop and its associated school into the trip. The word ‘feasible’ means the augmented trip should satisfy all the constraints: capacity, maximum ride time, the time window for every stop and every school on that route. Because an insertion could change the arrival time of all the following stops, the search procedure is very time-consuming (Park et al., 2012).

Based on the location, the school bus planning problem can also be categorized into the urban area school planning problem or rural area planning problem. Single-

load is usually adopted in urban schools, and mixed-load is more common in rural area schools. Generally, there are more students exist at one stop in urban schools, which makes the capacity constraints the binding constraint (Braca et al., 1997). Due to these characteristics, Shafahi et al. (2017) allowed that every stop could be served by several trips where each trip only picked up a portion of the students at that stop. Bowerman et al. (1995) even relaxed the maximum ride time constraints in the urban school problem. In the rural area, the stops are far away, and buses sometimes need to go to students' home to pick up the student, the maximum ride time is usually the dominant constraint. Under this condition, it is beneficial to use smaller vehicles to pick up students in a rural area (Ripplinger, 2005). This is often referred to as the heterogeneous fleet in comparison to the homogeneous fleet (in urban schools) where only one type of bus with the same capacity is used.

2.1.2 Objective

As explained above, the real objective for the school bus planning problem is to serve all students' transportation demand with the minimum number of buses and minimum total vehicle time/distance. But these two values are only revealed after solving the school bus scheduling (SBS) problem. At the routing stage, different surrogate objectives are proposed to describe the goodness of the routing plan with the hope that a good scheduling plan will be obtained. The most common surrogate objectives are: 1) minimizing the number of trips; or 2) minimizing the total travel time/distance (Schittekat et al., 2013; Faraj et al., 2014; Kinable et al., 2014); or 3) a mixture of these two (Díaz-Parra et al., 2012). Some papers also shed light on the balance between trips including the load balance and ride time balance (Bowerman et

al., 1995; Li and Fu, 2002; de Souza Lima et al., 2017). Bowerman et al. (1995) included the student walking distance in the objective as it incorporated the bus stop selection in the routing problem. In terms of the student ride time, many papers consider it as a level-of-service constraint. But Corberán et al. (2002) and Pacheco and Martí (2006) considered it as part of the objective. They both minimized the number of trips and the maximum student ride time.

2.1.3 Constraints

2.1.3.1 Logistic constraints

Logistic constraints are an expansion of the degree constraints from the Traveling Salesman Problem (TSP). The degree constraints require to find a path such that each node on that path has a degree of two. In the SBR, with two-index variables ($x_{i,j}, \forall i,j \in Stops | i \neq j.\ x_{i,j}$ is binary, which equals to 1 if stop *j* is visited right after stop i; 0, otherwise), the logistic constraints are: 1) each stop has exactly one preceding stop or school; 2) each stop has exactly one successive stop or school. With three-index variable ($x_{i,j}^t, \forall i,j \in Stops | i \neq j, t \in Trips.\ x_{i,j}^t$ is binary, which equals to 1 if stop *j* is visited right after stop *i* on trip *t*; 0, otherwise), the logistic constraints consist of the following: 1) each stop has a preceding stop (or the corresponding school); 2) the proceeding stop and the successive stop are assigned to the same trip; 3) each trip starts from the school. In the single-visit problem, each stop should be visited exactly once. For a multi-visit setting, each stop is visited at least once. For every stop, the sum of students picked up at all visits equals to the total number of students at this stop.

2.1.3.2 Capacity constraints

Capacity constraints are the ones that distinguish the VRP from the m-TSP. In a general VRP (if pick up and drop off both exist on the trip), the capacity constraints stipulate that the maximum load on a trip is less than or equal to the vehicle capacity. Instead of finding the maximum load, an equivalent way is to check whether the load after visiting every node satisfies the vehicle capacity constraint. In the afternoon school bus routing (SBR) problem, all students are picked up at the school and dropped off at bus stops. The maximum load is the sum of the students on a trip. There is no need to keep track of the load at every stop.

The capacity is not only determined by the number of seats on the bus but also the allowance of crowding. The general guideline for the school bus seating is that three younger students or two older students can fit in one standard 39-inch seat (National Association of State Directors of Pupil Transportation Services). Park et al. (2012) used 66 students as the capacity for a standard school bus. Another aspect is that whether overbooking is allowed. At high schools, many senior students drive to school themselves, which makes the bus occupancy rate much lower than the elementary schools and the middle schools. To fully use the bus capacity, Caceres et al. (2017) suggested to allow overbooking and considered the demand randomness in the routing problem.

2.1.3.3 Maximum ride time constraints

The Maximum Ride Time Constraint (MRT) is a level-of-service constraint in the SBR. The complete statement of the MRT is that the MRT of students cannot exceed a fixed limit. Because all students are picked up at the school (PM trips), the

students who go to the last stop will experience the longest ride time. Instead of tracking every student's ride time, we can simply limit the MRT from the school to the final stop. Russell and Morrel (1986) limited the MRT to be less than or equal 45 minutes. Chen et al. (1990) used 75 minutes as the MRT. Park et al. (2012) developed the benchmark problems using either 45 minutes or 90 minutes as the MRT.

An underlying assumption associated with travel time/distance is the triangle inequality (Solomon, 1987). It means that the direct path between every pair of the node is the shortest path in terms of the travel time or distance among all feasible paths. This assumption guarantees that inserting a stop into a route will never decrease the total travel time/distance. Furthermore, it means that when a trip violates the MRT, all the super-trips are infeasible. Trip 1 is a super-trip of trip 2 if and only if trip 1 contains trip 2. An application of this property is the insertion-based trip generation heuristics. If the MRT is violated after inserting the stop that has the least additional travel time, there is no need to check the feasibility of inserting other stops into this trip (Prins, 2004).

2.1.3.4 Time window constraints

Time window constraints are associated with school bell time. In the school bus planning problem, the time window at each stop is usually not constrained, but the start time of the trip (PM trips) is limited within a time window or equal to the school dismissal time. The time window is usually hard, which means that any violation of the time window constraint is prohibited.

A unique property with the time window constraints is that if every trip is assigned with a specific start time in the routing stage, the school bus scheduling

problem becomes a simple assignment problem (Kim et al., 2010). It is well known that the assignment problem can be solved efficiently using the Hungarian Algorithm with $O(n^3)$.

2.1.3.5 Sub-tour elimination constraints

Sub-tour elimination constraint eliminates the illegal trips that are not connected to the school (Bektaş and Elmastaş, 2007). The basic sub-tour elimination of the TSP is derived from the fact that for any nontrivial subset (say *S*) of the graph, there should exist at least one edge on the path connecting some vertices in that subset (*S*) and the vertices outside the subset. The above formulation is strong. However, it requires a huge number of constraints. In practice, researchers would dynamically add the constraints. That is, first solve the problem without the sub-tour elimination constraint, identify the illegal sub-tours, then add the corresponding constraints to eliminate the sub-tours and resolve the problem again until no sub-tour exists.

One disadvantage of the above approach is that the number of iterations is unknown. It may take a significant number of iterations if many illegal sub-tours exist. Another formulation called the Artificial Commodity Flow (ACF) is proposed to mitigate the drawback of the above approach. It considers the trip as water flow. The flow increases by one unit whenever it visits a stop. Therefore, the flow after one stop is strictly one unit greater than the flow before that stop (except the school). Therefore, if an illegal sub-tour exists (forms a loop without the school), there must exist a stop where the successive flow is less than the leading flow and violates the sub-tour elimination constraints (Bowerman et al., 1995). The benefit of this approach is that the problem only needs to be solved once. But the disadvantage is that an additional

$[(\text{number of stops})^2 \times \text{number of trips}]$ nonnegative continuous variables are required.

To address the additional variables problem, Miller et al. (1960) proposed another formulation which incorporated the sub-tour elimination constraints in the framework of the capacity or the ride time constraints. It is widely used in the general VRP because no additional variables are required. However, in the school bus planning problem where the bus load and ride time do not need to be tracked individually at each stop, this formulation would still require an additional $(\text{number of stops} \times \text{number of trips})$ nonnegative continuous variables. Still, it uses fewer additional variable than the Artificial Commodity Flow formulation. This formulation was adopted by Desrosiers et al. (1988) and Bektaş and Elmastaş (2007).

2.1.4 Algorithms

To solve the SBR, or the general VRP, the early attempts were made to use exact methods like modified assignment problem (Laporte et al., 1986), spanning tree search (Christofides et al., 1981a), dynamic programming (Christofides et al., 1981b; Desrosiers et al., 1986a), and labeling (Desrosiers and Soumis, 1988). However, due to the increase in the problem size, more efforts were put to develop heuristic algorithms.

2.1.4.1 Modified assignment Problem

One of the first attempts to solve the SBR using an exact method was the modified assignment problem. The idea was first developed to solve the TSP. Laporte et al. (1986) proposed to solve an m-TSP using the modified assignment formulation by duplicating the depots and expanding the distance matrix. The m-TSP can be converted into an assignment problem on the expanded distance matrix after relaxing

the sub-tour elimination constraints. It was also adopted by Laporte et al. (1988). Bektaş and Elmastaş (2007) applied this idea and transformed the SBR into a graph problem of finding the *m* node-disjoint paths (*m* equals to the number of trips). The SBR was formulated as an assignment problem. They tested it on a problem with 29 stops.

2.1.4.2 Set partition and column generation

The major task for the SBR is to design a set of trips to serve all stops. An intuitive way to formulate the problem is to decide the visiting sequence of the stops on the trips. Set partition formulation, on the other hand, solves the problem from another perspective. It assumes that we can list all possible trips or at least have a trip pool filled with good trip candidates, the decision then becomes finding the best combination of the trips to serve all stops. The set partition is a good mathematical formulation, and it is very easy to solve with a reasonable number of variables. However, the number of all possible trips is huge! In a complete graph with *n* stops, there are 2^n possible trips. Enumerating all feasible trips is impossible.

Two methods were proposed to handle this problem: 1) design a moderate-sized trip pool containing 'good' trip candidates; 2) dynamically add 'good' trip candidate into the pool. These two methods were usually used together. The Column Generation (CG) iteratively identify 'good' trip candidates and add them into the pool. A pool manager monitors the size of the pool and deactivate some 'bad' trips to have manageable restricted problem.

The Column Generation divides the problem into a master problem and a subproblem. The master problem is the LP (linear programming) relaxation of the set

partition formulation. This is because the subproblem needs the dual prices of the constraints (corresponding to the stops) from the master problem to find a good trip. The good news is that the master problem is near total unimodular and would often yield an integer solution if the problem size is relatively small (Toregas and ReVelle, 1972), which will be explained in more detail in section 5.3.1.4.

The subproblem is to find the best trip with the most negative reduced dual price under the current dual price (for the stops) from the master problem. After solving the subproblem, the trip is added to the trip pool. The master problem is solved again with the newly augmented trip pool. The subproblem can be considered as a Shortest Path Problem with the capacity and maximum ride time constraints (Laporte, 1992). The reduced dual price is the expected decrease of the objective function in the master problem if the new trip enters the basis. If the optimal solution in the subproblem has a nonnegative reduced dual price, there is no way to further reduce the objective function in the master problem. For a minimization problem, it means the optimality is reached. The Column Generation was adopted by Desrosiers et al. (1984), Desrochers et al. (1992), Xu et al. (2003), Riera-Ledesma and Salazar-González (2013), Caceres et al. (2018) and Santana and Carvajal (2015). A detailed discussion of the practical techniques of column generation can be found at a paper from Kinable et al. (2014) or the dissertation from Kinable (2014).

2.1.4.3 Lagrangian relaxation

Lagrangian relaxation is to relax some complicated constraints and put them in the objective function. The remaining constraints should be much simpler. The violation of the relaxed constraint will result in a huge penalty in the objective function.

Lagrangian relaxation can be used to find a lower bound for a (minimization) problem, and it can also be incorporated into the Branch and Bound algorithm. It was adopted by Desrosiers et al. (1988), Lamatsch (1992) and Mesquita and Paixão (1992). Due to the increased demand of the level of service and the problem size, it is less popular for solving the SBR than it was and as compared to other heuristic algorithms.

2.1.4.4 The RFCS and the CFRS

One widely used approach to solve complicated problems is clustering. In other words, break the problem into smaller pieces. In the SBR, based on the time to apply clustering, the heuristic algorithms can be classified into either the Route First, Cluster Second (RFCS) (Beasley, 1983) or the Cluster First, Route Second (CFRS) (Bodin and Berman, 1979).

Route First, Cluster Second develops one long trip as the solution to a 1-TSP after relaxing the capacity and maximum ride time constraints. It then partitions this long trip into several short sub trips where each sub trip satisfies the capacity and maximum ride time constraints. One application of the RFCS was Bowerman et al. (1994). The Cluster First, Route Second (CFRS) groups the bus stops into several clusters first where the total number of students in each cluster is less than or equal to the bus capacity. Each cluster is then solved as a 1-TSP problem. After solving all clusters, if some trips violate the maximum ride time constraints, the local search methods are applied to remove and re-insert some stops into their neighbors to find a feasible solution. This was adopted by Bowerman et al. (1995).

2.1.4.5 Insertion-based algorithms

Another family of heuristic algorithms is called the Insertion-Based Algorithms (IBA), which builds the trips gradually by iteratively inserting un-routed stops into the trips. The routing plan will be obtained after inserting all stops into trips. Based on the number of stops to insert at each iteration, the IBA is classified into either sequential or parallel IBA. The Sequential Insertion-Based Algorithms (SIBA) initialize trips one at a time and insert the un-routed stop with the least insertion cost into the current trip. When it is impossible to insert more stops into the current trip due to capacity or ride time, a new trip is initialized, and the insertion procedure is repeated until all the stops are routed. Bramel and Simchi-levi (1995) provided one realization of the SIBA called the Location Based Heuristic. The SIBA was also adopted by Braca et al. (1997) and Corberán et al. (2002).

One problem of the SIBA is that the last trip is usually of low quality (Potvin and Rousseau, 1993). The Parallel Insertion-Based Algorithms (PIBA), on the other hand, initialize several trips together. At every iteration, they can insert either one stop or multi-stops together. If they insert one stop, this stop-to-trip insertion must be the least-cost pair between every un-routed stop to every trip. If they insert multiple stops at one iteration, the sum of the insertion cost for the selected insertion pairs should be the least among all other possible combinations. Christofides et al. (1979) proposed a two-phase algorithm which applied the SIBA followed by the PIBA. The IBA is also the base of some complicated algorithms like Generalized Insertion Procedure (GENI), which was developed by Gendreau et al. (1992).

In the PIBA, the number of trips to initialize is a critical task. The most commonly used number is the ceiling integer of the total number of students divided by the bus capacity. It works well if capacity is the only binding constraint, but it may fail if the maximum ride time dominates the constraints. The insertion cost is usually the additional travel time due to the insertion of a stop. After constructing the insertion cost matrix, the least-cost insertion pair(s) are selected. An alternative selection strategy picks the insertion which has the highest gap between the least-cost insertion and the second-least cost insertion. The rationale is that if such insertion is not implemented immediately, the insertion cost will increase dramatically at the next iteration (Solomon, 1987).

A variance of insertion-based algorithms is called the Sweep Algorithm. Instead of choosing the insertion pair based on the insertion cost, it simply inserts the bus stops into trips based on their polar angles. This method is usually implemented several times at different starting point and in both clockwise and counterclockwise directions. Each trip is then solved as a 1-TSP problem and/or improved using some local search methods like 2-opt. It was first developed by Wren and Holliday (1972) and was adopted by Gillett and Miller (1974) and Park et al. (2012).

A simplified insertion-based algorithm is the Clarke and Wright Saving Method (CWSM). The CWSM calculates the savings (negative insertion cost) for every node pair at the beginning. The saving of a node pair is the sum of the travel time if serving two nodes separately minus the minimum travel time of serving two nodes by one trip. The CWSM builds the trips by sequentially adding the node pair in the decreasing order of their savings. The CWSM was shown to be a good method to generate an initial

solution. It was used by Tillman (1969) and Osman (1993). A more complicated version of the CWSM is the Greedy Randomized Adaptive Search Procedure (GRASP), which adds some randomness into the procedure. It was adopted by Kontoravdis and Bard (1995), Schittekat et al. (2013) and Faraj et al. (2014). An implementation of the GRASP in this dissertation will be presented in section 5.3.1.6.

2.1.4.6 Local search

Given the solutions from the insertion-based algorithms, many researchers apply the Local Search (LS) heuristics to improve the solution. The local search methods are universal and can be used to improve any solution obtained by any initialization method. The basic mechanism of a local search heuristic is to remove either edges or nodes from one location and reinsert them in another location based on an estimated cost. If edges are removed and reinserted, it is often referred to as the edge exchange. If nodes are removed and reinserted, it is known as node exchange or chain exchange (Babin et al., 2007). It is very rare to remove and reinsert edges and nodes together in one iteration. Based on the reconnection location, LS could be an in-trip algorithm if the reinsertion happens on the same trip. On the contrary, LS can be a between-trips method if the reconnection happens on another trip. The reconnected edges or nodes can be reinserted in the same order or the reversed order.

Based on different characters of the local search (re-inserting edges or nodes, number of edges or nodes to remove at a time, in-trip or between-trip move and the order of reconnection), different local search algorithms were developed such as the R-opt, the 2-opt, the 3-opt, the Or-opt etc. (Or, 1976). The most common local search algorithm is the 2-opt, which reverses the nodes order on one trip between two selected

break nodes. The 2-opt goes through every possible pair of the break nodes and finds the best-reconstructed trip. It is widely used to remove the cross in a trip. An illustration is shown in Figure 5. By reversing the order in which the nodes are visited between two break nodes (node b and c in this example), the cross is eliminated.

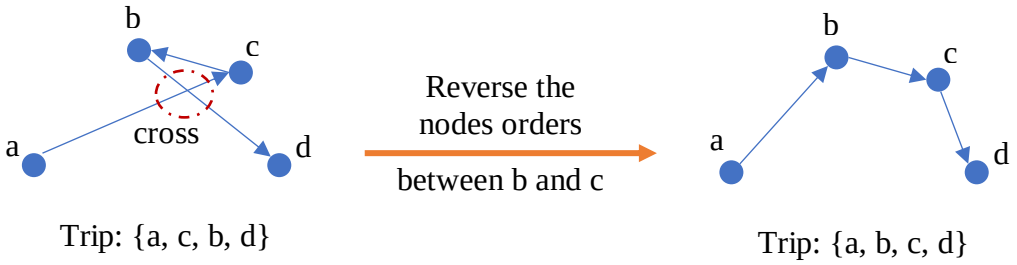


Figure 5 Illustration of the 2-opt

In terms of the acceptance strategies, there are two common selection methods: best-improvement and first-improvement (Osman, 1993). The best-improvement strategy enumerates all the possible reconnections and chooses the one that yields the best new trip. The first-improvement strategy accepts the first move that satisfies the acceptance criteria. The best-improvement strategy finds better reconnection at every iteration than the first-improvement strategy. But the first-improvement strategy is faster and can complete more iterations under the same running time. There is no general rule of the performance of these two strategies.

2.1.4.7 Tabu search

Randomness is the key point of the local search methods as it helps to find a better solution. But it could be wasteful if it is completely random and blind. Hence, the local search methods are usually implemented under the framework of meta-heuristics like the Tabu Search (TS), and/or the Simulated Annealing (SA). They can help to guide the local search direction and jump out of local optima. The core idea of

the Tabu Search is to improve the search efficiency by forbidding wasteful moves. The wasteful moves could be the ones that involve the nodes or the edges that are frequently moved or the moves of which the reserved moves were implemented a few iterations before.

The Tabu Search is quite flexible due to the various Tabu strategies, the different Tabu list size and different stopping criteria. The Tabu strategy could be applied to the edges or nodes as well as the moves. The Tabu could be hard such that any violation of the Tabu is forbidden or soft where the Tabu can be violated at the expense of a penalty. If an edge, node or a move becomes a Tabu, it could stay in the Tabu list forever or leave the Tabu list after a certain number of iterations. The most common stopping criteria for the Tabu Search are 1) the maximum number of iterations; 2) the maximum number of iterations where no change occurs; 3) the maximum running time, and 4) when the solution reaches the lower or upper bound. All trip initialization methods and the local search methods can be incorporated into the Tabu Search. Glover (1989), Potvin et al. (1992), Gendreau et al. (1994), Rochat and Semet (1994), Renaud et al. (1996), Badeau et al. (1997), Taillard et al. (1997), Fu et al. (2005), Pacheco and Martí (2006) and Mushi et al. (2015) all applied the Tabu Search to solve the VRP or the SBR.

2.1.4.8 Simulated annealing

Similar to the Tabu Search, the Simulated Annealing (SA) is quite flexible and can guide the direction of the local search using any trip initialization method and any local search algorithm. The major difference of SA compared to other methods is that SA might accept a move or a temporary solution even if it has a worse objective value

than the current best solution, while other methods would reject this move or the temporary solution directly. The control parameter in SA is the temperature. It starts at a high level and gradually drops until reaching the stopping value. The temperature determines the running time, the stopping criteria, and the current status (how likely we would accept a solution if it is worse than the current best solution). In the VRP, the initial temperature could be the number of trips multiplied by an adjustment coefficient. The idea is that the more trips it has, the larger solution space it has, and thus we should perform more local search iterations to find the best solution. Furthermore, the larger solution space is, the more tolerant we should be of accepting a worse solution at first such that the chance of jumping out of a local optimum is as large as possible.

At every iteration, the probability of accepting a move or a temporary solution is determined by the difference between the objective values of the new solution and the current best solution as well as the temperature. If the new solution has a better objective value, SA will accept this new solution. If the new solution has a worse objective value, the probability of accepting this new solution will decrease as the difference between the new and old solution increases and decrease as the temperature decreases. The SA will terminate if the temperature drops below the stopping temperature or the algorithm reaches the maximum number of iterations. It was adopted by Kuo (2010) and Chen et al. (2015).

2.1.4.9 Genetic algorithm

The Genetic algorithm (GA) was adopted by Thangiah and Nygard (1992) as the first step of the Cluster First, Route Second method. Prins (2004), on the other hand,

used the GA as the second step of Route First, Cluster Second method. Other applications included Diaz-Parra et al. (2012) and Kang et al. (2015).

2.2 Bell time adjustment

Compared to the school bus routing and scheduling problems, the bell time adjustment is significantly neglected. One of the reasons is that it is difficult to formulate the bell time adjustment problem solely due to the lack of a clear objective function. To the best of the author's knowledge, there are only two papers that optimized the school bell time. Both incorporated bell time adjustment in the school bus scheduling problem.

To optimize the school bell times, first, we assume school bell time windows are given. Kim et al. (2010) and Chen et al. (2015) held the assumption that each trip can start at any time within the bell time windows. Under this assumption, they solved the school bus scheduling problem. The actual the school (AM) bell time could be the latest trip arrival time at each school. The actual the school (PM) dismissal time could be the earliest trip departure time for each school.

However, this is rarely used in practice because holding a portion of the students after school involves many safety issues and requires huge labor. In practice, all (PM) trips need to depart right after school dismissal time or with a very small buffer. Braca et al. (1997) mentioned that the school buses should arrive at schools no earlier than 25 minutes before school bell times and no later than 5 minutes after school dismissal times. Fügenschuh (2009, 2011) requires that all trips arrive at school within 5 minutes of the school bell times.

The integrity constraints of the school bell times mean that the school bell times should be an integer multiple of a minimum time interval rather than a continuous variable. There also exists a synchronization constraint. One example is that all high schools need to dismiss at the same time such that extracurricular activities, like sports competitions, can start simultaneously.

2.3 School bus scheduling

The School Bus Scheduling (SBS) is the final stage of the SBPP. After solving the SBS, the final school bus planning will be revealed including the arrival time of the bus at every stop and school, the assignment of trips to buses, the student load on each trip, the travel time of the trips, the deadhead on each bus, etc.

2.3.1 Problem transformation

The most common way to solve the SBS is to transform the problem into a Time-dependent Directed Scheduling Graph (TDSG) and solve the problem as a time-constrained VRP (Haghani and Banihashemi, 2002). The first step is to construct the TDSG. In this graph, each node represents a trip containing all information like the travel time of this trip, the first and last stop of this trip, and earliest and latest start time of this trip and the number of students on that trip if using heterogeneous fleet. The directed edges of the graph imply the compatible trip pairs. The weights or the cost of each edge is the deadhead of each compatible trip pair. If there is no edge between two nodes (trips) or if the edge is one-way directed, the following trip cannot be served right after the preceding trip on the same bus.

One example of the TDSG is shown in Figure 6. There are three trips. Each trip has two properties – the start time window and the travel time.

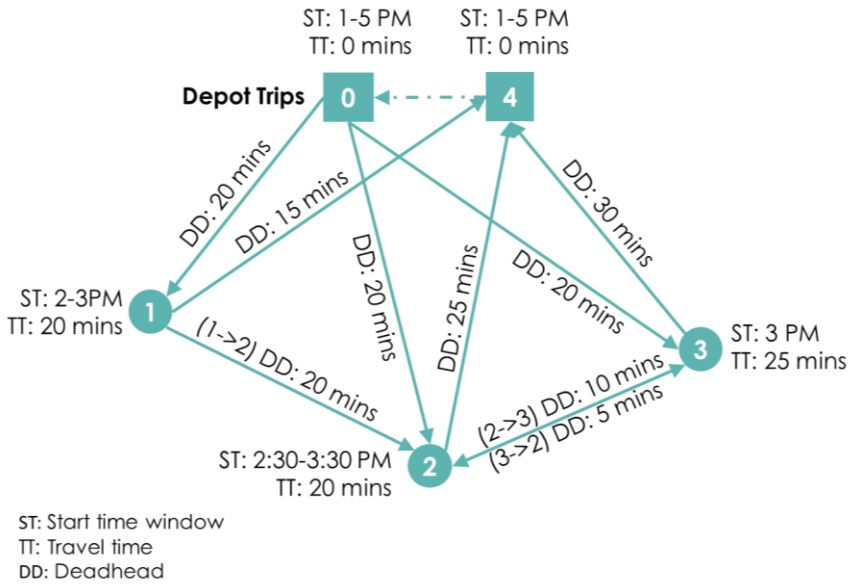


Figure 6 Illustration of the time-dependent directed scheduling graph

If a bus serves trip 1 at its earliest (2 PM), spends 20 minutes to finish trip 1 and travels 20 minutes to reach trip 2, the bus can start to serve trip 2 at 2:40 PM, which is between trip 2's start time window. It means that trip 1 is compatible with trip 2 and its weight (deadhead) is 20 minutes. For the completeness of the TDSD, the depot trip(s) are often added with zero travel time and very loose start-time window. Thanks to the homogeneous fleet assumption, each trip must satisfy the capacity constraint, and the only constraint which will raise incompatibility is the time window constraint. In a heterogeneous fleet problem, however, different trip capacities would also lead to incompatibility.

After constructing the TDSG, the SBS can be transformed into a time-dependent VRP. In the original VRP, nodes represent stops and paths are the trips. In

the TDSG, nodes represent the trips, and the paths are the bus routes. Because of the similarity, the formulation and algorithms developed to solve the VRP can be applied to the SBS with the consideration of time window constraints.

2.3.2 Objective

The goal of the SBPP is to transport students between their homes to the schools with the minimum number of buses and minimum total vehicle time/distance (Kim et al., 2010; Chen et al., 2015). The school bus scheduling problem, as the final stage of the SBPP, shares the same goal of the SBPP. The objective function for the SBS is to minimize the weighted sum of the number of buses used and the total vehicle time/distance. The total vehicle time/distance is the travel time for the trips (with students on board) and the deadhead (without student on board). Assuming the routing plan is fixed for the SBS, the total travel time/distance of the trips are fixed, thus minimizing the total vehicle time/distance is equivalent to minimizing the total deadhead.

2.3.3 Constraints

2.3.3.1 Logistic constraints

Due to the similarity between the SBS and the VRP, the basic logistic constraints in the SBS are similar to that from the SBR and the VRP. In the homogeneous fleet case, two-index variables are used (see section 2.1.3.1). But in a heterogeneous fleet setting, three-index variables are inevitable (see section 2.1.3.1). The only differences between the SBS and the SBR are 1) the SBR generates the trips while the SBS generates the routes; 2) the start and end node in the SBR is the school,

which becomes the depot in the SBS. Mathematically, the logistic constraints in the SBR and the SBS are the same.

2.3.3.2 Trip compatibility constraints

The major difference between the SBS and the VRP is that in the VRP, it is possible to travel from every node to every other node. If ignoring some specified constraints, the graph of the VRP is a complete graph. But this is not true in the SBS. The trip compatibility constraints make sure a preceding trip can only be served followed by a successive trip if the exact start time of the preceding trip plus the travel time of the preceding trip and the deadhead between these two trips is less than or equal to the exact start time of the successive trip.

2.3.3.3 Time window constraints

If all trips can start anytime within the time window, then the time window constraints need to make sure that no trip starts before or after its time window. However, as discussed above, the trip start times should be within a small buffer of the school bell/dismissal time. If the school bell/dismissal times are not predetermined, the trip start times for one school need to be synchronized to a small interval. In a more extreme case where no buffer is allowed, the trip start times would equal to the school bell times.

2.3.3.4 Maximum vehicle time constraints

In the SBR, the major level-of-service concern is the travel time of the students. Similarly, the level-of-service concern in the SBS is the Maximum Vehicle Time (MVT) of the buses. The vehicle time consists of the travel time of the trips (with students on board) and the deadhead between trips and deadhead between the trips and the depot

(without students on board). This is also important in the case where a contractor provides the school bus service. The service cost for one bus is usually a flat rate per day (within a maximum vehicle time, i.e., 6 or 8 hours per day) and the extra cost. The extra cost is charged when the vehicle time of a route exceeds its desired maximum flat rate time (6 or 8 hours). The extra cost is commonly proportional to the extra travel time with a per-hour-rate much higher than the flat rate cost. The maximum vehicle time constraints are added to make sure that no route violates the maximum operation time limit and that no extra cost is applied. However, this constraint is neglected by many papers. Shafahi et al. (2017) tried to balance the route length by replacing the last trip of a long route to a short route. This approach balances the route length to some extent, but the maximum vehicle time is not guaranteed.

2.3.4 Algorithms

2.3.4.1 Modified assignment Problem

If the start times of trips are fixed, the SBS becomes a modified assignment problem (Kim et al., 2010). The fixed trip start times may come from different scenarios: 1) the school bell times are fixed, and all trip start times equal to the school bell times or 2) the trips are generated with unique start times from an SBR problem with time window constraints. In either case, the trips have fixed start time, which means that the trip compatibilities are determined with fixed deadhead. Then it becomes an assignment problem, and it does not need the TDSG.

2.3.4.2 Set partition and column generation

Similar to the SBR, set partition changes the SBS into finding the best combination of possible bus routes to serve all trips. Obviously, listing all feasible

routes is impossible. Thus, column generation is adapted to generate good routes dynamically. The master problem is of the same form as it is in the SBR. But the subproblem is more complicated because of the trip compatibility and time window constraints. The set partition and column generation has been adopted for the SBS by Fügenschuh (2011).

2.3.4.3 Time-dependent insertion-based algorithm

Many efforts are put forth to expand the Insertion-based Algorithms (IBA) to solve the SBS. Solomon (1987) expanded several IBA (saving algorithm, insertion algorithm, nearest neighbor algorithm, sweep algorithm) to solve the school bus routing and scheduling problem simultaneously. The biggest difference between the time-dependent IBA and the traditional IBA is the consideration of time windows. It is applied in two ways: the insertion cost estimation and the feasibility check. In insertion cost estimation, these time-dependent algorithms would add the urgency of a node or the 'time closeness' of a node into the insertion cost estimation.

On the other hand, the feasibility check makes sure that the insertion will not violate the time window constraints. It is very time-consuming since inserting a stop to a trip or inserting a trip to a route would affect the service time for all successive stops or trips (Kim et al., 2010). Instead of explicitly checking the feasibility of every node and trip, Solomon (1987) proposed a faster necessary and sufficient condition to check the feasibility. Potvin and Rousseau (1993) also adopted time-dependent insertion-based algorithms. Similarly, the time-dependent IBA is usually followed by some post-improvement algorithms, and they could be guided by meta-heuristics like the Tabu

Search, the Simulated Annealing, etc. Brandao and Mercer (1997) adopted a Tabu Search algorithm to solve the school bus routing and scheduling problem.

2.4 Stochasticity

More recently, many efforts have been put forth to solve the SBPP with considerations of stochasticity in the system. In general, stochastic programming models can be categorized into chance-constrained programming (CCP, also written as chance constraints programming) and models with recourse (RM) (Gendreau et al., 1996). The chance-constrained programming models prohibit the violation of chance constraints at a certain probability. The models with recourse allow the violation of chance constraints. A recourse will be implemented in case of a failure at the expense of a penalty. The objective includes the cost under the satisfaction of chance constraints as well as the penalty of breaking these constraints. Usually, models with recourse are more sophisticated than chance-constrained programming models.

Laporte et al. (1992) proposed a chance-constrained model and a model with recourse for the general vehicle routing problem. The chance-constrained model upper-bounded the probability for every bus route to exceed a fixed duration limit. The model with recourse minimized the number of buses, the total travel distance (travel) as well as the expected penalty of excess bus route duration under different realization of the stochastic travel times. The models were solved using a branch-and-cut procedure.

Hvattum et al. (2006) proposed a multi-stage recourse-considered scenario-based stochastic programming model for a dynamic vehicle routing problem with time window constraints. They considered stochastic customers as well as stochastic travel time. The model was solved using the Dynamic Stochastic Hedging Heuristic (DSHH).

The core idea of the DSHH is gradually building a good solution using the common features from the solutions for all sample scenarios. A benchmark problem was solved for validation, which had 20 instances with up to 151 customers. A similar dynamic vehicle routing problem with stochastic customers was also solved by Ben and Hentenryck (2004) using a scenario-based approach. Their objective was to maximize the number of served customers.

Yan et al. (2015) proposed a multi-commodity network flow formulation for inner-school bus routing and scheduling problem with stochastic travel time. The uncertainty of the travel time was taking into account by branching out several edges from a node, where each edge has a unique travel time with a probability. The authors specifically considered the passenger flow separated from the vehicle flow for the purpose of transfer. Their objective was to minimize the bus operation cost; the passengers travel time and the unanticipated penalty cost for the scheduling. A decomposition-based matching algorithm was proposed to solve the stochastic programming model. The case study was conducted on a university system in Taiwan with 400 passengers.

Caceres et al. (2017) presented a chance-constrained programming (CCP) formulation for the stochastic school bus routing problem. They considered the stochastic demand at the bus stops as well as the stochastic travel time between the nodes. Three chance constraints were considered: 1) the probability of overcrowding of buses is upper-bounded by a fixed level; 2) the probability of being late for school must be less than or equal to a fixed level; 3) the expected maximum duration of the bus routes is bounded. With binomial attendance of students at bus stops and normal

distribution of the travel time between nodes, Caceres et al. (2017) linearized these chance constraints and formulated the problem as a Mixed Integer Programming (MIP) model. In terms of solving the model, Caceres et al. (2017) decomposed the whole system into many single-school, multi-depot subproblems, and each subproblem was solved using a heuristic algorithm. The heuristic algorithm found an initial solution using a Clarke and Writing Saving and Farthest First hybrid method. Then, a Column Generated-based algorithm was used to improve the initial solution. Caceres et al. (2017) tested the algorithm on the Williamsville Central School District in Western New York with 13 schools and showed that the original 86 buses could be reduced to 81 buses.

Samadi-Dana et al. (2017) presented a scenario-based robust optimization model to solve the school bus routing problem with stochastic travel time. They explicitly considered the expected pickup time for every student. The objective was to minimize the expected total travel time and the variance of the total travel time. The latter was used to improve the robustness of the system. The minimum and the maximum number of stops on a bus were both included but the maximum ride time constraint was ignored. A Simulated Annealing-based algorithm was proposed, and a case study of a university in Mazandaran, Iran with 115 students and six buses was solved.

2.5 Research gap

Upon the review of the literature, several fruitful areas of research emerge. First, as pointed out in Chapter 1, the integrated model for the whole school bus planning problem is missing. The importance of such a model does not need further justification

– the best possible integrated school bus routing, scheduling and bell time plan can only be obtained from this integrated model. Second, given the integrated model, how to solve it efficiently with high solution quality is another critical task. Decomposition is a common approach in the literature. However, since the integrated model considered in this dissertation is unbelievably complicated, traditional decomposition methods do not perform well. We must improve the traditional decomposition algorithm to better fit this problem.

Moreover, the bell time adjustment, which is essential in the school bus planning problem has been severely neglected in the literature. Similarly, the level-of-service such as maximum ride time (per trip) and maximum vehicle time (per route) also need more attention in the school bus transportation planning. They would both be considered in this dissertation. In addition, to make the problem more representative of the uncertain real world, stochastic travel times should be considered. The stochastic school bus routing problem has been formulated and solved, but the stochastic school bell time and bus scheduling problem lack research.

In summary, the tasks we will address in this dissertation include:

i. Exploring the possibility of formulating the integrated model to solve the school bus routing, bell time and scheduling simultaneously with the consideration of all level-of-service constraints;
ii. Finding a novel decomposition-based strategy to divide the integrated model into several subproblems and find good solutions for the whole system;

iii. Developing new meta-heuristic algorithms to solve the subproblems if they are still complicated and/or if solving the subproblems for large size problems is very time consuming;

iv. Providing several flexible combinations of the models and algorithms that can solve different school bus planning problems with different scopes, assumptions and need;

v. Developing new methods to solve the stochastic school bell time and bus scheduling optimization problem to provide a better and more reliable transportation plan under uncertain real-world cases.

2.6 Chapter summary

In this chapter, the literature of the school bus planning problem was reviewed and discussed. The literature was summarized based on the problem scope and the content – school bus routing, bell time adjustment, school bus scheduling, and the stochastic models. In each problem, the objective, the constraints, and algorithms used to solve the problem were discussed in detail.

This literature review focused on the underlying rationales of the formulations and algorithms. Thus, it spent more time on discussion than summarizing. The advantages, disadvantages, and applications of each approach and the relationship between these approaches were presented. Some ideas discussed in the literature review are also adopted in this dissertation. These discussions will become clear when the application of these ideas in this dissertation is presented.

Chapter 3 Deterministic School Bell Time, Bus Routing and Scheduling Optimization Models

The school bell time, bus routing and scheduling all have a huge impact on the school bus planning problem. Clearly, solving the integrated problem would provide the best solution. A Mixed Integer Programming (MIP) mathematical model is proposed to solve this integrated problem. This Multi-School Bell time, Bus Routing and Scheduling with the Maximum Vehicle Time Constraints problem (MBRSV[1]) is noted as the Model 1. It considers the Maximum Ride Time (MRT) and the Maximum Vehicle Time (MVT) constraints.

3.1 Assumptions

Here are the assumptions of the MBRSV (Model 1):

- The problem is designed for the afternoon (PM) bus transportation plan for a multi-school system. The task is to transport students from their schools to the stops;
- Each school has a number of bus stops. The locations of the stops and the number of students at the stops are fixed and known;
- Every bus stop must be visited exactly once (single-visit);
- Trips are dedicated to each school, and one trip would only pick up or drop off students for one school at the designated stops (single-load);
- All the buses have the same capacity (homogeneous fleet);

[1] Model naming rule: **M**: Multi-school, **S**: Single-school, **B**: Bell Time Adjustment, **R**: Bus Routing; **S**: Bus Scheduling, **V**: Maximum Vehicle Time Constraint, **C**: Compatibility. The same naming rule applies to all models throughout this dissertation.

- The school bell/dismissal times are constrained within a known time window. The bell/dismissal times are the integer multiple of a minimum time interval;
- All trips should depart at the school dismissal times with no buffer.

The multi-school problem commonly appears in a public-school district. The county's Department of Education would provide the school bus transportation service for all public schools in the county. The homogeneous fleet comes from the standardized school buses. The special aid service, which may require different size vehicles, is usually considered as a separate service from the general transportation service. This study mainly focuses on the urban public-school problems where the mixed load is prohibited.

3.2 Notations for model formulation

The notation used in the model formulation is summarized in Table 3. The potential trip set ($TP[k], \forall k \in SCH$) needs to be defined first. Clearly, a lower bound on the number of trips for a school can be calculated based on the bus capacity constraints. The minimum number of trips (MNT) is defined as follows:

The minimum number of trips (MNT) (per school): the smallest ceiling integer of the total number of students divided by the bus capacity. It is the minimum number of trips, with respect to the capacity, that is required to transport all students.

$$MNT[K] = \left\lceil \frac{\sum_{s \in SP[k]} STU_s}{Cap} \right\rceil, \forall k \in SCH \quad (3\text{-}1)$$

It is hard to determine a tight upper bound of the number of trips due to the maximum ride time constraint. In this dissertation, the upper bound is set to be the MNT pluses an artificial number – the Additional Allowed Trips (AAT).

Table 3 Notation summary for MBRSV (Model 1)

Variables for the integrated model	
Variable name	**Description**
sb_k	Integer variable, the school bell (dismissal) time
$st_{s,t}$	Binary variable equals 1 if stop s is assigned to trip t
ta_t	Binary variable equals 1 if trip t is in use (activate)
$x^t_{s1,s2}$	Binary variable equals 1 if stop $s1$ is followed by stop $s2$ on trip t
$y_{t1,t2}, y^b_{t1,t2}$	Binary variable equals 1 if trip $t1$ is followed by trips $t2$ on one bus (or on bus b)
$y_{t,k}$	Binary variable equals 1 if trips t is compatible with school k
$l_{s,t}$	Binary variable equals 1 if stop s is the last stop of trip t
tt_t	Nonnegative continuous variable, the travel/ride time of trip t
$dd_{t1,t2}, dd_{t1,k}$	Nonnegative continuous variable, the deadhead duration from the last stop of trip $t1$ to the first stop of trip $t2$ or to the school k
$ddb_{t1,t2}, ddb^b_{t1,t2}$	Nonnegative continuous variable, the deadhead duration from trip $t1$ to trip $t2$ on one bus (or on bus b). This is for linearization, which will be explained section 3.3.2.3.
$ddb_{t,k}, ddb^b_{t,k}$	Nonnegative continuous variable, the deadhead duration from trip t to school k on one bus (or on bus b). This is for linearization.
$ac^t_{s1,s2}$	Nonnegative continuous variable, the units of "artificial commodity" that is shipped from stop $s1$ to $s2$ by trip t (use for sub-tour elimination constraints, see section 2.1.3.5)
Parameters for the integrated model	
Parameter name	**Description**
SCH	Set of schools
$TP, TP[k]$	Set of all potential trips, the potential trips dedicated to school k
$SP, SP[k]$	Set of all stops, the stops belong to school k
B	Set of possible buses
Cap	The capacity of each bus
STU_s	The number of students at stop s
$PT(stu)$	The pickup time at schools with *stu* number of students
$DT(stu)$	The drop-off time at stops with *stu* number of students
MNT	The minimum number of trips (for each school)
AAT	The number of additional allowed trips (AAT)
$O\{t\}$	The school that trip *t* belongs to
SDT,EDT	The start/end depot trip
$D_{s1,s2}$	The duration to drive from stop $s1$ to $s2$
M	A large positive value (big-M)
$\alpha_B, \alpha_T, \alpha_D, \alpha_N, \alpha_C$	The coefficient for the number of buses (α_B), total ride time of the trips (α_T), total deadhead between trips (α_D), number of trips (α_N), trip-to-trip compatibilities (α_C)
MRT, MVT	The maximum ride time; the maximum vehicle time
ξ	The minimum bell time interval

The additional allowed trips (AAT): the maximum number of additional number of trips that can be used for each school above its MNT (e.g., school A has 180 students, bus capacity is 48, then the MNT $= \lceil 180/48 \rceil = 4$. If the AAT is set to be the same as the MNT, at most 8 (4 plus 4) trips can be used for school A).

Given the lower bound (the MNT) and the upper bound (MNT+AAT), the potential trip set for every school is defined. The depot trips are the dummy trips with a ride time of zero. For the completeness of the bus routes, buses depart from and return to the depot. The last stops ($l_{s,t}$) of the depot trips are also the depot.

3.3 The MBRSV (Model 1)

The mixed integer linear programming model is presented as follows.

3.3.1 Model formulation

The MBRSV (Model 1)

$$\min z1 = \alpha_B \sum_{b\in B} \sum_{t\in TP} y^b_{SDT,t} + \alpha_T \sum_{t\in TP} tt_t + \alpha_D \sum_{b\in B} \sum_{t1\in TP\cup SDT} \sum_{t2\in TP\cup EDT\setminus t1} ddb^b_{t1,t2} \quad (3\text{-}2)$$

Subject to

(Routing constraints)

$$\sum_{s2\in SP[k]\cup k\setminus s} x^t_{s,s2} = \sum_{s1\in SP[k]\cup k\setminus s} x^t_{s1,s}, \forall k \in SCH, s \in SP[k] \cup k,\ t \in TP[k] \quad (3\text{-}3)$$

$$\sum_{s1\in SP[k]\cup k\setminus s} x^t_{s1,s} = st_{s,t}, \forall k \in SCH, t \in TP[k], s \in SP[k] \quad (3\text{-}4)$$

$$\sum_{t\in TP[k]} st_{s,t} = 1, \forall k \in SCH, s \in SP[k] \quad (3\text{-}5)$$

$$st_{s,t} \le ta_t, \forall k \in SCH, s \in SP[k], t \in TP[k] \quad (3\text{-}6)$$

$$ta_t \le \sum_{s\in SP[k]} st_{s,t}\ \forall k \in SCH, t \in TP[k] \quad (3\text{-}7)$$

$$\sum_{t\in TP[k]} \sum_{s\in SP[k]} x^t_{k,s} = \sum_{t\in TP[k]} ta_t, \forall k \in SCH \quad (3\text{-}8)$$

$$\sum_{s\in SP[k]} STU_s \times st_{s,t} \leq Cap, \forall k \in SCH, t \in TP[k] \quad (3\text{-}9)$$

$$\sum_{i\in SP[k]\cup \mathrm{k}\setminus s} ac_{i,s}^{t} - \sum_{j\in SP[k]\cup \mathrm{k}\setminus s} ac_{s,j}^{t} = st_{s,t}, \forall k \in SCH, t \in TP[k], s \in SP[k] \quad (3\text{-}10)$$

$$ac_{s1,s2}^{t} \leq M \times x_{s1,s2}^{t}, \forall k \in SCH, t \in TP[k], s1, s2 \in SP[k] \cup \mathrm{k} \quad (3\text{-}11)$$

(Scheduling constraints)

$$x_{s,k}^{t} = l_{s,t}, \forall k \in SCH, t \in TP[k], s \in SP[k] \quad (3\text{-}12)$$

$$tt_t = \sum_{s1\in SP[k]\cup \mathrm{k}} \sum_{s2\in SP[k]\setminus s1} x_{s1,s2}^{t} \times D_{s1,s2} + PT\left(\sum_{s\in SP[k]} STU_s \times st_{s,t}\right) \times ta_{\mathrm{t}} + \sum_{s\in SP[k]} DT(STU_s) \times st_{s,t}, \forall k \in SCH, t \in TP[k] \quad (3\text{-}13)$$

$$dd_{t1,t2} = \sum_{s1\in SP[O\{t1\}]} D_{s1,\mathrm{O}\{t2\}} \times l_{s1,t1}, \forall t1 \in TP \cup \mathrm{SDT}, t2 \in TP \cup \mathrm{EDT}\setminus t1 \quad (3\text{-}14)$$

$$sb_{O\{t1\}} + tt_{t1} + dd_{t1,t2} - M \times \left(1 - \sum_{b\in B} y_{t1,t2}^{b}\right) \leq sb_{O\{t2\}}, \forall t1 \in TP, t2 \in TP\setminus t1 \quad (3\text{-}15)$$

$$\sum_{b\in B} \sum_{t1\in TP\cup SDT\setminus t} y_{t1,t}^{b} = ta_t, \forall t \in TP \quad (3\text{-}16)$$

$$\sum_{t1\in TP\cup SDT\setminus t} y_{t1,t}^{b} = \sum_{t2\in TP\cup EDT\setminus t} y_{t,t2}^{b}, \forall b \in B, t \in TP \quad (3\text{-}17)$$

$$\sum_{t\in TP} y_{SDT,t}^{b} \leq 1, \forall b \in B \quad (3\text{-}18)$$

$$2 \times \sum_{b\in B} \mathrm{y}_{\mathrm{t1,t2}}^{b} \leq ta_{t1} + ta_{t2}, \forall t1 \in TP, t2 \in TP\setminus t1 \quad (3\text{-}19)$$

$$ddb_{t1,t2}^{b} \geq M \times \left(y_{t1,t2}^{b} - 1\right) + dd_{t1,t2}, \forall b \in B, t1 \in TP \cup \mathrm{SDT}, t2 \in TP \cup \mathrm{EDT}\setminus t1 \quad (3\text{-}20)$$

(Level-of-service constraints)

$$tt_t \leq MRT, \forall t \in TP \quad (3\text{-}21)$$

$$\sum_{t\in TP} tt_t \times \left(\sum_{t1\in TP\cup SDT\setminus t} y_{t1,t}^{b}\right) + \sum_{t1\in TP\cup SDT} \sum_{t2\in N\cup EDT\setminus t1} dd_{t1,t2} \times y_{t1,t2}^{b} \leq MVT, \forall b \in B \quad (3\text{-}22)$$

(Efficiency constraints)

$$tt_{t1} \times ta_{t1} \geq tt_{t2} \times ta_{t2}, \forall k \in SCH, t1, t2 \in TP[k] | t1 \leq t2 \quad (3\text{-}23)$$

$$\sum_{t\in TP} tt_t \times \left(\sum_{t1\in TP\cup SDT\setminus t} y_{t1,t}^{b1}\right) \geq \sum_{t\in TP} tt_t \times \left(\sum_{t1\in TP\cup SDT\setminus t} y_{t1,t}^{b2}\right), \forall b1, b2 \in B | b1 \leq b2 \quad (3\text{-}24)$$

(Domain constraints)

$$\mathrm{LSB_k} \leq \mathrm{sb_k} \leq \mathrm{ESB_k} \text{ and } \mathrm{sb_k} \bmod \xi = 0, \forall \mathrm{k} \in \mathrm{SCH} \quad (3\text{-}25)$$

$$x_{s1,s2}^{t} \in \{0,1\}, \forall t \in TP, s1, s2 \in SP \quad (3\text{-}26)$$

$$st_{s,t} \in \{0,1\}, \forall s \in SP, t \in TP \quad (3\text{-}27)$$

$$ta_{t} \in \{0,1\}, \forall t \in TP \quad (3\text{-}28)$$

$$l_{s,t} \in \{0,1\}, \forall s \in SP, t \in TP \quad (3\text{-}29)$$

$$y_{t1,t2}^{b} \in \{0,1\}, \forall b \in B, \forall t1, t2 \in TP \quad (3\text{-}30)$$

$$ac_{s1,s2}^{t} \geq 0, \forall k \in SCH, t \in TP[k], s1, s2 \in SP[k] \cup \mathrm{k} \quad (3\text{-}31)$$

$$tt_{t} \geq 0, \forall t \in TP \quad (3\text{-}32)$$

$$dd_{t1,t2} \geq 0, \forall t1 \in TP \cup \mathrm{SDT}, t2 \in TP \cup \mathrm{EDT} \backslash t1 \quad (3\text{-}33)$$

$$ddb_{t1,t2}^{b} \geq 0, \forall b \in B, t1 \in TP \cup \mathrm{SDT}, t2 \in TP \cup \mathrm{EDT} \backslash t1 \quad (3\text{-}34)$$

3.3.2 Formulation description

3.3.2.1 Objective

The objective minimizes the weighted sum of the number of buses and total vehicle time/distance. The number of buses is expressed by the number of bus routes that depart from the depot. The total vehicle time/distance includes the ride time of the trips (with students on board) and the deadhead between the trips (without students on board). Since the travel time can better reflect the real time spent in traffic, this model (and the rest of the dissertation) uses the travel time instead of the travel distance. The number of buses, the total ride time (with students on board) and the deadhead are assigned different weights. The number of buses is of higher priority than the rest because the bus acquisition and drivers' wage can easily overwhelm the marginal cost of the travel time increase. So α_{B} is of higher magnitude than α_{T} and α_{D}. Also, the ride time with students on board is more important than the deadhead without students on board. Then, we have $\alpha_B \gg \alpha_T > \alpha_D$.

3.3.2.2 Routing constraints

The constraints in the MBRSV (Model 1) can be divided into five sets: the routing constraints, the scheduling constraints, the level-of-service constraints, the efficiency constraints, and the domain constraints.

The first set of constraints are routing constraints. Constraints (3-3) are the conservation of flow, which guarantee that for every stop on every trip, the number of preceding stops (or the school) equals to the number of successive stops (or the school). Constraints (3-4) to (3-7) are stop-to-trip assignment constraints. Constraints (3-4) say that a stop *s* belongs to trip *t* if there is a stop or the associated school (of stop *s*) visited right before stop *s* on trip *t*. Constraints (3-5) make sure that every stop is visited exactly once. Constraints (3-6) prevent the assignment of stops to the trips that are not in use. Constraints (3-7) deactivate a trip if no stops are assigned to it. Constraints (3-8) regulate that for each school, the number of active trips equals to the number of stops that are visited right after the school. Constraints (3-9) are the capacity constraints, which ensure that for every trip, the sum of the students at the stops (on that trip) is less or equal to the bus capacity. Constraints (3-10) and (3-11) are sub-tour elimination constraints. These are formulated based on an Artificial Commodity which was discussed in section 2.1.3.5.

3.3.2.3 Scheduling constraints

The second part of the constraints is the scheduling constraints. First, the last stop (3-12) and the ride time (3-13) of each trip is calculated. The ride time of the trips include the travel time between stops and the pickup and drop off time for students at the schools and the stops. A widely used pickup (PT at schools) and drop off (DT at

stops) time regression model was proposed by Braca et al. (1997) (calculated in seconds). It was used by Park et al. (2012) and Chen et al. (2015).

$$\mathrm{PT(stu)} = 29.0 + 1.9 \times \mathrm{stu} \tag{3-35}$$

$$\mathrm{DT(stu)} = 19.0 + 2.6 \times \mathrm{stu} \tag{3-36}$$

Due to the linearity of the regression, it can be directly incorporated into the model as follows:

$$tt_t = \sum_{s1 \in SP[k] \cup \mathrm{k}} \sum_{s2 \in SP[k] \setminus s1} x^t_{s1,s2} \times D_{s1,s2} + 29 \times ta_t + (19 + 4.5 \times STU_s) \times \sum_{s \in SP[k]} st_{s,t}, \forall k \in SCH, t \in TP[k] \tag{3-37}$$

Then, constraints (3-13) is replaced by (3-37). Constraints (3-14) calculate the deadhead between every pair of trips. Constraints (3-15) identify the compatible trip pairs if the finish time of the preceding trip (the bell time of the respective school plus the ride time of the trip) plus the deadhead between two trips is less than or equal to the start time (school dismissal times) of the successive trip. Constraints (3-16) says every active trip has exactly one preceding active trip or the depot trips. Constraints (3-17) assign a pair of preceding and successive trips on the same bus. Constraints (3-18) make sure that each bus departs from the depot. Constraints (3-19) guarantee that trips can only be led or followed by active trips or the depot trips. Constraints (3-20) calculate the deadhead duration on each specific bus route. This is for the purpose of linearization. Otherwise, the deadhead term in the objective function would be $\sum_{\mathrm{b} \in \mathrm{B}} \sum_{t1 \in TP \cup SDT} \sum_{t2 \in TP \cup EDT \setminus t1} dd_{t1,t2} \times y^b_{t1,t2}$, which is a quadric term.

3.3.2.4 Level-of-Service constraints

Due to the safety concerns and the minimum satisfaction requirement, level-of-service constraints need to be satisfied. Because the students' MRT equals to the MRT

of that trip, the MRT constraint can be formulated as trip-wise constraints (3-21). Similarly, the MVT for drivers can be formulated as route-wise constraints (3-22).

3.3.2.5 Efficiency constraints

Efficiency constraints are not necessary in defining the feasible range, but they can significantly improve the model's efficiency. The idea adopted is called breaking symmetries. Let's say a trip is generated. Labeling this trip as trip No.1 is identical to labeling this trip as trip No.2. To break the symmetries, constraints (3-23) label the longest trip (for every school) with the smallest ID. Constraints (3-24) assign the longest bus route to the bus with the smallest ID.

Moreover, a typical method to improve the efficiency of the model is the big M reduction. We find the minimum value for the big M at every constraint where the big M is used. In constraint (3-11), the big M can be reduced to the maximum number of stops for a school. In constraint (3-15), the big M can be reduced to the maximum difference of two schools' dismissal time plus the preceding school's maximum ride time and the maximum deadhead duration from the preceding school's stops to the successive school. In constraint (3-20), the big M can be reduced to the maximum deadhead duration from the preceding school's stops to the successive school. The big-M reduction technique is applied in the rest of the dissertation whenever a big M is used.

3.3.2.6 Domain constraints

Constraints (3-25) to (3-34) are domain constraints. Constraints (3-25) bound the school bell time within the time window. They also guarantee that the school bell time is an integer multiple of ξ. Constraints (3-26) to (3-30) are binary constraints.

Constraints (3-31) to (3-34) are the nonnegative continuous domain constraints. The ride time and deadhead, although defined as variables, can be saved. Whenever ride time and deadhead appear in the formulation, we can replace them with the right-hand side of the expression in constraints (3-37) and (3-14), respectively. Then, constraints (3-37) and (3-14) can be deleted as well. These two variables are kept to make the formulation clearer and more understandable.

3.4 Decomposition scheme

The MBRSV (Model 1) is an integrated model for the school bell time, bus routing and scheduling problem with all level-of-service constraints. Due to its complexity, it is nearly impossible to solve this model to optimality even for a medium-sized problem. There are a few strategies to handle complicated problems. One of them is to relax some complicated constraints to have a simpler problem. And these relaxed constraints will be considered later. An alternative way is the decomposition approach. There are usually two ways of decomposition:1) divide the tasks of the problem into several subproblems where each subproblem only handles one task; 2) break down the problem into smaller subproblems. Both relaxation and decomposition strategies are adopted in this dissertation.

It was first noticed that by relaxing the MVT constraints, the MBRSV (Model 1) could be much simpler in terms of the number of variables and constraints. The MVT relaxed model is called the Multi-School Bell time, Bus Routing and Scheduling Problem (MBRS, Model 2). The relationship between the solutions to the MBRSV (Model 1) and the solutions to the MBRS (Model 2) is stated in Lemma 1.

Lemma 1: If the optimal solution to the MBRS (Model 2) satisfies the maximum vehicle time constraints, it is the optimal solution to the MBRSV (Model 1).

Proof: The MBRSV (Model 1) and the MBRS (Model 2) share the same objective and the same constraints except for the MVT constraints. Thus, the MBRS (Model 2) is a relaxation of the MBRSV (Model 1). For minimization problems, the optimal solution to the MBRS (Model 2) is a lower bound of the MBRSV (Model 1). If this solution satisfies the MVT constraints, it is also a feasible solution to the MBRSV (Model 1). Hence, it is the optimal solution to the MBRSV (Model 1).

Given the Lemma 1, the MBRSV (Model 1) can be simplified as follows. First, solve the MBRS (Model 2). If all bus routes satisfy the MVT constraints, this is the optimal solution to the MBRSV (Model 1). Otherwise, given the routing solution from the MBRS (Model 2), solve a Multi-School Bell Time and Scheduling with the Maximum Vehicle Time Constraints Problem (MBSV, Model 5, in section 3.8) to handle the MVT constraints. A graphic representation of the decomposition scheme is shown in Figure 7.

Now the task is to solve the MBRS (Model 2), which is not easy for large-sized problems. The MBRS (Model 2) is further broken down into single-school routing problems. Because the MBRS (Model 2) requires some information from other schools, it cannot be decomposed directly. Thus, it is transformed into the Multi-School Bus Routing and Scheduling Problem (MSC, Model 3), which is decomposable based on schools. In the process of decomposition, an overestimation of compatibility might occur. To compensate for this inaccuracy, two compensation strategies are applied in the process of decomposition: 1) objective adjustment, and 2) compatibility assignment.

Based on the adoption of compensation strategies, we have three versions of the compatibility-considered single-school bus routing problem (in section 3.7). The detailed relaxation, decomposition, and reformulation are presented in sections 3.5 to 3.8. The algorithm developed to solve the compatibility-considered single-school bus routing problem and the MBSV (Model 5) will be presented in section 4.2 and section 4.3, respectively.

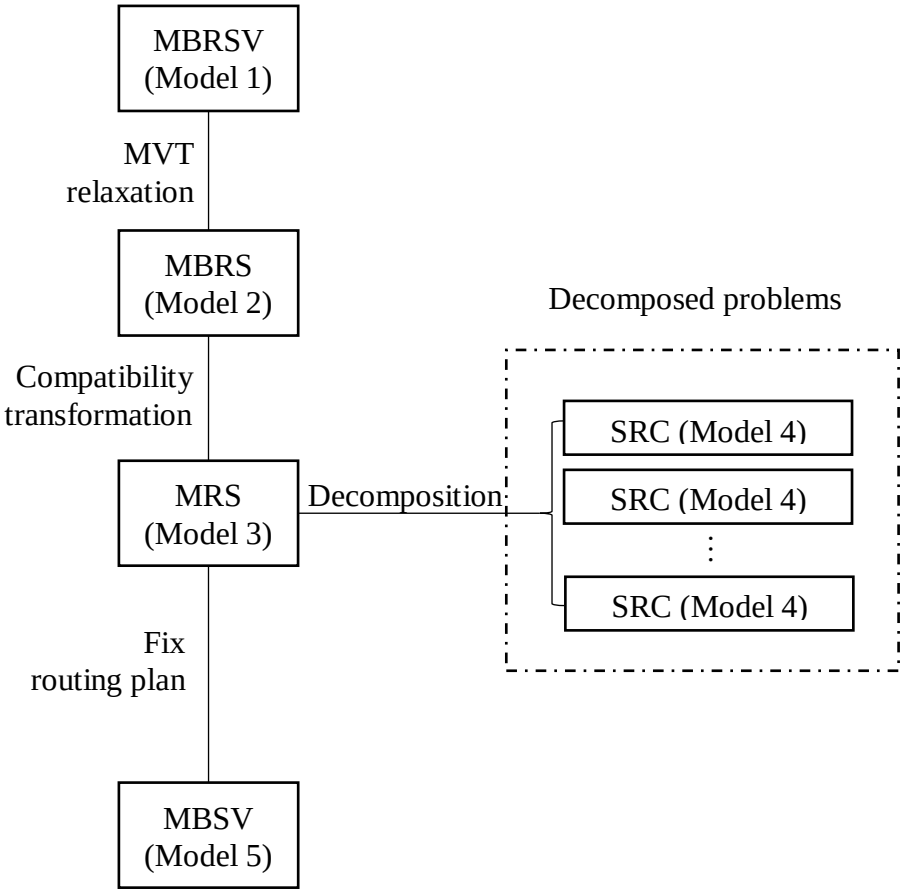


Figure 7 Illustration of the MBRSV decomposition

3.5 The MBRS (Model 2)

3.5.1 The MVT relaxation

The MBRS (Model 2) is the relaxation of the MBRSV (Model 1) after dropping the Maximum Vehicle Time constraints (MVT). This is defined as the MVT relaxation.

The MVT relaxation: Replace the three-index variable ($y^b_{t1,t2}$) by two-index variable ($y_{t1,t2}$) in the objective (3-2) and the constraints (3-15) to (3-20) and (3-30), drop constraints (3-22) and (3-24).

The rationale to drop the Maximum Vehicle Time constraint is that it requires many variables and constraints. The chance of violation this constraint is not extremely high in practice. It is not beneficial to add this constraint to the formulation. Simply by replacing the three-index variable ($y^b_{t1,t2}$) by two-index variable ($y_{t1,t2}$), $\{|\mathrm{TP}|^2 \times (|\mathrm{B}|-1)\}$ number of binary variables can be saved, where $|\mathrm{TP}|$ is the number of potential trips and $|\mathrm{B}|$ is the number of potential buses. Similarly, $\{2 \times |\mathrm{TP}|^2 \times (|\mathrm{B}|-1) + |\mathrm{TP}| \times (|\mathrm{B}|-2) + 0.5 \times |\mathrm{B}|^2 + 2 \times |\mathrm{B}|\}$ number of constraints can be discarded due to the MVT relaxation.

3.5.2 Model formulation

The objective (3-38) is the same as (3-2). It is written in a different form because it is easier to be decomposed into single-school problems. Clearly, the total vehicle time (ride time of the trips and the deadhead) is the same in two objectives. The only question is whether the number of buses can be expressed using the number of trips minus the trip-to-trip compatibility. The proof of equality is presented in Lemma 2. The constraints in the MBRS (Model 2) are similar to the MBRSV (Model 1) except that

the MVT constraints are dropped and that the three-index variable ($y^{b}_{t1,t2}$) is replaced by a two- index variable ($y_{t1,t2}$).

Lemma 2: The first two components in the objective function (3-38) with equal coefficients ($\alpha_N = \alpha_C$) equal to the minimum number of buses.

Proof: If no trips are compatible, the total number of buses equals to the number of trips ($\sum_{t\in TP} y_{SDT,t} = \sum_{k\in SCH} \sum_{t\in TP[k]} ta_t$). When any two trips are compatible and blocked together, a single bus can serve both trips together. Then each realized compatible trip pair ($y_{t1,t2} = 1$) would yield in one bus saving. The total realized compatible trip pairs ($\sum_{t1\in TP} \sum_{t2\in TP\setminus t1} y_{t1,t2}$) is the number of buses that can be saved by blocking. Hence, the first two terms in the objective function (3-38) equals to the number of buses at the best scheduling condition.

The MBRS (Model 2)

$$\min z2 = \alpha_N \sum_{k\in SCH} \sum_{t\in TP[k]} ta_t - \alpha_C \sum_{t1\in TP} \sum_{t2\in TP\setminus t1} y_{t1,t2} + \alpha_T \sum_{t\in TP} tt_t + \alpha_D \sum_{t1\in TP\cup SDT} \sum_{t2\in TP\cup EDT\setminus t1} ddb_{t1,t2} \quad (3\text{-}38)$$

Subject to

Constraints (3-3) to (3-14) (Routing and some scheduling constraints)

Constraints (3-21) (MRT constraints)

Constraints (3-23) (Efficiency constraints)

Constraints (3-25) to (3-29) (Binary domain constraints)

$$sb_{O\{t1\}} + tt_{t1} + dd_{t1,t2} - M \times (1 - y_{t1,t2}) \le sb_{O\{t2\}}, \forall t1, t2 \in TP | t1 \ne t2 \quad (3\text{-}39)$$

$$\sum_{t1\in TP\cup SDT\setminus t} y_{t1,t} = ta_t, \forall t \in TP \quad (3\text{-}40)$$

$$\sum_{t2\in TP\cup EDT\setminus t} y_{t,t2} = ta_t, \forall t \in TP \quad (3\text{-}41)$$

$$2 \times y_{t1,t2} \le ta_{t1} + ta_{t2}, \forall t1 \in TP, t2 \in TP\setminus t1 \quad (3\text{-}42)$$

$$ddb_{t1,t2} \geq M \times (y_{t1,t2} - 1) + dd_{t1,t2}, \forall t1 \in TP \cup \text{SDT}, t2 \in TP \cup \text{EDT} \backslash t1 \quad (3\text{-}43)$$

$$y_{t1,t2} \in \{0,1\}, \forall t1 \in TP \cup \text{SDT}, t2 \in TP \cup \text{EDT} \backslash t1 \quad (3\text{-}44)$$

3.6 The MRS (Model 3)

The MBRS (Model 2), although simpler than the MBRSV (Model 1), might still take some time to solve. Because each school has a set of exclusive bus stops, it is reasonable to decompose the MBRS (Model 2) into single-school problems. However, constraints (3-14), (3-40) to (3-44) in the MBRS (Model 2) need some information from other schools, making them indecomposable by schools. The Compatibility Transformation is designed to transform the MBRS (Model 2) into a new model that can be broken down into many single-school problems. The compatibility transformed model is called the Multi-School Routing and Scheduling Problem (MRS Model 3).

3.6.1 Compatibility transformation

The cross-school information that the MBRS (Model 2) required is just the departure time of the trips. And the departure time of trips equals to the school dismissal time (for PM trips). If the school dismissal times are fixed, the trip-to-trip compatibilities can be replaced by the trip-to-school compatibilities. The trip-to-school compatibility, defined between a trip and a school, equals to one, if the finish time of the preceding trip plus the deadhead from the trip to the successive school is less than or equal to the dismissal time of the successive school.

The compatibility transformation (CF): Fix the school bell time (sb_k), replace the trip-to-trip deadhead and compatibility by the trip-to-school deadhead and compatibility in constraints (3-14), (3-40) to (3-44).

$$dd_{t,k2} = \sum_{s \in SP[k]} D_{s,k2} \times l_{s,t}\,, \forall k \in SCH, t \in TP[k], k2 \in SCH \backslash \mathrm{k} \tag{3-45}$$

$$sb_k + tt_t + dd_{t,k2} - M \times \left(1 - y_{t,k2}\right) \leq sb_{k2}, \quad \forall k \in SCH, t \in TP[k], k2 \in SCH \backslash \mathrm{k} \tag{3-46}$$

$$\sum_{k2 \in SCH \backslash \mathrm{k}} y_{t,k2} \leq ta_t, \forall k \in SCH, t \in TP[k] \tag{3-47}$$

$$ddb_{t,k2} \geq M \times \left(y_{t,k2} - 1\right) + dd_{t,k2}, \forall k \in SCH, t \in TP[k], k2 \in SCH \backslash \mathrm{k} \tag{3-48}$$

$$\sum_{t1 \in TP \backslash TP[k]} y_{t1,k} \leq \sum_{t \in TP[k]} ta_t\,, \forall k \in SCH \tag{3-49}$$

Constraints (3-45) calculate the deadhead from the trips and other schools. Constraints (3-46) identify potential trip-to-school compatibility. After fixing the school bell times, sb_k is a parameter other than a variable. Constraints (3-47) regulate that each trip can have at most one successive school. Similar to constraints (3-20), constraints (3-48) calculate the deadhead from the trips to other schools on each bus for the purpose of linearization. Constraints (3-49) make sure that for every school, the number of preceding trips is less than or equal to the number of active trips that this school needs. Notice that although the trip-to-school variables ($dd_{t,k}, y_{t,k}, ddb_{t,k}$) use the same notation as their trip-to-trip counterparts ($dd_{t1,t2}, y_{t1,t2}, ddb_{t1,t2}$), they are defined in different domains.

3.6.2 Model formulation

The objective (3-50) is the same as (3-38) except replacing the trip-to-trip variables by the trip-to-school variables. So are the constraints.

The MRS (Model 3)

$$\min z3 = \alpha_N \sum_{k \in SCH} \sum_{t \in TP[k]} ta_t - \alpha_C \sum_{k \in SCH} \sum_{t \in TP[k]} \sum_{k2 \in SCH \backslash k} y_{t,k2} + \alpha_T \sum_{k \in SCH} \sum_{t \in TP[k]} tt_t + \alpha_D \sum_{k \in SCH} \sum_{t \in TP[k]} \sum_{k2 \in SCH \backslash k} ddb_{t,k2} \tag{3-50}$$

Subject to

Constraints (3-3) to (3-13) (Routing and some scheduling constraints)

Constraints (3-21) (MRT constraints)

Constraints (3-23) (Efficiency constraints)

Constraints (3-25) to (3-29) (Binary domain constraints)

Constraints (3-31) to (3-34) (Nonnegative continuous domain constraints)

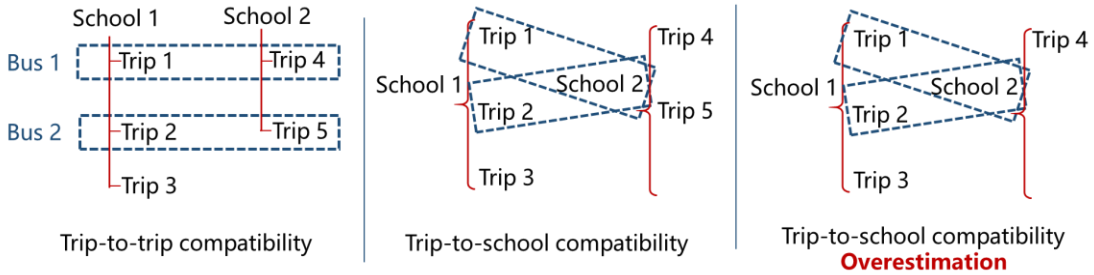
Constraints (3-45) to (3-49) (Compatibility transformed scheduling constraints)

$$y_{t,k2} \in \{0,1\}, \forall k \in SCH, t \in TP[k], k2 \in SCH \backslash k \quad (3\text{-}51)$$

3.6.3 Compatibility overestimation

The MRS (Model 3) can be broken down into many sing-school problems by simply separating the objective and the constraints based on the schools. Constraints (3-49) needs the information on the number of active trips for all other schools, which is unknown before solving the problem. Therefore, it needs to be dropped in the process of decomposition.

Due to the relaxation of constraints (3-49), the solutions from the decomposed single-school models may not be the optimal solution for the MRS (Model 3). An illustration is shown in Figure 8. Assume Trip 1 and Trip 2 are compatible to School 2 (i.e., the finish time of Trip 1 plus the deadhead from Trip 1 to School 2 is less than School 2's dismissal time). For every school, if the number of preceding trips is less than or equal to the number of trips this school actually uses, constraints (3-49) hold and dropping this constraint will not affect the optimality of the solution. This is the scenario in Figure 8(a) and Figure 8(b). The trip-to-trip compatibility and the trip-to-school compatibility are both equal to two.



(a) (b) (c)

Figure 8 Illustration of compatibility overestimation

However, it is possible to assign more preceding trips to a school with fewer trips after dropping constraints (3-49). This is the case in Figure 8(c). Under this circumstance, real trip-to-trip compatibility is one, either Trip 1 or Trip 2 can be assigned as a preceding trip to Trip 4. However, the trip-to-school compatibility is two, which is greater than the real trip-to-trip compatibility. It means that the trip-to-school compatibility in the objective (3-50) is overestimated. One trip-to-school compatibility does not yield one bus saving, which is the core idea of Lemma 2. Thus, Lemma 2 does not hold anymore and that the number of buses cannot be represented by the number of trips minus the trip-to-school compatibility. Similarly, the total deadhead using trip-to-school compatibility in objective (3-50) is also overestimated.

3.7 Decomposed single school routing problems

3.7.1 The SRC (Model 4)

Since the trip-to-school compatibility and total deadhead might be overestimated, a simple approach is to decrease the corresponding weights in the objective function.

The objective adjustment (OA): decrease the weights of trip-to-school compatibility ($\alpha_C^{OA} < \alpha_N$) and total deadhead ($\alpha_D^{OA} < \alpha_D$).

The decomposed single-school problems are called the Single-School Bus Routing Problems with the Compatibility (SRC, Model 4). They are defined for every school ($k \in SCH$). The solution to the MRS (Model 3) is the combination of the solutions to all the SRCs (Model 4). The sequence of solving the SRCs (Model 4) will not affect the final solution.

The SRC (Model 4)

$$\min z4 = \alpha_N \sum_{t\in TP[k]} ta_t - \alpha_C^{OA} \sum_{t\in TP[k]} \sum_{k2\in SCH\setminus k} y_{t,k2} + \alpha_T \sum_{t\in TP[k]} tt_t + \alpha_D^{OA} \sum_{t\in TP[k]} \sum_{k2\in SCH\setminus k} ddb_{t,k2} \quad (3\text{-}52)$$

Subject to

Constraints (3-3) to (3-13) (Routing and some scheduling constraints for school k)

Constraints (3-21) (MRT constraints for school k)

Constraints (3-23) (Efficiency constraints for school k)

Constraints (3-25) to (3-29) (Binary domain constraints for school k)

Constraints (3-31) to (3-34) (Nonnegative continuous domain constraints for school k)

Constraints (3-45) to (3-49) (Compatibility transformed scheduling constraints for school k)

3.7.2 The SRCA (Model 4')

We define compatibility assignment as an alternative to handle the overestimation of the trip-to-school compatibility and the total deadhead. The decomposed single-school problems adopting this idea are referred to as the Single-School Bus Routing Problem with the Compatibility Assignment (SRCA, Model 4').

The compatibility assignment (CA): dynamically update the school Unassigned Trip Capacity (UTC) and only assign trip-to-school compatibility to the schools with a positive UTC.

The unassigned trip capacity (UTC, per school): The number of trips to a school that has not been assigned to a compatible preceding trip.

The underlying reason for overestimating trip-to-school compatibility and the total deadhead is the assignment of more compatible preceding trips to schools with fewer active trips. Compatibility assignment will prevent this from happening with the help of the UTC. When solving each single-school problem (*k*), the SRCA (Model 4') only assigns trips to other schools (*k2*) with a positive UTC[*k2*]. This constraint is expressed as follows:

$$\sum_{t\in TP[k]} y_{t,k2} \leq UTC[k2], \forall k2 \in SCH \backslash k \quad (3\text{-}53)$$

Note that trips can only be compatible with schools with later dismissal times. It is desired to solve the SRCA (Model 4') for the latest dismissed school first. Then, when solving the SRCA (Model 4') for early dismissed schools, we can guarantee that the number of compatible preceding trips is less than or equal to the exact number of active trips to the successive school. In the beginning, the initial UTC is set to be the Minimum Number of Trips (MNT, see section 3.2). Whenever one SRCA (Model 4') (for school k) is solved, the UTC for all schools needs to be updated. The update rules are:

1. For the newly solved school (*k*), the updated UTC for school k is the number of active trips to that school;

$$UTC[k] \leftarrow \sum_{t\in TP[k]} ta_t \quad (3\text{-}54)$$

2. For all other schools with later bell times (*k2*), the updated UTC is the old UTC minus the number of compatible preceding trips (from the newly-solved school)

$$UTC[k2] \leftarrow UTC[k2] - \sum_{t \in TP[k]} y_{t,k2}\,, \forall k2 \in SCH \backslash k \tag{3-55}$$

Note the SRCA (Model 4') is solved from the latest-dismissed school to the earliest-dismissed school. When school k's problem is newly solved, only schools with equal or later bell times are solved. Because the trips from later dismissed schools are incompatible to school k, at this moment, no compatible preceding trip is assigned to school k, and the UTC[k] is equal to the number of active trips of school k. Constraints (3-53) to (3-55) limit the assignment of trips to a school based on the successive school's UTC. In this case, trip-to-school compatibility and total deadhead will not be overestimated.

The SRCA (Model 4')

$$\begin{aligned} \min z4' = {} & \alpha_N \sum_{t \in TP[k]} ta_t - \alpha_C \sum_{t \in TP[k]} \sum_{k2 \in SCH \backslash \mathrm{k}} y_{t,k2} + \\ & \alpha_T \sum_{t \in TP[k]} tt_t + \alpha_D^{OA} \sum_{t \in TP[k]} \sum_{k2 \in SCH \backslash \mathrm{k}} ddb_{t,k2} \end{aligned} \tag{3-56}$$

Subject to

Constraints (3-3) to (3-13) (Routing and some scheduling constraints for school k)

Constraints (3-21) (MRT constraints for school k)

Constraints (3-23) (Efficiency constraints for school k)

Constraints (3-25) to (3-29) (Binary domain constraints for school k)

Constraints (3-31) to (3-34) (Nonnegative continuous domain constraints for school k)

Constraints (3-45) to (3-49) (Compatibility transformed scheduling constraints for school k)

Constraints (3-53) (Compatibility assignment constraints for school k)

3.7.3 The SRCW (Model 4")

The SRCA (Model 4') prevents the assignment of more preceding trips to a school with fewer trips. Still, it does not guarantee to output an optimal solution. One example is shown in Figure 9.

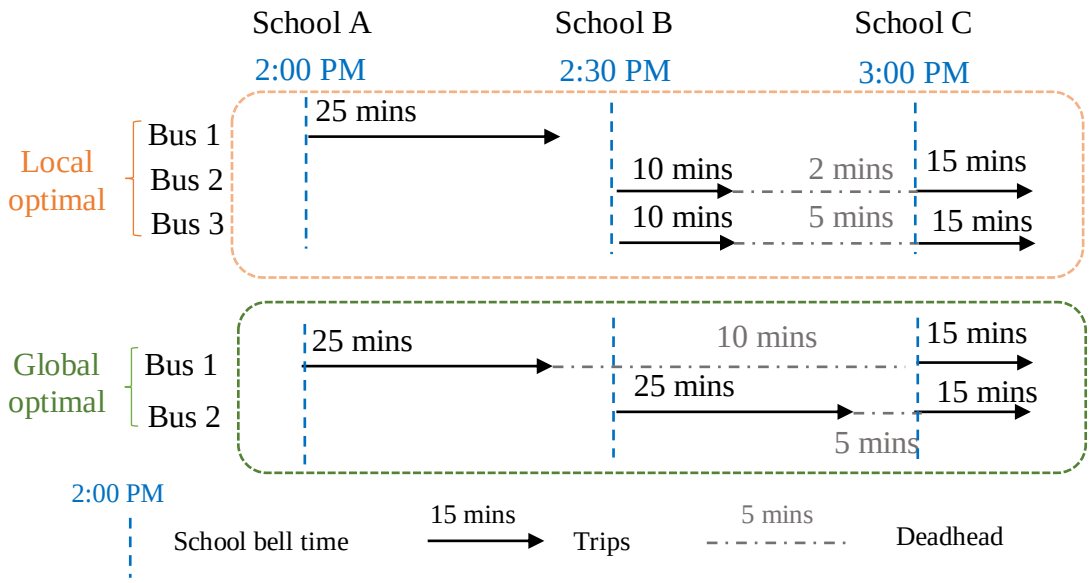


Figure 9 Illustration of the SRCA (Model 4')

Suppose we are trying to solve the SRCA (Model 4') for school B. The optimal solution for school C has already been obtained with two trips. By solving the SRCA (Model 4'), the optimal solution for school B is two short trips where both trips are assigned as preceding trips to school C (number of trips and trip-to-school compatibility both equal to 2, with the total vehicle time of 27 minutes). In the end, when school A is solved with one trip, there is no UTC left (assume the trip from school A is not compatible with school B), and three buses are needed. However, a globally optimal solution can be obtained if we have one long trip for school B. This is not the optimal solution to the SRCA (Model 4') for school B because the objective value of this long trip solution is worse than the two short trips solution (number of trips and

trip-to-school compatibility both equal to 1 with total vehicle time of 30 minutes). However, this solution will yield the best scheduling plan with only two buses.

The underlying reason causing the problem is that once a trip is assigned to a compatible successive school, there is no cost of adding such trip apart from its ride time and the deadhead. However, the trip-to-school compatibility assignment would consume a UTC, which might be used later. A simple solution is to add a small cost of adding a trip even if it has a compatible successive school. To accomplish this goal, the weights of all components in the objective function need to be carefully tuned such that each component has a different priority in the objective. A proposal for the weights is $\alpha_B = \alpha_T = \alpha_C > \alpha_C^{CA} > \alpha_C^{OA} \gg \alpha_T > \alpha_D$ where α_C^{CA} is the adjusted weights for trip-to-school compatibly in the SRCA (Model 4'). In this case, the objectives of the SRCA (Model 4') in the descending order of importance are:

- minimizing the number of trips with no compatible schools (with weight α_T);
- minimizing the number of trips with compatible school (with weight $\alpha_T - \alpha_C^{CA}$);
- minimizing the weighted sum of the ride time and deadhead (with weight α_T, α_D).

A potential drawback for this structure is when the school with a later dismissal time has enough active trips, the total vehicle time might increase to reduce the number of trips. However, even under such circumstances, the number of buses will not increase.

We denote the decomposed Single-School Bus Routing Problem with the Compatibility Assignment and the Weight Adjustment as the SRCW (Model 4"). Note

the SRCW (Model 4") and the SRC (Model 4) both need the weights adjustment. But the purpose is different. The weight adjustment in the SRC (Model 4) is due to the overestimation of trip-to-school compatibility. But the weight adjustment in the SRCW (Model 4") is to add a small cost to the trip even if it can be assigned to a compatible successive school. The small cost for such a trip is $\alpha_N - \alpha_C^{CA}$.

3.8 The MBSV (Model 5)

A few parameters used in the MBSV (Model 5) is summarized in the Table 4. Other notations used above are kept as well.

Table 4 Parameters for the MBSV (Model 5)

Parameters	
Name	**Description**
$dd_{t1,t2}$	The deadhead duration from the last stop of trip $t1$ to the first stop of trip $t2$.
ATP	Set of all active trips
E	Set of possible compatible trip pairs

After solving the MRS (Model 3) (either by the MRS itself or by a set of decomposed single-school routing problems), the routing plan and the trips are generated. The trips are treated as the input for the Multi-School Bell Time and Scheduling with the Maximum Vehicle Time Constraints Problem (MSBV, Model 5). This would yield the final school bus transportation plan with all level-of-service constraints.

Because the routing plan is given, the sequence of visiting stop on each trip ($x_{s1,s2}^t$), the first and last stop on each trip ($l_{s,t}$) and the ride time of each trip (tt_t) are known. Note that in the MBRSV (Model 1), the MBRS (Model 2), the MRS (Model 3), and the SRC (Model 4), the trip set is unknown. They all used a potential trip set

TP. But in the MSBV (Model 5), the trip set is known so that only the Active Trip Set (*ATP*) needs to be considered. The ATP consists of the trips in *TP* that are in use.

Also, the deadhead ($\mathrm{dd}_{\mathrm{t1,t2}}$) between every trip pair (*t1*, *t2*) becomes a parameter. To save notation and improve the efficiency, the possible compatible trip pairs set ***E*** is defined in the preprocessing step. A trip pair (*t1*, *t2*) exists in ***E*** if it satisfies: 1) *t1* and *t2* are both in the ATP or the start depot trip (for *t1*), and end depot trip (for *t2*); 2) *t1* and *t2* are not the same trip nor the depot trip simultaneously; 3) trip *t1* to *t2* is potentially compatible. The trip pair (*t1*, *t2*) is potentially compatible if the earliest start of trip *t1* pluses the ride time of trip *t1* and the deadhead time between trip *t1* and *t2* is less than or equal to the latest start time of trip *t2*.

To consider the MVT constraints, the trip-to-bus ($\mathrm{z}_{\mathrm{t}}^{\mathrm{b}}$) assignment is necessary. Hence, the three-index trip variable ($y_{t1,t2}^{b}$) is used. Because the ride time for the trips (with students on board) is fixed, the objective is reduced to minimizing the number of buses and the total deadhead.

The MBSV (Model 5)

$$\min z5 = \alpha_B \sum_{b\in B} \sum_{t\in ATP} y_{SDT,t}^{b} + \alpha_D \sum_{b\in B} \sum_{(t1,t2)\in E} dd_{t1,t2} \times y_{t1,t2}^{b} \quad (3\text{-}57)$$

Subject to

Constraints (3-24) - (3-25) (school bell time window and efficiency constraints)

$$sb_{O\{t1\}} + tt_{t1} + dd_{t1,t2} - M \times \left(1 - \sum_{b\in B} y_{t1,t2}^{b}\right) \le sb_{O\{t2\}}, \forall (t1,t2) \in E \quad (3\text{-}58)$$

$$\sum_{b\in B} \sum_{t1:(t1,t)\in E} y_{t1,t}^{b} = 1, \forall t \in ATP \quad (3\text{-}59)$$

$$\sum_{t1:(t1,t)\in E} y_{t1,t}^{b} = \sum_{t2:(t,t2)\in E} y_{t,t2}^{b}, \forall b \in B, t \in ATP \quad (3\text{-}60)$$

$$\sum_{t2:(t,t2)\in E} y_{SDT,t2}^{b} \le 1, \forall b \in B \quad (3\text{-}61)$$

$$\sum_{t\in ATP} tt_t \times \left(\sum_{t1:(t1,t)\in E} y_{t1,t}^{b}\right) + \sum_{(t1,t2)\in E} dd_{t1,t2} \times y_{t1,t2}^{b} \le MVT, \forall b \in B \quad (3\text{-}62)$$

$$y_{t1,t2}^{b} \in \{0,1\}, \forall b \in B, (t1, t2) \in E \quad (3\text{-}63)$$

3.9 Chapter summary

In this chapter, the Multi-School Bell time, Bus Routing and Scheduling with the Maximum Vehicle Time Constraints Problem (MBRSV, Model 1) was proposed. The problem is very complicated, and it is nearly impossible to solve to optimality for medium- or large-sized problems. An innovative relaxation and decomposition method was proposed to break down the problem into smaller and simpler subproblems. The Multi-School Bell time, Bus Routing and Scheduling Problem (MBRS, Model 2) can be obtained by dropping the MVT constraints. This problem is further transformed into the Multi-School Routing and Scheduling Problem (MRS, Model 3), which is suitable for the decomposition.

Three decomposed compatibility-considered single-school bus routing problems (the SRC, Model 4; the SRCA, Model 4'; and the SRCW, Model 4") were presented using different mitigation strategies to account for the compatibility overestimation in the decomposition process. In the end, the violation of the MVT constraints is managed by the Multi-School Bell time and Scheduling with the Maximum Vehicle Time Constraints Problem (MBSV, Model 5) using the trips generated by solving the MRS (Model 3) or a set of decomposed compatibility-considered single-school bus routing problems.

Chapter 4 Deterministic School Bell Time, Bus Routing and Scheduling Optimization Algorithms

The integrated model MBRSV (Model 1) can solve the school bell time, bus routing and scheduling problem simultaneously. But the complexity of this model makes it nearly impossible to be solved to optimality directly. The Decomposition-Based Multi-School Bell Time, Bus Routing and Scheduling Algorithm (DMBRS, Algorithm 1) is the decomposition framework to solve the MBRSV (Model 1). There are three versions of Algorithm 1 given the different decomposed problems it uses (the SRC, Model 4; the SRCA, Model 4'; and the SRCW, Model 4"). In section 4.2, an insertion-based heuristic algorithm (Algorithm 2) is presented to solve the decomposed SRC (Model 4). It consists of the trip generation and post improvement. Then, Algorithm 3 is proposed to solve the MBSV (Model 5) using a dynamic update strategy. By following these steps, the MBRSV (Model 1) can be solved.

4.1 Algorithm 1 to solve the MBRSV (Model 1)

Algorithm 1 decomposes the MBRSV (Model 1) into many subproblems, which is the same argument in section 3.4. Based on the choice of solving the SRC (Model 4), the SRCA (Model 4') and the SRCW (Model 4"), Algorithm 1 has three versions: Algorithm 1-1 (solving the SRC, Model 4) Algorithm 1-2 (solving the SRCA, Model 4') and Algorithm 1-3 (solving the SRCW, Model 4"). Note that Algorithm 1-2 and Algorithm 1-3 are similar except that Algorithm 1-2 solves the SRCA (Model 4'), which does not adjust the coefficients of the trip-to-school compatibility while

Algorithm 1-3 solves the SRCW (Model 4"), which adjusts the coefficients of the trip-to-school compatibility. The pseudo-code of Algorithm 1-1, 1-2 and 1-3 are as follows:

Algorithm 1-1

1. Implement the MVT relaxation on the MBRSV (Model 1) to obtain the MBRS (Model 2)
2. Implement compatibility transformation on the MBRS (Model 2) to obtain the MRS (Model 3)
3. Drop constraints (3-49), implement objective adjustment and decompose the MRS (Model 3) to obtain a set of SRCs (Model 4)
4. Initialize trip set = Φ
5. For k in SCH
   5.1 Solve the SRC (Model 4) for school k
   5.2 Augment trip set with the trips for school k
6. Solve the MBSV (Model 5) given the trip set

Algorithm 1-2

1. Implement the MVT relaxation on the MBRSV (Model 1) to obtain the MBRS (Model 2)
2. Implement compatibility transformation on the MBRS (Model 2) to obtain the MRS (Model 3)
3. Drop constraints (3-49), implement compatibility assignment and decompose the MRS (Model 3) to obtain a set of SRCAs (Model 4)
4. Sort the school solving sequence with the descending school bell times
5. Initialize trip set = Φ and initialize the UTC as the MNT
6. For k in sorted school solving sequence:
   6.1 Solve the SRCA (Model 4') for school k
   6.2 Augment trip set with the trips for school k, Update the UTC for all schools
7. Solve the MBSV (Model 5) given the trip set

The MBRSV (Model 1) is first relaxed to obtain the MBRS (Model 2) by relaxing the MVT constraints. Then the MBRS (Model 2) is transformed into the MRS (Model 3) by compatibility transformation. The MRS (Model 3) is decomposed into many single-school problems after dropping constraints (3-49). To compensate for the inaccuracy of the decomposition, the SRC (Model 4), the SRCA (Model 4') and the SRCW (Model 4") adopt different strategies. A set of single-school problems (the SRC, Model 4; the SRCA, Model 4'; or the SRCW, Model 4") are solved. The routing plan

to the MRS (Model 3) is the combination of the solutions to the single-school problems (the SRC, Model 4; the SRCA, Model 4'; or the SRCW, Model 4"). In the end, the MBSV (Model 5) is solved using the generated routing plan from the decomposed single-school bus routing problems.

Algorithm 1-3

1. Implement the MVT relaxation on the MBRSV (Model 1) to obtain the MBRS (Model 2)
2. Implement compatibility transformation on the MBRS (Model 2) to obtain the MRS (Model 3)
3. Drop constraints (3-49), implement compatibility assignment, coefficient adjustment and decompose the MRS (Model 3) to obtain a set of SRCWs (Model 4")
4. Sort the school solving sequence with the descending school bell times
5. Initialize trip set = Φ and initialize the UTC as the MNT
6. For k in sorted school solving sequence:
   6.1 Solve the SRCW (Model 4") for school k
   6.2 Augment trip set with the trips for school k, Update the UTC for all schools
7. Solve the MBSV (Model 5) given the trip set

The difference between Algorithm 1-1, Algorithm 1-2, and Algorithm 1-3 comes from different compensation strategies. Algorithm 1-1 does not need to assign trips to schools based on the number of trips that the school actually uses. There is no need to update the Unassigned Trip Capacity (UTC). However, choosing an adjusted weight could be challenging. Algorithm 1-2, on the other hand, assigns trips to school based on the UTC. The UTC needs to be dynamically updated when an SRCA (Model 4') is solved. Algorithm 1-3, which considers the UTC and weight adjustment, is the most complicated one. Algorithm 1-2 and Algorithm 1-3 need to solve the schools with the descending order of the dismissal times. The sequence in which the SRCs (Model 4) are solved will not affect the final solution in Algorithm 1-1.

4.2 Algorithm 2 to solve the SRC (Model 4)

The SRCs (Model 4) can be solved to optimality or with a relatively low optimal gap using a commercial solver for a medium-sized problem. Based on the experiments in Chapter 6, it would take more than one hour to solve an SRC (Model 4) with 30 stops on a regular desktop computer. In practice, one school could have many more stops. In these larger cases, simply using commercial solver cannot find a good or even a feasible solution to the SRC (Model 4). Therefore, a fast heuristic algorithm is needed to solve the SRC (Model 4). The heuristic algorithm proposed is called the Minimum Cost Matching with the Post Improvement Algorithm (MCMPI, Algorithm 2). It consists of two parts: 1) the Minimum Cost Matching algorithm (MCM, Algorithm 2-1), which generates a good feasible routing plan; and 2) the Post Improvement algorithm (PI, Algorithm 2-2), which combines the Tabu Search, and the Simulated Annealing to improve the initial solution. Note that the MCMPI does not involve the dynamic trip-to-school compatibility assignment, this heuristic algorithm can only be used to solve the SRC (Model 4) not the SRCA (Model 4') or the SRCW (Model 4").

4.2.1 Notations for Algorithm 2

The notation used in Algorithm 2 is presented in Table 5. Other notations from the model formulation are inherited. Algorithm 2 solves the SRCs (Model 4) one at a time. Thus, all the sets (trips set, stops set, etc.) and parameters (number of stops, number of trips, etc.) are restricted to the current school (k) for which the problem is being solved. The only exception is the school set (the number of schools and the school bell/dismissal times). The school bell/dismissal times are assumed as known in all SRCs (Model 4).

Table 5 Notation summary for Algorithm 2

Name	Description
MCM	
$IC_{s,t}$	Insertion cost of inserting stop *s* into trip *t*
$MQ_{s,t}$	The remaining vehicle capacity of trip *t* after inserting stop *s*
$MC_{s,t}$	The compatibility criterion of trip *t* after inserting stop *s*
$MT_{s,t}$	The augmented ride time of trip *t* after inserting stop *s*
$SP\{t\}$	The stops that form the trip *t*
$IX(s1, s2)$	Indicator function, which equals 1 if stop *s2* is visited right after *s1* on one trip, it is updated after the matching problems are solved.
$m_{s,t}$	Binary variable, equals to 1 if trip *s* is selected to be inserted into trip *t*
R	Set of un-routed (un-assigned) stops
$\beta_Q, \beta_C, \beta_T$	The coefficient of the remaining vehicle capacity (β_Q), the compatibility (β_C), and the total ride time (β_T) in insertion cost
$LS_{s,t}$	Last stop in trip *t* after inserting stop *s*
PI	
temp	Temperature
$temp_{end}$	Terminal temperature
$temp_{cool}$	Temperature cooling rate
iter	Number of iteration that no changes occur
$iter_{max}$	Maximum number of iterations that no changes occur
TL	The tabu list
CR_s	The cost of removal of stop *s*
IS_s	Number of stops in the trip to which stop *s* belongs
IT_s	The travel time of the trip to which stop *s* belongs
$IP_{p,s}$	The closeness of stop *p* with respect to stop *s*
IQ_s	Number of students for the trip to which stop *s* belongs
$SP(s)$	Stops building the trip to which stop s belongs
ϵ	Random variables
$\gamma_s, \gamma_T, \gamma_Q, \gamma_D$	The coefficient of the number of stops (γ_s), the travel time (γ_T), the number of students (γ_Q) in 'cost of remove' and the distance (γ_D) in 'closeness'
EL	Vertex exchange list
NL	Neighborhood list
Nnei	Maximum number of stops in the neighborhood list
move(p,s)	Move stop *p* to the place right after stop *pl* where p and pl belongs to different trips
SC	Surrogate cost of a routing plan
TN, TC, TT	Number of trips (*TN*), the trip compatibility (*TC*) and the total travel time (*TT*) in a routing plan
$l\{t\}$	The last stop of trip *t*
$\delta_N, \delta_C, \delta_T$	The coefficient of the number of trips (δ_N), the trip compatibility (δ_C), and the total travel time (δ_T) in surrogate cost

4.2.2 Algorithm 2-1: minimum cost matching algorithm

The MCM algorithm is an insertion-based trip generation algorithm where the stop insertion decisions are made by solving the minimum cost matching problems. The inputs for each SRC (Model 4) problem are the stop locations, the number of students at each stop and the school bell/dismissal times for all schools.

The MCM trip generation can be done either sequentially or in parallel. The pseudo-code for the MCM is shown in Algorithm 2. A graphical example of the MCM in illustrated in Figure 10.

Algorithm 2-1 MCM

1. Initialize the predefined number of trips (PNT)
2. Cluster stops into PNT clusters [by k-means algorithm]
3. Find the 'furthest' stop to the school in each cluster
4. Initialize the trip set (TP[k]) and the un-routed (unassigned) stop set (R)
5. **While** $R \neq \emptyset$ **do**
   - 5.1 Construct the insertion cost matrix
   - 5.2 Solve the minimum cost matching problem
   - 5.3 **If** (feasible insertion exists) **then**
     - A. Implement the insertion, update TP[k], R

   Else:
     - B. Initialize a new trip using the furthest stop in R, update TP[k], R

4.2.2.1 Initialization

The initialization refers to steps 1 – 4 of the MCM algorithm. The MCM algorithm starts with a Predefined Number of Trips (PNT). Different ways to initialize this predefined number of trips would affect the MCM. Two strategies are adopted in this dissertation. One way is to initialize PNT to one, making it the Sequential Minimum Cost Matching (SMCM) trip generation algorithm. An alternative is to initialize PNT to the Minimum Number of Trips (MNT) that is required to transport students based on the bus capacity (see section 3.2). In this way, multiple trips will be

initialized simultaneously, making it the Parallel Minimum Cost Matching (PMCM) trip generation algorithm. Both the PMCM and the SMCM initialize (PM) trip sets by having the school location as the first stop and adding a stop that is the furthest away from the school location. Since the trips are gradually generated, the trip set (TP[k]) is the actual trip set for school k rather than the potential trip set as used in Chapter 3.

In the PMCM, several trips are initialized together. There might be a chance that two close stops are initialized to two trips because they are far away from the school. A Cluster First, Initial Second (CFIS) scheme is adopted to prevent this from happening. The stops are first clustered into PNT number of clusters. Then, the 'furthest' stops in each cluster are used to initialize the trip set. This CFIS helps to scatter the initialized trips. Various clustering methods could be used. The widely used k-means algorithm (Pedregosa et al., 2011) is used in this dissertation where k equals to the PNT.

4.2.2.2 Trip construction and stop assignment

This subsection refers to step 5 of the MCM algorithm. Once the trips are initialized, the remaining task is to insert the remaining (un-routed) stops into the trips. To achieve this goal, the minimum cost perfect matching problems (Model 6) are solved iteratively until all stops are inserted. During each matching round, the MCM either inserts some un-routed stop into the trips or add a new trip. Which case happens depends on whether there exists a feasible insertion.

An illustration of a hypothetical bi-partite graph for the matching problem is shown in Figure 10. In this example, there are five remaining (un-routed) stops and two trips. There are $5 \times 2 = 10$ possible assignment (insertion) pairs in total.

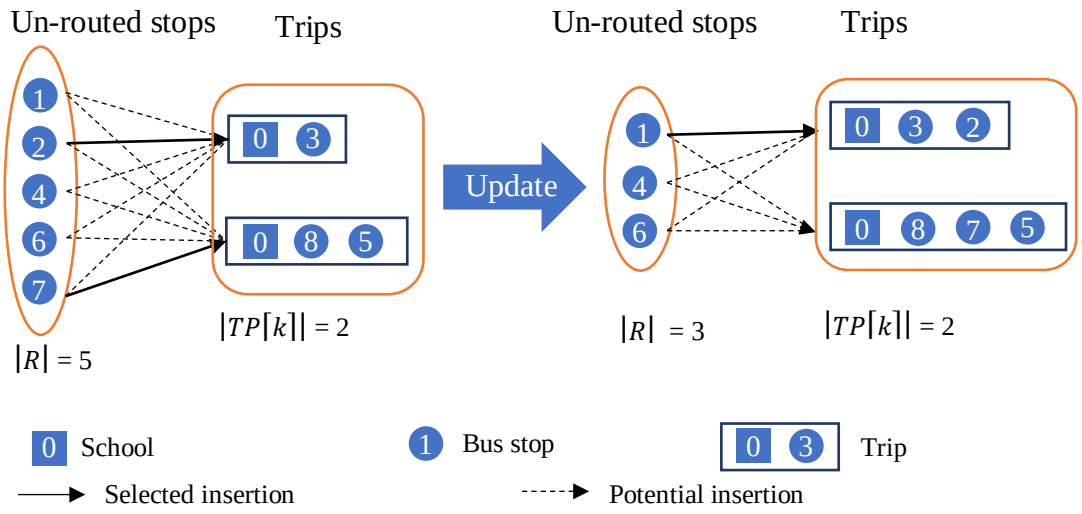


Figure 10 Illustration of the bipartite graph for matching problem

After solving the matching problem, two pairs are selected (inserting stop 2 into trip 1 and inserting stop 7 into trip 2) because the total insertion cost of these two pairs is the minimum among all other combinations. After updating the insertion pairs, three remaining stops and two trips are left. Since only one insertion is feasible (inserting stop 1 into trip 1), only this insertion will be selected and updated in the next round. The algorithm will continue with two remaining stops and two trips. The costs/weights needed to solve the matching can be stored in a matrix called the insertion cost matrix. Section 4.2.2.3 explains how to calculate the insertion cost. Section 4.2.2.4 explains how to find the best matching by solving a matching problem under the satisfaction of the feasibility constraint. An example is presented in 4.2.2.5.

4.2.2.3 Constructing the insertion cost matrix.

A core task in the minimum cost matching is constructing the $|R| \times |TP[k]|$ insertion cost matrix, which evaluates the cost of inserting an un-routed stop s into every trip $t \in TP[k]$. This construction is step 5-1 of the MCM algorithm. For the PMCM, the insertion cost is a $|R| \times |TP[k]|$ matrix while in the SMCM, it is a $|R| \times 1$

vector. The Insertion Cost (*IC*) (shown in equation 4-1) depends on factors such as the augmented ride time, the student load (or remaining bus capacity) and compatibility.

$$IC_{s,t} = \beta_Q \times MQ_{s,t} + \beta_C \times MC_{s,t} + \beta_T \times MT_{s,t} \; \forall s \in R, t \in TP[k] \tag{4-1}$$

$$MQ_{s,t} = Cap - \sum_{s1 \in SP\{t\} \cup s} STU_{s1} \; \forall s \in R, t \in TP[k] \tag{4-2}$$

$$MT_{s,t} = \sum_{s1 \in SP\{t\} \cup s} \sum_{s2 \in SP\{t\} \cup s \setminus s1} IX(s1, s2) \times D_{s1,s2} + PT\left(\sum_{s1 \in SP\{t\} \cup s} STU_{s1}\right) + \sum_{s1 \in SP\{t\} \cup s} DT(STU_{s1}), \forall s \in R, t \in TP[k] \tag{4-3}$$

$$MC_{s,t} = |SCH| - \sum_{k2 \in SCH \setminus k} 1\{sb_k + MT_{s,t} + D_{LS_{s,t},k2} \leq sb_{k2}\} \quad \forall s \in R, t \in TP[k] \tag{4-4}$$

The insertion cost between a stop and a trip is a weighted sum of the remaining bus capacity, the trip-to-school compatibility and the total ride time. The remaining bus capacity defined in (4-2) is the bus capacity minus the sum of students at the stops in trip *t* and stop *s*. The intuition is that it is preferred to assign a stop to a trip if this assignment helps in filling up the trip (bus). The total ride time (4-3) is the sum of travel time between the stops and the school and the pickup and drop-off times. Obviously, short trips can save total ride time and are easier to contribute to compatibility, which can further help to save buses.

To calculate the trip-to-school compatibility criterion ($MC_{s,t}$), an indicator function ($1\{Condition\}$) is used in constraint (4-4). It equals to one if the *Condition* is satisfied and zero otherwise. In constraint (4-4), if the condition is satisfied $\left(sb_k + MT_{s,t} + D_{LS_{s,t},k2} \leq sb_{k2}\right)$, then the trip-to-school compatibility indicator $1\{sb_k + MT_{s,t} + D_{LS_{s,t},k2} \leq sb_{k2}\}$ is evaluated to 1, which means that the augmented trip *t*, after inserting a remaining stop *s*, is still compatible to school *k2*. The MCM is a minimization problem but the trip-to-school compatibility is preferred to be as large as possible. Therefore, the trip-to-school compatibility criterion ($MC_{s,t}$) is the total

number of schools ($|SCH|$) minus the sum of the trip-to-school compatibility indicators $1\{sb_k + MT_{s,t} + D_{LS_{s,t},k2} \leq sb_{k2}\}$. The stops are inserted in the position where the ride time of the augmented trip is minimized.

Some constraints like the conservation of flow and the sub-tour eliminations are not listed because they are implicitly guaranteed when the trips are generated under the insertion-based procedure. The maximum ride time and capacity constraints are guaranteed by adding a huge violation penalty to the weight if the insertion of that stop would cause any violation.

4.2.2.4 Solving the minimum cost matching problem and maintaining feasibility

The formulation of the minimum cost perfect matching (MCPM) problem is shown below. For the cases when $|R| \neq |TP[K]|$, dummy stops or trips are added to make it a perfect matching problem.

The MCPM (Model 6)

$$\text{Min z6} = \sum_{s \in R} \sum_{t \in TP[k]} IC_{s,t} \times m_{s,t} \quad (4\text{-}5)$$

Subject to

$$\sum_{s \in R} m_{s,t} = 1, \forall t \in TP[k] \quad (4\text{-}6)$$

$$\sum_{t \in TP[k]} m_{s,t} = 1, \forall s \in R \quad (4\text{-}7)$$

$$m_{s,t} \in \{0,1\}, \forall s \in R, t \in TP[k] \quad (4\text{-}8)$$

Objective (4-5) minimizes the total sum of the insertion cost. Constraints (4-6) and (4-7) enforce that one stop is inserted into exactly one trip and that each trip has exactly one stop to be inserted. If an insertion pair involves the dummy stops or dummy trips, it will be discarded. The Hungarian (Munkres) algorithm is used to solve the minimum cost matching problem with a time complexity of $O(n^3)$. For the SMCM, the

problem is to find the stop that has the minimum insertion cost for the current trip. If these insertions are feasible, the stops will be inserted into their corresponding trips. The trip set will be updated by these augmented trips, and these inserted stops will be removed from the un-routed stop set. This is step 5.2 in the MCM algorithm.

The feasibility check (step 5.3 in the MCM algorithm) consists of the vehicle capacity and the maximum ride time constraints. An insertion is feasible only if both constraints are satisfied. At the point when no feasible insertions exist, a new trip is initialized by connecting the school to the furthest stop in the un-routed stop set. In the PMCM, if all $\min(|R|, |TP[K]|)$ insertions are feasible, these stops will be inserted simultaneously at one iteration in comparison to one stop-insertion for the SMCM. Theoretically, the PMCM is faster than the SMCM.

4.2.2.5 Example

A simple graphical example of the PMCM with one school and 11 stops is shown in Figure 11. Assume the Predefined Number of Trips (PNT) is 3. In Figure 11(a), stops are grouped into three clusters using the k-means method. Then, the furthest stops are selected for each cluster (Figure 11(b)). The trip set (TP[k]) is initialized (Figure 11(c)). In Figure 11(d), a minimum cost matching problem is solved. Three feasible insertions are obtained, which are implemented in Figure 11(e). In Figure 11(f), no feasible insertion exists, and a new trip (Trip 4) is initialized. Then, the algorithm continues until all stops are inserted into the trips.

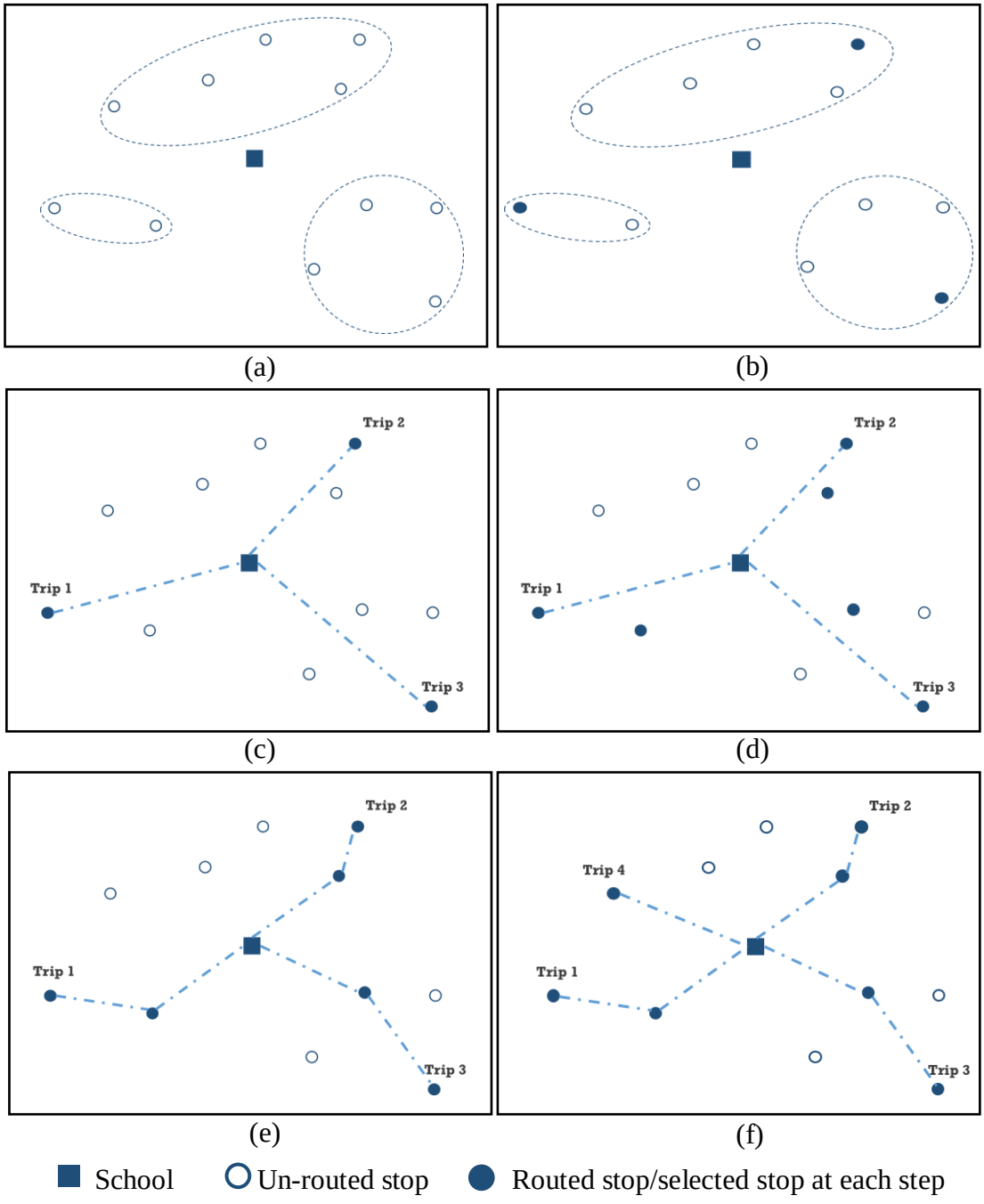


Figure 11 Graphical example of the minimum cost matching trip generation algorithm

4.2.3 Algorithm 2-2: post improvement algorithm

The Post Improvement (PI) algorithm is applied to improve the initial solution generated by the Minimum Cost Matching (MCM) trip generation algorithm. It is a

combination of the Tabu Search, the Simulated Annealing, and the chain exchange algorithm. The pseudo code for the PI algorithm is presented in Algorithm 2-2.

Algorithm 2-2 PI

1. Solve the SBS using the initial routing plan, store as the initial scheduling plan (ISP)
2. Initialize *t*, *it*, N, TL
3. **while** ($t \geq t_{end}$ and $it \leq it_{max}$) **do**
   - 3.1 construct vertex exchange list (EL)
   - 3.2 **for** (p in EL) **do**
     - 3.2.1 construct neighborhood list (NL[p]) of stop p
     - 3.2.2 **for** (pl in NL[p]) **do**
       - A. **If** (*move(pl,p)* is feasible) **then**
         - B. **If** (*move(p,pl)* is not in the Tabu list) **then**
           - C. **If** (*move(pl,p)* is acceptable at *t*) **then**
             - D. Implement *move(pl,p)*, update N, TL, *it* = 0, **Break**;
   - 3.3 *it* += 1; *t* = *t** *t_cool*
4. Solve the SBS using the new routing plan, store as new scheduling plan (NSP)
5. **If** (Number of buses of NSP < number of buses of ISP) **then**
   - 5.1 **Return** NSP

   Else
   - 5.2 **Return** ISP

4.2.3.1 Vertex exchange list

The input trip set *TP[k]* (for school k) is the output from the MCM. Two iterative count variables are used to track the iterations: 1) the temperature *(temp)* and 2) the number of iterations *(iter*) at which no changes occur. The idea of temperature *(temp)* comes from the Simulated Annealing, which is used to decide whether a new solution is acceptable or not. When the temperature is higher, there is a high probability of accepting the moves that worsen the objective function. However, when the temperature cools down, it is very rare to accept such moves but instead, focusing on the moves that improve the objective. The vertex (chain) exchange operator (step 2 in Algorithm 2-2) will be activated only if both of the following conditions are met: 1) the temperature *(temp)* is greater than or equal to the terminal temperature ($temp_{end}$) and 2) the number of iterations with no changes *(iter*) is less than or equal to $iter_{max}$.

The underlying mechanism for the vertex exchange operator is that when moving some stops from short trips to long trips, some short trips will become empty and then are deleted. As a result, the trip set (TP[k]) will be reduced, and the bus scheduling problem is more likely to come up with a solution with fewer buses. More accurately, other attributes like the travel time and the trip-to-school compatibility also contribute to the 'goodness' of a routing plan. The complete surrogate cost to evaluate the 'goodness' of a move will be discussed in section 4.2.3.3. The vertex exchange operator starts with constructing the vertex exchange list based on their 'cost of removal' as defined in (4-9):

$$CR_s = \gamma_s \times IS_s + \gamma_T \times IT_s + \epsilon \quad (4\text{-}9)$$

where IS_s, IT_s are the number of stops and travel time of the trip to which stop *s* belongs, and γ_s, γ_T are the associated coefficients. The vertex exchange list is the set of all stops arranged in the increasing order of their 'cost of removal'.

The 'cost of removal' of a stop describes the negative likelihood of reaching a better routing plan after moving this stop to another trip. For instance, if a trip *t* only contains one stop *s*, this stop will have a low 'cost of removal'. It means that if stop *s* is moved to another trip, trip *t* becomes empty and we have a higher chance to find a better routing and scheduling solution. Generally, the underlying mechanism for estimating the 'cost of removal' is that it is desired to move the stops belonging to shorter trips (with respect to the fewer number of stops and smaller travel time) to other trips. After removing these stops, the short trips are more likely to be deleted. In this way, the total number of trips is more likely to be reduced, and the number of buses in the scheduling stage is more likely to be reduced.

If a trip *t* has two stops *s1* and *s2*, these two stops will have the same 'cost of removal' because they belong to the same trip (*t*). Thus, ϵ is used to introduce some randomness to help distinguish these stops that belong to the same trip.

The travel time is also considered because the higher the travel time is, the less likely that we can find a feasible re-insertion due to the maximum ride time constraint. The travel time is calculated in the same way as in (4-3), including the pickup and drop-off time. The PI algorithm sequentially goes through the vertex exchange list and tries to move these stops and insert them to their best reconnection positions.

4.2.3.2 Neighborhood list

For each stop *s* in the vertex exchange list (*EL*), its neighborhood list (*NL[s]*) contains the stops that are close to stop *s* such that stop *s* has a higher probability to be inserted after them. The stops in neighborhood list are sorted in the ascending order of the 'closeness' (*IP*) defined in (4-10):

$$IP_{p,s} = \gamma_Q \times IQ_p + \gamma_T \times IT_p + \gamma_D \times D_{p,s} \; \forall p \in SP \backslash SP(s) \tag{4-10}$$

where $IQ_p, IT_p, D_{p,s}$ are the number of students, the travel time of the trip to which stop p belongs, and the distance between stop *s* and *p*, and $\gamma_Q, \gamma_D, \gamma_T$ are the associated coefficients. Equation (4-10) states that stop *s* has a higher probability to be inserted after stop *p* which is close to stop *s* and belongs to a 'short' trip (with respect to few students and short travel time). It is inefficient to check the feasibility of every vertex move and it is undesirable to insert stop *s* into a faraway trip even if it is feasible. Therefore, the size of the neighborhood is limited by Nnei ($< |\mathrm{SP}|$). The neighborhood does not contain the stops that are in the same trip with stop *s* since the main goal of

the vertex exchange is to move stop *s* to another trip such that the old trip of stop *s* might be deleted.

For each stop *p* in the neighborhood list (*NL[s]*) of stop *s*, the PI checks whether move (s, p) is in the Tabu List (TL) and whether move (p, s) is feasible. The feasibility check consists of the maximum ride time and vehicle capacity constraints. A move (s, p) would be in the Tabu list if it were accepted within some previous iterations. The PI only considers the feasible move (p, s) of which its reverse move (s, p) is not accepted recently. This idea is from the Tabu Search, and it is implemented to avoid frequent reverse moves. In the PI algorithm, the acceptance of a move (p, s) is based on the Simulated Annealing criteria where a move (p, s) might be acceptable even if it yields a worse cost function. It means that move (p, s) and move (s, p) may both be acceptable. One consequence is that once a move is removed from the *TL*, it is very likely to be implemented again even if it yields a worse cost function. Hence, the maximum forbidden iteration should be large enough to prevent the frequent removal of forbidden moves.

4.2.3.3 Surrogate cost

As explained above, the real estimation of the goodness of a routing plan is the number of buses and total vehicle time after solving a bus scheduling problem. But this is unknown at the routing stage. The Surrogate Cost (SC) (shown in equation 4-11) is used to assess the goodness of the routing plan and the goodness of a move without solving the bus scheduling problem. The cost function is the objective function of the SRC (3-52), see section 3.7.1.

$$SC = \delta_N \times |TP[k]| - \delta_C \times TC + \delta_T \times TT \quad (4\text{-}11)$$

$$TC = \sum_{t \in TP[k]} \sum_{k1 \in SCH \setminus O[k]} 1\{SB_{O\{t\}} + tt_t + D_{l\{t\},k1} \leq SB_{k1}\} \quad (4\text{-}12)$$

$$TT = \sum_{t \in TP[k]} tt_t \quad (4\text{-}13)$$

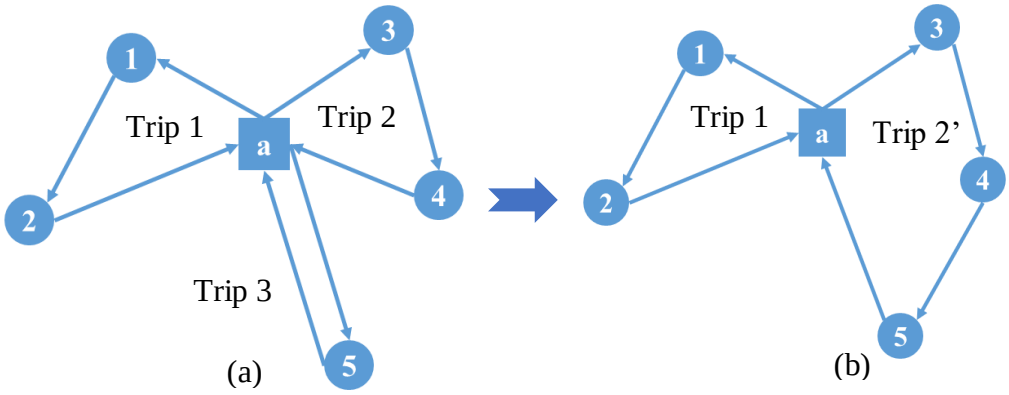
where $|TP[k]|, TC, TT$ are the total number of trips, the trip-to-school compatibilities and the total travel time with coefficients $\delta_N, \delta_C, \delta_T$. Constraints (4-12) calculate the total trip-to-school compatibility (*TC*). The total travel time (4-13) is the sum of the travel time and the pickup and drop-off time for all trips (for the current school *k*). The surrogate cost implies that a routing plan is a good one if the number of trips and the total travel time is small while the trip-to-school compatibility is as large as possible. Because (4-11) is just a surrogate estimate of the goodness of a routing plan, it is possible that a routing plan with smaller SC needs more buses than another one with higher SC. The PI adopts the acceptance criterion from the Simulated Annealing in (4-14) to help jump local optima by introducing some randomness.

$$P(accept\ a\ move) = 1/(1 + e^{(SC_{new} - SC_{old})/temp}) \quad (4\text{-}14)$$

Equation (4-14) ensures that when the new surrogate cost after a move is smaller than the old one (which is preferred), this move will be accepted. On the other hand, if the new surrogate cost is larger than the old one (which is undesired), the probability of accepting this move is small if the difference between two surrogate costs is large and the temperature is low. The PI will be iteratively implemented until either the maximum number of iterations without change is reached, or the temperature drops below the terminal temperature.

4.2.3.4 Example

One example of the PI is presented in Figure 12. At some iteration, the parameters are $temp = 10,\ iter = 2, TL = \{(1,4)\}$.



(a)

1. $temp = 10, iter = 2, TL = \{(1,4)\}$
2. **While** $temp \geq temp_{end}$ *and* $iter \leq iter_{max}$ **do**
   - 2.1 *EL* = {5, 3, 1, 4, 2}
   - 2.2 **for** (5 in EL) **do**
     - 2.2.1 *NL*[5] = {4, 2, 1}
     - 2.2.2 **for** (4 in *NL*[5]) **do**
       - move(4,5) satisfies conditions (A, B, C)
       - Implement move(4,5), update TP; $TL = \{(1,4), (4,5)\}$;**Break**;
   - 2.3 $temp = 10 \times 0.9 = 9; iter = 0$

(b)

1. $temp, iter = 0, TL = \{(1,4), (4,5)\}$
2. **While** $temp \geq temp_{end}$ *and* $iter \leq iter_{max}$ **do**
   - 2.1 *EL* = {1, 2, 3, 4, 5}
   - 2.2 **for** (1 in EL) **do**
     - 2.2.1 NL[1] = {3, 4, 5}
     - 2.2.2 **for** (3 in NL[1]) **do**
       - C. move(3,1) is not acceptable at *temp*
     - **for** (4 in NL[1]) **do**
       - B. move(1,4) is in in the TL
     - **for** (5 in NL[1]) **do**
       - A. move(5,1) is infeasible
   - **for** (2,3,4,5 in EL), no move is accepted
   - 2.3 $temp = 9 \times 0.9 = 8.1; iter = 1$

a School 1 Bus stop Trip

Figure 12 Illustration of the Post Improvement Algorithm

There are three trips connecting five stops for school *a*. In Figure 12(a), the PI first constructs the vertex exchange list (*EL*) as {5, 3, 1, 4, 2}. For the first stop (5) in

the *NL*, its neighborhood list is *NL*[5]={4, 2, 1} with $Nnei = 3$. The first move (4, 5) satisfies all conditions (A: feasibility, B: its reverse move is not in the Tabu list, C: it is acceptable given the surrogate cost estimate). Then stop 5 is moved from trip 3 to the position right after stop 4 in trip 2. Trip 3 becomes empty and is deleted while trip 2 is augmented to {3, 4, 5}. This iteration is terminated, and control parameters are updated: $temp = 10 \times 0.9 = 9;\ iter = 0;\ TL = \{(1,4),(4,5)\}$.

At the next iteration, the PI re-constructs the vertex exchange list (*EL*) as {1, 2, 3, 4, 5}. For the first stop (1), the PI finds its neighborhood list *NP*[1]={3, 4, 5}. Starting from the first stop 3, move (1, 3) is not acceptable due to the worse surrogate cost at the current temperature. The PI goes to the next stop 4. Its reverse move (1, 4) is in the *TL*, making it unacceptable. The PI continues and finds move (5, 1) infeasible. The PI then tries to move stop 2. The process is repeated, and no move is accepted for stop 2, 3, 4, and 5. This iteration terminates without any change. The temperature cools down, and iteration count is incremented by one ($temp = 9 \times 0.9 = 8.1;\ iter = 1$). The PI continues to the next iteration. Notice that due to the randomness of constructing a vertex exchange list and the randomness of acceptance criterion, some unacceptable moves might become acceptable at different iterations. Therefore, it is worth continuing the search even if no change occurs at one iteration.

4.3 Algorithm 3 to solve the MBSV (Model 5)

The hardest constraints in the MBSV (Model 5) are the MVT constraints. After dropping the MVT constraints, the Multi-School Bell Time and Scheduling Problem (MBS, Model 7) is obtained, which is much simpler than the MBSV (Model 5). Based

on Lemma 3, the MBSV (Model 5) can be simplified to the MBS (Model 7) if the solution to the MBS (Model 7) satisfies the MVT constraints.

Lemma 3: If the optimal solution to the MBS (Model 7) satisfies the maximum vehicle time constraints, it is the optimal solution to the MBSV (Model 5).

Proof: The MBS (Model 7) is a relaxation of the MBSV (Model 5). For minimization problems, the optimal solution to the MBS (Model 7) is a lower bound of the MBSV (Model 5). If all routes in that solution satisfy the maximum vehicle time constraints, this solution is also a feasible solution to the MBSV (Model 5). Hence, it is the optimal solution to the MBSV (Model 5).

4.3.1 The MBS (Model 7)

The MBS (Model 7)

$$\min z7 = \alpha_B \sum_{t \in ATP} y_{SDT,t} + \alpha_D \sum_{(t1,t2) \in E} dd_{t1,t2} \times y_{t1,t2} \quad (4\text{-}15)$$

Subject to

Constraints (3-25) (school bell time window constraints)

$$sb_{O\{t1\}} + tt_{t1} + dd_{t1,t2} - M \times (1 - y_{t1,t2}) \leq sb_{O\{t2\}}, \forall (t1, t2) \in E \quad (4\text{-}16)$$

$$\sum_{t1:(t1,t) \in E} y_{t1,t} = 1, \forall t \in ATP \quad (4\text{-}17)$$

$$\sum_{t2:(t,t2) \in E} y_{t,t2} = 1, \forall t \in ATP \quad (4\text{-}18)$$

$$y_{t1,t2} \in \{0,1\}, \forall (t1, t2) \in E \quad (4\text{-}19)$$

4.3.2 Illegal route update (IRU)

An intuitive way to solve the MBSV (Model 5) is by solving the MBS (Model 7) while satisfying the MVT constraints. The satisfaction of the MVT constraints can be guaranteed by excluding all the Illegal Bus Routes (IBR) which violate the MVT

constraints. Since the number of all illegal bus routes are huge, it can be generated dynamically. Given the routing plan, the MBS (Model 7) is solved. Then, the solution to the MBS (Model 7) is checked to ascertain whether the MVT constraints are satisfied. If all bus routes satisfy the MVT constraints, the solution is optimal, and the algorithm terminates. If some routes violate the MVT constraints, they are added to the Illegal Bus Route (IBR). For every illegal bus route *r* in the IBR, the following constraints are added to exclude these illegal bus routes.

$$\sum_{(t1,t2)\in r} y_{t1,t2} \leq |r| - 2, \forall r \in \text{IBR} \quad (4\text{-}20)$$

where $|r|$ is the number of bus stops in route *r*. $(t1, t2) \in r$ refers to all trip pairs that were served on bus route *r*. For instance, an illegal bus route {*t1, t5, t6, t8*} is identified (excluding the start and end depot trip). Constraints (4-20) is $y_{t1,t5} + y_{t5,t6} + y_{t6,t8} \leq 2$. Bus route {*t1, t5, t6, t8*} would become illegal after adding this constraint because $y_{t1,t5} + y_{t5,t6} + y_{t6,t8}$ is evaluated to 3. The pseudo-code of Algorithm 3, which is developed to solve the MBSV (Model 5), is as follows.

Algorithm 3

1. IBR = ϕ
2. **While** True **do**
    3. Solve the MBS (Model 7) with constraints (4-20)
    4. **For** (route *r* in the current solution) **do**
        4.1 **If** (route *r* violates the MVT) **then**
            A. IBR_update(r)
        4.2 Else
            B. **Return** the current solution;

Intuitively, Algorithm 3 can be represented in Figure 13. Assume the blue vertical solid line corresponds to the optimal solution to the MBSV (Model 5).

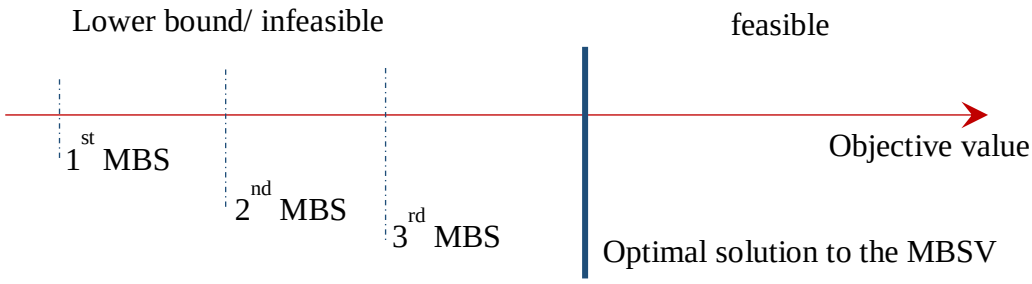


Figure 13 Illustration of Algorithm 3

For a minimization problem, all the solutions with objective values smaller than the optimal solution are infeasible and they are the lower bounds of the optimization problem. All feasible solutions have an objective value no smaller than the optimal solution, which corresponds to the right-hand side of the blue vertical line in Figure 13. Algorithm 3 starts by solving the MBS (Model 7), which does not consider the Maximum Vehicle Time Constraint. Its solution is represented by the 1st MBS in Figure 13. Then, Algorithm 3 identifies the illegal bus routes and re-solves the MBS (Model 7) with the updated IBR. The new solution, demonstrated by the 2nd MBS in Figure 13, has a bigger objective value but it gets closer to the feasible region. Algorithm 3 continues this process iteratively until reaching the feasible region, which is the optimal solution to the MBSV (Model 5).

4.3.3 Discussion

4.3.3.1 Super-routes and sub-routes

Constraints (4-20) exclude all super-routes of an illegal bus route but keep all sub-routes of this illegal bus route. The super-routes of a route *r* are the routes that contain *r*. And the sub-routes of a route *r* are the routes which are a portion of route *r*. From the super-routes' perspective, due to the triangle inequality (see section 2.1.3.3),

adding more trips into a bus route would not decrease the total vehicle time. It means that the super-routes of illegal bus routes (in terms of the MVT) are also illegal bus routes. And they should be excluded. On the other hand, the subset of illegal bus routes might be legal and should be kept in the model.

4.3.3.2 The IBR update strategy

At every iteration, the easiest way to update the IBR (Step 4.1.A in Algorithm 3) is simply adding the newly identified (by solving the MBS, Model 7) illegal bus route *r* (say *base illegal routes*). But whenever a base illegal route is identified, similar illegal routes can be found by replacing part of the base illegal route to longer trips. Those newly generated illegal routes are called the *replaced illegal routes*. Base and replaced illegal routes are both illegal and should be included in the IBR set. The only difference is the way they are identified. Base illegal routes are found by solving the MBS (Model 7), and the replaced illegal routes are found by replacing a portion of the base illegal routes.

Take the last trip on a base illegal route for example. If route $r = \{t1, t5, t6, t8\}$ is identified as a base illegal route, route $\{t1, t5, t6, t_j\}$ is a replaced illegal route if it satisfies the following constraints:

$$dd_{t6,t8} + tt_{t8} + dd_{t8,EDT} \leq dd_{t6,t_j} + tt_{t_j} + dd_{t_j,EDT}, \forall t_j: (t6, t_j) \in E | t_j \neq r \qquad (4\text{-}21)$$

Therefore, new illegal bus routes can be identified by replacing some trip on an illegal bus route by a trip with longer vehicle time (the sum of the travel time of this trip and the deadhead before and after this trip). This replacement strategy could go much further. Multiple trips can be replaced by one longer trip or several trips as long as the replacing trip has longer total vehicle time. The replaced illegal routes can also

be treated as base illegal routes and used to generate more illegal routes. The merit of the replacement is that it finds illegal bus routes in a much cheaper way than solving the MBS (Model 7) repeatedly. But the tradeoff is that it might take some time to loop through all possible replacements and calculate (4-20) for every possible replacement. Moreover, some of the replaced illegal routes might be unnecessary. If there are too many wasteful illegal bus routes, the MBS (Model 7) will slow down with no benefit. In the extreme case where all possible replacements are considered, it is the same as finding all possible illegal bus routes from the beginning. In this case, the benefit of the MBS (Model 7), which is only taking the subset of all illegal routes, is gone. It is a tricky task to determine how many possible replacements to explore at every iteration.

4.3.3.3 Running time

The advantage of Algorithm 3 is that if only a few illegal bus routes exist, it is extremely fast. A few illegal bus routes could come from two cases: either the problem size is relatively small, or the MVT is relatively loose. In both, Algorithm 3 can find the optimal solution very fast. Furthermore, starting from the second iteration, we can specify an initial solution based on the solution from the prior iteration. We serve each trip in the illegal route by a bus and use this modified solution as the initial solution to the next iteration. In this way, Algorithm 3 takes much less time to solve the Model 7 starting from the second iteration than the first iteration. The information from the prior iteration benefits the following iteration.

However, the disadvantage of Algorithm 3 is that if there exist lots of illegal bus routes (the problem size is large, or the MVT is super tight or both), Algorithm 3 will take many iterations to solve.

Another problem is that if Algorithm 3 is terminated in the middle of the process where some illegal bus routes still exist, this solution is not even a feasible solution to the MBSV (Model 5) with respect to the MVT constraints. So, the solution from Algorithm 3 is either optimal[1] or infeasible[2]. The good news is that in practice, the maximum vehicle time cannot be extremely tight because the school bell times and the bus schedule both follow (at least roughly) the regular eight-hour day schedule. This property makes Algorithm 3 more practical.

4.4 Model comparison

The models discussed above solve different problems with different level-of-service constraints. An illustration of the number of variables and constraints in all models is presented in Table 6. The two test problems are from section 6.1.4. It shows the huge reduction of variables and constraints from the integrated model to the decomposed models. The comparison of these models is summarized in Table 7.

Table 6 Model comparison in two test problems

		MBRSV Model 1	MBRS Model 2	MRS Model 3	SRC Model 4	MBSV Model 5	MCPM Model 6	MBS Model 7
Problem size		**2 schools / 20 stops**						
NOV	Integer	2	2	0	0	2	0	2
	Binary	**1,152**	**1,040**	**992**	**580**	**128**	**18**	**12**
	Continuous	1,545	897	856	492	12	0	0
# of constraints		**2,488**	**1,432**	**1,175**	**759**	**194**	**11**	**22**
Problem size		**30 schools / 300 stops**						
NOV	Integer	30	30	0	0	30	0	30
	Binary	**1,758,432**	**30,432**	**19,632**	**1,704**	**217,680**	**33**	**1,754**
	Continuous	13,586	28,193	20,872	1,698	60	0	0
# of constraints		**3,513,088**	**49,888**	**32,384**	**2,397**	**225,180**	**14**	**1,904**

Note: **NOV**: Number of variables

[1] Barrier constraints (optimal gap and/or running time) can be set when solving each MBS (Model 7). In these cases, the solution might not be optimal. Assume such barrier constraints do not exist.
[2] Infeasible means the solution from Algorithm 3 (if terminated in the middle) violates the MVT in the MBSV (Model 5). Not that the MBSV (Model 5) or the MBS (Model 7) is infeasible.

Table 7 Analytical model comparison

		MBRSV (Model 1)	MBRS (Model 2)	MRS (Model 3)	SRC (Model 4)	MBSV (Model 5)	MCPM (Model 6)	MBS (Model 7)
Scope		R, B, S	R, B, S	R, S	R	B, S	R	B, S
# of schools		M	M	M	S	M	S	M
Level-of-service		MRT, MVT	MRT	MRT	MRT	MVT	-	-
# of variable	**Integer**	$\|SCH\|$	$\|SCH\|$	0	0	$\|SCH\|$	0	$\|SCH\|$
	Binary	$\|B\| \times (\|TP\|^2 + \|TP\|) + \|TP\| + \sum_{k\in SCH}(\|TP[k]\| \times \|SP[k]\|^2 + 2 \times \|TP[k]\| \times \|SP[k]\|)$	$\|TP\|^2 + \|TP\| + \sum_{k\in SCH}(\|TP[k]\| \times \|SP[k]\|^2 + 2 \times \|TP[k]\| \times \|SP[k]\|)$	$\|TP\| \times (\|SCH\| + 1) + \sum_{k\in SCH}(\|TP[k]\| \times \|SP[k]\|^2 + 2 \times \|TP[k]\| \times \|SP[k]\|)$	$\|TP[k]\| \times (\|SP[k]\|^2 + 2 \times \|SP[k]\| + \|SCH\|)$	$\|B\| \times (\|E\| + \|ATP\|)$	$\|R\| \times \|TP[k]\|$	$\|E\|$
	Continuous	$(\|TP\| + 1)^2 \times (\|B\| + 1) + \|TP\| + \sum_{k\in SCH}(\|TP[k]\| \times \|SP[k]\|^2)$	$(\|TP\| + 1)^2 + \|TP\| + \sum_{k\in SCH}(\|TP[k]\| \times \|SP[k]\|^2)$	$2 \times \|TP\| \times (\|SCH\| + 1) + \sum_{k\in SCH}(\|TP[k]\| \times \|SP[k]\|^2)$	$\|TP[k]\| \times (\|SP[k]\|^2 + 2 \times \|SCH\|)$	$\|ATP\|$	0	0
# of constraints		$2 \times \|TP\| \times \|B\| \times (\|TP\| + 1) + \|B\| \times (0.5 \times \|B\| + 2) + 5 \times \|TP\| + 2 \times \|SCH\| + \sum_{k\in SCH}(\|TP[k]\| \times \|SP[k]\|^2 + 0.5 \times \|TP[k]\|^2 + 5 \times \|TP[k]\| \times \|SP[k]\| + \|SP[k]\|)$	$2 \times \|TP\|^2 + 7 \times \|TP\| + 2 \times \|SCH\| + \sum_{k\in SCH}(\|TP[k]\| \times \|SP[k]\|^2 + 0.5 \times \|TP[k]\|^2 + 5 \times \|TP[k]\| \times \|SP[k]\| + \|SP[k]\|)$	$\|TP\| \times (3 \times \|SCH\| + 7) + \|SP\| + \sum_{k\in SCH}(\|TP[k]\| \times \|SP[k]\|^2 + \|TP[k]\|^2 + 5 \times \|TP[k]\| \times \|SP[k]\|)$	$\|TP[k]\| \times (\|TP[k]\| + \|SP[k]\|^2 + 5 \times \|SP[k]\| + 3 \times \|SCH\| + 4) + \|SP[k]\|$	$\|B\| \times (\|E\| + 2 \times \|ATP\| + 1) + 2 \times (\|ATP\| + \|SCH\|)$	$\|R\| + \|TP[k]\|$	$\|E\| + 2 \times \|ATP\| + \|SCH\|$

Note: Scope: R (Routing), B (Bell time adjustment), S (Scheduling);
# of school: M (Multi-school), S (Single-school)
Level-of-service: MRT (Maximum ride time), MVT (Maximum vehicle time)
$|set|$: the dimension of the set
Number of constraints do not include the domain (integrity, binary, nonnegative continuous) constraints
Dimension relationships:
$|SCH| \leq \sum_{k\in SCH} MNT[k] \leq |ATP| \leq \sum_{k\in SCH} MNT[k] + AAT[k] = |TP|$;
$|R| < |SP[k]|; |ATP| \leq |SP|; |E| < 0.5 \times |ATP|^2; |B| \leq |ATP|$
$\forall k \in SCH$: $MNT[k] \leq |ATP[k]| \leq MNT[k] + AAT[k] = |TP[k]|$; $|ATP[k]| \leq |SP[k]|$;

4.5 Chapter summary

In this chapter, the relaxation, the transformation, and the decomposition framework to solve the school bell time, bus routing and scheduling problem was presented in Algorithm 1. The underlying rationale is to relax and decompose the integrated model into many smaller and simpler subproblems. Based on the decomposed single-school problem used (the SRC, Model 4; the SRCA, Model 4'; and the SRCW, Model 4"), Algorithm 1 has three versions.

Then, Algorithm 2 was proposed to solve each SRC (Model 4). Algorithm 2 is a two-stage heuristic algorithm. It first finds an initial solution by solving a set of minimum cost matching problems and then improves the initial solution using the vertex exchange method under the framework of the Tabu Search and the Simulated Annealing.

In the end, Algorithm 3 was proposed to solve the MBSV (Model 5) iteratively. The combination of Algorithms 1, 2 and 3 provides a complete scheme to solve the MBRSV (Model 1). The pseudo codes and examples were presented for each algorithm as well as the proofs and the discussions. The models proposed in Chapter 3 and Chapter 4 were compared in section 4.4.

Chapter 5 Stochastic School Bell Time and Bus Scheduling Optimization Models and Algorithms

In a general Vehicle Routing Problem (VRP), the stochasticity comes from the uncertainty of the customer demand and the uncertainty of the travel time. In a school bus planning problem, the students are assigned to their designated bus stops. Thus, customer demand is deterministic. We only consider the uncertainty of the travel time. More specifically, we consider the Chance-Constrained Multi-School Bell Time and Bus Scheduling Problems. The Stochastic On-Time Chance Constraint (SOT) in this stochastic model is:

For each bus route, the probability of all trips on that route being served on time is at least ρ.

This constraint is easier to understand in an ensemble method. Assume that we have 1000 sets of observations of the travel time and the deadhead durations. Each set is one realization of the stochastic system. A bus route $r = \{t1, t2, t3\}$ satisfies the SOT only if in at least 980 realizations, the bus arrives at the initial stop of all trips $t1, t2, t3$ before their scheduled start times. Note, even if the bus fails to arrive on-time for one trip, this whole realization is marked as a failure.

The stochastic programming is usually computationally expensive. We first consider a simpler version of the problem in section 5.1, which still considers the traffic congestion into the system. Then, we present a chance-constrained programming model under the assumption of normally distributed travel times in section 5.2. In the end, a

Column Generation-based Stochastic School Bell Time and Bus Scheduling problem is proposed in section 5.3 without any assumption of the travel time distribution.

5.1 Time-dependent multi-school bell time and scheduling

A major contributor to the stochasticity of the travel time is the congestion, which in general shows a daily pattern dependent on the time of the day. Thus, we present a degraded version of the stochastic model – a Time-Dependent Multi-School Bell Time and Bus Scheduling Problem with the Maximum Vehicle Time Constraints (TD_MBSV, Model 8). In this model, the ride time of the trips and the deadhead duration (between trips) vary at different departure times. But at a specific departure time, the ride time and the deadhead are deterministic. Clearly, this is not a stochastic model. Yet, the TD_MBSV (Model 8) can account for the traffic congestion under different departure times, which gives more flexibility to the system.

Given all the trips $\mathrm{t} \in \mathrm{TP}$, we know the start and end stop of each trip. First, we define a possible compatible trip set *E* if it satisfies one of the conditions: 1) $(\mathrm{t1}, \mathrm{t2}) \in \mathrm{E}, \forall \mathrm{t1}, \mathrm{t2} \in \mathrm{TP} | \mathrm{t1} \neq \mathrm{t2}$ and $(\mathrm{t1}, \mathrm{t2})$ is compatible under at least one departure time; 2) $(\mathrm{SDT}, \mathrm{t2}) \in E, \forall \mathrm{t2} \in \mathrm{TP}$ and 3) $(\mathrm{t1}, \mathrm{EDT}) \in E, \forall \mathrm{t1} \in \mathrm{TP}$. The trip pairs that are not in *E* are not compatible under all departure times. Thus, only trip pairs in *E* are considered in our formulation. Moreover, all the afternoon trips depart at the school dismissal time and the schools only dismiss at a certain discrete time stamp (i.e., 2:15 PM). Thus, we can list a discrete set of the ride time and deadhead under finite discrete departure times.

5.1.1 Notations

The notations used in the TD_MBSV (Model 8) are summarized in Table 8.

Table 8 Notation summary for the TD_MBSV (Model 8)

Set and Index	
$\omega \in \Omega$	Set of discrete time space
$t \in TP[k]$	Set of trips for school k
$k \in SCH$	Set of schools
$(t1, t2) \in E$	Set of possible compatible trip pairs
Variables	
$y_{t1,t2}^{b}$	Binary variable equals 1 if trip $t2$ is served right after trip $t1$ on bus b
sb_{k}^{ω}	Binary variable equals 1 if school k dismisses at time ω
ttb_{t}^{b}	Nonnegative continuous variable, the actual travel time of trip t if it is served by bus b. This is an auxiliary variable for linearization.
$ddb_{t1,t2}^{b}$	Nonnegative continuous variable, the deadhead between trip $t1$ to trip $t2$ on bus b. This is an auxiliary variable for linearization.
Parameters	
$O\{t\}$	The school that trip t belongs to
$dd_{t1,t2}^{\omega}$	The deadhead from trip $t1$ to $t2$ if $t1$ departs at time ω
tt_{t}^{ω}	The ride time of trip $t1$ at department time ω
α_B, α_D	The coefficient for the number of buses (α_B), total deadhead (α_D) in the objective function
M	A large positive value (big-M)

5.1.2 The TD_MBSV (Model 8)

The Time-Dependent Multi-School Bell Time and Bus Scheduling with the Maximum Vehicle Time Constraints Problem (TD_MBSV, Model 8) is as follows:

The TD_MBSV (Model 8)

$$\min z8 = \alpha_B \sum_{b\in B} \sum_{t2\in TP} y_{SDT,t2}^{b} + \alpha_D \sum_{b\in B} \sum_{(t1,t2)\in E} ddb_{t1,t2}^{b} \quad (5\text{-}1)$$

Subject to:

$$\sum_{b\in B} \sum_{t1:(t1,t)\in E} y_{t1,t}^{b} = 1, \forall t \in TP \quad (5\text{-}2)$$

$$\sum_{t1:(t1,t)\in E} y_{t1,t}^{b} = \sum_{t2:(t,t2)\in E} y_{t,t2}^{b}, \forall b \in B, t \in TP \quad (5\text{-}3)$$

$$\sum_{t2:(t,t2)\in E} y_{SDT,t2}^{b} \leq 1, \forall b \in B \quad (5\text{-}4)$$

$$\sum_{\omega\in\Omega} sb_{k}^{\omega} = 1, \forall k \in SCH \quad (5\text{-}5)$$

$$\omega_1 + tt_{t1}^{\omega_1} + dd_{t1,t2}^{\omega_1} - M \times \left(2 - sb_{O\{t1\}}^{\omega_1} - \sum_{b\in B} y_{t1,t2}^{b}\right) \leq \omega_2 + M \times \left(1 - sb_{O\{t2\}}^{\omega_2}\right), \forall (t1, t2) \in E, \omega 1 \in \Omega, \omega 2 \in \Omega | \omega 1 < \omega 2 \quad (5\text{-}6)$$

$$ddb^{b}_{t1,t2} \geq \sum_{\omega\in\Omega} dd^{\omega}_{t1,t2} sb^{\omega}_{O\{t1\}} - M \times \left(1 - y^{b}_{t1,t2}\right), \forall b \in B, (t1,t2) \in E \quad (5\text{-}7)$$

$$ttb^{b}_{t} \geq \sum_{\omega\in\Omega} tt^{\omega}_{t} sb^{\omega}_{O\{t\}} - M \times \left(1 - z^{b}_{t}\right), \forall b \in B, t \in TP \quad (5\text{-}8)$$

$$\sum_{t\in TP} ttb^{b}_{t} + \sum_{(t1,t2)\in E} ddb^{b}_{t1,t2} \leq MVT, \forall b \in B \quad (5\text{-}9)$$

$$\sum_{t\in TP} ttb^{b1}_{t} \times \left(\sum_{t1:(t1,t)\in E} y^{b1}_{t1,t}\right) \geq \sum_{t\in TP} ttb^{b2}_{t} \times \left(\sum_{t1:(t1,t)\in E} y^{b2}_{t1,t}\right), \quad \forall b1, b2 \in B | b1 \leq b2 \quad (5\text{-}10)$$

$$y^{b}_{t1,t2} \in \{0,1\}, \forall b \in B, (t1,t2) \in E \quad (5\text{-}11)$$

$$sb^{\omega}_{k} \in \{0,1\}, \forall k \in SCH, \omega \in \Omega \quad (5\text{-}12)$$

$$ddb^{b}_{t1,t2} \geq 0, \forall b \in B, (t1,t2) \in E \quad (5\text{-}13)$$

$$ttb^{b}_{t} \geq 0, \forall b \in B, t \in TP \quad (5\text{-}14)$$

Objective (5-1) minimizes the number of bus routes and the total deadhead. Constraints (5-2) guarantee that each trip is served exactly once. Constraints (5-3) assign a pair of preceding and successive trips on one bus. Constraints (5-4) regulate that every bus departs from the depot. Constraints (5-5) determine the schools' dismissal times. Constraints (5-6) are the trip compatibility constraints under the time-dependent case. Constraints (5-7) calculate the time-dependent deadhead in the final scheduling plan, which is for the purpose of linearization. Similarly, constraints (5-8) calculate the actual travel time of the trips on a bus under a specific departure time. Constraints (5-9) are the maximum vehicle time constraints under the time-dependent case. Constraints (5-10) break the symmetries, which assign the longest bus route to the bus with the smallest ID number. Constraints (5-11) and (5-12) are binary domain constraints. Constraints (5-13) and constraints (5-14) are the nonnegative continuous domain constraints. Similar big-M reduction technique is applied to constraints (5-8) where the big-M is reduced to the maximum ride time of a trip across all departure times.

5.2 Chance-constrained multi-school bell time and scheduling

In this section, we consider that the ride time of the trips (TT_t) and the deadhead durations ($DD_{t1,t2}$) are random variables instead of constant parameters. In this way, we can solve the chance-constrained programming model. Recall the Stochastic On-Time Chance Constraint (SOT) is:

For each bus route, the probability of all trips on that route being served on time is at least ρ.

In theory, we must write down the joint probability of all trips on a route to calculate the upper probability, which is extremely hard and causes the formulation to become highly nonlinear. Thus, we use the following Trip-based Stochastic On-Time Constraint (TSOT) in this section:

For each trip, the probability that this trip is served on time is at least ρ.

With this assumption, we present a Trip-based Stochastic On-Time Constrained Multi-School Bell Time and Bus Scheduling with the Maximum Vehicle Time Constraints Problem (TS_MBSV, Model 9). The validity of the Trip-based Stochastic On-Time Constraint (TSOT) will be tested in the computational experiments.

5.2.1 Notations

The notations used to formulate the TS_MBSV (Model 9) are summarized in Table 9. Generally, they are very similar to the ones in section 5.1.1. But since the travel time is uncertain and if it has an infinite density tail like normal distribution, in theory any pair of trips (if belong to different schools) are potentially compatible. However, the travel time in real world cannot be smaller than the free flow travel time. Thus, we still define the compatible trip set E the same way as in the section 5.1 except

that the previously deterministic travel time is now the mean of the stochastic travel time under different departure times.

Table 9 Notation summary for the TS_MBSV (Model 9)

Set and Index	
$t \in TP[k]$	Set of trips for school k
$k \in SCH$	Set of schools
$(\mathrm{t1}, \mathrm{t2}) \in \mathrm{E}$	Set of compatible trips
Variables	
$y_{t1,t2}^{b}$	Binary variable equals 1 if trip $t2$ is served right after trip $t1$ on bus b
sb_k	Binary variable equals 1 if school k dismisses at time ω
$\hat{\mu}_{A_t}$	Nonnegative continuous variable, the estimated mean of the arrival time of a bus at the initial stop of trip t
$\hat{\sigma}_{A_t}$	Nonnegative continuous variable, the estimated standard deviation of the arrival time of a bus at the initial stop of trip t
$\gamma_{h_t}^{A_t}$	Binary variable equals 1 if the auxiliary integer h_t is the estimated standard deviation of the arrival time for trip t
Parameters	
$O\{t\}$	The school that trip t belongs to
ξ	The minimum bell time interval
$\mu_{DD_{t1,t2}}$	The mean of the deadhead from trip $t1$ to $t2$
μ_{TT_t}	The mean of the ride time of trip t
α_B, α_D	The coefficient for the number of buses (α_B), total deadhead (α_D) in the objective function
ρ	The minimum probability that a trip is served on time
M	A large positive value (big-M)

5.2.2 The TS_MBSV (Model 9)

The Trip-based Stochastic On-Time Constraint (TSOT) requires that the arrival time ($\mathrm{A_t}$) of a bus at the initial stop of trip t is less than or equal to the scheduled departure time of trip t with probability ρ. Mathematically, it is:

$$\mathrm{P}\left(\mathrm{A_t} \le sb_{O\{t\}}\right) \ge \rho, \forall t \in TP \tag{5-15}$$

$$\mathrm{A_{t2}} = \max\left(A_{t1}, sb_{O\{t1\}}\right) + TT_{t1} + DD_{t1,t2} \quad \text{if t1 is followed by t2 on one bus} \tag{5-16}$$

where A_t is a random variable. If a bus arrives at the initial stop of preceding trip *t1* on time, the actual departure time of *t1* is the school dismissal time ($sb_{O\{t1\}}$) with no uncertainty. In addition, if the ride time of trip *t1* (TT_{t1}) and the deadhead duration ($DD_{t1,t2}$) are both normally distributed, the arrival time at the successive trip *t2* (A_{t2}) is also normally distributed due to the additive property.

On the other hand, if a bus arrives at the initial point of trip *t1* later than the school's dismissal time, the actual departure time of trip *t1* is the arrival time (A_{t1}). If the arrival time of trip *t1* is normally distributed, due to additive property, the arrival time of trip *t2* is again normally distributed. The assumption that the arrival time of trip *t1* is normally distributed can be derived by induction because it is also a successive trip of another preceding trip (or the depot trip). Considering both cases, the arrival time of any trip *t* (A_t) is approximately normally distributed with the mean μ_{A_t} and the standard deviation σ_{A_t}. Hence, we standardize (5-15), and the TSOT constraint becomes:

$$\Phi\left(\frac{sb_{O\{t\}}-\mu_{A_t}}{\sigma_{A_t}}\right) \geq \rho, \forall t \in TP \tag{5-17}$$

Taking the inverse, (5-17) becomes

$$sb_{O\{t\}} - \mu_{A_t} \geq \Phi^{-1}(\rho) \times \sigma_{A_t}, \forall t \in TP \tag{5-18}$$

Note μ_{A_t} and σ_{A_t} are estimated by the continuous non-negative decision variables $\hat{\mu}_{A_t}$ and $\hat{\sigma}_{A_t}$ in our formulation for the purpose of linearization.

$$sb_{O\{t\}} - \hat{\mu}_{A_t} \geq \Phi^{-1}(\rho) \times \hat{\sigma}_{A_t}, \forall t \in TP \tag{5-19}$$

$$\hat{\mu}_{A_t} \geq sb_{O\{t\}} - M \times \left(1 - \sum_{b \in B} y^b_{SDT,t}\right), \forall t \in TP \tag{5-20}$$

$$\mu_{A_{t2}} \geq sb_{O\{t1\}} \times \rho + \hat{\mu}_{A_t} \times (1-\rho) + \mu_{TT_{t1}} + \mu_{DD_{t1,t2}} - M \times \left(1 - \sum_{b \in B} y_{t1,t2}^{b}\right), \forall (t1, t2) \in E \quad (5\text{-}21)$$

$$\sum_{h_{t2}=1}^{h_{t2}^{+}} h_{t2}^{2} \gamma_{h_{t2}}^{A_{t2}} \geq \sum_{h_{t1}=1}^{h_{t1}^{+}} h_{t1}^{2} \gamma_{h_{t1}}^{A_{t1}} + \sigma_{TT_{t1}}^{2} + \sigma_{DD_{\mathrm{t1,t2}}}^{2} - M \times \left(1 - \sum_{b \in B} y_{t1,t2}^{b}\right), \forall (t1, t2) \in E \quad (5\text{-}22)$$

$$\sum_{h_t=1}^{h_t^{+}} h_t \gamma_{h_t}^{A_t} = \hat{\sigma}_{\mathrm{A}_t}, \forall t \in TP \quad (5\text{-}23)$$

$$\sum_{h_t=1}^{h_t^{+}} \gamma_{h_t}^{A_t} = 1, \forall t \in TP \quad (5\text{-}24)$$

Constraints (5-19) are the Trip-based Stochastic On-Time Constraints (TSOT). Constraints (5-20) assume that a bus can always arrive at the first trip on a route on time. This is because the bus has no prior work and it can depart from the depot earlier in a highly stochastic system. Constraints (5-21) calculate the estimated mean arrival time at the trips that are not the first trip in each bus route. It assumes that its prior trip is on time with probability ρ and late with probability $1-\rho$. The actual departure time of the prior trip is different under the on time and late cases.

Constraints (5-22) - (5-24) calculate the standard deviation of the arrival time at the initial stop of each trip. The linearization of constraints (5-23) and (5-24) mostly derives from the method proposed by Caceres et al. (2017). Because $\gamma_h^{A_t}$ is a binary variable, the assignment constraints (5-24) make sure that $\hat{\sigma}_{A_t}$ will take one integer value of $\mathrm{h_t} \in [0, \mathrm{h_t^+}]$ where $\mathrm{h_t^+}$ is the largest possible standard deviation of A_t. The $\mathrm{h_t^+}$ is approximately estimated given historical data. In this study, we used a numerical estimation. For every trip *t*, we find its preceding trip *t1* with the largest $\sigma_{TT_{t1}}^2 + \sigma_{DD_{\mathrm{t1,t}}}^2$. Then, we find the preceding trip for trip *t1* until no possible preceding trip exists. Let route r be the longest possible preceding route for trip *t*, then $\mathrm{h_t^+} = \left\lfloor \sqrt{\sum_{(t1,t2) \in r} \sigma_{TT_{t1}}^2 + \sigma_{DD_{\mathrm{t1,t2}}}^2} \right\rfloor$ is the largest possible standard deviation of the arrival

time of trip *t*. Now, $\Phi^{-1}(\rho), \mu_{T_t}, \mu_{dd_{t1,t2}}, \sigma^2_{T_{t1}}, \sigma^2_{\mathrm{dd}_{\mathrm{t1,t2}}}, M, h_t$ are all constant parameters for a fixed ρ and can be estimated from the historical sample data. Constraints (5-19) - (5-24) are linear with respect to the variables $sb_{O\{t\}}, \mu_{A_t}, \widehat{\sigma}_{\mathrm{A_t}}, y^b_{t1,t2}, \gamma^{A_t}_{h_t}$.

The Trip-based Stochastic On-Time Constrained Multi-School Bell Time and Bus Scheduling with the Maximum Vehicle Time Constraints Problem (TS_MBSV, Model 9) is defined as follows:

The TS_MBSV (Model 9)

$$\min z9 = \alpha_B \sum_{b\in B} \sum_{t\in ATP} y^b_{SDT,t} + \alpha_D \sum_{b\in B} \sum_{(t1,t2)\in E} \mu_{DD_{t1,t2}} \times y^b_{t1,t2} \quad (5\text{-}25)$$

Subject to:

Constraints (5-19) - (5-24)

$$\sum_{b\in B} \sum_{t1:(t1,t)\in E} y^b_{t1,t} = 1, \forall t \in TP \quad (5\text{-}26)$$

$$\sum_{t1:(t1,t)\in E} y^b_{t1,t} = \sum_{t2:(t,t2)\in E} y^b_{t,t2}, \forall b \in B, t \in TP \quad (5\text{-}27)$$

$$\sum_{t2:(t,t2)\in E} y^b_{SDT,t2} \leq 1, \forall b \in B \quad (5\text{-}28)$$

$$\sum_{t\in TP} \mu_{TT_t} \times \left(\sum_{t1:(t1,t)\in E} y^b_{t1,t}\right) + \sum_{(t1,t2)\in E} \mu_{DD_{t1,t2}} \times y^b_{t1,t2} \leq MVT, \forall b \in B \quad (5\text{-}29)$$

$$\sum_{t\in TP} \mu_{TT_t} \times \left(\sum_{t1:(t1,t)\in E} y^{b1}_{t1,t}\right) \geq \sum_{t\in TP} \mu_{TT_t} \times \left(\sum_{t1:(t1,t)\in E} y^{b2}_{t1,t}\right), \quad \forall b1, b2 \in B | b1 \leq b2 \quad (5\text{-}30)$$

$$\mathrm{LSB_k} \leq \mathrm{sb_k} \leq \mathrm{ESB_k} \text{ and } \mathrm{sb_k} \bmod \xi = 0, \forall \mathrm{k} \in \mathrm{SCH} \quad (5\text{-}31)$$

$$y^b_{t1,t2} \in \{0,1\}, \forall b \in B, (t1,t2) \in E \quad (5\text{-}32)$$

$$\mu_{A_t} \geq 0, \forall t \in TP \quad (5\text{-}33)$$

$$\widehat{\sigma}_{\mathrm{A}_t} \geq 0, \forall t \in TP \quad (5\text{-}34)$$

$$\gamma^{A_t}_{h_t} \in \{0,1\}, \forall t \in TP, h_t \in [1, h^+_t] \quad (5\text{-}35)$$

Objective (5-25) minimizes the number of buses and the total expected deadhead duration. Constraints (5-19) - (5-24) are the Trip-based Stochastic On-Time

Constraint, which ensures the trip compatibility constraints as well. Constraints (5-26) guarantee that each trip is served exactly once. Constraints (5-27) assign a pair of preceding and following trips on the same bus. Constraints (5-28) regulate that every bus departs from the depot. Constraints (5-29) are the maximum vehicle time constraint based on the mean ride time of the trips and the mean deadhead duration. Constraints (5-30) break the symmetries. Constraints (5-31) are the school bell/dismissal time constraints. Finally, constraints (5-32) - (5-35) are domain constraints.

5.3 Stochastic multi-school bell time and scheduling (Algorithm 4)

A widely used method for stochastic and robust optimization is scenario-based optimization. With the discrete assumption of travel time, we can list all possible realizations of the system with a specific probability. Treating each realization as a deterministic model, a scenario-based stochastic programming model can be formulated with the objective of minimizing the expected operations cost over all possible realizations. The drawback of this scenario-based formulation is that the number of possible realizations is huge, especially if we consider the stochastic travel time on every edge.

Alternatively, we propose a two-stage Stochastic Multi-School Bell Time and Bus Scheduling Algorithm (SMBS, Algorithm 4). The first stage is the Column Generation-Based Stochastic Multi-School Bus Scheduling Algorithm (CG_MS, Algorithm 4-1) and the second stage is the Stochastic Multi-School Bell Time Adjustment Algorithm (SMB, Algorithm 4-2). Algorithm 4 is a sample-based method, which is very flexible. It can solve problems with discrete or continuous distribution of

the travel time and deadhead or even directly with the samples without any assumption of the distribution. The notations used in Algorithm 4 are summarized in Table 10.

Table 10 Notation summary for Algorithm 4

Set and Index	
$r \in R$	Set of feasible bus routes
$ir \in IR$	Set of infeasible bus routes
$t \in TP$	Set of trips
$(t1, t2) \in E$	Set of compatible trips
Parameters	
it_{max}, it_{bt}	The maximum number of iterations that the scheduling plan is not improved (it_{max}) in the otter loop (under the change of bell times); The maximum number of iterations that no beneficial school bell time plan is found (it_{bt});
it_{ss}	The maximum number of iterations that the scheduling plan is not improved (it_{ss}) in the inner loop (stochastic scheduling with fixed bell time)

The pseudo-code for Algorithm 4 is as follows:

Algorithm 4

1. Find an initial solution by solving the multi-school bell time and scheduling problem (the MBSV, Model 5; the MBS, Model 7; the TMBSV, Model 8; Algorithm 3)
2. Initial solution feasible check and update; initialize *E, R, IR*
3. While (otter stop criteria not satisfied) do
   3.1 Column generation-based stochastic bus scheduling (Algorithm 4-1)
   3.2 Stochastic school bell time adjustment (Algorithm 4-2)
4. Return the best solution

In the initialization step, a multi-school bell time and scheduling problem is solved to provide an initial bell/dismissal time and scheduling solution for Algorithm 4. Preferably, the TD_MBSV (Model 8) and/or the TS_MBSV (Model 9) are solved with the consideration of time-dependent and/or normally distributed travel time and deadhead. Under the circumstance where the TD_MBSV (Model 8) and the TS_MBSV (Model 9) are hard to solve, the MBSV (Model 5), the MBS (Model 7) and Algorithm 3 can all be used to obtain an initial solution given the computing power and running time we have.

Since some bus routes may violate the Stochastic On-Time (SOT) and the MVT constraints (if initialized from the MBS, Model 7), we need to check the bus routes' feasibility in the initialization specifically. If a bus route violates the SOT or the MVT, we extract the last trip in that route and assign the last trip to a new bus route. Again, we check whether the remaining bus route is feasible. If not, we continue the extraction until all bus routes are feasible. We initialize a feasible bus route set (*R*) as all bus routes in this updated initial plan and the infeasible bus route set (*IR*) as the routes that violate either the SOT or the MVT constraints. Note that we have a preliminary school bell/dismissal times in the initial solution. The compatible trip set (*E*) is defined under this preliminary school bell/dismissal times. The main tasks of Algorithm 4 are 1) stochastic bus scheduling and 2) bell time adjustment. They are conducted iteratively. Algorithm 4 will terminate if one of the following stop criteria is satisfied:

I. Cannot find new beneficial school bell/dismissal time plans for more than it_{bt} iterations;

II. The best bus scheduling plan is not improved for more than it_{max} iterations (under the change of the bell/dismissal times).

5.3.1 Column generation-based stochastic bus scheduling (Algorithm 4-1)

The inputs of the stochastic scheduling problem include the current school bell/dismissal times, the compatible trip set (*E*), the feasible bus routes (*R*) and infeasible bus routes (*IR*). The notations used in Algorithm 4-1 are summarized in Table 11.

Using the similar argument in the Lemma 3 (section 4.3), we can get the solution to the chance-constrained stochastic problem by solving its corresponding

deterministic model while excluding all illegal bus routes that violate the stochastic chance constraints. Using this idea, we can re-formulate the problem as a Set Partitioned Multi-School Bus Scheduling Problem (PMS, Model 10). In the PMS (Model 10), we assume that all feasible bus route set (*AR*) are known. The PMS (Model 10) finds the best combination of the bus routes to serve all trips with the minimum cost (number of buses and total deadhead). The optimal solution to the PMS (Model 10) is the optimal solution to the chance-constrained problem as long as every route in the bus route set (*AR*) satisfies the SOT and the MVT constraint.

Table 11 Notation summary for Algorithm 4-1

Index and Set	
followTrip[t]	Set of trips could be served right after trip *t* on a bus
leadTrip[t]	Set of trips could be served right before trip *t* on a bus
Variable	
η_{r}	Binary variable equals to 1 if bus route *r* is used
$y_{t1,t2}$	Binary variable equals 1 if trip $t2$ is served after trip $t1$ on a bus
Parameters	
π_t	The dual variable associated the constraint in the master problem
a_{t}^{r}	Equals to 1 if trip *t* is served on bus route *r*
$dd_r, dd_{t1,t2}$	Total deadhead for bus route *r*, and deadhead for trip pair $(\mathrm{t1}, \mathrm{t2})$
tt_t	The travel time of trip t
$\alpha_B, \alpha_{\mathrm{D}}$	The coefficient of the number of buses (α_B) and the total deadhead (α_{D})
it_{cut}, it_{sub}, it_r	The number of iterations the master problem is solved (it_{cut}); The number of iterations that no new beneficial bus routes is found for the sub-problem (it_{sub}); The number of iterations that a route is not been used (it_r)
$it_{cut}^{max}, it_{sub}^{max}, it_r^{max}$	The maximum number of iterations for it_{cut}, it_{sub}, it_r
maxRLen	The desired maximum length of R
r_{act}	The number of iterations that a route $r \in R$ has not been used
ρ	The minimum probability that a trip is served on time

5.3.1.1 The PMS (Model 10) and the CMS (Model 11)

The Set Partitioned Multi-School Bus Scheduling Problem (PMS, Model 10) is defined as follows:

The PMS (Model 10)

$$\min z10 = \alpha_B \sum_{r \in AR} \eta_r + \alpha_D \sum_{r \in AR} E[dd_{\mathrm{r}}]\eta_{\mathrm{r}} \quad (5\text{-}36)$$

Subject to:

$$\sum_{\mathrm{r} \in AR} a_t^r \eta_r = 1, \forall t \in TP \quad (5\text{-}37)$$

$$\eta_r \in \{0,1\}, \forall r \in AR \quad (5\text{-}38)$$

Objective (5-36) minimizes the number of buses and the expected total deadhead. Constraints (5-37) regulate that each trip is served exactly once. Constraints (5-38) are binary domain constraints.

The major downside of this idea is that the complete list of all feasible bus routes (*AR*) is huge! Thus, the Column Generation (CG) algorithm is widely used to solve the set partition formulation. The CG simplified the problem by only using a restricted bus route set (*R*), which is much smaller than the *AR* ($|R| \ll |AR|$) and it contains "good" bus routes. The Column Generation-based Multi-School Bus Scheduling Problem (CMS Model 11) is defined as follows:

The CMS (Model 11)

Master problem

$$\min z11 = \alpha_B \sum_{r \in R} \eta_r + \alpha_D \sum_{r \in R} E[dd_{\mathrm{r}}]\eta_{\mathrm{r}} \quad (5\text{-}39)$$

Subject to:

$$\sum_{\mathrm{r} \in \mathrm{R}} a_t^r \eta_r \geq 1, \forall t \in TP \quad (5\text{-}40)$$

$$\eta_r \geq 0, \forall r \in R \quad (5\text{-}41)$$

Subproblem:

$$\min w = \alpha_B + \alpha_D \sum_{(t1,t2)\in E} E\left[dd_{t1,t2}\right] y_{t1,t2} - \sum_{t\in TP} \sum_{t2:(t,t2)\in E} y_{t,t2}\pi_t \quad (5\text{-}42)$$

Subject to:

$$\sum_{t1:(t1,t)\in E} y_{t1,t} = \sum_{t2:(t,t2)\in E} y_{t,t2}\,, \forall t \in TP \quad (5\text{-}43)$$

$$\sum_{(t1,t2)\in E} E\left[dd_{t1,t2}\right] y_{t1,t2} + \sum_{t\in TP} \sum_{t2:(t,t2)\in E} E[tt_t] y_{t,t2} \leq MVT \quad (5\text{-}44)$$

$$Stochastic\ On - Time\ Constraint \quad (5\text{-}45)$$

$$y_{t1,t2} \in \{0,1\}, \forall (t1,t2) \in E \quad (5\text{-}46)$$

The Column Generation (CG) algorithm separates the problem into the master problem and the subproblem. The master problem is the linear relaxation of the PMS (Model 10). The set partition is changed to the set cover problem (will be explained in section 5.3.1.4). The subproblem dynamically generates a new bus route given the dual prices of all trips (π_t) from the master problem. Whenever a subproblem is solved, the newly generated bus route is added to the bus route set (*R*) and the master problem is solved again using the augmented bus route set (*R*). The CG solves the master and subproblems iteratively until no new bus route with a negative reduced price (w defined in (5-42)) is found. The current solution is optimal because no other column would further reduce the objective in the master problem (5-39) if entering the basis.

Objective (5-39) in the master problem minimizes the number of buses and expected total deadhead. Constraints (5-40) regulate that each trip is served at least once (set cover). Constraints (5-41) are nonnegative continuous domain constraints. Constraints (5-40) relax constraints (5-37) because the bus route set (*R*) is not the complete list of all feasible bus routes now. There might be cases that serving every trip exactly once is infeasible using the current restricted bus route set (*R*). To

compensate for this problem, a solution refinement is implemented after the master problem is solved (will be explained in section 5.3.1.4).

In the subproblem, objective (5-42) minimizes the reduced dual prices (w) of a new bus route under the current dual prices of the trips (which corresponds to the dual prices of the constraints in the master problem). The reduced dual price consists of the cost of using a new bus and the cost of expected deadhead on that route minus the dual price of the served trips on that route. Constraints (5-43) find the trip serving sequence of the new bus route. Constraints (5-44) make sure that the expected vehicle time of the new bus route satisfies the maximum vehicle time constraints. Constraints (5-45) are the stochastic on-time constraints. It can be either be formulated as a set of Mixed Integer constraints as (5-19) to (5-24) or as a sample-based method, which will be discussed in the next subsection. Constraints (5-46) are binary domain constraints.

5.3.1.2 The procedure of Algorithm 4-1

The pseudo code of the framework of the stochastic bus scheduling is shown in Algorithm 4-1. The detailed procedure will be discussed from section 5.3.1.4 to 5.3.1.7.

Algorithm 4-1

1. Initialize $a_t^r \ \forall r \in R, \pi_t = 0$
2. While (inner stop criteria not satisfied) do
   - 2.1 Solve the master problem of the CMS (Model 10)
   - 2.2 Refine the master problem solution, update the current solution, R, IR, π_t, a_t^r
   - 2.3 Solve the subproblem of the CMS (Model 10) under the current π_t (Algorithm 4-1I or Algorithm 4-1G)
   - 2.4 Deactivate inefficient route
3. Return the solution; Go to Algorithm 4-2

We used the Column Generation algorithm in Algorithm 4-1 to solve the stochastic bus scheduling problem. Algorithm 4-1 iteratively solves the master and the subproblems. Because the subproblem only generates bus routes that satisfy the SOT

constraint, the final solution satisfies the stochastic chance constraint even if the master problem is a deterministic model and that the SOT is not explicitly considered in the master problem. Algorithm 4-1 will terminate if one of the following stop criteria is satisfied:

I. Cannot find a new bus route with a negative reduced dual price in the subproblem;

II. Cannot find new beneficial bus route for more than it_{sub} iterations.

5.3.1.3 An illustration of the column generation

A simple illustration of the Column Generation is shown in Figure 14. In this example, three trips need to be served, each of which has a unique start/departure time and travel time. Assume in the first iteration, five potential routes exist: bus route 1, 2, and 3 serve trip 1, 2, and 3, respectively. Bus route 4 serve trip 1 and trip 2 and bus route 5 serve trip 2 and trip 3. By solving the master of the CMS (Model 11), the optimal solution under the current potential bus routes is using route 3 and route 4 to serve all trips.

Given the dual price of each trip under the current optimal solution, the sub problem in the Column Generation finds a new potential bus route with the most negative reduced dual price. The new route is marked as route 6, which serves trip 1 followed by trip 3. If this route satisfies the Maximum Vehicle Time and the Stochastic On-Time Constraint, we add it into the potential bus route set and resolve the master problem. If the deadhead duration of route 6 is less than that of route 4, a new optimal solution is obtained. One bus serves trip 2 and another bus serves trip 1 followed by trip 3. The Column Generation repeats this procedure until the subproblem cannot find

a new route with a negative reduced dual price. The name, Column Generation, comes from the fact that in each iteration, a new column is generated and added into the master problem. In our case, each column represents a new bus route.

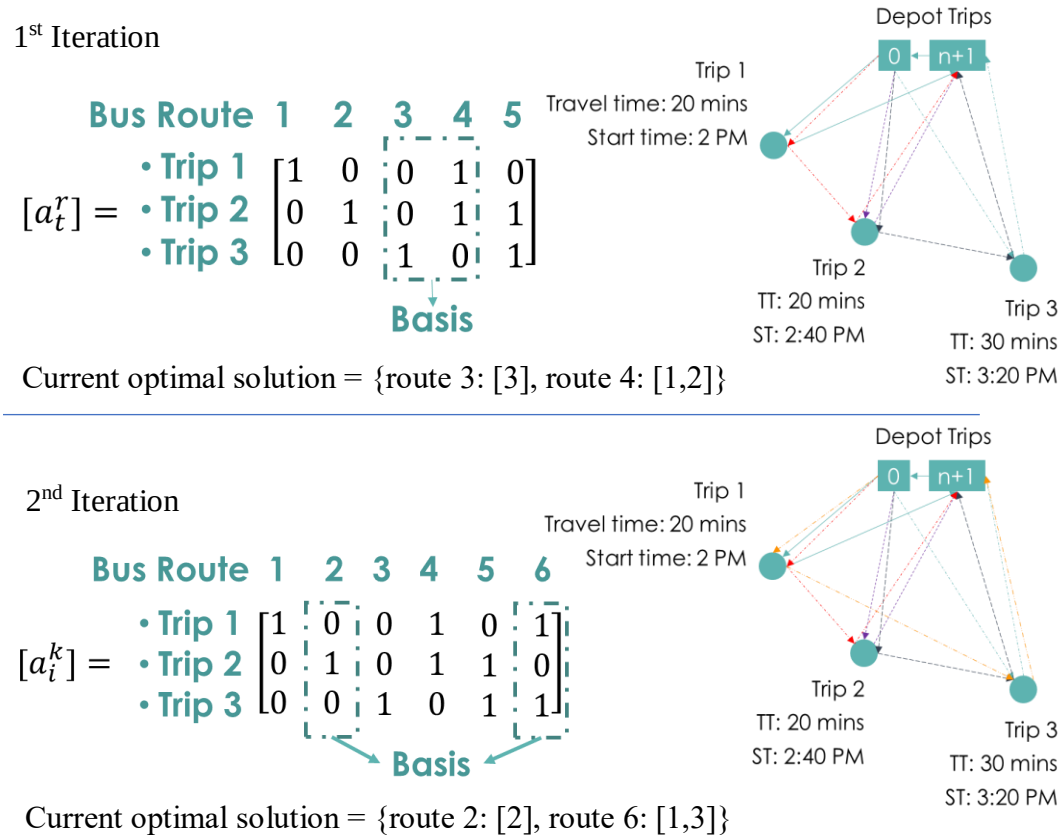


Figure 14 Illustration of the Column Generation

5.3.1.4 Master problem solution refinement

As explained in section 5.3.1.1, the master problem of the CMS (Model 11) is a relaxation of the PMC (Model 10). Therefore, it is possible that a trip is served twice or even multiple times. If this happens, we eliminate these multi-served trips from some of the bus routes and make sure this trip is only served once. The elimination is selected such that the saving (the number of buses and expected total deadhead) is maximized.

More precisely, if a trip is served multiple times on a one-trip bus route and a multi-trip route, we eliminate the one-trip route. If trip *t* is served multiple times in several multi-trip routes $\{r_1 = (\dots, t_1^+, t, t_1^-, \dots), r_2 = (\dots, t_2^+, t, t_2^-, \dots), \dots\}$, we only keep the trip in the bus route that minimizes the additional insertion cost (5-47) and eliminate the trip on all other routes:

$$\min_{r:t\subset r} dd_{t_r^+,t_r} + dd_{t_r,t_r^-} - dd_{t_r^+,t_r^-} \quad \forall \text{multi served } t \in TP \tag{5-47}$$

t_r^+, t_r^- is the preceding and successive trip of trip *t* on bus route *r*. The insertion cost of inserting trip *t* between t_r^+, t_r^- is defined as the deadhead from t_r^+ to *t* plus the deadhead from *t* to t_r^- minus the deadhead from t_r^+ to t_r^-.

In addition, the master problem of the CMS (Model 11) is a linear relaxation of the PMC (Model 10). This is for obtaining the dual prices associated with the constraints as well as improving the efficiency of the whole algorithm. However, the solution to the master problem of the CMS (Model 11) may not always be an integer solution. In case we obtain a fractional solution, we add the following cut to the master problem (Laporte, 1992):

$$\sum_{r\in R} \eta_r \geq \lceil \sum_{r\in R} \eta_r^* \rceil \tag{5-48}$$

where η_r^* is the optimal solution of η_r. After adding the cuts, we resolve the master problem again until finding an integer solution. As proved by Ryan and Foster (1981), the set partition problem with a balanced matrix (an 0-1 matrix that does not contain a submatrix that is the incidence matrix of any odd cycle), will yield integer solutions. Thus, only a few cuts are needed to reach integrality (Orloff, 1976). An illustration of an unbalanced matrix is shown in Figure 15.

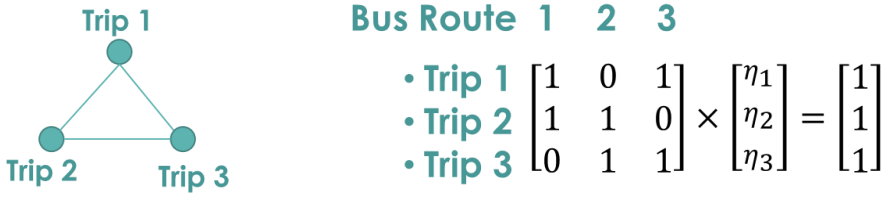


Figure 15 Illustration of an unbalanced matrix

In this example, three potential bus routes are examined. Bus route 1 visits trip 1 and trip 2, bus route 2 visits trip 2 and trip 3 and bus route 3 visits trip 1 and trip 3. Trip 1, 2, and 3 forms an odd cycle and matrix is the incident matrix of this odd cycle. Hence, this is an unbalanced matrix. This is the only unbalanced 0-1 three-by-three matrix (up to permutation of the rows and columns). Without the integrality constraint, one possible fractional solution is $\eta_1 = \eta_2 = \eta_3 = 0.5$. However, as long as the upper case does not happen, the master problem of the CMS (Model 11) will always yield an integer solution. We also set an upper bound for the number of iterations that the master problem is solved by it_{cut} in case of an excessive number of iterations.

5.3.1.5 An insertion-based heuristic for the subproblem

The subproblem can be solved as a shortest-path problem constrained in the same manner as the original problem (Laporte, 1992). However, the level of service constraint (the MVT) and the stochastic constraint (the SOT) make the subproblem very hard to solve. Thus, we develop heuristics algorithms (the Insertion and the GRASP) to solve the subproblem.

We present an Insertion-based heuristic (Algorithm 4-1I) to find several "good" bus routes simultaneously in contrast to solving the subproblem to optimality and obtain one bus route to enter the bus route set. Although the new bus routes from Algorithm 4-1I might not be the best under the current dual prices, more bus routes can

be explored in one iteration. Overall, it is much faster and might output a better solution. The pseudo-code for the Insertion-based is as follows:

Algorithm 4-1I

1. Initialize π_t, $\text{it}_{\text{sub}} = 0, R, IR, AR$
2. While ($it_{sub} \leq it_{sub}^{max}$) do
   - 2.1 Sort the trips in the descending order of the dual prices as sortTrip
   - 2.2 For (trip *t* in the top three in sortTrip) do
     - 2.2.1 searchForward(t, it_{sub}), update *R*, *IR*
     - 2.2.2 searchBackward(t, it_{sub}), update *R*, *IR*
     - 2.2.3 searchBoth(t, it_{sub}), update *R*, *IR*
   - 2.3 If (new routes are generated) then
     - 2.3.1 Return *R*, *IR*; Go to Algorithm 4-1
   - 2.4 Else:
     - 2.4.1 Add white noise to π_t given it_{sub}; $it_{sub} += 1$; Go to step 2
3. Return *R*, *IR*; Go to Algorithm 4-2

The Insertion-based algorithm first finds the trip with the highest dual price (step 2.1). For each high dual-priced trip, Algorithm 4-1I has three bus route construction methods: the searchForward, the searchBackward, and the searchBoth. The searchForward (step 2.2.1) uses the highest-dual-priced trip as the first trip on a bus route and sequentially adds succeeding trips until no further trip insertion is feasible due to the trip compatibility, the maximum vehicle time constraints or the stochastic on-time constraints. The searchBackward (step 2.2.2) is similar to the searchForward except using the highest-dual-priced trip as the last trip in a bus route and sequentially adds preceding trips. The searchBoth (step 2.2.3) performs the searchForward and the searchBackward alternatively. It first adds a succeeding trip then adds a preceding trip and so on.

Algorithm 4-1I repeats the route generation process (the searchForward, the searchBackward, and the searchBoth) for the top three trips with the highest dual prices (step 2.2). If one generated bus route has a negative reduced dual price (w) and does

not exist in the bus route set *R* (step 2.3), we augment this new bus route to the *R*, terminate the subproblem and resolve the master problem in Algorithm 4-1. However, if no new bus route is generated (step 2.4), we add white noise to the dual prices of the trips and repeat the search process in step 2. The white noise (of a trip) is randomly generated from a normal distribution with the mean of 10% of its dual price. Its standard deviation increases with the number of iterations (when no new route is added). In another scope, if the number of iterations exceeds $\mathrm{it}_{\mathrm{sub}}^{\max}$, we also terminate the algorithm and directly go to bell time adjustment (Algorithm 4-2) instead of the stochastic bus scheduling (Algorithm 4-1).

The pseudo-code the searchForward is as follows. Take the searchForward for instance, given a high dual priced trip *t*; the searchForward constructs a new bus route *r* starting with trip *t* in step 1. Then, it sequentially adds successive trips to the end of the new bus route *r*. In every iteration, it sorts the compatible successive trip $t2 \in \mathrm{followTrip}[r_{last}]$ (r_{last} is the last trip in route r) in the descending order of the expected value of their deadhead $E[dd_{r_{last},t2}]$ (step 2.1). The searchForward goes through the sorted list of the successive trips. The tentative augmented bus route augR is the route *r* augmented by one successive trip $\mathrm{t2} \in \mathrm{followTrip}[r_{last}]$. There are in total five different cases of the augmented route augR:

I. (Step 2.1.2) If augR exists in AR and is inactive, reactive augR in R and reset it_{r} to zero (inactive and it_r will be explained in section 5.3.1.7;

II. (Step 2.1.3) If augR exists in R and is active, we accept the tentative augment route ($r = \mathrm{augR}$) with the probability $1/\exp(it_{sub})$; or go to step 2.1 to search

for the next possible successive trip (rand is a uniform random generator between 0 and 1);

III. (Step 2.1.4) If augR exists in IR, go to step 2.1 to search for the next possible successive trip;

IV. (Step 2.1.5) If augR satisfies the MVT and the SOT constraints, we accept the tentative augment route ($r = \text{augR}$) with the probability $1/\exp(w_{avg} - w_{augR})$; or go to step 2.1 to search for the next possible successive trip;

V. (Step 2.1.6) If augR does not satisfy the MVT or the SOT, add augR in IR (step 2.1.6.1) and go to step 2.1 to search for the next possible successive trip.

The searchForward(t, it_{sub}, followTrip**)**

1. Initialize a new route *r* starting with trip *t*
2. While (∃ followTrip$[r_{last}]$) do
   - 2.1 For ($t2$ in the descending order of $E[dd_{r_{last},t2}] \forall t2 \in$ followTrip$[r_{last}]$) do
     - 2.1.1 Augment t2 to the end of r as augR
     - 2.1.2 If (augR in *R* and inactive) then
       - 2.1.2.1 Reactive augR, reset $it_r = 0$, $r =$ augR; Go to step 2
     - 2.1.3 Else-if (augR in R and active) then
       - 2.1.3.1 If (rand $\leq 1/\exp(it_{sub})$) then
         - 2.1.3.1.1 $r =$ augR; Go to step 2
       - 2.1.3.2 Else:
         - 2.1.3.2.1 Go to step 2.1
     - 2.1.4 Else-if (augR in IR) then
       - 2.1.4.1 Go to step 2.1
     - 2.1.5 Else-if (augmented route satisfies the MVT and the SOT) then
       - 2.1.5.1 If (rand $\leq 1/\exp(w_{avg} - w_{augR})$) then
         - 2.1.5.1.1 $r =$ augR; Go to step 2
       - 2.1.5.2 Else:
         - 2.1.5.2.1 Go to Step 2.1
     - 2.1.6 Else:
       - 2.1.6.1 Add augR to IR; Go to step 2.1
   - 2.2 If ($t2$ is the last in the sorted list of successive trips of r_{last}) then
     - 2.2.1 Return *R*, *IR*; Go to step 2.2.2 in Algorithm 4-1I
3. Return new route *r*; Go to step 2.2.2 in Algorithm 4-1I

The searchForward repeats this process until there is no successive trip for the new generated bus route *r* (step 2) is acceptable (step 2.2). Algorithm 4-1I then tries to find new bus routes by the searchBackward and the searchBoth. The process of the searchBackward and searchBoth are similar to the searchForward. Thus, the pseudo-code for the searchBackward and the searchBoth are omitted in the dissertation.

Note that Algorithm 4-1I incorporates the expected deadhead in the route generation procedure along with the dual price of the trips. This helps to overcome the extremely unbalanced distribution of the dual prices in some scenarios. Sometimes, a few trips will take most of the total dual price and the rest of them only occupy a small portion of the total dual price. Under this circumstance, purely estimating the reduced dual price may not be efficient. The consideration of the expected deadhead and the white noise helps to mitigate the extremely unbalanced dual price distribution if it happens. The same logic applies to the following Greedy Randomized Adaptive Search Procedure.

5.3.1.6 A GRASP heuristic for the subproblem

Now, we propose a Greedy Randomized Adaptive Search Procedure (GRASP)-based heuristic (Algorithm 4-1G) to solve the subproblem. The GRASP was adopted by Kontoravdis and Bard (1995), Schittekat et al. (2013) and Faraj et al. (2014) to solve the school bus routing and scheduling problem. The pseudo-code of Algorithm 4-1G is shown as follows:

Algorithm 4-1G

1. Initialize π_t, $\mathrm{E}\left[dd_{t1,t2}\right], E, R, IR, it_r, \mathrm{it_{sub}} = 0$
2. Initialize $remTP = TP, newR = \mathrm{ftTP} = \mathrm{lsTP} = \Phi$
3. While ($it_{sub} \leq it_{sub}^{max}$) do
   - 3.1 Construct and sort saving list (π_t, $\mathrm{E}\left[dd_{t1,t2}\right], E$) in the descending order
   - 3.2 For (($t1, t2$) in the sorted saving list) do
     - 3.2.1 If (t1 in remTP and t2 in remTP) then
       - 3.2.1.1 augR = [t1, t2]
       - 3.2.1.2 new route update(augR, remTP, newR, ftTp, lsTp, $\mathrm{it_r}$, it_{sub})
     - 3.2.2 Else-if (t1 in lsTP and t2 in remTP) then
       - 3.2.2.1 Augment t2 to the end of the route that t1 belongs to as augR
       - 3.2.2.2 new route update(augR, remTP, newR, ftTp, lsTp, $\mathrm{it_r}$, it_{sub})
     - 3.2.3 Else-if (t1 in remTP and t2 in ftTP) then
       - 3.2.3.1 Augment t2 to the first of the route that t1 belongs to as augR
       - 3.2.3.2 new route update(augR, remTP, newR, ftTp, lsTp, $\mathrm{it_r}$, it_{sub})
     - 3.2.4 Else:
       - 3.2.4.1 Go to step 3.2
   - 3.3 If ($newR$ contains route does not in R) then
     - 3.3.1 Add new routes from newR to R
     - 3.3.2 Return *R*, *IR*; Go to step 2.4 in Algorithm 4-1
   - 3.4 Else:
     - 3.4.1 add white noise to π_t given it_{sub}; it_{sub}+= 1; Go to step 2
4. Return *R*, *IR*; Go to Algorithm 4-2

The new route update(augR, remTP, newR, ftTp, lsTp, $\mathrm{it_r}$, $\mathrm{it_{sub}}$)

1. If (augR in R and inactive) then
   - 1.1 Reactive augR and reset $\mathrm{it_r}$; Go to step 3.2 in Algorithm 4-1G
2. Else-if (augR in R and active) then
   - 2.1 If (rand $\leq 1/\exp(it_{sub})$) then
     - 2.1.1 Add augR to newR and delete the old route in newR which has trips in augR
     - 2.1.2 Update remTP, ftTp, lsTp
   - 2.2 Go to step 3.2 in Algorithm 4-1G
3. Else-if (augR in IR) then
   - 3.1 Go to step 3.2 in Algorithm 4-1G
4. Else-if (augR satisfies the MVT and the STO) then
   - 4.1 If (rand $\leq 1/\exp(w_{avg} - w_{augR})$) then
     - 4.1.1 Add augR to newR and delete the old route in newR which has trips in augR
     - 4.1.2 Update remTP, ftTp, lsTp
   - 4.2 Go to step 3.2 in Algorithm 4-1G
5. Else
   - 5.1 Add augR to IR; Go to step 3.2 in Algorithm 4-1G

The GRASP is a randomized version of the CWSM (Clarke and Wright Saving Method). The CWSM constructs trips by iteratively merging pairs of bus stops into the same trip based on the descending order of their savings. The saving of a pair of bus stops is the travel time reduction if serving two stops into the same trip from serving them separately on two trips. The GRASP randomizes the CWSM by adding some white noise to the saving list and/or accepting each merge with some probability. The white noise is usually randomly drawn from a random distribution.

Algorithm 4-1G initializes a remaining trip set (remTP) as the set of all trips. In each iteration of Algorithm 4-1G, the newly generated route set (newR), the first trips in each route in newR (ftTP) and last trips in each route in newR (lsTP) are all re-initialized as empty sets (step 2). Note that the feasible route set (R) and the infeasible route set (IR) contain all bus routes explored throughout all iterations and the newR, ftTP, lsTP are within one iteration. The purpose of separating these sets is to keep the subproblem (route generation) relatively simple. But if the same route is generated again in another iteration, we can easily check its feasibility in R and IR without calculating the MVT and the SOT from the historical sample data.

In the main route generation loop (step 3), Algorithm 4-1G constructs the saving list of every pair of compatible trips based on their descending order of the savings (step 3.1). The saving (5-49) is defined as the sum of the reduced price of two routes that serve these two trips individually minus the reduced price of the route that serves these two trips on one route in an optimal manner:

$$\text{saving}(\text{t1},\text{t2}) = w(\text{t1}) + w(\text{t2}) - w(t1,t2)\ \forall (t1,t2) \in E \tag{5-49}$$

where $w(t1, t2)$ is the reduced price of the route that serves trip $t1, t2$. The higher this saving is, the more beneficial the merging of these two trips is. Algorithm 4-1G goes through the saving list to construct new bus routes (step 3.2). For a specific pair of trips $(\mathrm{t1}, \mathrm{t2})$, there are four possible conditions:

I. Trip $\mathrm{t1}, \mathrm{t2}$ are both not served (step 3.2.1): we set the tentative new route (augR) to serve trip $\mathrm{t1}, \mathrm{t2}$ (step 3.2.1.1), update the newR (step 3.2.1.2) given the feasibility of augR (will be explained in the *new route update*);

II. Trip t1 is the last trip of a route *r* in newR, and trip t2 is not served (step 3.2.2): augment the trip t2 to the end of t1 on route *r*, and mark this tentative new route as augR (step 3.2.2.1). The new augR is checked in the same manner (step 3.2.2.2);

III. Trip t1 is not served, and trip t2 is the first trip of a route *r* in newR (step 3.2.3): augment the trip t2 to the leading position of t1 on route *r*, and mark this tentative new route as augR (step 3.2.3.1). The new augR is checked in the same manner (step 3.2.3.2);

IV. If t1, t2 do not satisfy any of the above conditions, the algorithm goes to step 3.2 to check the next pair of trips in the saving list.

Given a tentative newly augmented route augR, Algorithm 4-1G checks whether augR follows any of the following cases (*new route update*):

I. augR exists in the feasible bus routes set (R) and is inactive: the algorithm reactivates route augR and resets the number of iterations not used ($\mathrm{it_r}$) to zero and then goes back to step 3.2 to continue the search on the saving list;

II. augR exists in the feasible bus routes set (R) and is active: with a certain probability, the algorithm accepts augR and uses augR to replace the old route in newR that contains the same trip as augR (if it exists) and updates remTP, ftTP, lsTP accordingly; then it goes back to step 3.2;

III. augR exists in the infeasible bus routes set (IR): the algorithm rejects this merging and goes back to step 3.2;

IV. augR satisfies the MVT and the SOT constraint: with the probability of $\exp(w_{avg} - w_{augR})$, the algorithm accepts augR and uses augR to replace the old route in newR that contains the same trip as augR (if it exists) and updates remTP, ftTP, lsTP accordingly; then it goes back to step 3.2;

V. augR does not satisfy either the MVT or the SOT constraint: the algorithm rejects this merging, adds augR to the infeasible bus routes set (IR) and goes back to step 3.2.

After looping through the entire saving list (step 3.2), Algorithm 4-1G has a preliminary set of new bus route set newR. In step 3.3, if there exist some routes in newR that are not in the feasible routes set (R), the algorithm adds them to the R. With the newly updated feasible route set (R) and infeasible route set (IR), Algorithm 4-1G goes back to step 2.4 in Algorithm 4-1, and the next iteration of the stochastic scheduling problem continues. However, if no new route is generated (step 3.4), white noise is added to the dual prices. The white noise is randomly generated from a normal distribution with the mean of 10% of its dual price. Its standard deviation increases with the number of iterations (when no new route is added). Then the algorithm goes back to step 2 to re-initialize remTP, newR, ftTP, and lsTP and re-generate the bus

routes. If after it_{sub}^{max} iterations, no new route is generated, Algorithms 4-1G and 4-1 (the stochastic scheduling) terminate and the process goes to Algorithm 4-2 to solve the bell time adjustment problem.

5.3.1.7 Inefficient route deactivation

Although the master problem in the CMS (Model 11) is quite efficient, it might still take some time to solve for a large-scale problem size, especially considering the problem size for the master problem could be as large as the all possible routes set (AR), which could be exponentially large as the trip set. Moreover, the master problem would have a higher chance to output a fractional solution with a large route pool size. Thus, we restrict the size of the feasible route set (R) much smaller than AR ($|R| \ll |AR|$). When the route (R) size exceeds our designed maximum size, we deactivate some "useless" bus routes and store these inactive routes in another set. The "useless" bus routes are defined as the bus routes that have not been used in a number iteration ($\mathrm{it}_{\mathrm{r}}^{max}$). Also, if a route serves a trip that no other routes in R will serve, this route cannot be deleted since such deletion would lead to the infeasibility of the master problem. The choice of the desired route set (R) size and the number of iterations to deactivate a trip is subject to the characteristics of the specific problems. The inactive route will be activated if the route is generated again in the sub problem.

5.3.2 Stochastic school bell time adjustment (Algorithm 4-2)

The Bell Time Adjustment Algorithm (Algorithm 4-2) aims to optimize the school bell times such that more feasible bus routes exist, which could further benefit the stochastic scheduling problem. Instead of checking all possible bus route (AR), we have a potential good route pool that consists of the feasible route set (R) and the

infeasible bus route (IR) from the scheduling problem. The infeasible routes are included because these routes are still "good" routes since they are generated in a greedy fashion in the subproblem. They are "good" routes in terms of small deadhead and/or negative reduced dual price. The core idea of Algorithm 4-2 is to change the bell times such that more infeasible bus routes (in IR) become feasible and as little as feasible routes (in R) become infeasible after the adjustment.

The pseudo-code of Algorithm 4-2 is presented below. Algorithm 4-2 first initializes the feasible bus route set (R), the infeasible bus route set (IR), the current school bell times (SB). The solvedSB contains the school bell times at which the bus scheduling problem is solved. The number of iterations of bell times adjustment ($\mathrm{it}_{\mathrm{bt}}$) is set to zero (step 1). Then Algorithm 4-2 sorts the schools in the order such that if the school's bell time adjusted, there is a higher chance that more feasible routes would exist (step 2). The school for which the corresponding trips occur the most in IR would be the top one the sorted list.

Algorithm 4-2

1. Initialize R, IR, SB, solvedSB, $\mathrm{it}_{\mathrm{bt}} = 0$
2. Sort the schools in the descending order of their occurrence that their associated trip appear in IR
3. While ($\mathrm{it}_{\mathrm{bt}} \leq \mathrm{it}_{\mathrm{bt}}^{\max}$) do
   - 3.1 For (sch in the sorted school list) do
     - 3.1.1 Change sch's bell time to maximize the number of feasible bus route of R, IR with some probability
     - 3.1.2 If (new bell time is not in solvedSB) then
       - 3.1.2.1 If ($\mathrm{rand} \leq \exp(\text{net increased feasible routes}/\mathrm{it}_{\mathrm{bt}})$) then
         - A. Update SB, solvedSB, R, IR; Go to step 3 in Algorithm 4
     - 3.1.3 Go to step 3.1
   - 3.2 Add white noise to the sorted school list ($\mathrm{it}_{\mathrm{bt}}$); $\mathrm{it}_{\mathrm{bt}} \mathrel{+}= 1$; Go to step 2
4. Return *R*, *IR*, SB, solvedSB; Go to step 4 in Algorithm 4

In the main loop (step 3), Algorithm 4-2 goes through the schools in the sorted list. For each school, its bell time is changed such that the number of feasible routes under the new bell times is maximized. To make the search step manageable, Algorithm 4-2 only checks the routes in R and IR (step 3.1). If the new proposed bell time is not solved before (step 3.1.2), with the probability $\exp(\text{net increased feasible routes}/\text{it}_{\text{bt}})$, Algorithm 4-2 accepts the newly proposed bell times and return to step 3 in Algorithm 4 with the updated solvedSB, R, IR. On the contrary, if the new bell time is solved before or if it decreases the net increased feasible routes (with some probability), Algorithm 4-2 goes back to step 3.1 to search for the next school (step 3.1.3). If Algorithm 4-2 reaches the end of the school list without accepting any new bell times (step 3.2), it randomly switches the sequence of some schools in the sorted school list and repeats the search procedure in step 3 with the incremented iteration count (it_{bt}). The number of schools switched increases with the number of iterations (when no new bell time is accepted). If Algorithm 4-2 cannot find any bell times worth solving, Algorithm 4 is terminated, and the best solution is reported.

One important merit of Algorithm 4 is that all the route-wise constraints like the maximum vehicle time, and the stochastic on-time constraint are taken care of using a sample-based feasibility check in the subproblem of the Column Generation-based stochastic bus scheduling (Algorithm 4-1). The benefits of the sample-based route-wise feasibility check are:

- The route-wise constraints are explicitly checked in the subproblem. Such constraints are relaxed in the master problem. Thus, the master problem always remains relatively easy;

- Other route-wise constraints can be easily added into the subproblem, for instance, the minimum travel time of a route. We can prevent a short and inefficient route from entering the master problem and this route will never be used in the final solution;
- Even if a route violates the route-wise constraint, we still store this route and try to make it feasible in the bell time optimization. It provides more information to the bell time optimization; thus, this constraint feasibility check will never be wasteful;
- On the other hand, Algorithm 4 can be downgraded to solve the deterministic problem by dropping the stochastic on-time constraint. After the downgrade, it can serve as a substitute to Algorithm 3 to solve the multi-school bell time and bus scheduling problem. This is especially beneficial when many routes violate the Maximum Vehicle Time Constraint because Algorithm 3 takes too much time to solve.

5.4 Chapter summary

In this chapter, three models were proposed to solve the school bell time and bus scheduling problem with considerations of traffic congestion and uncertainty of the travel times. The Time-Dependent Multi-School Bell Time and Bus Scheduling Problem with Maximum Vehicle Time Constraints (TD_MBSV, Model 8) assume that actual travel time of a trip depends on its departure time and that the travel time at a departure time is fixed. The Trip-based Stochastic On-Time Constrained Multi-School Bell Time and Bus Scheduling Problem with Maximum Vehicle Time Constraints (TS_MBSV, Model 9) assume that the actual travel time of a trip is a random variable,

which is normally distributed and is independent of its departure time. Both the TD_MBSV (Model 8) and the TS_MBSV (Model 9) are converted into Mixed Integer Programming models that can be directly solved by commercial solvers.

Then, a Column Generation-based algorithm (Algorithm 4) was proposed to efficiently solve the same problem without any assumption of travel time. An insertion and a GRASP based heuristic algorithm were proposed to solve the subproblem in Algorithm 4. The benefits of Algorithm 4 are 1) it is more efficient than solving the MIP models using exact methods; 2) it considers the impact of the departure time on trips' travel time; 3) it considers the uncertainty of travel time even at a fixed departure time; 4) it directly works with the empirical distribution of the observed data without any assumption regarding the distribution of the travel time.

Chapter 6 Computational Experiment

The entire school bus planning problem is a complicated problem. Thus, the computational experiments on the entire problem would be very difficult to analyze. Moreover, the MBRSV (Model 1) is solved by solving a set of decomposed subproblems. Therefore, the experiments are designed to test the performance of each subproblem including the correctness and efficiency of the formulations and the algorithms.

6.1 Experiment 1: test on Algorithm 1

6.1.1 Experiment target

Algorithm 1 involves three layers of model manipulations: 1) the relaxation from the MBRSV (Model 1) to the MBRS (Model 2) and the MRSV (Model 5); 2) the transformation from the MBRS (Model 2) to the MRS (Model 3); and 3) the decomposition of the MRS (Model 3) into many compatibility-considered single-school bus routing problems (the SRC, Model 4; the SRCA, Model 4'; and the SRCW, Model 4"). Given the Lemma 1, the optimal solution to the MBRSV (Model 1) can be obtained by solving the MBRS (Model 2) if the solution to the MBRS (Model 2) satisfies the MVT constraints. The MVT constraints are then taken care of in the MBSV (Model 5). Therefore, the top-layer relaxation is obviously valid. Given Lemma 2, the transformation from the MBRS (Model 2) to the MRS (Model 3) holds. These top two layers of model manipulation are tested in experiment 1-1 in section 6.1.3. And the bottom layer decomposition from the MRS (Model 3) into many decomposed compatibility-considered single-school bus routing problems is tested in experiment 1-

2 in section 6.1.4. The sensitivity analysis of the SRC (Model 4) is conducted in experiment 1-3 in section 6.1.5. Section 6.1.6 analyzes the marginal cost of a school bus in terms of the total vehicle time. The procedure to generate all the test problems are specified in section 6.1.2.

6.1.2 Test data generation

In this chapter, apart from the benchmark problems, all the deterministic random test problems are generated in the following way. For each test problem, the problem size, the number of schools and number of stops are given. All nodes (including stops and schools) are assumed to be located within a 2-dimensional square with a length of 20 miles (105,600 ft). The locations of all nodes are designated using their longitude and latitude, which are both randomly generated as a uniformly distributed random variable between 0 and 105,600. Then, the k-means algorithm is applied to group these nodes together with the number of clusters (k) equal to the number of schools. The closest nodes to each cluster centroid are selected to be the school locations, and the rest of the nodes in each cluster are treated as bus stops for the respective school. One depot is assumed to be located in the middle of this 2-dimensional square. The number of students at each stop is randomly generated as a uniformly distributed random variable between 1 and 20. The buses are assumed to have the same capacity of 66. The preliminary bell/dismissal times for the schools are uniform random variables between 12:00 PM and 16:00 PM and they are integer multiples of 15 minutes (i.e., 2:00 PM, 3:15 PM). The buses are assumed to run at the constant speed of 20 miles per hour. The distance is Euclidean. The Additional Allowed

Trips (AAT) is assumed to be equal to the Minimum Number of Trips (MNT) for every school (see section 3.2).

6.1.3 Experiment 1-1: model relaxation and transformation

There are a few scopes of assessment of the integrated models namely the MBRSV (Model 1), the MBRS (Model 2) and the MRS (Model 3):

- Solvability: whether these models can be solved within a reasonable amount of time for at least a small-sized problem;
- Correctness: whether these three models output the correct solutions;
- Efficiency: whether the running time to solve these models using exact methods is reasonable.

Experiment 1-1 testes the performance of these models on two small-sized randomly generated problems using the following methods:

1) Model 1: Directly solve the MBRSV (Model 1) to optimality using a commercial solver;
2) Model 2: Directly solve the MBRS (Model 1) to optimality using a commercial solver;
3) Model 3: Directly solve the MRS (Model 3) to optimality using a commercial solver.

Model 1 and Model 2, which have the ability to optimize the school bell time, solve three bell time adjustment instances for each test problem. The maximum bell time adjustment (postpone from the current bell time) is one of the following: 0 minutes, 30 minutes or 60 minutes. The results are shown in Table 12.

The first observation of the result is that all three models are solvable. They can be solved in a reasonable amount of time. Apart from the solvability, the correctness of these models is our main concern. We need to show two things: 1) these models can find a feasible solution if it exists; 2) the solutions that these models find are the optimal solution. We manually check the solutions generated by solving these models, and they are all meaningful and feasible solutions that satisfy all the constraints. Also, the fact that the optimal gap reaches zero means that these solutions are optimal.

Table 12 Result summary for experiment 1-1 (Models 1, 2, and 3)

Scenario		**1**			**2**		
Problem size		2 schools /20 stops			2 schools /30 stops		
Maximum Bell time adjustment (mins)		0	30	60	0	30	60
Model 1	**NOB**	**4**	**4**	**3***	**5**	**5**	**4***
	NOT	4	4	4	5	5	5
	TVT	309.5	309.5	310.7	370.1	370.1	370.2
	RT	**2.4**	**125.4**	**2554**	**284.3**	**36000**	**36000**
	Gap	0	0	0	0	19.99	24.9
Model 2	**NOB**	**4**	**4**	**3***	**5**	**5**	**4***
	NOT	4	4	4	5	5	5
	TVT	309.5	309.5	310.7	370.1	370.1	370.2
	RT	**2.5**	**11.4**	**11.6**	**37.9**	**162.3**	**1060.2**
Model 3	**NOB**	**4**	-		**5**	-	
	NOT	4			5		
	TVT	309.5			370.1		
	RT	**1.3**			**38.8**		

Note: ***** minimum number of buses; **Gap**: Optimal gap (%); **NOB**: Number of buses; **NOT**: Number of trips; **TVT**: Total vehicle time (minutes); **RT**: Running time (second)

In terms of efficiency, the MRS (Model 3) is the most efficient model, the MBRS (Model 2) is the second, and the MBRSV (Model 1) is the slowest. These running times are consistent with the model comparison in section 4.4. On the other hand, the MRS (Model 3) does not have the ability to adjust the bell times. Hence, in the instances where the bell time can be postponed up to 60 minutes, Model 1 and

Model 2 can find better solutions using a fewer number of buses than what Model 3 finds. This shows the benefit of adjusting the bell time in the MBRSV (Model 1) and the MBRS (Model 2).

Furthermore, the results demonstrate the tradeoff between model integration and efficiency. The MBRSV (Model 1) is the most integrated model, which makes it the hardest model. It has the highest solution quality but also requires the longest solution time. On the contrary, the MRS (Model 3) is the least integrated model among these three, which makes it the simplest model. It requires the least solution time, but the solution quality is the worst among these three models.

6.1.4 Experiment 1-2: model decomposition

There are four major questions that need to be answered regarding the decomposition from the MRS (Model 3) into many SRCs (Model 4).

i. Can the decomposed problems provide solutions as good as what the MRS (Model 3) finds?
ii. Is the proposed decomposition method better than the traditional decomposition methods?
iii. How much running time can the decomposed methods save in comparison to solving the MRS (Model 3) directly?
iv. What is the rational coefficient range in the SRC (Model 4)? Is the SRC (Model 4) sensitive to the adjusted coefficients?
v. Which decomposed problem (the SRC, Model 4; the SRCA, Model 4'; the SRCW, Model 4") is the best in terms of the solution quality and efficiency?

6.1.4.1 Competing methods

The newly proposed decomposition method in Algorithm 1 is tested in comparison to the traditional decomposition methods. The traditional decomposition methods simply decompose the MRS (Model 3) into the Single-School Bus Routing Problem (without compatibility), defined as the SR (Model 12) and the Multi-School Bus Scheduling Problem (MS, Model 13).

The SR (Model 12):

$$\min z12 = \sum_{t \in TP[k]} ta_t \quad (6\text{-}1)$$

$$\min z12' = \alpha_T \sum_{t \in TP[k]} tt_t \quad (6\text{-}2)$$

$$\min z12'' = \alpha_N \sum_{t \in TP[k]} ta_t + \alpha_T \sum_{t \in TP[k]} tt_t \quad (6\text{-}3)$$

Subject to

Constraints (3-3) to (3-11) (Routing constraints for school k)

Constraints (3-21) and (3-37) (MRT constraints for school k)

Constraints (3-23) (Efficiency constraints for school k)

Constraints (3-25) to (3-29) (Binary domain constraints for school k)

Constraints (3-31) to (3-34) (Nonnegative continuous domain constraints for school k)

The objective (6-1) - (6-3) are the most widely used routing objective for the routing problem: minimize the number of trips (MinN, (6-1) (Bodin and Berman, 1979) or minimize the total ride time (MinTT, (6-2) (Schittekat et al., 2013; Faraj et al., 2014; Kinable et al., 2014) or a combination of these two (MinNT, (6-3)) (Díaz-Parra et al., 2012). After solving the routing problems (with different surrogate objectives), the scheduling problems are solved using the trips generated from the routing problems. The objective of the scheduling problem is to minimize the number of buses and the

total deadhead. These three methods (using different routing objectives) are considered as traditional research methods.

The scheduling problem is very similar to the MBS (Model 7) except that the school bell time (sb_k) is a parameter rather than a variable. Also, since school bell times are known, the compatible trip pairs (E) can be estimated for certain. The problem reduces to an assignment problem, which can be solved very efficiently using the Hungarian (Munkres) algorithm with a time complexity of $O(n^3)$.

The MS (Model 13):

$$\min z13 = \alpha_B \sum_{t \in TP} y_{SDT,t} + \alpha_D \sum_{(t1,t2) \in E} dd_{t1,t2} \times y_{t1,t2} \quad (6\text{-}4)$$

Subject to

Constraints (4-17) to (4-19)

To make a fair comparison, the same scheduling problems are also solved using the trips generated by solving the MRS (Model 3) or the SRC (Model 4). The number of buses and the total vehicle time are the major criteria to evaluate the solution quality.

6.1.4.2 Experiment setup

Eight randomly generated test problems are solved using five methods. The approaches are as follows:

1. **Model 3**: Directly solve the MRS (Model 3) using commercial solver;
2. **MinN**: Traditional routing and scheduling decomposition method, use minimizing the number of trips (6-1) as the routing objective;
3. **MinTT**: Traditional routing and scheduling decomposition method, use minimizing the total ride time (6-2) as the routing objective;

4. **MinNT**: Traditional routing and scheduling decomposition method, use minimizing the number of the trips and total ride time (6-3) as the routing objective;
5. **Alg1-1**: Solve a set of SRCs (Model 4) with $\alpha_N = 1e5, \alpha_C^{OA} = 5e4$ and the scheduling problem using the trips from the SRCs (Model 4);
6. **Alg1-2**: Solve a set of SRCAs (Model 4') with $\alpha_N = \alpha_C = 1e5$ and the scheduling problem using the trips from the SRCAs (Model 4');
7. **Alg1-3**: Solve a set of SRCWs (Model 4") with $\alpha_N = 1e5, \alpha_C^{CA} = 9e4$ and the scheduling problem using the trips from the SRCWs (Model 4");

The MRS (Model 3) and all subproblems are solved by the commercial solver Gurobi on a computer with Intel® Core™ i7-840 CPU, 2.93 GHz with 8 GB RAM. The code is written in Python 2.7. The other parameters are $\alpha_B = 1e5, \alpha_T = 1, \alpha_D = 0.5, MRT = 90\ minutes$.

6.1.4.3 Comparison of the results

The solutions are summarized in Table 13. In theory, the solution from Model 3 (if solved to optimality) should be the best among all solutions in terms of all criteria. It is demonstrated in the experiments as Model 3 found the solutions using the minimum number of buses for all scenarios and the minimum vehicle time for the scenarios if solved to optimality. Based on the number of times that each approach found the best solution (with respect to the minimum number of buses), the approach which has the highest solution quality is: Model 3 (8 times) = Alg1-3 (8 times) > Alg1-2 (7 times) > Alg1-1 (4 times) = MinNT (4 times) > MinTT (1 time) = MinN (1 time).

Table 13 Result summary for experiment 1-2 (decomposition in Algorithm 1)

Scenario		1	2	3	4	5	6	7	8
# of schools		2	4	6	8	10	15	20	30
# of stops		20	40	60	80	100	150	200	300
Model 3	**NOB**	**4***	**4***	**7***	**9***	**8***	**8***	**11***	**16***
	NOT	4	8	11	18	26	35	44	67
	TVT	**310**	**437**	**583**	**720**	**814**	**1089**	**1259**	**1715**
	RT	1.27	18.73	1192	2585	1585	23795	5697	36000
	Gap	0	0	0	0.05	0.03	0.02	0.01	12.31
Min N	**NOB**	**4***	**5**	**8**	**11**	**10**	**12**	**17**	**22**
	NOT	4*	8	11	18	21	30	41	60
	TVT	**354**	**703**	**971**	**1212**	**1238**	**1638**	**2094**	**2635**
	RT	1.21	1.87	2.92	7.25	10.09	1254	13.74	13.98
Min TT	**NOB**	**5**	**6**	**12**	**9***	**11**	**13**	**15**	**21**
	NOT	5	12	18	21	26	44	55	82
	TVT	**330**	**503**	**707**	**733**	**877**	**1174**	**1381**	**1918**
	RT	1.11	1.91	5.91	12.15	27.52	23.59	43.19	33.42
Min NT	**NOB**	**4***	**4***	**7***	**9***	**10**	**11**	**14**	**18**
	NOT	4*	8	11	18	21	30	41	60
	TVT	**313**	**474**	**652**	**773**	**859**	**1136**	**1391**	**1759**
	RT	1.34	4.56	5.98	9.8	14.85	18.22	31.68	28.38
Alg1-1	**NOB**	**4***	**4***	**7***	**9***	**10**	**9**	**13**	**18**
	NOT	4	8	11	18	21	30	41	60
	TVT	**332**	**467**	**604**	**730**	**819**	**1056**	**1327**	**1651**
	RT	1.07	3.39	12.65	38.11	41.71	50.57	60.18	83.43
Alg1-2	**NOB**	**4***	**4***	**7***	**9***	**8***	**8***	**13**	**16***
	NOT	4	8	11	18	22	35	43	62
	TVT	**332**	**451**	**604**	**732**	**813**	**1092**	**1296**	**1690**
	RT	1.11	4.61	10.06	48.92	193.37	58.91	60.15	85.01
Alg1-3	**NOB**	**4***	**4***	**7***	**9***	**8***	**8***	**11***	**16***
	NOT	4	8	11	18	22	34	44	61
	TVT	**313**	**465**	**589**	**732**	**816**	**1087**	**1290**	**1659**
	RT	1.23	5.15	9.77	137.97	543.58	71.96	177.92	208.57

Note: NOB: Number of buses; **NOT**: Number of trips; **TVT**: Total vehicle time (minutes); **RT**: Running time (second)

Alg1-3 is the best in all decomposition approaches. It is as good as Model 3 for finding the solution with the minimum number of buses. Alg1-2 has the second highest solution quality in the decomposed methods. It finds solutions with the same number of buses as what Model 3 and Alg1-3 found in seven out of eight scenarios. Although Alg1-1 is worse than Alg1-2 and Alg1-3, it finds a better solution than the traditional

decomposition method. Alg1-1 finds solutions using at most two more buses than the best-known solution, which are better than the solutions from traditional methods (MinN, MinTT, and MinNT). The difference between Alg1-3 and the best traditional method is up to 27% in scenario 6 (8 buses from Alg1-3 as compared to 11 buses from the MinNT).

In terms of the total vehicle time (sum across all scenarios), the approaches are ranked in the order of: Model 3 (6927 min) < Alg1-3 (6951 min) < Alg1-1 (6986 min) < Alg1-2 (7010 min) < MinNT (7357 min) < MinTT (7623 min) < MinN (10845 min). Model 3 is the best among all methods. In comparison to the traditional decomposition methods, all proposed decomposition methods (Alg1-1, Alg1-2, Alg1-3) find shorter vehicle time than the traditional decomposition methods (MinN, MinTT, and MinNT). It means that the proposed method (Alg1-1, Alg1-2, and Alg1-3) outperforms the traditional methods in terms of all criteria (the number of buses and total vehicle time).

The objective of the MinNT shares the same terms of the SRC (Model 4) as minimizing the number of trips and total ride time except for the trip-to-school compatibility and the deadhead. And the result in Table 13 shows that Alg1-1 always finds better or equal solutions than what the MinNT does. This demonstrates the importance of the compatibility and deadhead in the routing problems.

The comparison between Alg1-1, Alg1-2, and Alg1-3 is clear. Alg1-3 can find the best solutions in these three approaches with respect to the number of buses and total vehicle time for all scenarios. Alg1-2 is the second, and Alg1-1 is the worst of these three. But the difference between the solutions is not significant, especially for small problems. Thanks to the additional compatibility assignment, Alg1-2

outperforms Alg1-1. Furthermore, the weight adjustment in Alg1-3 improves the model and helps to find better solutions than what Alg1-2 does. Alg1-3 takes more time to solve than Alg1-1 and Alg1-2. The small running time increase is still beneficial considering the huge financial benefit of saving even just one bus. Overall, Alg1-3 is the best. Also, Alg1-3 only takes as little as 0.6% of the time in comparison to solving the MRS (Model 3) directly (in scenario 6 where the MRS is not even solved to optimality). It is beneficial to use the decomposed compatibility-considered single-school bus routing problem than solving the MRS (Model 3) directly.

6.1.5 Experiment 1-3: sensitivity of the SRC (Model 4)

One argument of the difference between Alg1-1 and Alg1-3 comes from the choice of the adjusted weight in the objective of the SRC (see section 3.7.1). Clearly, the weight of the trip-to-school compatibility would have an impact on the solution to the SRC (Model 4) and thus Algorithm 1-1. In this experiment, the weight of the number of trips, the total ride time and deadhead are fixed ($\alpha_{\mathrm{N}} = 1\mathrm{e}5, \alpha_{\mathrm{T}} = 1, \alpha_D = 0.5$) and the weight of the trip-to-school compatibility (α_{C}^{OA}) changes from 1 to 90,000. The sensitivity is conducted based on scenarios 7 and 8.

The result is presented in Table 14. It shows that the solutions found with different weights are very similar with respect to the number of buses, the number of trips and the total vehicle time. It means that the SRC (Model 4) is not extremely sensitive to the adjusted weight. Still, different solutions are obtained under different weights of the trip-to-school compatibility. Yet the solutions to the SRC (Model 4, Alg1-1) are worse that the ones from the SRCW (Model 4", Alg1-3). The latter only needs 11 and 16 buses in scenario 7 and 8, respectively.

Table 14 Result summary for experiment 1-3 (sensitivity of the SRC, Model 4)

α_C^{OA}	Scenario 7				Scenario 8			
	NOB	NOT	TVT	RT	**NOB**	NOT	TVT	RT
1	**15**	41	1391	46.47	**18**	60	1776	39.41
10	**15**	41	1369	51.11	**18**	60	1698	40.13
100	**14**	41	1337	52.42	**18**	60	1722	43.21
1000	**13**	41	1376	40.33	**18**	60	1710	46.11
10000	**13**	41	1319	36.15	**18**	60	1745	48.38
20000	**13**	41	1322	36.80	**18**	60	1726	55.17
30000	**13**	41	1295	61.92	**18**	60	1693	79.13
40000	**13**	41	1328	70.72	**18**	60	1667	89.00
50000	**13**	41	1327	60.18	**18**	60	1651	83.43
60000	**13**	42	1332	82.09	**18**	61	1698	146.81
70000	**13**	42	1303	105.62	**18**	61	1676	97.52
80000	**13**	42	1303	532.15	**17**	61	1692	213.54
90000	**13**	42	1314	143.91	**17**	61	1688	295.00

Note: NOB: Number of buses; **NOT**: Number of trips; **TVT**: Total vehicle time (minutes); **RT**: Running time (second)

6.1.6 Experiment 1-4: the marginal cost of a school bus

To better determine the coefficients of the number of buses and total vehicle time in a more scientific way, an experiment is conducted to calculate the marginal cost of a school bus. Using the same test problems in section 6.1.4, a revised MRS (Model 3) (noted as RModel 3) is solved using the commercial solver Gurobi on Python.

The RModel 3 solves the multi-school bus routing and scheduling problem with the objective of minimizing the total vehicle time and uses up to one more bus than the minimum number of buses (already calculated in section 6.1.4). The marginal cost (in terms of total vehicle time increase) of a school bus is the difference of the total vehicle time in two solutions: using the minimum number of buses or one more bus than the minimum number of buses. The results are shown in Table 15.

Table 15 Result summary for experiment 1-4 (the marginal cost of a school bus)

Scenario		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
# of schools		2	4	6	8	10	15	20	30
# of stops		20	40	60	80	100	150	200	300
Model 3	**NOB**	**4***	**4***	**7***	**9***	**8***	**8***	**11***	**16***
	NOT	4	8	11	18	26	35	44	67
	TVT	**310**	**437**	**583**	**720**	**814**	**1089**	**1259**	**1715**
	RT	1.27	18.73	1192	2585	1585	23795	5697	36000
	Gap	0	0	0	0.05	0.03	0.02	0.01	12.31
RModel 3	**NOB**	**4***	**4***	**7***	**10**	**9**	**9**	**12**	**17**
	NOT	4	8	11	18	21	32	42	63
	TVT	**310**	**437**	**583**	**716**	**787**	**1013**	**1241**	**1648**
	RT	5.51	41.03	644.8	4204	36000	64805	86401	14823
	Gap	0	0	0	0	6.63	5.43	1.3	4.68

Note: * minimum number of buses; **Gap**: Optimal gap (%); **NOB**: Number of buses; **NOT**: Number of trips; **TVT**: Total vehicle time (minutes); **RT**: Running time (second)

The results show that in small cases (scenarios 1 to 3), allowing one more bus does not save vehicle time. This comes from the fact that dispatching a new bus would generate new deadhead trips that depart from and return to the depot as well as the change of bus trips and bus routes. Given the huge amount of financial saving due to the bus and small vehicle time increase, the solutions from Model 3 is significantly better than the one from RModel 3. In case 8, a significant vehicle time reduction is realized by allowing one more bus. However, if we consider the vehicle time reduction per bus and the difference of the optimal gap, such vehicle time saving is still not strong enough to justify the cost increase associated with the new bus. On average, by allowing one more bus, the total vehicle time is only reduced by 38.4 minutes, which corresponds to 3% of the total vehicle time.

6.2 Experiments 2: test on Algorithm 2

6.2.1 Experiment target

The experiments above show that it is beneficial to solve a set of single-school problems rather than the integrated model. However, some large-sized single-school problem might be computationally expensive to solve. A fast heuristic algorithm called the Minimum Cost Matching with the Post Improvement Algorithm (MCMPI, Algorithm 2) is proposed to solve the large-sized single-school problems. A few questions arise with respect to the necessity and the performance of the MCM.

i. Is it necessary to have the MCM? How much running time can the MCM reduce in comparison to the exact method?
ii. Which one is better, the SMCM or the PMCM, in terms of the solution quality and efficiency?
iii. How good are the solutions found by the MCM in comparison to the solution found by exact methods?
iv. Can the MCM find better solutions than state-of-the-art methods?
v. Is it necessary to perform the PI after the MCM? How much can the PI improve the solution from the MCM?

In section 6.2.2, the experiment 2-1 tests the running time and solution quality of the exact method and the heuristic algorithm on the randomly generated problems to address the necessity of the fast heuristic algorithm in large scale problem. The Experiment 2-2 in section 6.2.3 is the preliminary experiment of the SMCM and the PMCM on benchmark problems. A sensitivity analysis is conducted in the experiment 2-3 in section 6.2.4 to determine the best parameters of the PMCM in benchmark

problems. The experiment 2-4 in section 6.2.5 presents the full solutions of the PMCMPI on the benchmark problems.

6.2.2 Experiment 2-1: test on exact and heuristic algorithm

The first preliminary experiment sheds light on the necessity of fast heuristic to solve single-school bus routing problems (the SRCW, Model 4"). Because the necessity mainly comes from the huge computational expense, this experiment tests how the running time of the SRCW and the MCM would increase in response to the increase of the problem size. In this experiment, the test data is generated the same way as it was in section 6.1.2 except that the number of schools is set to two and the number of stops increases from 20 to 60. The approaches used are:

1) SRCW: Directly solve the SRCW (Model 4") to optimality by a commercial solver with $\alpha_{\mathrm{N}} = 1e5, \alpha_C^{CA} = 9e4$;
2) SMCM: Solve the single-school bus routing problem using the sequential minimum cost matching algorithm;
3) PMCM: Solve the single-school bus routing problem using the parallel minimum cost matching algorithm;

The scheduling problems are solved given the routing plans generated by these methods. The results are shown in Table 16 and Figure 16. In small cases (scenarios 1, 3 and 4), both the SMCM and the PMCM can find the solutions as good as the SRCW (with respect to the number of buses) in a much shorter time (as low as 0.16% in scenario 4).

Table 16 Result summary for the experiment 2-1 (the SRC, the PMCM, and the SMCM)

Scenario		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
# of schools		2	2	2	2	2	2	2	2
# of stops		20	24	28	32	36	40	50	60
SRCW	NOB	**4***	**2***	**3***	**4***	**3***	**4***	**5***	**9***
	NOT	4	5	5	6	6	8	8	10
	TVT	313	327	375	393	433	540	488	721
	RT	**2.36**	**4.60**	**155**	**191**	**143**	**610**	**1654**	**4464**
PMCM	NOB	**4***	**3**	**3***	**4***	**5**	**5**	**7**	**9***
	NOT	4	5	5	6	6	9	8	10
	TVT	446	468	474	535	618	724	762	921
	RT	**0.07**	**0.10**	**0.68**	**0.61**	**0.11**	**0.50**	**1.02**	**0.22**
SMCM	NOB	**4***	**3**	**3***	**4***	**4**	**4***	**6**	**9***
	NOT	4	5	5	6	7	8	8	10
	TVT	443	488	566	560	660	763	806	1013
	RT	**0.13**	**1.45**	**1.84**	**0.30**	**0.26**	**0.48**	**0.63**	**0.79**

Note: * minimum number of buses; **NOB**: Number of buses; **NOT**: Number of trips; **TVT**: Total vehicle time (minutes); **RT**: Running time (second)

The difference arises when the problem size gets larger. The SRCW can find solutions using a fewer number of buses than the PMCM and the SMCM at the expense of huge running time (scenarios 5 and 7). The difference between the SMCM and the PMCM is also observed. Out of eight scenarios, the SMCM finds the solutions using the same number of buses as the SRCW in five scenarios where the PMCM only has four. Still, the solutions from the PMCM and the SMCM are both quite good. The SMCM can find solutions using at most one more bus than the exact method, and the PMCM only uses at most two more buses.

The total vehicle times from the PMCM and the SMCM are higher than that of the SRCW. But the vehicle time is much less important than the number of buses. Such difference can be ignored in large scenarios where the exact method is nearly unsolvable. The running time of the exact method becomes a critical problem with the

increase in the problem size. The running time for heuristic barely increases with the increase in the problem size. But the running time for the exact methods increases exponentially. With such an exponential trend, the problem can easily become unsolvable using exact methods. This shows the necessity of fast heuristics to solve large single-school bus routing problems.

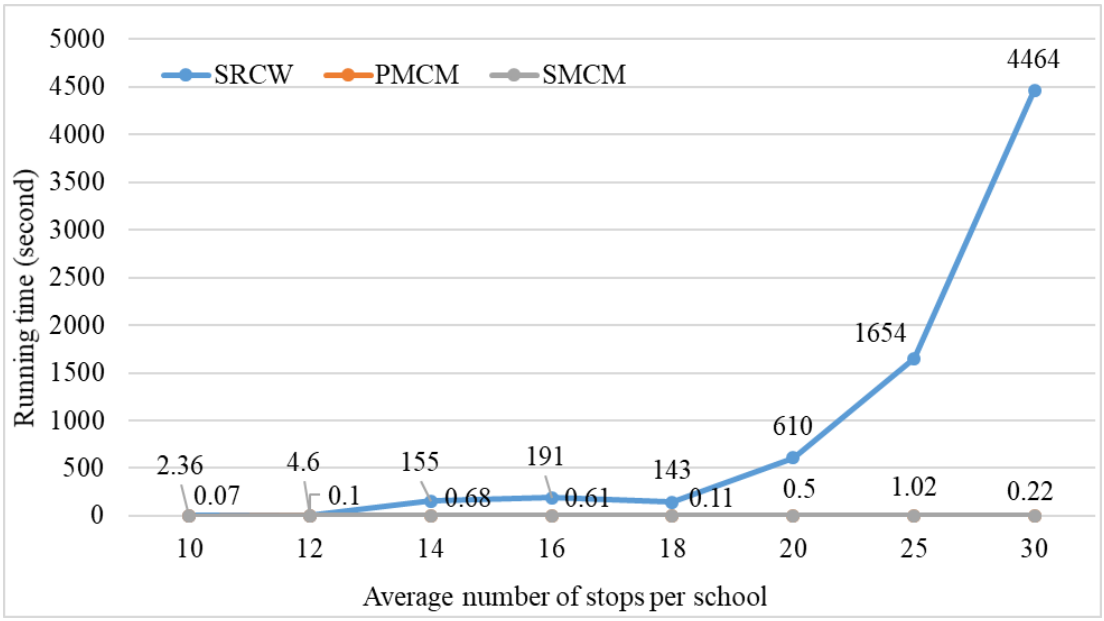


Figure 16 Comparison of the running times for the SRCW, the PMCM, and the SMCM

6.2.3 Experiment 2-2: preliminary benchmark problems

Another question that arises between the SMCM and the PMCM is which one is better? A set of benchmark problems are used to make the comparison. This set of benchmark problems were developed by Park et al. (2012) (Abbreviated as the Park). There are two sets of test scenarios: 1) the RSRB, where stops and schools are randomly distributed, and 2) the CSCB, where stops and schools are gathered together. There are eight test problems in RSRB and 16 test problems in the CSCB. The Maximum Ride Time (MRT) is set to be either 2700 seconds (45 minutes) or 5400 seconds (90 minutes).

The bus fleet is assumed to be homogeneous with the fixed capacity of 66. The distance is assumed to be a grid (Manhattan) distance with a constant bus speed of 20 miles per hour. All Park's results are adopted using the corrected version published on Aug 10th, 2012. These benchmark problems are designed for morning schools, and our algorithm is adjusted accordingly. For example, the trip-to-school compatibility used in afternoon trips is now changed to school-to-trip compatibility for the morning problem.

This comparison is made between the SMCM and the PMCM algorithms using the same naïve parameters setting ($\beta_Q = \beta_C = \beta_T = 1$). The experiments are limited to the first five scenarios for both the RSRB and the CSCB under the Maximum Ride Time (MRT) of 4500 seconds. The results are summarized in Table 17.

Table 17 Result summary for experiment 2-2 (preliminary benchmark problem)

Scenario	# schools	# stops	Braca	Park		SMCM			PMCM		
			NOB	**NOB**	RT	NOT	**NOB**	RT	NOT	**NOB**	RT
RSRB01	6	250	**31**	**31**	0.5	60	**32**	2.9	56	**31**	0.6
RSRB02	12	250	**29**	**30**	0.5	68	**27**$^{+*}$	1.4	71	**31**	0.9
RSRB03	12	500	**61**	**61**	0.8	126	**52**$^{+*}$	6.2	124	**49**$^{+*}$	2.2
RSRB04	25	500	**60**	**56**	0.9	140	**56**$^{+}$	3.7	138	**54**$^{+*}$	2.4
RSRB05	25	1000	**86**	**106**	1.5	234	**92***	24.7	235	**92***	4.8
CSCB01	6	250	**33**	**35**	0.5	69	**34***	4.3	69	**34***	0.9
CSCB02	12	250	**26**	**27**	0.5	67	**30**	3.2	66	**28**	0.9
CSCB03	12	500	**50**	**52**	0.7	133	**58**	8.7	127	**53**	2.9
CSCB04	25	500	**67**	**57**	0.9	150	**54**$^{*+}$	4.8	159	**54**$^{*+}$	3.1
CSCB05	25	1000	**134**	**115**	1.6	291	**119**$^{+}$	31.7	308	**116**$^{+}$	6.6

Note: RT: running time (seconds) without the scheduling problem; **NOB**: Number of buses; **NOT**: Number of trips; *: the result was better than Park et al. (2012); $^{+}$: the result was better than Braca et al. (1997).

The comparison is made between the solution from Park et al. (2012) with single-load and the solution that Park et al. (2012) reported using Braca et al. (1997) algorithm (abbreviated as the Braca). The SMCM and the PMCM have very similar results based on the number of buses. The differences between the SMCM and the PMCM are within four buses. When compared to the solution from Braca et al. (1997)

and the Park et al. (2012), both the SMCM and the PMCM find better solutions for about one-half of the cases even under this naïve parameter setting. The running time for the MCM is higher than the Park because 1) the MCM is coded in Python while the Park algorithm was coded in C, which is inherently faster than Python; and 2) the MCM code is not optimized since the running time is not a critical issue for a planning problem. We also notice that under the same coding structure and language, the PMCM is much faster than the SMCM. Therefore, the complete computational tests use PMCM.

Note that in the benchmark problem, the school bell time is a time window and the trips can depart anytime within that time window. In our implementation of the MCM, the bell time is assumed to be the early school bell time window, and the school-to-trip compatibility is a rough estimation of the real school-to-trip compatibility. The actual trip start time is optimized in the SBS.

6.2.4 Experiment 2-3: sensitivity of the PMCMPI

Clearly, parameters have a significant impact on the algorithms. A sensitivity analysis experiment is conducted to find the best combination of the meta-parameters of both the Minimum Cost Matching (MCM) and the Post Improvement (PI) algorithms. Due to the huge combination of different parameters in the MCM and the PI, only a set of parameters candidates are tested. The sensitivity analysis results shown in Table 18 are obtained based on the RSRB01 with the Maximum Ride Time (MRT) equals to 2700 seconds. The most influential meta-parameters in the MCM (ParaMCM) are the coefficient for the remaining (waste) vehicle capacity (β_Q), the school-to-trip compatibility (β_C) and the travel time (β_T) for evaluating insertion cost in (4-1). Eight sets of meta-parameters for the ParaMCM are tested in the form of $(\beta_Q: \beta_C: \beta_T)$. The

most important meta-parameters in the post improvement (ParaPI) are the coefficients for the total number of trips (δ_N), the trip compatibilities (δ_C) and the travel time (δ_T) in the surrogate cost estimation (4-11). Four sets of meta-parameter values are examined in the shape of ($\delta_N : \delta_C : \delta_T$). In total, 32 different combinations of meta-parameter sets are examined. The criterion is still the number of buses (NOB).

Table 18 Sensitivity analysis for experiment 2-3 (the MCMPI)

NOB		ParaPI							
		(10000:1250:1)		(1000:125:1)		(100:125:1)		(10:10:1)	
		PMCM	PI	PMCM	PI	PMCM	PI	PMCM	PI
Para MCM	(300:20:1)	34	31	34	33	32	33	33	32
	(200:100:1)	33	32	33	33	33	33	34	33
	(100:100:1)	**30***	**30***	32	32	33	**30***	33	33
	(200:10:1)	33	31	33	31	33	31	33	33
	(100:10:1)	32	32	34	33	33	34	33	33
	(1:1:1)	32	31	32	31	33	32	32	31
	(10:10:1)	32	**30***	34	33	34	34	34	33
	(10:100:1)	34	31	33	33	33	32	33	32

Note: **PI**: number of buses of the solution generated by the PMCM followed by the PI; **NOB**: number of buses; **PMCM**: number of buses of the solution generated by the PMCM; **ParaMCM**: parameter of the MCM in the form of ($\beta_Q : \beta_C : \beta_T$); **ParaPI**: parameters for the PI in the form of ($\delta_N : \delta_C : \delta_T$); *: minimum number of buses

By changing the meta-parameters, the MCM performs quite differently. The best solution (with 30 buses) is gained when $\beta_Q : \beta_C : \beta_T = 100 : 100 : 1$ and $\delta_N : \delta_C : \delta_T = 10000 : 1250 : 1$, which is used in the following test.

6.2.5 Experiment 2-4: full benchmark problem

This is the complete experiment of the PMCMPI on all the benchmark problems from Park. For each of the 48 test scenarios, the PMCM ran for ten iterations. The scheduling problems are solved using the trips generated by the PMCM. The best scheduling solution from these ten iterations is reported. Then, all trips generated from

ten iterations of the PMCM are improved using the Post Improvement (PI) algorithm. Similarly, the best scheduling solution using the trips from the PMCM with the Post Improvement (PMCMPI) is reported. There are three major sets of comparisons:

1) solutions from the PMCM are compared to the existing solutions (the Park), for both the absolute difference of the number of buses and the relative bus saving percentage;

2) solutions from the PMCMPI are compared to the existing solutions (the Park);

3) performance of the PI, which is the solution improvement of the PMCMPI with respect to the solutions from the PMCM.

The results of these three sets of comparisons are summarized in Table 19 (maximum ride time equals to 2700 seconds) and Table 20 (maximum ride time equals to 5400 seconds). Because the Park reported better results than Braca et al. (1997) for almost all cases, in this dissertation, only the solutions from Park are reported.

The improvement of the PMCMPI with respect to the Park can be observed in Table 19 and Table 20. Among these 48 test scenarios, the PMCMPI finds strictly better results than the Park in all 48 scenarios. The highest saving of the PMCMPI reaches 37 buses out of 215 buses in CSCB15 with the MRT=5400 seconds, which corresponds to 17% improvement. The highest saving ratio of the PMCMPI is 25% at RSRB03 with the MRT=4500 seconds where 61 buses can be reduced to 46 buses. On average, compared to Park, the PMCMPI can save 11.7 buses, which roughly equals 13%.

Table 19 Result summary for experiment 2-4 (benchmark with the MRT=2700)

Scenario	# of schools	# of stops	Park			PMCM			PMCMPI			PMCM2Park[1]		PMCMPI2Park[2]		PI2PMCM[3]	
			NOT	**NOB**	RT	NOT	**NOB**	RT	NOT	**NOB**	RT	BS	%	BS	%	BS	%
RSRB01	6	250	77	**35**	0.6	72	**30**	1.4	72	**30**	10.0	5	14%	5	14%	0	0%
RSRB02	12	250	88	**32**	0.5	86	**30**	1.4	85	**28**	6.4	2	6%	4	13%	2	7%
RSRB03	12	500	154	**66**	0.8	155	**56**	4.2	154	**54**	25.8	10	15%	12	18%	2	4%
RSRB04	25	500	192	**68**	1.0	185	**62**	4.3	188	**61**	18.0	6	9%	7	10%	1	2%
RSRB05	25	1000	315	**124**	1.6	303	**111**	13.9	297	**104**	93.2	13	10%	20	16%	7	6%
RSRB06	50	1000	366	**103**	1.7	349	**94**	19.7	345	**92**	440.9	9	9%	11	11%	2	2%
RSRB07	50	2000	637	**185**	2.6	595	**178**	56.4	593	**171**	1372.0	7	4%	14	8%	7	4%
RSRB08	100	2000	807	**178**	3.1	769	**157**	55.5	760	**154**	1032.0	21	12%	24	13%	3	2%
CSCB01	6	250	85	**39**	0.5	84	**38**	5.2	82	**35**	139.4	1	3%	4	10%	3	8%
CSCB02	12	250	96	**33**	0.5	96	**31**	3.3	93	**29**	11.6	2	6%	4	12%	2	6%
CSCB03	12	500	176	**66**	0.8	178	**64**	13.4	178	**63**	209.1	2	3%	3	5%	1	2%
CSCB04	25	500	214	**72**	1.0	220	**65**	9.7	220	**65**	131.0	7	10%	7	10%	0	0%
CSCB05	25	1000	398	**135**	1.8	388	**123**	28.0	380	**119**	457.5	12	9%	16	12%	4	3%
CSCB06	50	1000	471	**138**	2.0	452	**127**	19.9	451	**126**	398.8	11	8%	12	9%	1	1%
CSCB07	50	2000	724	**201**	2.9	734	**201**	75.9	718	**194**	858.9	0	0%	7	3%	7	3%
CSCB08	100	2000	840	**183**	3.4	840	**168**	78.5	833	**167**	864.3	15	8%	16	9%	1	1%
CSCB09	6	250	82	**32**	0.4	81	**28**	2.8	77	**25**	185.3	4	13%	7	22%	3	11%
CSCB10	12	250	116	**33**	0.6	112	**32**	3.6	112	**31**	90.9	1	3%	2	6%	1	3%
CSCB11	12	500	158	**58**	0.7	166	**58**	13.1	163	**54**	278.1	0	0%	4	7%	4	7%
CSCB12	25	500	181	**77**	1.0	196	**79**	8.9	187	**73**	229.9	-2	-3%	4	5%	6	8%
CSCB13	25	1000	347	**165**	2.0	348	**150**	29.5	350	**148**	600.9	15	9%	17	10%	2	1%
CSCB14	50	1000	431	**137**	2.1	434	**131**	20.5	431	**129**	423.8	6	4%	8	6%	2	2%
CSCB15	50	2000	708	**247**	3.3	698	**238**	83.9	686	**231**	1204.0	9	4%	16	6%	7	3%
CSCB16	100	2000	846	**208**	3.6	808	**185**	87.3	808	**185**	707.9	23	11%	23	11%	0	0%
Average												**7.5**	**7%**	**10.3**	**10%**	**2.8**	**4%**

Note: **NOB**: number of buses; **RT**: running time (seconds); **NOT**: Number of trips; **BS**: Bus saving (e.g., NOB(Park)– NOB(PMCM)); **%**: bus saving percentage (e.g. (NOB(Park)– NOB(PMCM))/ NOB(Park)×100%); [1]**PMCM2Park**: solution of the PMCM w.r.t the Park; [2]**PMCMPI2Park**: solution of the PMCMPI w.r.t. the Park; [3]**PI2PMCM**: solution of the PMCMPI w.r.t. the PMCM

Table 20 Result summary for experiment 2-4 (benchmark with the MRT=5400)

Scenario	# of schools	# of stops	Park			PMCM			PMCMPI			PMCM2Park[1]		PMCMPI2Park[2]		PI2PMCM[3]	
			NOT	**NOB**	RT	NOT	**NOB**	RT	NOT	**NOB**	RT	BS	%	BS	%	BS	%
RSRB01	6	250	65	**31**	0.5	55	**29**	1.0	57	**27**	15.0	2	6%	4	13%	2	7%
RSRB02	12	250	72	**30**	0.5	67	**24**	1.1	66	**23**	13.2	6	20%	7	23%	1	4%
RSRB03	12	500	129	**61**	0.8	121	**47**	2.9	120	**46**	56.1	14	23%	15	25%	1	2%
RSRB04	25	500	149	**56**	0.9	130	**48**	2.5	129	**46**	52.9	8	14%	10	18%	2	4%
RSRB05	25	1000	268	**106**	1.5	228	**86**	8.1	229	**83**	193.9	20	19%	23	22%	3	3%
RSRB06	50	1000	275	**82**	1.6	226	**78**	9.3	226	**72**	183.4	4	5%	10	12%	6	8%
RSRB07	50	2000	554	**159**	2.4	455	**147**	29.9	457	**143**	911.0	12	8%	16	10%	4	3%
RSRB08	100	2000	714	**158**	2.9	586	**135**	41.3	578	**131**	1330.0	23	15%	27	17%	4	3%
CSCB01	6	250	72	**35**	0.5	69	**33**	2.2	68	**28**	188.5	2	6%	7	20%	5	15%
CSCB02	12	250	66	**27**	0.5	59	**25**	2.4	59	**23**	114.8	2	7%	4	15%	2	8%
CSCB03	12	500	133	**52**	0.7	121	**47**	5.3	120	**44**	297.9	5	10%	8	15%	3	6%
CSCB04	25	500	151	**57**	0.9	149	**50**	5.3	145	**47**	224.2	7	12%	10	18%	3	6%
CSCB05	25	1000	341	**115**	1.6	301	**109**	9.0	296	**96**	199.3	6	5%	19	17%	13	12%
CSCB06	50	1000	384	**123**	1.9	327	**116**	8.9	326	**105**	142.3	7	6%	18	15%	11	9%
CSCB07	50	2000	560	**164**	2.6	494	**160**	48.9	491	**153**	2083.8	4	2%	11	7%	7	4%
CSCB08	100	2000	619	**156**	2.9	549	**136**	65.6	544	**131**	1964.0	20	13%	25	16%	5	4%
CSCB09	6	250	78	**26**	0.4	69	**26**	1.1	69	**23**	15.7	0	0%	3	12%	3	12%
CSCB10	12	250	86	**30**	0.5	78	**29**	1.1	77	**26**	10.5	1	3%	4	13%	3	10%
CSCB11	12	500	118	**50**	0.7	106	**46**	6.1	104	**43**	309.4	4	8%	7	14%	3	7%
CSCB12	25	500	133	**55**	0.9	118	**55**	3.0	116	**54**	52.0	0	0%	1	2%	1	2%
CSCB13	25	1000	284	**135**	1.7	252	**130**	9.7	248	**115**	198.6	5	4%	20	15%	15	12%
CSCB14	50	1000	329	**117**	1.9	300	**107**	12.7	300	**101**	546.1	10	9%	16	14%	6	6%
CSCB15	50	2000	577	**215**	2.8	476	**189**	42.9	473	**178**	1894.0	26	12%	37	17%	11	6%
CSCB16	100	2000	582	**158**	2.9	479	**145**	56.8	479	**145**	872.6	13	8%	13	8%	0	0%
Average												**8.4**	**9%**	**13.1**	**15%**	**4.8**	**6%**

Note: **NOB**: number of buses; **RT**: running time (seconds); **NOT**: Number of trips; **BS**: Bus saving (e.g., NOB(Park)– NOB(PMCM)); **%**: bus saving percentage (e.g. (NOB(Park)– NOB(PMCM))/ NOB(Park)×100%); [1]**PMCM2Park**: solution of the PMCM w.r.t the Park; [2]**PMCMPI2Park**: solution of the PMCMPI w.r.t. the Park; [3]**PI2PMCM**: solution of the PMCMPI w.r.t. the PMCM

Without the Post Improvement, the PMCM can find relatively good solutions as well. Among these 48 test scenarios, the PMCM finds better than or equal results to the Park in 47 scenarios. The highest percentage saving of the PMCM is 23% at RSRB03 with the MRT=4500 seconds. On average, the PMCM can save 7.9 buses from the Park and improve the solution by 8%. The Post Improvement can save up to 15% in the number of buses from the solution from the PMCM. On average, the Post Improvement (PI) can improve the solution from the PMCM by 4.9%. Considering that the solution from the PMCM is already very good, the average 4.9% improvement is remarkable. The exact and the percentage of saving in the number of buses from the PMCMPI compared to Park for the MRT=5400 cases are shown in Figure 17.

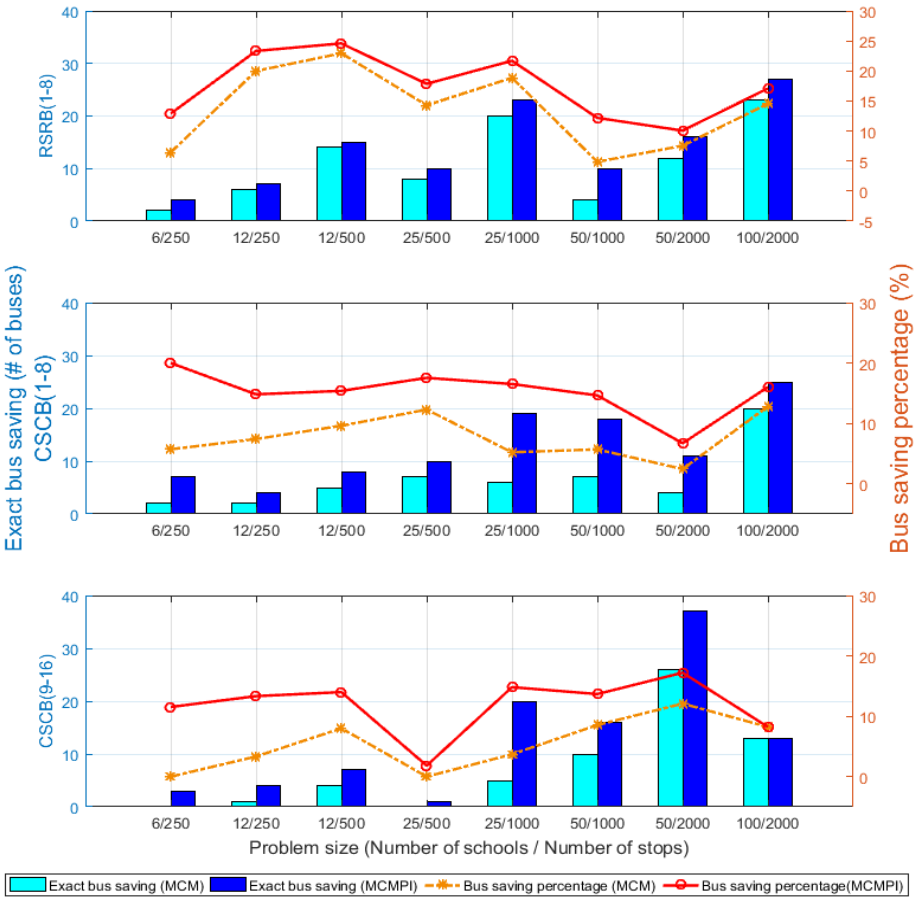


Figure 17 Savings in the number of buses compared to Park (MRT=5400 seconds)

In the benchmark problems, there are eight different problem sizes which differ by the number of stops and schools for all 24 test scenarios. They are arranged in an increasing order based on the problem sizes in Figure 17.

6.3 Experiments 3: test on Algorithm 3

This experiment tests the performance of Algorithm3. The trips data used in this experiment is generated using the PMCM. And the input problem for the PMCM are randomly generated in the same way in section 6.1.2. The minimum bell time interval (σ) is set to be 15 minutes.

6.3.1 Experiment setup

Some characteristics of Algorithm 3 should be examined, including the running time performance, the algorithm performance under different Illegal Route Update (IRU) strategies and different MVT constraints. The following approaches are adopted to solve the same test problems:

1) Alg3S: Using Algorithm 3 with the simple IRU update strategy;
2) Alg3L: Using Algorithm 3 with the late trip replacement IRU update strategy;
3) Model 5: Solving the MBSV (Model 5) using the exact method in a commercial solver.

The MVT is first set to be 8 hours and then tightened to 6 hours and 4 hours. In Alg3S, we only add the base illegal bus route found by the MBS (Model 7) into the illegal bus route (IBR) set. But in Alg3L, after identifying the base illegal bus routes, we try to find more illegal bus routes by replacing the last trip on the base illegal bus

routes to longer trips. The base and replaced illegal bus routes will all be included in the IBR set.

6.3.2 Comparison of the results

The solutions obtained from different approaches are summarized in Table 21. The first thing to notice is that the solutions from Model 5 and Alg3L are the same (the MVT=8 in scenario 1 to 5) in terms of the number of buses and the total deadhead. In scenario 7 and 8 where Model 5 is not solved to optimality, Alg3L finds solutions with fewer buses and shorter total deadhead than Model 5. It demonstrates the argument that the solution of Algorithm 3 is either optimal or infeasible (see section 4.3.3.3). Also, the running time of Alg3L is only 0.003% of Model 5 (at scenario 8 with the MVT=8 hours where Model 5 is not even solved to optimality). This demonstrates the benefit of using Algorithm 3 to solve the MBSV (Model 5) especially at the condition where the MVT constraints are loose.

The MVT constraints are gradually tightened to test how Algorithm 3 will perform with different MVT constraints. When the MVT constraint is tightened to 6 hours per bus route, all the buses still satisfy the MVT constraints. Thus, the solution to the MBSV (Model 5) is the same as the MBS (Model 7). The running time (of Algorithm 3) increases when the MVT constraint is tightened to 4 hours per bus route. This is expected because the tighter the MVT constraints are, the more illegal bus routes exist and thus the more iterations Algorithm3 needs to run before finding the final solution.

An important property of Algorithm 3 is the IRU strategy (detailed discussion can be found in section 4.3.3.2). Two methods are tested in this experiment: Alg3S and

Alg3L. The solution shows that Alg3L uses a shorter time and fewer iterations to find the optimal solution than Alg3S. This means that the last trip replacement strategy is better than the simple IRU strategy.

Table 21 Result summary for experiment 3

Method	MVT	Scenario	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
		# of schools	2	4	6	8	10	15	20	30
		# of stops	20	40	60	80	100	150	200	300
		NOT	9	12	17	20	23	32	42	61
Alg3S	4	NOB	9	7	11	10	12	14	19	23
		TDD	520	574	827.27	741	930	1078	1459	1964
		RT	**0.04**	**0.93**	**0.91**	**1.35**	**21.94**	**57.01**	**164**	**454,750**
		IT	1	6	8	8	81	95	190	4806
		$\|IBR\|$	0	17	13	13	232	229	561	17302
Alg3L	4	NOB	9	7	11	10	12	14	19	23
		TDD	520	574	827	741	930	1078	1459	1964
		RT	**0.02**	**0.35**	**0.60**	**0.61**	**3.89**	**18.63**	**52.64**	**22,281**
		IT	1	5	7	5	18	48	106	428
		$\|IBR\|$	0	25	19	21	354	635	1589	28265
	6	NOB	9	7	11	9	10	13	14	18
		TDD	520	521	771	687	752	951	1,100	1,399
		RT	**0.04**	**0.11**	**0.21**	**0.15**	**0.21**	**0.34**	**0.46**	**1.07**
		IT	1	1	1	1	1	1	1	1
		$\|IBR\|$	0	0	0	0	0	0	0	0
	8	NOB	9	7	11	9	10	13	14	18
		TDD	520	521	771	687	752	951	1,100	1,399
		RT	**0.04**	**0.12**	**0.15**	**0.19**	**0.18**	**0.23**	**0.46**	**1.07**
		IT	1	1	1	1	1	1	1	1
		$\|IBR\|$	0	0	0	0	0	0	0	0
Model 5	8	NOB	9	7	11	9	10	13	16	20
		TDD	520	521	771	687	752	951	1,325	1,545
		RT	**0.03**	**0.47**	**19.36**	**8.82**	**29.12**	**1,920**	**18,000**	**31,348**
		Gap	0	0	0	0	0	0	12.5	10

Note: **MVT**: Maximum vehicle time per route (hour); **NOB**: Number of buses; **NOT**: Number of trips; **TDD**: Total deadhead (minute); **RT**: Running time (second); **IT**: Number of iterations; $|\boldsymbol{IBR}|$: Number of illegal routes; **Gap**: Optimality gap (%)

Alg3L explores additional illegal bus routes, which makes the number of illegal bus route higher than that from Alg3S. Although some of these illegal bus routes might

be unnecessary for finding the solution, spending some time to update the IBR using the replacement strategy is better than blindly solving the MBS (Model 7) repeatedly. This difference is more significant in large cases. In scenario 8, Alg3S finds the final solution after five days while Alg3L finds the solution in 6.19 hours.

6.4 Experiments 4: test on stochastic models

6.4.1 Test problem generation

Collecting the stochastic data in the real world is difficult. Many of the research conducted computational experiments on randomly generated test problems. So does our research. The deterministic part of the data, including the locations of the schools and the bus stops, the number of students at bus stops and school bell time windows, is generated the same way as in section 6.1.2. The essential task of generating the stochastic test problem is to generate the stochastic travel time of every edge at different departure time on the network.

In practice, the travel time on a road is influenced by the time of the day – the travel time increases during a peak period and decreases during a non-peak period. Also, the travel time at each specific time of the day acts as a random variable. To account for these two parts, we assume that the expected travel time of a trip varies at different departure time. Moreover, at a specific departure time, the real travel time of that trip follows a normal distribution. In the school bus bell time and scheduling problem, because the schools must dismiss at an integer multiple of a minimum time interval (i.e., 2:00 PM) instead of a fraction time, we can analytically write down the expected travel times for all edges at a given departure time. Then we can draw a

collection of real travel times at a given departure time from a normal distribution. The first collection of generated travel time is assumed to be the observed data, which is known beforehand. The second collection of travel times is generated in the same way, which is used for validation. The same method is applied to generate the stochastic travel times and the stochastic deadhead durations. To compare the stochastic models and the deterministic models, we generated the stochastic data in a way that the means of the stochastic travel time and deadhead are close to the deterministic travel time and deadhead (section 6.1.2)

We assume that the expected travel time at different times of the day (from 12 PM to 4 PM, which is the bell time window for all schools), forms an increasing linear line (shown in Figure 18). The minimum and maximum equal to 80% and 120% of the deterministic travel time (amplitude=20%). Due to the symmetry, the mean of the expected travel time over the bell time window (12 PM to 4 PM) equals the deterministic travel time (flat dash line in Figure 18). Moreover, the real travel time of a trip at a specific departure time follows a normal distribution with the mean equal to the expected travel time on this increasing line. The standard deviation of the stochastic travel time of the trips is 20% of the expected travel time at a specific departure time (stdTT=20%). The standard deviation of the stochastic deadhead is 10% of the expected deadhead at a specific departure time (stdDD=10%). According to the above assumption, the real travel time of a trip has a smaller mean and a smaller variance at non-peak periods (close to 12 PM) and a higher mean and a higher variance at peak periods (close to 4 PM). Given the mean and the standard deviation, we draw 1000 sample data from the given normal distribution function (samSize=1000) as the

observed data for each trip or each deadhead at each specific departure time. Then, we draw another 1000 test data from the same normal distribution function (testSize=1000) as the test data. The parameters including amplitude, the stdTT, and the stdDD are subjected to change in sensitivity analysis in section 6.4.4.

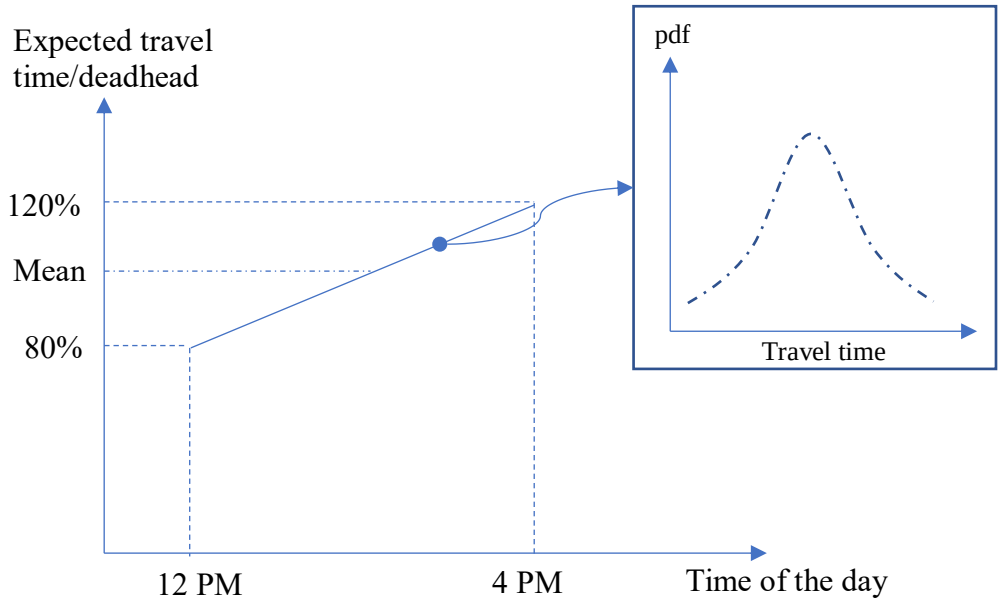


Figure 18 Illustration of the generation of stochastic travel times and deadheads

6.4.2 Experiment setup

In this set of experiment, the deterministic model, the time-dependent model, and the stochastic model are used to solve the multi-school bell time and bus scheduling problem based on eight randomly generated test problems. As explained in section 6.4.1, the stochastic test problems are generated very close to the deterministic test problems from section 6.1.2. Thanks to the similarity, we can ask the following questions:

i. How do the deterministic solution, the time-dependent solution, and the stochastic solutions perform under a certain level of stochastic travel time and

deadhead in terms of the cost (number of buses, total deadhead) and the probability of being served on time?

ii. Whether it is more beneficial to solve the time-dependent and the stochastic models than solving the deterministic model in terms of the solution quality and computing cost?

iii. Which stochastic model is better: TS_MBSV (Model 9) or Algorithm 4, the insertion-based or the GRASP in Algorithm 4?

Five approaches are used to solve the eight randomly generated test problems:

i. Model 5: Deterministic model, the MBSV (Model 5);

ii. Model 8: Time-dependent model, the TD_MBSV (Model 8);

iii. Model 9: Chance Constrained model, the TS_MBSV (Model 9);

iv. Alg4I: Column Generation-based Stochastic model, using the Insertion-based heuristics to solve subproblems in Algorithm 4-1;

v. Alg4G: Column Generation-based stochastic model, using the GRASP to solve subproblems in Algorithm 4-1.

The solutions from these five methods are compared in the following aspects:

1) Cost: the number of buses (NOB), and the total deadhead (TDD);

2) Stochasticity:

2.1) Minimum probability of being served on time (MinProb);

2.2) Mean probability of being served on time (MeanProb);

2.3) Number of routes with the probability of less than 98% to be served on time (N<.98);

2.4) Ratio of routes with the probability of less than 98% to be served on time (R<.98);

3) Efficiency: Optimal gap, the running time (RT, in second).

6.4.3 Comparison of the solutions

6.4.3.1 Necessity of stochasticity

The solutions obtained from the five models are summarized in Table 22. The first thing noticed is the low on-time service rate for Model 5 and Model 8 that do not consider any uncertainty. The most unreliable bus route from Model 5 has only a 26% probability to be on time in scenario 6. Up to 36% of the routes have a probability less than or equal to 98% to be on time for Model 8 in case 3. This means although the time-dependent model (Model 8) could consider the traffic congestion at different departure times, it cannot account for the uncertainty of the travel time and the deadhead under a specific departure time. The result in Table 22 demonstrates the importance of a stochastic model in a real stochastic world.

Model 9, which considers the uncertainty of the travel time under the normal assumption, yields better solutions than Model 5 and Model 8. But it could also violate the Stochastic On-Time Constraint (SOT) in some cases (scenario 4). This results from several reasons. First, the normal assumption may not be a good assumption for the general trips travel times and deadheads, especially when ignoring the departure time. Second, we used the estimated mean arrival time and the estimated standard deviation of the arrival time for linearization, which might diminish the on-time performance. Finally, Model 9 used the Trip-based Stochastic On-Time Constraint (TSOT) instead of the Route-based SOT. The former is slightly more relaxed than the latter.

Table 22 Result summary for Model 5, 8, 9 and Algorithm 4

Scenario		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Number of schools		2	4	6	8	10	15	20	30
Number of stops		20	40	60	80	100	150	200	300
Number of trips		9	12	17	20	23	32	42	61
Model 5	**NOB**	**9**	**7**	**11**	**9**	**10**	**13**	**16**	**20**
	TDD	**520**	**521**	**771**	**687**	**752**	**951**	**1,325**	**1,545**
	Gap (%)	0	0	0	0	0	0	12.5	10.0
	RT	0.03	0.47	19.36	8.82	29.12	1,920	18,000	31,348
	MinProb	100%	99%	100%	99%	99%	26%	57%	80%
	MeanProb	100%	100%	100%	100%	100%	91%	97%	98%
	N<.98	0	0	0	0	0	3	3	3
	R<.98	0	0	0	0	0	23%	19%	15%
Model 8	**NOB**	**9**	**7**	**11**	**9**	**10**	**13**	**14**	**20**
	TDD	**520**	**512**	**731**	**663**	**733**	**968**	**1,096**	**2,300**
	Gap (%)	0	0	0	0	0	0	0	40.0
	RT	0.02	3.32	31.89	33.52	945	18,000	18,000	432,003
	MinProb	100%	100%	71%	60%	99%	98%	86%	60%
	MeanProb	100%	100%	92%	95%	100%	100%	98%	96%
	N<.98	0	0	4	1	0	1	2	5
	R<.98	0	0	36%	11%	0	8%	14%	17%
Model 9	**NOB**	**9**	**6**	**12**	**9**	**11**	**13**	**16**	**19**
	TDD	**521**	**458**	**871**	**700**	**942**	**975**	**1169**	**1403**
	Gap (%)	0	0	0	0	0	0	0	5
	RT	0.02	0.27	7.03	18.32	36.74	1084.8	8365	28800
	MinProb	100%	100%	100%	92%	100%	100%	100%	100%
	MeanProb	100%	100%	100%	99%	100%	100%	100%	100%
	N<.98	0	0	0	1	0	0	0	0
	R<.98	0%	0%	0%	11%	0%	0%	0%	0%
Alg4I	**NOB**	**9**	**7**	**11**	**9**	**10**	**13**	**15**	**21**
	TDD	**520**	**512**	**771**	**665**	**733**	**968**	**1,106**	**1,527**
	RT	0	0	0	20.46	0	236.14	1,008	379.04
	MinProb	100%	100%	100%	100%	99%	100%	100%	100%
	MeanProb	100%	100%	100%	100%	100%	100%	100%	100%
	N<.98	0	0	0	0	0	0	0	0
	R<.98	0	0	0	0	0	0	0	0
Alg4G	**NOB**	**9**	**7**	**11**	**9**	**10**	**13**	**14**	**20**
	TDD	**520**	**512**	**771**	**687**	**733**	**951**	**1,096**	**1,426**
	RT	0	0	0	0	0	85.23	225.67	1466.3
	MinProb	100%	100%	100%	99%	99%	100%	99%	98%
	MeanProb	100%	100%	100%	100%	100%	100%	100%	100%
	N<.98	0	0	0	0	0	0	0	0
	R<.98	0	0	0	0	0	0	0	0

Note: **NOB**: Number of buses; **TDD**: Total deadhead duration (minute); **RT**: Running time (second); **MinProb**: Minimum probability of being served on time; **MeanProb**: Mean probability of being served on time; **N<.98**: Number of routes with the probability of less than 98% to be served on time; **R<.98**: Ratio of routes with the probability of less than 98% to be served on time.

The Column Generation-based stochastic algorithms (Alg4I and Alg4G) conquer all the problems with an sample-based method to check the Route-based Stochastic On-Time Constraint (SOT) on empirical distribution from the observed data. The solutions from Alg4I and Alg4G satisfy the route-based stochastic on-time constraints. The minimum probability of being served on time for all routes in all scenarios is larger than 98%.

In small cases (scenarios 1 to 5 for Model 5 and scenarios 1, 2, 5 for Model 8), the stochastic on-time constraint is satisfied even if the models do not consider uncertainty. This happens because in small cases, each bus route only serves one or two trips and there is a high probability that the route is served on time. The benefit of considering the stochasticity is not significant in small cases. Note these tests are conducted based on the data generated under the uncertainty specified in section 6.4.1. The benefits of stochastic models will be more significant in scenarios with higher uncertainty.

6.4.3.2 Solution quality

In terms of the number of buses, which dominates the cost of the transportation plan, Alg4G find the best solution with the least number of buses. Model 5 and Model 8 use more buses in scenarios 7 and 8 respectively than the best-known solution. It comes from the high optimality gap, which is the nature of solving complicated MIP models using exact methods. Model 9 uses more buses in scenario 7 even when its optimality gap is 0. This comes from the fact that Model 9 ignores the different travel time distribution at different departure time.

Similar results are observed for the total deadhead. For the large scenarios when the models cannot be solved to optimality, the solutions from the heuristics (Alg4I and Alg4G) are better than those from the MIP models (Model 5, Model 8, and Model 9) with smaller total deadhead. In the small cases, the total deadhead of the solutions from Alg4I and Alg4G are within 10% of the best solutions from Model 5, Model 8, and Model 9. Due to the nature of the MIP, it is impossible to solve Model 5, Model 8, and Model 9 to optimality using the exact method for large scale real-world problems. The time-dependent model is the most complicated one due to the additional departure time index added to each trip and deadhead. It took five days to reduce the optimal gap to 40% in scenario 8, which has only 30 schools, 300 stops, and 61 trips. Model 5 and Model 9, although are more efficient than Model 8, cannot find optimal solutions for scenario eight after one day. Thus, the Column Generation-based stochastic algorithms (Alg4I and Alg4G) are the best practical way with good solution quality to solve large-scale real-world school bus scheduling and bell time adjustment problem with stochastic travel times and deadheads.

In these limited tests, Alg4G outperforms Alg4I with a fewer (or equal) number of buses and smaller (or equal) total deadhead in all eight test problems. Nevertheless, the difference is not significant (at most one bus). When looking at the efficiency, both Alg4I and Alg4G outperform Model 5, Model 8 and Model 9 significantly. In scenario 8, Alg4G only uses 0.3% of the running time of Model 8 and finds a solution with 33% fewer buses than Model 8. Overall, this shows the benefits of using the stochastic model, especially Algorithm 4 over the deterministic and time-dependent models in the real world.

6.4.4 Sensitivity analysis

6.4.4.1 Level of uncertainty

The benefits of the stochastic models depend on the uncertainty of the system. In a free flow system without any traffic, the traffic time is fixed and the same regardless of the departure time. The deterministic model (Model 5) can fully capture all features in this system, and no route will violate the stochastic constraint. In a highly predictable system (the standard deviation at a fixed departure time is small), even if the travel time varies at different departure time, the time-dependent model (Model 8) can find good solutions with a high probability of being served on time. In a system where the travel time of a trip is approximately normally distributed regardless of its departure time, Model 9 could be a good fit, even if its standard deviation is high. Algorithm 4 directly works with the historical data without any distribution assumption of the travel time. Algorithm 4 should be able to find solutions satisfy the Stochastic On-Time constraint as long as the historical sample data capture the trend of the stochasticity regardless of its distribution.

In stochastic data generation (section 6.4.1), the amplitude determines the variation of the maximum and minimum expected travel time (deadhead) away from the mean deterministic travel time (deadhead), which in some sense implies the severity of the congestion. The standard deviation of travel time (stdTT) and the deadhead (stdDD) specify the uncertainty of the travel time and deadhead under a specific departure time, which highly influences the predictability of the system. To test how Model 5, Model 8, Model 9 and Algorithm 4 GRASP (Alg4G) perform under different levels of uncertainty, the amplitude is changed from 0.1 to 0.4 and three set of standard

deviation of travel times and deadheads (stdTT:stdDD) are examined: (0.1:0.05), (0.2:0.1), and (0.3:0.15). The results are shown in Table 23.

Table 23 Solutions under different level of uncertainty

stdTT: stdDD	Model	Criteria	Amplitude			
			0.1	0.2	0.3	0.4
0.10:0.05	Model 5	**MinProb**	**94%**	**64%**	**28%**	**8%**
		MeanProb	100%	98%	96%	94%
		R<.98	6%	6%	6%	6%
	Model 8	**MinProb**	**96%**	**98%**	**95%**	**98%**
		MeanProb	98%	100%	100%	100%
		R<.98	14%	7%	7%	7%
	Model 9	**MinProb**	**100%**	**99%**	**100%**	**99%**
		MeanProb	100%	100%	100%	100%
		R<.98	0%	0%	0%	0%
	Alg4G	**MinProb**	**100%**	**100%**	**100%**	**100%**
		MeanProb	100%	100%	100%	100%
		R<.98	0%	0%	0%	0%
0.20:0.10	Model 5	**MinProb**	**78%**	**57%**	**36%**	**23%**
		MeanProb	97%	97%	96%	95%
		R<.98	31%	19%	13%	6%
	Model 8	**MinProb**	**85%**	**86%**	**85%**	**85%**
		MeanProb	96%	98%	99%	99%
		R<.98	14%	14%	7%	7%
	Model 9	**MinProb**	**100%**	**100%**	**100%**	**100%**
		MeanProb	100%	100%	100%	100%
		R<.98	0%	0%	0%	0%
	Alg4G	**MinProb**	**100%**	**99%**	**100%**	**100%**
		MeanProb	100%	100%	100%	100%
		R<.98	0%	0%	0%	0%
0.30:0.15	Model 5	**MinProb**	**69%**	**55%**	**43%**	**31%**
		MeanProb	93%	95%	95%	95%
		R<.98	44%	31%	19%	19%
	Model 8	**MinProb**	**75%**	**76%**	**75%**	**73%**
		MeanProb	96%	97%	98%	98%
		R<.98	21%	21%	14%	7%
	Model 9	**MinProb**	**100%**	**100%**	**100%**	**100%**
		MeanProb	100%	100%	100%	100%
		R<.98	0%	0%	0%	0%
	Alg4G	**MinProb**	**100%**	**100%**	**100%**	**100%**
		MeanProb	100%	100%	100%	100%
		R<.98	0%	0%	0%	0%

Note: **MinProb**: Minimum probability of being served on time; **MeanProb**: Mean probability of being served on time; **R<.98**: Ratio of routes with the probability of less than 98% to be served on time.

The test data is generated the same way as in section 6.4.1 using the new stochastic parameters. The MinProb, the MeanProb, and the R<.98 (defined in section 6.4.2) are all reported for the solution from scenario 7. From Table 23, we can see that with the increase of the uncertainty (amplitude and the standard deviation), Model 5 performs the worst. The probability of being on time could be as low as 8% (amplitude = 0.4, stdTT:stdDD = 0.1:0.05) and 44% of bus routes have a probability less than 98% to be on time (amplitude = 0.1, stdTT:stdDD = 0.3:0.15).

For Model 8 (which considers different travel times at different departure times), the increase of the amplitude does not affect its performance as much as the change of the standard deviation does. For example, when the stdTT:stdDD is fixed to be (0.1:0.05), the MinProb, the MeanProb, the N<.98 and the R<.98 are almost the same even if the amplitude is changed from 0.1 to 0.4. But when the stdTT:stdDD is changed to (0.2:0.1), even if the amplitude remains unchanged, the MinProb decreases by more than 10%. Moreover, when the stdTT:stdDD reaches (0.3:0.15), the N<.98 and R<.98 also increase. Overall, the time-dependent Model 8 find more robust solutions than the deterministic Model 5.

The Model 9 and Alg4G have better solutions than the deteministic Model 5 and the time-dependent Model 8. It comes from the consideration of the real stochastic travel time. Among these tests, no matter how different the stochastic and the deterministic travel times are, the Model 9 and Alg4G never fail to find a solution that satisfies the stochastic on-time constraint, which stipulate that the routes have to be served on time with at least 98% probability.

6.4.4.2 Simulation size

Another parameter that may impact the performance of Algorithm 4 is the simulation/sample size. To show the influence of sample size, we gradually increase the sample size (for each trip and each deadhead pair) from 10 to 1000. The experiment is conducted based on scenario 7 with amplitude = 0.2, stdTT:stdDD = 0.2:0.1, and the problem is solved by Alg4G. The test sample size always remains at 1000. The results are shown in Table 24.

Table 24 Solutions of Alg4G with different sample sizes

Sample size	**10**	**50**	**100**	**500**	**1000**
NOB	15	14	15	14	14
TDD	1089	1104	1078	1105	1096
RT (second)	161.09	92.88	142.65	427.98	225.67
Iter	121	66	99	121	121
MinProb	100%	98%	98%	99%	99%
MeanProb	100%	100%	100%	100%	100%
N<.98	0	1	1	0	0
R<.98	0%	7%	7%	0%	0%

Note: **NOB**: Number of buses; **TDD**: Total deadhead duration (minute); **RT**: Running time (second); Iter: Total number of iterations; **MinProb**: Minimum probability of being served on time; **MeanProb**: Mean probability of being served on time; **N<.98**: Number of routes with the probability of less than 98% to be served on time; **R<.98**: Ratio of routes with the probability of less than 98% to be served on time.

As we see, when the sample size is relatively small (less than or equal to 100), the solutions are not stable. Some routes may violate the Stochastic On-Time Constraint, and some solutions may use more buses than the minimum needed number. When the sample size is large enough (greater than or equal to 500), the solutions are quite stable, meaning that the solutions have the minimum number of buses and all bus routes satisfy the Stochastic On-Time Constraint. Furthermore, the running time and the number of iterations do not increase with the increase in the sample size, which means increasing the sample size does not introduce a computational burden for

Algorithm 4. Thus, in the real implementation of Algorithm 4, it is desired to use all the training data to find better solutions without worrying about the computational burden due to the training data size.

6.5 Chapter summary

In this chapter, many experiments were conducted on randomly generated test problems to test the correctness and the performance of the models and algorithms in the stochastic cases. Due to the complexity of the school bus planning problem (SBPP), each subproblem and algorithm was tested separately in different experiments. These experiments proved that the SBPP could be solved using these algorithms very efficiently with high solution quality.

The experiments in section 6.1 showed that the integrated models are correct and solvable. It was also shown that proposed decomposed single-school problems could find solutions using the same number of buses as the integrated model in much shorter time, which outperforms traditional decomposition methods. The experiments in section 6.2 showed the necessity and efficiency of Algorithm 2 in large scale single-school problems. It showed that Algorithm 2 outperforms state-of-the-art trip generation methods in all instances of the benchmark problems by up to 25%. The experiment in section 6.3 showed that Algorithm 3 could find the optimal solution in a much shorter time than solving the MBSV (Model 5) using an exact method.

Section 6.4 demonstrated the need for stochastic models and algorithms in the world full of uncertainty, and further showed that the Column Generation-based method (Algorithm 4) is a better choice than the time-dependent model and the chance-

constrained model with respect to the better capability of capturing the un-parameterized uncertainty, fast running time and good solution quality.

Chapter 7 Real-World Case Study

To test our proposed models and algorithms in real-world applications, we collected the real-world data from a public-school system. Due to the confidentiality reason, we will not reveal the school system's name. But all the data we use in this chapter, unless otherwise stated (i.e., some data are from Google Map Distance Application Programming Interface), are real data from this public-school system. Two sets of data were collected from this school system: general-purpose bus transportation (case 1) and bus transportation with special needs (case 2). Given the different data available to us, we present the bell/dismissal time in both AM and PM cases. Please note that the AM and PM cases are identical. We can easily obtain the AM (or PM) solution by reversing the other. Similarly, because the school duration is fixed, once the AM bell time (or PM dismissal time) is determined, the other one is fixed.

7.1 Case 1: general-purpose bus transportation

7.1.1 Experiment setup

In case 1, there are in total 40 elementary schools (ES), 20 middle schools (MS), 12 high schools (HS) and 6 other institutes. In total 994 trips are designed to serve all schools. For each trip, the following information is given: the trip name, the trip duration, the last stop of each trip and its location (longitude and latitude), the school name, and the school bell/dismissal time. The deadhead duration (the travel time from the last stop of every trip to all other trips/schools) are queried from Google Distance Application Programming Interface (API). The travel distance (in meters) and travel time without traffic (in seconds) are both obtained. In this dissertation, we used the

travel time without traffic in minutes, which will be subjected to further sensitivity analysis in section 7.2.2.4.

As changing the school bell times will have an impact on the students, the teachers, the parents, and the drives, we worked with the school board and listed three potential scenarios of the school bell time adjustment. Note that the bus routes are optimized along with the bell times in each scenario. Adding the case of optimizing bus routes while keeping the current school bell time, we have in total of four test scenarios:

- Sche only (optimize the bus routes only)
- ES first (elementary schools start first)
- MS first (middle schools start first)
- HS first (high schools start first)

The detailed range of school bell times for each scenario is shown in Table 25. Because high school students usually have extracurricular activities like sports, it is demanded by the school board that all high schools must start and dismiss at the same time.

Table 25 Proposed school bell time range for case study 1

Scenario	School bell time range			
	ES	**MS**	**HS**	**Others**
Current/Sche only	8:10 a.m. - 9:25 a.m.	7:40 a.m. - 8:25 a.m.	7:25 a.m.	8:30 a.m. - 9:15 a.m.
ES first	7:30 a.m. - 8:00 a.m.	8:30 a.m. - 9:15 a.m.	8:30 a.m.	8:00 a.m. – 9:00 a.m.
MS first	8:30 a.m. - 9:15 a.m.	7:30 a.m.	8:15 a.m.	7:45 a.m. – 9:00 a.m.
HS first	8:10 a.m. - 9:10 a.m.	7:40 a.m. - 8:55 a.m.	7:25 a.m.	6:55 a.m. - 8:25 a.m.

7.1.2 Results

We used Model 13 and Algorithm 3 to solve the Multi-School Bus Scheduling and the Multi-School Bell Time and Bus Scheduling problem respectively. All the results with and without bell time adjustment are shown in Figure 19. As we can see, the current operation uses 324 buses, which is the highest and most inefficient among all scenarios. By optimizing the bus routes (without the changing the bell times), the number of buses can be reduced to 304. As the average bus annual cost is approximately $60,000-$70,000 (including the bus acquisition, driver's salary, bus maintenance, gas, etc.), reducing 20 buses could result in a significant saving of 1.2-1.4 million dollars annually. Yet, this solution does not change the school bell time, which means that except for the drivers, the students' and the teachers' schedule will not be affected. The students still go to the same bus stops to get picked up and dropped off at the same time; the students still arrive at and depart from the school at the same time; and the teachers still go to and leave the schools at the same time.

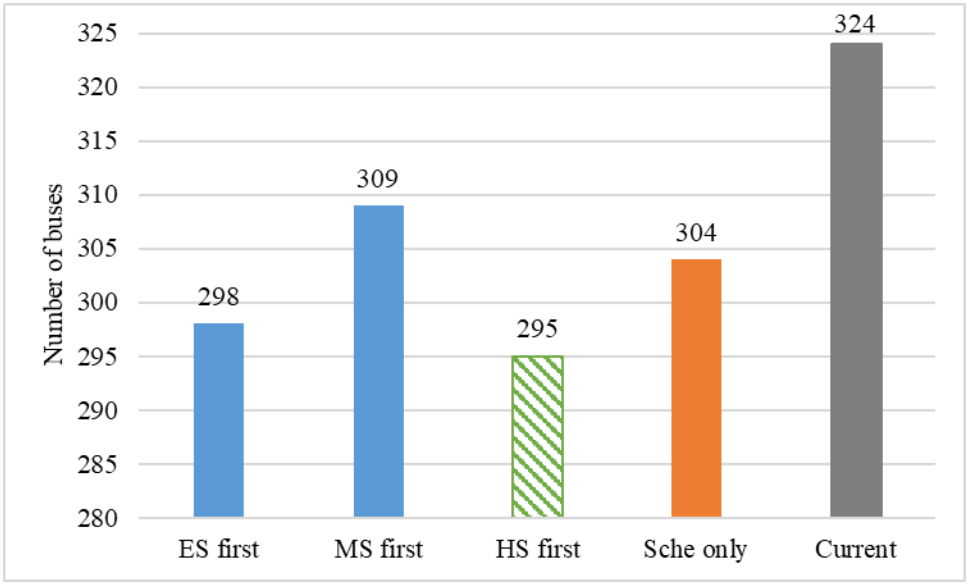


Figure 19 Solution summary of case study 1

After changing the school bell times, the number of buses can be further reduced. The HS first scenario yields the best solution among all other proposals: 1) the school bell times have the least change from the current bell times and 2) it requires the least number of buses. Under the HS first assumption, all high schools start at 7:25 a.m., which is the same as the current bell times; all middle schools start between 7:40 a.m. and 8:45 a.m., of which the earliest bell time remains the same and the latest bell time is postponed by 30 minutes; and all elementary schools start between 8:00 a.m. and 9:15 a.m., of which the earliest bell time remains unchanged and the latest bell time is postponed by 15 minutes.

Moreover, the HS first scenario gains a significant bus saving. The number of buses is reduced from 324 to 295. This 29-bus saving corresponds to 8.9% improvement of the solution, and it saves approximately 1.7-2.0 million dollars annually. We realize that the change of school bell times is a complex issue and does not only depend on the transportation cost. However, this case study provides strong mathematical evidence that changing the school bell times wisely could greatly reduce the transportation cost, which is definitely worth consideration in real world. Furthermore, it finds the scenario with the highest cost reduction and the smallest deviation from the current bell times.

7.2 Case 2: bus transportation with special needs

7.2.1 Experiment setup

In this case study, we collect the public-school transportation system data with special need from the same school district as in case 1. This set of data consists of the

information for special schools and students who have special needs at general schools. This dataset not only contains the Trip Name, School Code, School Name but also the stop-related information including the Stop Time, Number (of) Students (at each stop), X Coord (longitude), Y Coord (latitude) (of each stop), and Sequence. An illustration of one trip in the dataset is shown in Table 26. Again, due to the confidentiality reason, some data is masked. Some summary statistics of the data are summarized in Table 27.

Table 26 Illustration of trip information of case study 2

Trip Name	Stop Time	Number Students	School Code	School Name	X Coord	Y Coord	Sequence
A_ School PM 179	3:45:00 PM	9	A	A_School	-76.xxxx	39. xxxx	1
	3:51:00 PM	-1			-76. xxxx	39. xxxx	2
	3:52:00 PM	-1			-76. xxxx	39. xxxx	3
	3:55:00 PM	-1			-76. xxxx	39. xxxx	4
	4:00:00 PM	-1			-76. xxxx	39. xxxx	5
	4:01:00 PM	-1			-76. xxxx	39. xxxx	6
	4:04:00 PM	-1			-76. xxxx	39. xxxx	7
	4:06:00 PM	-1			-76. xxxx	39. xxxx	8
	4:10:00 PM	-1			-76. xxxx	39. xxxx	9
	4:12:00 PM	-1			-76. xxxx	39. xxxx	10

Note that the minimum number of students per school is 1. This is because some students who attend special schools outside of this school district are also transported by this school district if they live in this school district.

The earliest and latest school dismissal times are 2:00 pm and 3:55 pm respectively. The school's dismissal times are integer multiples of 5 minutes. All trips for one school depart five minutes after the school dismissal time. The travel time of every pair of nodes, including the schools and bus stops, are queried from Google Map Distance API. The travel distance (in meters) and travel time without traffic (in seconds) are both obtained. The students' pick up and drop-off times are estimated using the regression model in (3-35) and (3-36).

Table 27 Summary statistics for case study 2

			Value
Schools	Number of	All schools	93
		Element Schools	40
		Middle Schools	20
		High Schools	12
		Special Schools	20
		Community College	1
	Number of students		1,711
	Students / school	Maximum	135
		Minimum	1
		Average	18.4
		Median	12
Bus stops	Number of bus stops		1,501
	Students / stop	Maximum	8
		Minimum	1
		Average	1.1
		Median	1
Trips (Current)	Number of trips		286
	Students / stop	Maximum	18
		Minimum	1
		Average	5.9
		Median	5
	Trip duration (min)	Maximum	151
		Minimum	4
		Average	38.1
		Median	34
	Stops / trip	Maximum	14
		Minimum	1
		Average	5.2
		Median	5

7.2.2 Deterministic school bell time, bus routing, and scheduling

The routing data we got for case 2 only contains the trip information. The route information, in other words, the scheduling plan is not given. Thus, we solve the Multi-School Bus Scheduling Problem (Model 13) using the trips of case 2. This scheduling plan is noted as the current plan. Notice that the scheduling problem is solved to optimality given the current routing plan. The difference of the current plan to other new proposed plans, in terms of the number of buses and the total vehicle time, purely

comes from the goodness of the routing plans (and school bell time adjustment in the next subsection). Our new routing plans significantly differ from the current plan due to the consideration of trip compatibility.

Since the problem of case 2 is large, the exact method is not applicable, and the parallel method outperforms the sequential method, only the Parallel Minimum Cost Matching with the Post Improvement Algorithm (Algorithm 2) is used to generate the trips. Then, the Multi-School Bus Scheduling Problem (Model 13) is solved to generate the bus routes given the routing plan from the Algorithm 2. Different scenarios are tested, including:

- CUR: Solve the scheduling problem to optimality using the current routing plan;
- MRT45: Generate the trips using Algorithm 2 with Maximum Ride Time = 45 minutes and with current bell times;
- MRT60: Generate the trips using Algorithm 2 with Maximum Ride Time = 60 minutes and with current bell times;
- MRT90: Generate the trips using Algorithm 2 with Maximum Ride Time = 90 minutes and with current bell times.

Note, the current routing plan provided by the school district has slightly different estimation of travel times than Google Map API. For consistency, we keep the sequence of visiting stops in the current routing plan (CUR) and use the travel time estimation from Google Map API as the true travel time for all scenarios. The validity of Google Map API is subject to sensitivity analysis in section 7.2.2.4.

7.2.2.1 Sensitivity analysis - routing coefficients

First, a sensitivity analysis is conducted to find the best combination of the coefficients. As in section 6.2.4, the most influential parameters in Algorithm 2 are the coefficients for the remaining (waste) vehicle capacity (β_Q), the trip-to-school compatibility (β_C) and the travel time (β_T) for evaluating the insertion cost in (4-1). They are noted as the ParaMCM $= (\beta_Q : \beta_C : \beta_T)$. The total number of trips (δ_N), the trip compatibilities (δ_C) and the travel time (δ_T) are the essential coefficients in the surrogate cost estimation (4-11) in the post improvement, which are noted as the ParaPI $= (\delta_N : \delta_C : \delta_T)$. In total, 9 different combinations of the parameters are solved for case 2 with the MRT=90 minutes. The result of the sensitivity test is shown in Table 28. For each of the test scenarios, two sets of results from Algorithm 2 are reported, either with or without the Post Improvement (PI). Overall, the parameter set ParaMCM $= (200:10:1)$ and ParaPI $= (1000:1000:1)$ provides the best and most reliable solutions. Thus, this set of parameters is used for other later analysis.

Table 28 Sensitivity analysis for the coefficients in Algorithm 2 for case study 2

Number of buses		ParaPI					
		(10,10,1)		(100,100,1)		(1000,1000,1)	
		MCM	PI	MCM	PI	MCM	PI
Para MCM	(100,10,1)	94	90	95	90	86	85
	(200,10,1)	96	90	94	90	**85**	**84***
	(500,100,1)	94	90	94	90	86	85

Note: **PI**: number of buses of the solution generated by the MCM followed by the PI; **MCM**: number of buses of the solution generated by the MCM; **ParaMCM**: parameter of the MCM in the form of $(\beta_Q : \beta_C : \beta_T)$; **ParaPI**: parameters for the PI in the form of $(\delta_N : \delta_C : \delta_T)$; *: minimum number of buses

7.2.2.2 School bus routing and scheduling

The solutions for all test scenarios are summarized in Table 29. Clearly, our solutions obtained by Algorithm 2 are much better than the current solutions in terms

of the cost and the level of service. In our solution to the MRT90 with the PI (noted as MRT90PI), only 84 buses are needed compared to the 119 buses in the current operations.

Table 29 Summary of deterministic school bus routing and scheduling plans of case study 2

		CUR	MRT45		MRT60		MRT90	
			MCM	PI	MCM	PI	MCM	PI
Number of buses		**119**	**122**	**117**	**98**	**97**	**85**	**84**
Number of trips		286	251	250	201	201	166	165
RT (sec)	Routing	-	38.9	321.3	31.0	897.0	1504	23.7
	Scheduling	-	39.5	40.0	14.8	17.5	8.2	5.8
Total vehicle time		**10,157**	**9,547**	**10,000**	**9,107**	**9,752**	**9,652**	**8,747**
Total deadhead		887.6	645.0	707.8	605.6	594.6	516.6	540
Vehicle Time /route (min)	**Max**	**137.9**	**138.3**	**141.5**	**157.6**	**153.1**	**176.8**	**185.5**
	Min	20.9	25.3	30.5	27.6	27.6	41.4	35.1
	Mean	81.6	75.6	82.4	89.8	97.5	110.5	100.9
	Median	79.7	77.8	84.6	92.7	106.4	108.5	99.6
Number of trips /route	Max	4	4	4	3	4	4	3
	Min	1	1	1	1	1	1	1
	Mean	2.4	2.1	2.1	2.1	2.1	1.9	1.9
	Median	2	2	2	2	2	2	2
Number of stops /route	Max	29	31	34	29	31	33	31
	Min	1	1	1	2	2	2	2
	Mean	12.6	12.3	12.8	15.3	15.5	17.7	17.9
	Median	11	12	12	17	15	18	19
Number of students /route	Max	37	46	50	41	45	41	39
	Min	1	1	1	2	2	2	2
	Mean	14.4	14.0	14.6	17.5	17.6	20.1	20.4
	Median	12	13	14	18	17	20	19
Ride time /trip (min)	**Max**	**104.2**	**53.8**	**53.8**	**59.9**	**60.0**	**89.8**	**89.8**
	Min	4.4	3.4	3.4	3.4	3.4	3.4	3.4
	Mean	30.5	34.1	35.7	40.7	44.0	53.4	48.0
	Median	27.9	38.9	40.1	44.1	51.2	57.2	43.3
Number of stops /trip	Max	14	17	16	17	16	17	17
	Min	1	1	1	1	1	1	1
	Mean	5.2	5.9	6.0	7.5	7.5	9.0	9.1
	Median	5	6	6	8	8	10	10
Number of students /trip	Max	18	18	18	18	18	18	18
	Min	1	1	1	1	1	1	1
	Mean	5.9	6.8	6.8	8.5	8.5	10.3	10.4
	Median	5	6	6	8	8	11	11

Note: **RT**: running time (second); **CUR:** Current operation plan; **MCM**: the solution generated by the MCM; **PI**: the solution generated by the MCM followed by the PI

Notice that the current operation is based on optimal scheduling. The difference of the number of buses purely results from the routing plans. The total vehicle time is also significantly reduced to 8,747 (MRT90PI) as compared to 10,157 in the current operation. In terms of the cost, the solution from the MRT90PI saves 29% of buses and 14% of the total vehicle time than the current operation.

The level of service from the MRT90PI is also better than the current operation. The maximum ride time per trip is 89.8 minutes in the MRT90PI as compared with the 104.2 minutes in the current operation. The maximum vehicle time in the MRT90PI is slightly higher, which is the tradeoff of reducing the number of buses and the total vehicle time. We also notice that the new plan tends to utilize the bus capacity more wisely than the current operation because the mean and the median bus load almost double their counterparts in the current operation.

Then, we try to find solutions with better service to students by shortening the maximum ride time per trip to 60 minutes and 45 minutes. As expected, the number of buses and the total vehicle time increase. When the MRT=60, the solution still uses fewer buses and the total vehicle time is lower than the current operation, with better service (shorter ride time for students). Even when the MRT is limited to 45 minutes, we still only need 117 buses, which is fewer than the current operation. In plan MRT45, the maximum student ride time is 53.8 minutes. This is because a few bus stops are quite far away from the school such that even if the bus directly travels from the school to the bus stop, the ride time is still higher than 45 minutes. Although there is no way to further reduce the maximum student ride time, the student ride time is kept within 45 minutes if possible. As the maximum student ride time (53.8 minutes) is about half

of the current operation (104.2 minutes), our new routing and scheduling problem outperform the current operation from all perspectives.

After applying the PI, the solutions are even better than the ones without, in terms of fewer number of buses by up to 4% (in the MRT45, 117 buses after the PI as compared to 122 before the PI). It again demonstrates the validity of the Post Improvement algorithm. As the maximum vehicle times in all cases are relatively small (less than 3.1 hours), we did not further constrain the maximum vehicle time constraint.

The running time for a planning problem is not quite important, and the focus of this experiment is not the efficiency of the algorithm. Thus, different test cases are solved on different computers and/or servers with different configurations. The running time are not truly comparable. This holds true for the rest of the experiments.

7.2.2.3 School bell time, bus routing and scheduling

In this section, we optimize the school bell times along with the bus schedules. We assume that a school's dismissal time can be changed within a time window of up to 15 minutes before, and 15 minutes after its original dismissal time while not exceeding the earliest and latest dismissal time (2:00 pm and 3:55 pm). The parameters and the algorithm use the same setup as in the previous section. Notice that we also optimize the bell times and schedules for the current operation assuming the current routing (trips) plan is fixed.

We also consider the case of the MRT45, the MRT60, the MRT90, with and without the Post Improvement (PI). In each of the test problem, the Parallel Minimum Cost Matching with the Post Improvement Algorithm (Algorithm 2) is used to generate the trips and the Multi-School Bus Bell Time, and Scheduling Problem (Model 7) is

solved to generate the bus routes while optimizing the school bell times. The summary of the solutions is summarized in Table 30. The comparison of the solutions with and without the bell time adjustment is summarized in Table 31.

Table 30 Summary of deterministic school bell time, bus routing and scheduling plans of case study 2

		CUR	MRT45		MRT60		MRT90	
			MCM	PI	MCM	PI	MCM	PI
Number of buses		**89**	**101**	**99**	**79**	**78**	**72**	**70**
Number of trips		286	267	266	211	211	172	172
RT (sec)	Routing	-	125.3	681.7	7466	34.7	8415.9	34.8
	Scheduling	-	89.7	106.9	3024	4351	1010.7	4430.7
Total vehicle time		**10,979**	**11,333**	**11,903**	**11,264**	**10,926**	**11,173**	**10,114**
Total deadhead		1,526	1,349	1,488	1,187	1,357	923.4	943
Vehicle Time /route (min)	**Max**	**152.7**	**147.2**	**155.2**	**169.4**	**166.4**	**201.6**	**186.3**
	Min	70.1	45.9	41.9	96.5	56.7	80.5	83.4
	Mean	114.8	105.5	112.7	135.1	131.4	148.8	137.8
	Median	116.9	106.6	111.4	131	132.6	149.5	138.4
Number of trips /route	Max	5	5	4	4	5	4	4
	Min	2	1	1	2	1	1	1
	Mean	3.2	2.6	2.7	2.7	2.7	2.4	2.5
	Median	3	3	3	3	3	2	2
Number of stops /route	Max	37	35	28	35	38	40	38
	Min	5	1	1	5	3	5	4
	Mean	16.9	14.9	15.2	19	19.3	20.8	21.4
	Median	15	13	16	18	19	21	23
Number of students /route	Max	47	40	37	46	44	51	51
	Min	5	1	1	5	3	5	4
	Mean	19.2	16.9	17.3	21.6	21.9	23.8	24.4
	Median	16	15	18	19	20	23	25
Ride time /trip (min)	**Max**	**104.2**	**53.8**	**53.8**	**60.0**	**59.9**	**89.9**	**89.6**
	Min	4.4	3.9	3.9	3.9	3.9	4.4	4.4
	Mean	30.5	34.9	36.4	45	42.2	56.9	50.6
	Median	27.9	39.3	41.2	51.8	47.4	62	45.7
Number of stops /trip	Max	14	15	14	16	16	17	18
	Min	1	1	1	1	1	1	1
	Mean	5.2	5.6	5.6	7.1	7.1	8.7	8.7
	Median	5	5	5	7	7	9	9
Number of students /trip	Max	18	18	18	18	18	18	18
	Min	1	1	1	1	1	1	1
	Mean	5.9	6.4	6.4	8.1	8.1	9.9	9.9
	Median	5	5	6	7	8	10	10

Note: **RT**: running time (second) and the running time in scheduling including the bell time adjustment; **CUR:** Current operation plan; **MCM**: the solution generated by the MCM; **PI**: the solution generated by the MCM followed by the PI; The school bell time and bus scheduling model is terminated when the optimality gap is less than or equal to 2%.

By adjusting the school bell times, even if the routing (trips) remains the same, the number of buses can still be reduced to 89 as compared to 119. The total vehicle time increases a little, which is the side effect of saving in the number of buses. And this 30-bus saving, which is equivalent to 25% improvement, purely results from the maximum change of 15 minutes to the school bell times. Similarly, with the bell time adjustment, the new plan can serve all the students using as little as 70 buses while making sure the maximum ride time is less than 90 minutes. Comparing to the current operations without bell time optimization (119 buses), the MRT90PI with bell time optimization only needs 70 buses, which is 41% improvement.

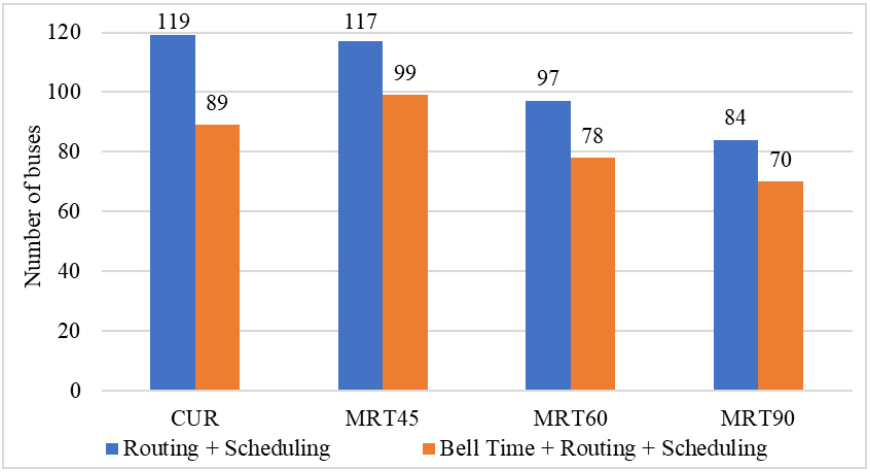
Table 31 Comparison of the deterministic solutions with and without the bell time adjustment of case study 2

Reduction		CUR	MRT45		MRT60		MRT90	
			MCM	PI	MCM	PI	MCM	PI
NOB	Value	**30**	**21**	**18**	**-3**	**19**	**12**	**15**
	Ratio	25%	17%	15%	-3%	20%	14%	18%
TVT	Value	**-756**	**-1,725**	**-1,836**	**-2,105**	**-1,121**	**-1,481**	**-1,326**
	Ratio	-7%	-19%	-19%	-23%	-11%	-17%	-14%

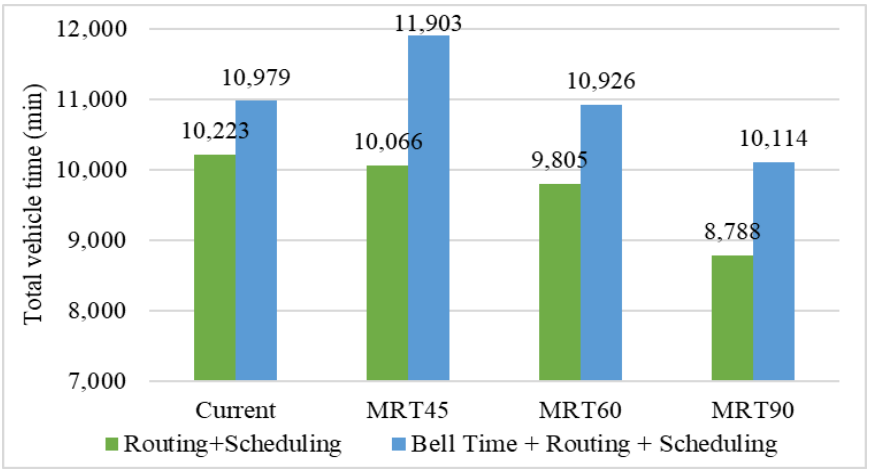
Note: **NOB Value** = number of buses without the bell time adjustment – number of buses with the bell time adjustment; **NOB Ratio** = NOB Value / number of buses without the bell time adjustment; **TVT Value** = total vehicle time without the bell time adjustment – total vehicle time with the bell time adjustment; **TVT Ratio** = TVT Value / total vehicle time without the bell time adjustment; **CUR:** Current operation plan; **MCM**: the solution generated by the MCM; **PI**: the solution generated by the MCM followed by the PI.

For the MRR90PI, as compared to the plan without the bell time adjustment, the only tradeoff in the bell time optimized plan (Table 31) is that the total vehicle time increases from 8,788 minutes to 10,114 minutes (Table 30), which corresponds to a 14% increase. However, the huge financial saving due to the bus reduction (15 buses, 18%) can easily justify the small increase in the vehicle time.

The summary of the solutions using the current routing plan (CUR), and the optimized routing plans (MRT45, MRT60, MRT90) with and without the bell time optimization are summarized in Figure 20.



(a) Number of buses



(b) Total vehicle time

Figure 20 Comparison of the solutions for case study 2 under deterministic condition

With the improvement of the service (the MRT60 and the MRT45), the number of bused required increases. Under the best service (the MRT45), after the bell time optimization, 99 buses are required. It is slightly higher the solution if optimizing the bell time while keeping current routing (89 buses). However, as the maximum ride time

of the current routing plan is 104 minutes, the MRT45 provides a significant better service where the students will not stay on a bus more than 45 minutes (unless the direct travel time from the school to the bus stop is higher than 45 minutes). From another perspective, the total vehicle times of our proposed solutions (the MRT45, the MRT60, and the MRT90) are all smaller than the current operation.

7.2.2.4 Sensitivity analysis - travel time buffer coefficients

One question is how accurate the "travel time without traffic (from Google Map Distance API)" is. As the real-world operation is fully uncertain, several concerns may arise:

1) the "travel time without traffic" may not fully represent the true free-flow traffic condition;

2) the "travel time without traffic" may not fully represent the true traffic condition at different departure times with different levels of congestion;

3) the "travel time without traffic" may not fully represent the true traffic condition (even if we specify the departure time) due to the uncertainty of the traffic.

Thus, we consider multiplying a buffer coefficient to the "travel time without traffic" from Google Map Distance API and treating this new value as the true travel time. We observe the travel time fluctuation of several node pairs throughout the whole day and their most congested travel times are approximately twice their free-flow travel times. Hence, we change the buffer coefficient from 1.0 to 2.0. For each of the buffer adjusted scenario, the new travel times of every pair of nodes is the product of the "travel time without traffic" from Google Map Distance API and the buffer coefficient.

The results of the Multi-School Bell Time, Bus Routing and Scheduling Optimization with the MRT=90 minutes using buffer adjusted travel time is shown in Table 32.

Table 32 Sensitivity analysis for the buffer coefficients of travel time from the Google Map API

		CUR	Buffer coefficient					
			1.0	1.2	1.4	1.6	1.8	2.0
Number of buses		**89**	**70**	**84**	**97**	**113**	**131**	**150**
Number of trips		286	172	185	196	216	233	261
RT (sec)	Routing	-	34.8	2255.9	27	820.5	556.1	680.8
	Scheduling	-	4430.7	169.5	282.1	18.3	16.1	24.1
Total vehicle time		**10,979**	**10,114**	**12,759**	**13,647**	**16,826**	**18,480**	**20,965**
Total deadhead		1,526	943	954	1,106	1,211	1,136	1,430
Vehicle Time /route (min)	**Max**	**152.7**	**186.3**	**199.3**	**190.4**	**196.6**	**200.1**	**200.5**
	Min	70.1	83.4	66.5	61.5	62.1	55.8	68.1
	Mean	114.8	137.8	146.1	134.9	143.5	136.7	134.9
	Median	116.9	138.4	151.5	137.2	152.5	144.1	137.3
Number of trips /route	Max	5	4	4	3	3	3	3
	Min	2	1	1	1	1	1	1
	Mean	3.2	2.5	2.2	2	1.9	1.8	1.8
	Median	3	2	2	2	2	2	2
Number of stops /route	Max	37	38	36	29	28	25	24
	Min	5	4	3	2	1	1	1
	Mean	16.9	21.4	17.8	15.5	13.3	11.5	10
	Median	15	23	18.5	16	14	12	9.5
Number of students /route	Max	47	51	43	40	31	31	27
	Min	5	4	3	2	1	1	1
	Mean	19.2	24.4	20.4	17.6	15.1	13.1	11.4
	Median	16	24.5	21	18	16	14	11
Ride time /trip (min)	**Max**	**104.2**	**89.6**	**89.6**	**89.8**	**89.9**	**96.3**	**106.9**
	Min	4.4	4.4	5	5.6	5.7	4.5	5.2
	Mean	30.5	50.6	61.2	61.2	69.5	71.9	72.1
	Median	27.9	45.7	70.5	66.9	79.6	82.3	82.1
Number of stops /trip	Max	14	18	17	17	16	17	16
	Min	1	1	1	1	1	1	1
	Mean	5.2	8.7	8.1	7.6	6.9	6.4	5.8
	Median	5	9	8	8	7	6	5
Number of students /trip	Max	18	18	18	18	18	18	18
	Min	1	1	1	1	1	1	1
	Mean	5.9	9.9	9.2	8.7	7.9	7.3	6.5
	Median	5	10	9	8.5	7	7	6

Note: **RT**: running time (second) and the running time in scheduling including the bell time adjustment; **CUR:** Current operation plan; **MCM**: the solution generated by the MCM; **PI**: the solution generated by the MCM followed by the PI; The school bell time and bus scheduling model is terminated when the optimality gap is less than or equal to 2%.

Since the solutions after the Post Improvement (Algorithm 2-2) are better than the ones without the Post Improvement, only solutions after the Post Improvement are reported in Table 32.

With the increase of the buffer coefficient, the transportation cost (the number of buses and the total vehicle time) increases. When the travel time is increased by 20%, 40%, 60%, 80% and 100%, additional 14, 27, 43, 61 and 80 buses are required as compared to the case where the buffer coefficient is 1.0, which corresponds to 20%, 39%, 61%, 87% and 114% respectively. The comparison of the solutions, in terms of the number of buses and the total vehicle time, under different buffer coefficients are summarized in Figure 21.

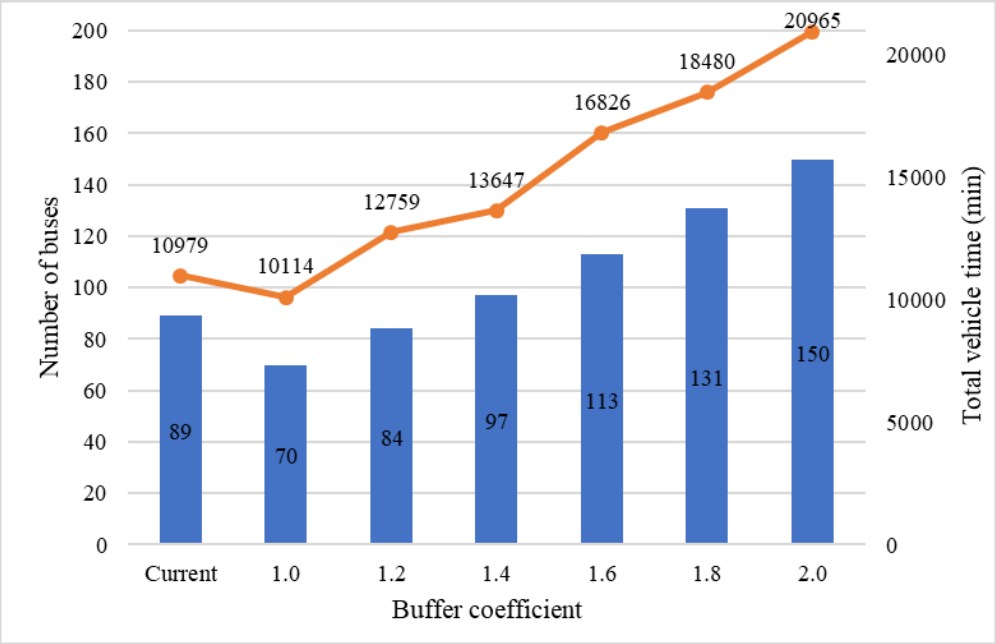


Figure 21 Comparison of the solutions for case study 2 under different buffer coefficients

The result shows that the number of buses increases at a relatively same rate compared with the increase in travel time. Note the travel times for every pair of nodes in the network is multiplied by the buffer coefficient. The real situation may not be as

bad. Other changes also occur along with the increase of the buffer coefficients. First, total vehicle time significantly increases. To satisfy the maximum ride time constraint under the buffer adjusted conditions, the trips and the bus routes tend to be shorter:

- The maximum number of trips per route decreases from 5 to 3;
- The mean number of trips per route decreases from 3.2 to 1.8,
- The maximum number of stops per route decreases from 38 to 24;
- The mean number of stops per route decreases from 21.4 to 10.0;
- The maximum number of students per route decreases from 51 to 27;
- The mean number of students per route decreases from 24.4 to 11.4.

The estimated travel time provided by the school district on average is 13% higher than the "travel time without traffic" from Google Map Distance API. Looking at our buffer coefficient adjusted solutions, even if every edge's travel time in the network is 20% higher than the "travel time without traffic" from Google Map Distance API, our solution is still better than the current operation – 84 buses in comparison to 89 buses while the student maximum ride time is 90 minutes as compared to 104 minutes. Generally, our solution is more efficient than the current operations even under more unreliable and congested conditions.

7.2.2.5 Stochastic on-time test for deterministic results

Because we do not have the historical observation of the travel times for case 2, we generate the stochastic trip travel times and the deadhead duration the same way as in section 6.4.1. The only difference is that we assume the travel times and deadheads obtained from the Google Map Distance API are the minimum travel times and deadheads under the free flow traffic condition. Under the stochastic case, the real

travel times and deadheads would be the same or higher than the travel times and deadheads from Google Map Distance API. They would never be smaller than the travel times and deadheads under the free flow traffic.

At a specific departure time, we generate 1000 (sample size) observations of a trip's travel time from a normal distribution. The mean of this normal distribution is changed at different departure times. For instance, if a school originally dismisses at 2:15 PM. At 2:00 PM (the earliest possible dismissal time), the mean of the normal distribution is the travel time directly obtained from the Google Map Distance API. At 2:30 PM (the latest possible dismissal time), the mean of the normal distribution is 110% (Amplitude=10%) of the travel time from the Google Map Distance API. Between 2:00 PM and 2:30 PM, the mean of this normal distribution linearly increases. The standard deviation of this normal distribution is 20% of the travel time from Google Map Distance API. The stochastic deadhead duration between pairs of trips are generated in the same way we generate the stochastic trip travel time except that the standard deviation is 10% of the deadhead duration from Google Map Distance API. Another 1000 observations (test size) of the travel time and the deadhead are generated along with the sample data. And the second 1000 observation is reserved as the test data.

The first question is, under this level of uncertainty how does the deterministic bus routing and scheduling plan perform? We use the test data to simulate its performance, and the results are summarized in Table 33. Clearly, without consideration of the uncertainty of travel time, all deterministic scheduling plans perform poorly. In the current operation without the bell time adjustment, the minimum probability of being on time is as little as 35%. The same trend occurs in other scenarios.

In the most cost-saving scenario (the MRT90PI with bell time adjustment), up to 96% of the bus routes have a probability of being on time less than or equal to 98%. Although 98% seems a pretty high standard, the school board is expected to provide the best transportation service, which includes the punctuality, to students. The bus routes need to be on time almost surely. Moreover, even if we reduce the probability of being on time to 80%, still 73% of the bus routes in the MRT90PI with bell time adjustment would violate this stochastic on-time constraint.

Table 33 Stochastic on-time test for deterministic bus routing and scheduling plans of case study 2

		CUR	MRT45		MRT60		MRT90	
			MCM	PI	MCM	PI	MCM	PI
Without the bell time adjustment								
Prob	Max	100%	100%	100%	100%	100%	100%	100%
	Min	**35%**	**38%**	**26%**	**34%**	**27%**	**27%**	**36%**
	Mean	88%	85%	81%	79%	78%	79%	74%
ρ=0.8	N<ρ	35	40	50	45	47	37	47
	R<ρ	29%	33%	43%	46%	49%	44%	56%
ρ=0.98	N<ρ	52	64	68	61	59	49	56
	R<ρ	44%	53%	58%	62%	61%	58%	67%
With the bell time adjustment								
Prob	Max	100%	100%	100%	100%	100%	100%	100%
	Min	**30%**	**27%**	**18%**	**34%**	**35%**	**31%**	**38%**
	Mean	74%	80%	79%	69%	62%	71%	68%
ρ=0.8	N<ρ	54	45	45	53	68	44	51
	R<ρ	61%	45%	46%	67%	87%	61%	73%
ρ=0.98	N<ρ	70	65	63	70	76	62	67
	R<ρ	79%	64%	64%	89%	97%	86%	96%

Note: **Prob**: the probability of being on time; **N<ρ**: Number of bus routes with the probability of being on time less than or equal to ρ; **R<ρ**: Ratio of bus routes with the probability of being on time less than or equal to ρ; **CUR:** Current operation plan; **MCM**: the solution generated by the MCM; **PI**: the solution generated by the MCM followed by the PI.

There is a clear tradeoff between reducing the number of buses and providing the most punctual bus service. When we focus on reducing the number of buses, we prefer short idle time (the time difference between the bus arrival time and its departure

time). The shorter the idle time is, the more efficient the bus routes are, and fewer number of buses are required. However, the shorter idle time is, the smaller buffer the routes have to recover to the original schedule in case of a delay. Hence, it has a higher chance to be late. Furthermore, reducing the cost tends to have more trips on a route while improving punctuality results in fewer trips in a route.

Note when generating the stochastic data, we assume everything pair of nodes has some uncertainty and the travel time is higher than the free flow travel time. The real-world condition may not be this bad. The analysis shows the on-time performance of the deterministic solutions under the worst case. Further sensitivity analysis is conducted based on different level of uncertainty.

7.2.3 Stochastic school bell times and bus scheduling

7.2.3.1 School bus scheduling

First, we optimize the school bus schedules using the Column Generation-based Stochastic Multi-school Bell Time and Scheduling (Algorithm 4) except that the bell time remains unchanged in this section. All trips are assumed to be fixed, either the current trips or the deterministic trips from section 7.2.2. Since the solutions with the Post Improvement (PI) outperforms the ones without the PI, only trip plans resulting from the PI are adopted. Two levels of punctuality are checked: 80% and 98%. The stochastic bus scheduling solutions are summarized in Table 34.

Obviously, the new stochastic bus scheduling plans significantly improve the punctuality of the bus service. There are some cases that the minimum probability of being on time is slightly lower than the required level. However, the difference is quite small. Such a small difference is acceptable because Algorithm 4 is performed based

on the sample data while the test is conducted on the test data. On average, the minimum probability of being on time is improved from 31% (deterministic solutions) to 78% and 97%, given the different levels of required punctuality (ρ), respectively.

Table 34 Summary of stochastic bus scheduling plans of case study 2

		CUR		MRT45		MRT60		MRT90	
ρ		0.8	0.98	0.8	0.98	0.8	0.98	0.8	0.98
Number of buses		**127**	**131**	**131**	**146**	**116**	**127**	**100**	**108**
Number of trips		286	286	250	268	201	201	165	165
Total vehicle time		**10007**	**10793**	**10510**	**10989**	**10199**	**10121**	**9074**	**9077**
Total deadhead		912.2	1,047	729.4	720.4	556.2	503.8	468.2	469.8
RT (sec)		25499	4486	905.4	710.3	329.2	239.2	135.7	150.2
Vehicle Time /route	Max	**128.9**	**122.1**	**150.1**	**134.7**	**144.3**	**143.2**	**152.9**	**149.9**
	Min	17.3	17.1	22.6	29.9	29	16.7	27.3	16.7
	Mean	75.2	78.4	77.4	72.8	85.5	77.8	88.4	81.9
	Median	80.1	80.4	84.9	80.2	83.9	62.9	87.3	86.2
Number of trips /route	Max	4	4	3	4	3	3	3	3
	Min	1	1	1	1	1	1	1	1
	Mean	2.1	2.2	1.9	1.8	1.7	1.6	1.7	1.5
	Median	2	2	2	2	2	2	2	1
Number of stops /route	Max	29	26	31	28	29	32	30	31
	Min	1	1	1	1	2	1	2	1
	Mean	10.8	11.5	11.5	10.3	12.9	11.8	15	13.9
	Median	11	11	11	10	12	11	15	14
Number of students /route	Max	34	37	39	40	37	39	42	42
	Min	1	1	1	1	2	1	2	1
	Mean	12.4	13.1	13.1	11.7	14.8	13.4	17.1	15.8
	Median	11.5	12	13	11	13.5	12	16	15
Prob	Max	100%	100%	100%	100%	100%	100%	100%	100%
	Min	**84%**	**97%**	**78%**	**97%**	**78%**	**98%**	**78%**	**97%**
	Mean	98%	100%	97%	100%	97%	100%	97%	100%
ρ	N<ρ	0	4	2	5	2	2	2	3
	R<ρ	0%	3%	2%	3%	2%	2%	2%	3%

Note: **RT**: running time (second); **CUR:** Current operation plan; **MCM**: the solution generated by the MCM; **PI**: the solution generated by the MCM followed by the PI; **Prob**: the probability of being on time; **N<$\boldsymbol{\rho}$**: Number of bus routes with the probability of being on time less than or equal to ρ; **R<$\boldsymbol{\rho}$**: Ratio of bus routes with the probability of being on time less than or equal to ρ.

The side effect of the improvements in punctuality is the increase in the number of buses. To maintain an 80% probability of punctuality, additional 8, 14, 19, 16 buses are needed as compared to the deterministic CUR, MRT45, MRT60, and MRT90 bus scheduling solutions, respectively. Furthermore, additional 12 (for the CUR), 29 (for

the MRT45), 30 (for the MRT60) and 24 (for the MRT90) buses are needed for a more reliable service (98% probability of being on time). On average, additional 14 buses (15%) are needed to maintain 80% probability of being on time as compared to the deterministic solutions and 24 buses (24%) are needed to maintain 98% confidence level to be on time.

The balance of the total cost and the level of service is an essential question that the school board and the bus service provider need to consider. Again, the above conclusion is based upon the level of uncertainty we considered in generating the stochastic data, which may not fully represent the real-world conditions. Still, it demonstrates the validity of Algorithm 4 and the difference between the deterministic and stochastic models.

7.2.3.2 School bell times and bus schedules

Now, we try to optimize the school bell times along with the bus schedules under the stochastic travel times using the Column Generation-based Stochastic Multi-school Bell Time and Scheduling (Algorithm 4). The trips from the deterministic cases with the Post Improvement are considered fixed and the bell times can be adjusted to no more than 15 minutes before and after the current bell times. Also, the bell times cannot exceed the current earliest and latest bell times (2:00 PM and 3:55 PM). The solutions are summarized in Table 35.

As expected, the stochastic school bell times and bus schedules satisfy the Stochastic On-Time constraints (SOT) with only a few exceptions. Punctuality is significantly improved as compared to the deterministic models. On average, 19 additional buses (19%) are required to maintain the 80% probability of being on time

compared to the deterministic school bell times and bus scheduling optimization plans (section 7.2.2.3). To further satisfy the 98% confidence of being on time, additional 35 buses (42%) are required. Such difference is similar to the case without the bell time adjustment (section 7.2.3.1).

Table 35 Summary of stochastic school bell time and bus scheduling plans of case study 2

		CUR		MRT45		MRT60		MRT90	
ρ		0.8	0.98	0.8	0.98	0.8	0.98	0.8	0.98
Number of buses		**102**	**114**	**119**	**136**	**105**	**121**	**84**	**104**
Number of trips		286	286	266	266	211	211	172	172
Total vehicle time		**11257**	**11040**	**11660**	**11428**	**10433**	**10314**	**10055**	**9858**
Total deadhead		1364	1220	969.0	814.6	759.0	660.8	619.8	488.6
RT (sec)		66452	91998	13972	36659	3276	812.3	4028	3000
Vehicle Time /route	Max	**160.4**	**145.4**	**150.7**	**138.2**	**167.2**	**150.4**	**185.1**	**176.5**
	Min	37	30.2	9.8	9.8	19.7	19.9	44.6	29.8
	Mean	103.7	91.5	93.9	81	95.7	82.5	116.0	92.4
	Median	103.6	93.6	95.5	88.3	101.8	78.3	120.4	88.8
Number of trips /route	Max	5	4	4	4	4	4	4	3
	Min	1	1	1	1	1	1	1	1
	Mean	2.8	2.5	2.2	11	2	1.7	2.0	1.6
	Median	3	3	2	11	2	1	2	1
Number of stops /route	Max	35	32	27	34	40	30	37	29
	Min	1	1	1	1	2	2	2	1
	Mean	14.7	13.2	12.6	12.6	14.3	12.4	17.9	14.4
	Median	14.5	13	12	12	14	12	18	14
Number of students /route	Max	42	37	33	34	51	38	48	45
	Min	1	1	1	1	2	2	2	1
	Mean	16.8	15	14.4	12.6	16.3	14.1	20.4	16.5
	Median	16	14	13	12	16	14	19	16
Prob	Max	100%	100%	100%	100%	100%	100%	100%	100%
	Min	**75%**	**97%**	**76%**	**97%**	**77%**	**97%**	**78%**	**97%**
	Mean	95%	100%	96%	100%	94%	100%	93%	100%
ρ	N<ρ	4	7	8	4	6	7	3	2
	R<ρ	4%	6%	7%	2%	6%	6%	4%	2%

Note: **RT**: running time (second); **CUR:** Current operation plan; **MCM**: the solution generated by the MCM; **PI**: the solution generated by the MCM followed by the PI; **Prob**: the probability of being on time; **N<ρ**: Number of bus routes with the probability of being on time less than or equal to ρ; **R<ρ**: Ratio of bus routes with the probability of being on time less than or equal to ρ.

The comparison of the solutions with and without the bell time adjustment is summarized in Table 36. When comparing the stochastic bus schedule without the bell time adjustment (section 7.2.3.1), on average 13 buses (10%) can be saved due to the bell time adjustment. Comparing the small perturbation caused by changing the bell times (no more than 15 minutes) and the huge financial benefits (including the bus acquisition, drivers' salary, maintenance cost, etc.), the bell time adjustment is certainly beneficial.

Table 36 Comparison of the stochastic solutions with and without the bell time adjustment of case study 2

Reduction		CUR		MRT45		MRT60		MRT90	
		0.8	0.98	0.8	0.98	0.8	0.98	0.8	0.98
NOB	Value	**25**	**76**	**12**	**10**	**11**	**6**	**16**	**4**
	Ratio	20%	40%	9%	7%	9%	5%	16%	4%
TVT	Value	**-1250**	**-942**	**-1150**	**-439**	**-234**	**-193**	**-981**	**-781**
	Ratio	-12%	-9%	-11%	-4%	-2%	-2%	-11%	-9%

Note: **NOB Value** = number of buses without the bell time adjustment – number of buses with the bell time adjustment; **NOB Ratio** = NOB Value / number of buses without the bell time adjustment; **TVT Value** = total vehicle time without the bell time adjustment – total vehicle time with the bell time adjustment; **TVT Ratio** = TVT Value / total vehicle time without the bell time adjustment; **CUR:** Current operation plan; **MCM**: the solution generated by the MCM; **PI**: the solution generated by the MCM followed by the PI.

The comparison, with respect to the number of buses of all stochastic solutions of case 2 with and without the bell time adjustment, is shown in Figure 22. In general, our proposed plans are better than the current operation. The number of required buses decreases when the maximum ride time is increased. By optimizing the school bell times, the number of buses decreases. To maintain a certain level of punctuality, a few more buses are needed in the stochastic case as compared to its counterpart in the deterministic case. As the school board needs to balance the total cost (dominated by the number of buses) and the level of service (punctuality and maximum ride time),

Figure 22 provides an overview of different options from which the school board can choose one to fit their needs.

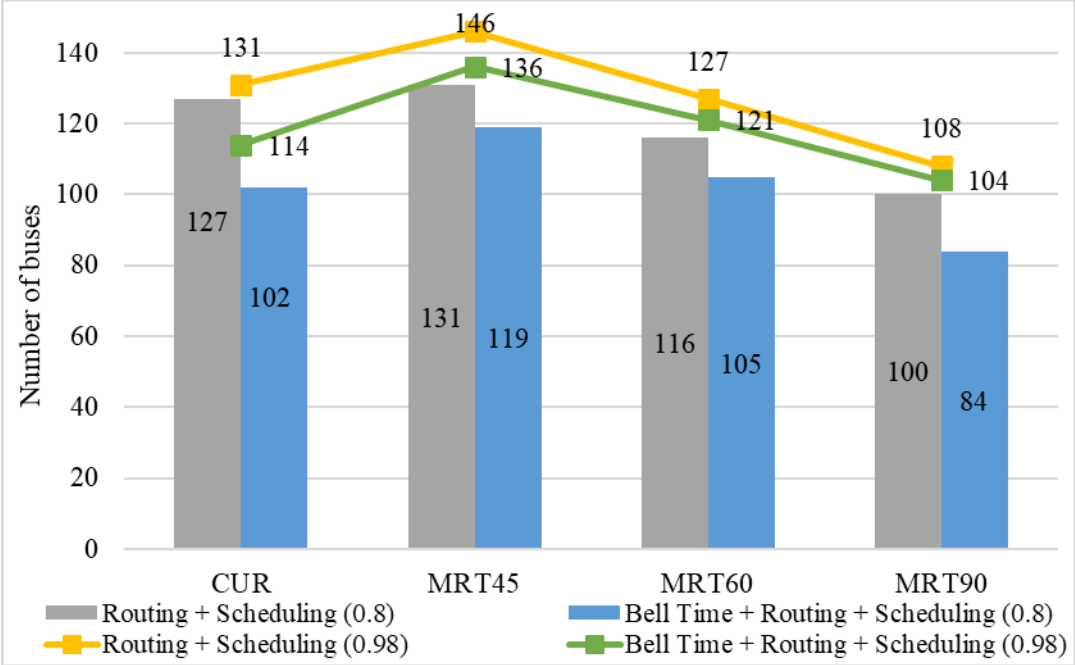


Figure 22 Comparison of all stochastic solutions for case study 2

7.2.3.3 Sensitivity analysis – level of uncertainty

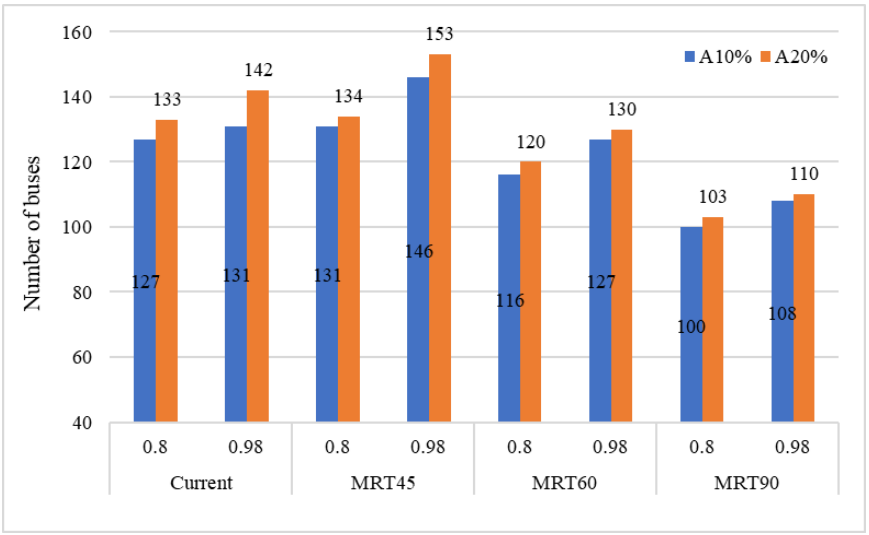
In section 7.2.2.5, we are generating the stochastic travel times and deadhead durations assuming the mean of the normal distribution could be 10% higher than the free flow travel times and deadhead durations (Amplitude = 10%). Now, we assume the amplitude is 20%. Other parameters remain the same. A new set of stochastic sample and test data are generated. We optimize the school bus schedules under the new stochastic travel times and deadheads with the amplitude equals to 20%. The results are shown in Table 37 and Figure 23.

Table 37 Summary of stochastic bus scheduling plans of case study 2 (Amplitude=20%)

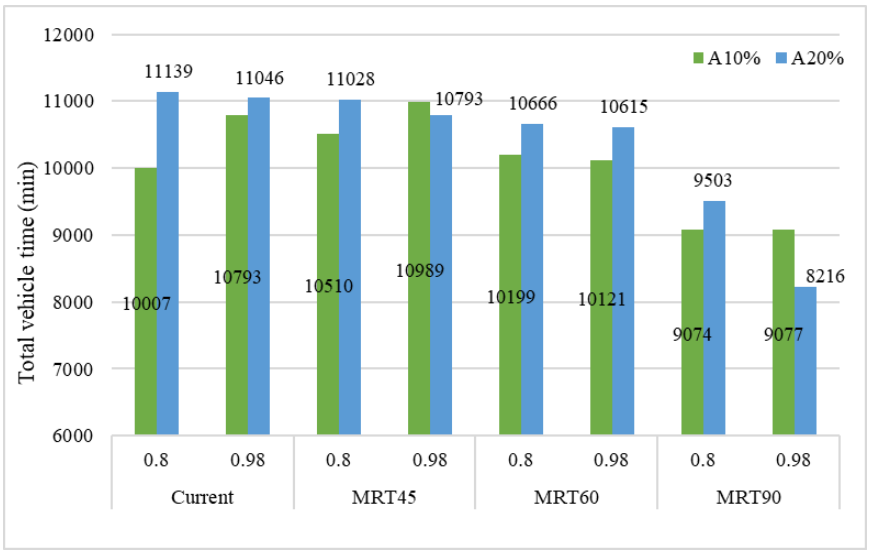
		CUR		MRT45		MRT60		MRT90	
ρ		0.8	0.98	0.8	0.98	0.8	0.98	0.8	0.98
Number of buses		**133**	**142**	**134**	**153**	**120**	**130**	**103**	**110**
Number of trips		286	286	250	250	201	201	165	165
Total vehicle time		**11138**	**11046**	**11027**	**10792**	**10665**	**10614**	**9502**	**8215**
Total deadhead		949.8	890.6	737.0	580.4	527.8	494.2	459.2	314.6
RT (sec)		2602.6	33169	1031	688.8	241.8	220.5	221.5	137.5
Vehicle Time /route	Max	**136.5**	**131.6**	**147.1**	**124.7**	**159.6**	**148.2**	**191.7**	**159.4**
	Min	17.9	18.3	23.4	14.7	29.5	16.9	17.2	17.4
	Mean	80.2	74.6	79.5	68.6	86.7	79.8	90	73.3
	Median	80.6	73.6	86.7	64.2	77.5	65.4	91.8	89.7
Number of trips /route	Max	4	4	3	3	3	3	3	3
	Min	1	1	1	1	1	1	1	1
	Mean	2.1	2	1.8	1.6	1.7	1.5	1.6	1.2
	Median	2	2	2	2	2	1	2	1
Number of stops /route	Max	32	32	28	24	38	31	37	32
	Min	1	1	1	1	2	1	1	1
	Mean	22.2	10.6	11.2	9.8	12.5	11.5	14.5	11.5
	Median	10	10	10	9	12	10	15	13
Number of students /route	Max	37	37	34	33	43	39	43	43
	Min	1	1	1	1	2	1	1	1
	Mean	12.8	12	12.7	11.2	14.3	13.2	16.6	12.9
	Median	11	10	11	10	13	11	16	14
Prob	Max	100%	100%	100%	100%	100%	100%	100%	100%
	Min	**80%**	**98%**	**78%**	**96%**	**79%**	**97%**	**79%**	**97%**
	Mean	98%	100%	97%	100%	97%	100%	97%	100%
ρ	N<ρ	1	1	2	7	1	6	3	2
	R<ρ	1%	1%	1%	5%	1%	5%	3%	2%

Note: **RT**: running time (second); **CUR:** Current operation plan; **MCM**: the solution generated by the MCM; **PI**: the solution generated by the MCM followed by the PI; **Prob**: the probability of being on time; **N<ρ**: Number of bus routes with the probability of being on time less than or equal to ρ; **R<ρ**: Ratio of bus routes with the probability of being on time less than or equal to ρ.

As the traffic becomes more congested (amplitude = 20%), on average, 5 more buses (4%) are needed compared with the less congested scenarios (amplitude = 10%). In terms of the total vehicle time, an average of 279 minutes (3%) increase is observed.



(a) Number of buses



(b) Total vehicle time

Figure 23 Comparison of the stochastic bus scheduling plans of case study 2 (Amplitude=10% and 20%)

7.3 Chapter summary

In this chapter, we used both the deterministic and stochastic models and algorithm to optimize the bus transportation plan for a real-world school district. Both the general-purpose bus transportation (section 7.1) and bus transportation with special needs (section 7.2) were tackled. The results provided strong evidence of the capability of the proposed algorithms in optimizing the large-scale real-world problems. It demonstrated the benefits of considering trip compatibility in the routing stage, the bell time adjustment as well as the reliability of the stochastic algorithms.

From the operation's point of view, the general-purpose bus transportation can reduce the number of buses from 324 to 304 (without the bell time adjustment) and 295 (with the bell time adjustment), which could save 1.2 – 2.0 million dollars annually. For the bus transportation with special needs, the same transportation task can be carried out using 84 buses (without the bell time adjustment) and 70 (with the bell time adjustment) with better service (smaller maximum ride time per trip) as compared to its current 119 buses. When the considering the stochasticity of the travel time, 104 buses are enough to provide service with maximum ride time per trip less than or equal to 90 minutes and 98% probability of punctuality with the optimization of the school bell times.

Chapter 8 Conclusions and Future Research

This dissertation shed light on the problem of designing the school bus routing and scheduling plan while simultaneously optimizing the school bell times. Both deterministic and stochastic travel times were considered. In the deterministic case, one integrated model and several relaxed models were developed to solve the problem with different scopes and assumptions. Three heuristic algorithms were proposed to solve these models. Furthermore, two Mixed Integer Programming (MIP) models and one Column Generation-based (CG) heuristic algorithm were proposed to solve the stochastic problem. Still, there are places where the models and algorithms can be improved.

8.1 Summary

The main contribution of this dissertation lies in the following aspects. First, it demonstrates the importance of the coordination between the school bell times and transportation system including the bus routing and scheduling plans. The development of the mathematical model for the integrated problem proves that it is possible to simultaneously optimize the school bell time, the school bus routing plan and the scheduling plan and that the integrated model can provide a better plan than optimizing the subproblems separately.

Second, this dissertation shows that the decomposition approach can help to solve the complicated integrated problem. A good decomposition approach can significantly reduce the problem size without diminishing the solution quality. This novel decomposition approach in Algorithm 1 blurs the boundaries between the school

bus routing and scheduling problem. Information from scheduling problem (namely the compatibility) is considered in the routing problem. This 'look ahead' strategy is proved to be effective and provides a successful example for similar decomposition approaches or sequential decisions.

Third, a fast heuristic algorithm (Algorithm 2) is proposed to solve the Single-School Bus Routing Problem with the adoption of the 'look ahead' strategy. It combines the Insertion Algorithm, the Simulated Annealing, and the Tabu Search. This algorithm is shown to outperform state-of-the-art methods on the benchmark problems.

This dissertation provides a practical approach to tackle complicated level-of-service constraints. Those level-of-service constraints, if costly to formulate, can be relaxed at the beginning. Then, the post-improvement can be applied to adjust the solution (to the relaxed problem) to satisfy the level-of-service constraints. One example is Algorithm 3, which solves the complicated problem (the MBSV, Model 5) by iteratively solving the simple, relaxed problem (the MBS, Model 7) with the exclusion of illegal solutions that violate the level-of-service constraints.

Furthermore, this dissertation addresses the concern of the stochasticity. A time-dependent Mixed Integer Programming model is proposed to incorporate the traffic congestion at different departure times. A chance-constrained Mixed Integer Programming model is then developed under the assumption of normally distributed travel times. A more general Column Generation-based method (Algorithm 4) is proposed, which simultaneously considers the traffic congestion at different times and the uncertainty at a particular departure time.

From a practical point of view, this dissertation provided several mathematical models and algorithms. Each model and algorithm can be used separately to solve different school bus planning problems with different problem scope and assumptions with different computing power. From an engineer's perspective, these models and algorithms are verified to be able to solve real-world large-scale school bus planning problems.

In this dissertation, we concentrate on the afternoon school bus transportation plan. But these methodologies are applicable to optimize the morning bus transportation plan. As there are some subtle differences, for instance, the traffic might be more congested in the morning peak, they can be easily incorporated into the methods. The core mechanism remains the same. In the afternoon plan, the start time of each trip is fixed once the school dismissal time is determined. We iteratively add successive bus stops until reaching the vehicle capacity and/or the maximum ride time constraint. While in the morning plan, the finish time of each trip is fixed along with the school bell time. Preceding bus stops are iteratively added. In the estimation of trip compatibility, the trip-to-school compatibility in the afternoon plan is substituted by the school-to-trip compatibility in the morning plan. From the methodological point of view, the morning and the afternoon school bus planning optimization problems are identical. All the models and algorithms we proposed in this dissertation can be extended to optimize the morning school bus planning problem.

8.2 Conclusion

Application of operations research techniques is very powerful in improving the efficiency of the operations of complicated transportation systems, yet its

importance is neglected. It is beyond a human brain's computing ability to optimize such a huge and complex multi-school bell time, bus routing and scheduling problem with hundreds of schools, thousands of bus stops and tens of thousands of students. But with well-developed Operations Research models and algorithms, this seemingly untouchable problem, although not to optimality, can be solved with remarkably high quality. Millions of dollars can be saved, and better service can be provided with higher punctuality and shorter travel time for students and drivers. Other transportation systems can also be optimized using operations research.

Moreover, previous research concentrated on the school bus routing and scheduling while treating the school bell time as a fixed parameter. In this research, however, we showed that the entire school bus transportation system could be significantly improved if we make a moderate adjustment to the school bell times. In practice, the school boards can embrace the idea of optimizing the school bell times if it shows enough financial benefits. The bottom line is everything is open to change if the benefit is higher than the cost, or at very least open for discussion.

Another lesson is how to tackle a new sophisticated problem that does not have a well-developed solution method. Decomposition is often a promising start point. The integrated multi-school bell time, bus routing and scheduling problem is unbelievably hard, but it can certainly be divided into small pieces where each piece is much easier to solve. The essence is how to divide the problem such that the relation between the subproblems is kept as much as possible. There is no common rule, and it should be tailored based on the problems. In this dissertation, we used the 'look ahead' strategy, joining the routing and scheduling by the trip compatibility. There would be other

decomposition ideas that suit better for other problems. The balance between the model integration and decomposition is as much an art as a science.

In addition, borrowing and combining several existing methods properly could lead to a new method that is more powerful than the simple combination of the individuals. The hybrid Simulated Annealing and Tabu Search algorithm incorporates the merits of the two, expanding the search area by accepting temporary worse solutions with some probability and improving efficiency by forbidding wasteful searches. This judicious combination results in a better method than the state-of-the-art method.

Furthermore, we are amazed by the drastic change due to the tiny modifications of the model inputs and assumptions, in almost every aspect: model formulation, algorithm development, and the output. Both the relaxation of the maximum vehicle time constraint and the change from the trip-to-trip compatibility to the trip-to-school compatibility, for instance, radically simplify the problem. The takeaway is that a certain level of simplification may be a good approach for framing an elaborate problem and solving it efficiently. Post improvement can be further used to adjust the solution to address the previous simplifications. In real-world applications, completion is greater than perfection.

Last, we observe the trade-off between the transportation cost and the level of service in the system design. Usually, better service in terms of higher punctuality and shorter student ride time, would lead to higher transportation cost, namely more buses and higher total travel time/distance. Operations research techniques are powerful to find a solution that optimizes a certain objective, either minimizing the transportation

cost or improving the level of service. But to balance between these objectives, especially in the real-world with the budget, labor, and time constraints, a human's judgment is needed to make the mathematical models applicable to improve real-world life. It is equally important to understand the limitations of the operations research techniques as well as their capability.

8.3 Future research

There are some avenues for future research where the models and algorithms proposed in this dissertation can be further studied and improved. First, two level-of-service constraints are considered in this dissertation, namely the maximum ride time (per trip) and the maximum vehicle time (per route). There are other level-of-service requirements that can be considered as well. For example, the balance between trips and bus routes, including travel times and the number of students. One possible approach is to minimize the variance or the standard deviation of the ride/vehicle time and the student load of the trip/route. The challenge would be how to linearly approximate the variance or the standard deviation in the current model with the least number of additional auxiliary variables as possible.

Second, we would like to further increase the efficiency of Algorithm 3 by parallel computing. Algorithm 3 iteratively solves the Multi-School Bell Time and Scheduling Problem (MBS, Model 7) with the exclusion of the illegal routes that violate the Maximum Vehicle Time constraint. After finishing each iteration, Algorithm 3 can either directly move to the next iteration with the newly identified illegal routes or perform the Illegal Route Update (IRU) and then perform the next iterations with more illegal routes at once.

An intuitive parallelization approach is to adopt a different IRU strategy:

1) Do not perform the IRU;
2) Have each core perform a different number of iterations for IRU;
3) Have each core perform the IRU by replacing different portions of an illegal route: the first trip, the first two trips, the last trip, the last two trips, the middle trip etc.

One difficulty is that at some point, we need to collect the illegal routes from all cores in order to find the optimal solution. How often the illegal routes should be integrated, how often the search procedure should be parallelized, and how different illegal route update strategies should be balanced are subject to further research.

In addition, we are interested to improve the Trip-based Stochastic On-Time Constrained Multi-School Bell Time and Bus Scheduling Problem with the Maximum Vehicle Time Constraints (Model 9) to account for the non-normally distributed travel times. Due to the symmetry and infinite tail, the normal distribution may not be the best representation of the real-world travel times. It is of our interest to see whether it is possible to formulate this problem with more general distributions or even empirical distributions of the historical data.

Finally, in our stochastic cases (especially Model 9 and Algorithm 4), we concentrate on the worst performance, the minimum probability of being on time. These model and algorithm are guaranteed to find solutions that satisfy the minimum service requirement. However, it is also valuable to consider the average performance in some cases. For instance, the mean probability of being on time. Moreover, in the school bus planning, the buses need to be on time almost surely (with the probability

of no less than 98% in this dissertation). Such constraint may not have to be so strict for the general vehicle routing and scheduling problem. Instead, we can extend this model and algorithm by allowing for the violation of the punctuality constraint at the expense of a penalty. It is reasonable to consider some recourse to mitigate the impact of the violation of chance constraints because such violation in a stochastic world to some extent is unavoidable. One proposal to tackle this task is when generating new bus routes in the subproblem of Algorithm 4, although some bus routes fail to satisfy the stochastic on-time constraints at a fixed confidence level, they are still accepted and will be added to the master problem with additional recourse cost for failure. Thus, the overall objective is to minimize the expected transportation cost under satisfying chance constraints as well as the additional correction cost in case of failure. As other methods may be applicable to consider the recourse cost of failure, this problem is subject to further study.

Bibliography

[1] Angel, R.D., Caudle, W.L., Noonan, R., Whinston, A., 1972. Computer-assisted school bus scheduling. Management Science 18 (6), 279–288.

[2] Babin, G., Deneault, S. and Laporte, G., 2007. Improvements to the Or-opt heuristic for the symmetric travelling salesman problem. Journal of the Operational Research Society, 58(3), pp.402-407.

[3] Badeau, P., Guertin, F., Gendreau, M., Potvin, J.Y. and Taillard, E., 1997. A parallel Tabu search heuristic for the vehicle routing problem with time windows. Transportation Research Part C: Emerging Technologies, 5(2), pp.109-122.

[4] Beasley JE. 1983. Route-first cluster-second methods for vehicle routing. Omega;11: 403–8.

[5] Bektaş, T. and Elmastaş, S., 2007. Solving school bus routing problems through integer programming. Journal of the Operational Research Society, 58(12), pp.1599-1604.

[6] Bennett, B.T., and Gazis, D.C., 1972. School bus routing by computer. Transportation Research, 6(4), pp.317-325.

[7] Bent, R.W. and Van Hentenryck, P., 2004. Scenario-based planning for partially dynamic vehicle routing with stochastic customers. Operations Research, 52(6), pp.977-987.

[8] Bodin, L.D. and Berman, L., 1979. Routing and scheduling of school buses by computer. Transportation Science 13 (2), 113–129.

[9] Bögl, M., Doerner, K.F., and Parragh, S.N., 2015. The school bus routing and scheduling problem with transfers. Networks, 65(2), pp.180-203.

[10] Bowerman, R., Calamai, P.H., and Hall, G.B., 1994. The space filling curve with optimal partitioning heuristic for the vehicle routing problem. European Journal of Operational Research, 76(1), pp.128-142.

[11] Bowerman, R., Hall, B., and Calamai, P., 1995. A multi-objective optimization approach to urban school bus routing: Formulation and solution method. Transportation Research Part A: Policy and Practice, 29(2), pp.107-123.

[12] Braca, J., Bramel, J., Posner, B. and Simchi-Levi, D., 1997. A computerized approach to the New York City school bus routing problem. IIE Transactions, 29(8), pp.693-702.

[13] Bramel, J. and Simchi-Levi, D., 1995. A location based heuristic for general routing problems. Operations research, 43(4), pp.649-660.

[14] Brandao, J. and Mercer, A., 1997. A Tabu search algorithm for the multi-trip vehicle routing and scheduling problem. European journal of operational research, 100(1), pp.180-191.

[15] Caceres, H., Batta, R. and He, Q., 2017. School bus routing with stochastic demand and duration constraints. Transportation science, 51(4), pp.1349-1364.

[16] Caceres, H., Batta, R. and He, Q., 2018. Special need students school bus routing: Consideration for mixed load and heterogeneous fleet. Socio-Economic Planning Sciences.

[17] Chapleau, L., Ferland, J.A., Rousseau, J.-M., 1985. Clustering for routing in densely populated areas. European Journal of Operational Research 20 (1), 48–57.

[18] Chen, D.S., Kallsen, H.A., Chen, H.C. and Tseng, V.C., 1990. A bus routing system for rural school districts. Computers & Industrial Engineering, 19(1-4), pp.322-325.
[19] Chen, X., Kong, Y., Dang, L., Hou, Y. and Ye, X., 2015. Exact and metaheuristic approaches for a bi-objective school bus scheduling problem. PloS one, 10(7), p.e0132600.
[20] Christofides, N., Mingozzi, A. and Toth, P., 1979. Vehicle routing problem. In N. Combinatorial Optimization. Wiley. Chichester, 313-338
[21] Christofides, N., Mingozzi, A. and Toth, P., 1981a. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. Mathematical Programming, 20(1), pp.255-282.
[22] Christofides, N., Mingozzi, A. and Toth, P., 1981b. State-space relaxation procedures for the computation of bounds to routing problems. Networks,11(2), pp.145-164.
[23] Corberán, A., Fernández, E., Laguna, M. and Marti, R.A.F.A.E.L., 2002. Heuristic solutions to the problem of routing school buses with multiple objectives. Journal of the operational research society, 53(4), pp.427-435.
[24] de Souza Lima, F.M., Pereira, D.S., da Conceição, S.V., and de Camargo, R.S., 2017. A multi-objective capacitated rural school bus routing problem with heterogeneous fleet and mixed loads. 4OR, 15(4), pp.359-386.
[25] Desrochers, M., Desrosiers, J., and Solomon, M., 1992. A new optimization algorithm for the vehicle routing problem with time windows. Operations Research, 40(2), pp.342-354.
[26] Desrochers, M., Soumis, F. 1988. A Generalized Permanent Labeling algorithm for the shortest path problem with time windows. INFOR 26, 191-192
[27] Desrosiers, J., Dumas, Y. and Soumis, F., 1986. A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. American Journal of Mathematical and Management Sciences, 6(3-4), pp.301-325.
[28] Desrosiers, J., Dumas, Y., Solomon, M.M. and Soumis, F., 1995. Time constrained routing and scheduling. Handbooks in operations research and management science, 8, pp.35-139.
[29] Desrosiers, J., Ferland, J.A., Rousseau, J.-M., Lapalme, G., Chapleau, L., 1981. An overview of a school busing system. In: Jaiswal, N.K. (Ed.), Scientific Management of Transport Systems. North-Holland, Amsterdam, pp. 235–243.
[30] Desrosiers, J., Ferland, J.A., Rousseau, J.M., Lapalme, G., and Chapleau, L., 1986. TRANSCOL: A multi-period school bus routing and scheduling system. TIMS Studies in the Management Sciences, 22, pp.47-71.
[31] Desrosiers, J., Sauvé, M., and Soumis, F., 1988. Lagrangian relaxation methods for solving the minimum fleet size multiple traveling salesman problem with time windows. Management Science, 34(8), pp.1005-1022.
[32] Desrosiers, J., Soumis, F. and Desrochers, M., 1984. Routing with time windows by column generation. Networks, 14(4), pp.545-565.
[33] Díaz-Parra, O., Ruiz-Vanoye, J.A., Buenabad-Arias, A. and Cocon, F., 2012, November. A vertical transfer algorithm for the school bus routing problem. In

Nature and Biologically Inspired Computing (NaBIC), 2012 Fourth World Congress on (pp. 66-71). IEEE.
[34] Dulac, G., Ferland, J.A., Forgues, P.A., 1980. School bus routes generator in urban surroundings. Computers and Operations Research 7 (3), 199–213.
[35] Faraj, M.F., Sarubbi, J.F.M., Silva, C.M., Porto, M.F. and Nunes, N.T.R., 2014, October. A real geographical application for the school bus routing problem. In Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on (pp. 2762-2767). IEEE.
[36] Fügenschuh, A., 2009. Solving a school bus scheduling problem with integer programming. European Journal of Operational Research, 193(3), pp.867-884.
[37] Fügenschuh, A., 2011. A set partitioning reformulation of a school bus scheduling problem. Journal of Scheduling, 14(4), pp.307-318.
[38] Gavish, B., Shlifer, E., 1979. An approach for solving a class of transportation scheduling problems. European Journal of Operational Research 3 (2), 122–134.
[39] Gendreau, M., Hertz, A. and Laporte, G., 1992. New insertion and postoptimization procedures for the traveling salesman problem. Operations Research, 40(6), pp.1086-1094.
[40] Gendreau, M., Hertz, A. and Laporte, G., 1994. A tabu search heuristic for the vehicle routing problem. Management Science, 40(10), pp.1276-1290.
[41] Gendreau, M., Laporte, G., and Séguin, R., 1996. Stochastic vehicle routing. European Journal of Operational Research, 88(1), pp.3-12.
[42] Gillett, B.E., and Miller, L.R., 1974. A heuristic algorithm for the vehicle dispatch problem. Operations Research, 22(2), pp.340-349.
[43] Glover, F., 1989. Tabu search-part I. ORSA Journal on Computing, 1(3), pp.190-206.
[44] Golden, B.L., Wasil, E.A., Kelly, J.P. and Chao, I.M., 1998. The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results. In Fleet management and logistics (pp. 33-56). Springer, Boston, MA.
[45] Haghani, A. and Banihashemi, M., 2002. Heuristic approaches for solving large-scale bus transit vehicle scheduling problem with route time constraints. Transportation Research Part A: Policy and Practice, 36(4), pp.309-333.
[46] Hargroves, B.T., Demetsky, M.J., 1981. A computer-assisted school bus routing strategy: a case study. Socio-Economic Planning Sciences 15 (6), 341–345.
[47] Hvattum, L.M., Løkketangen, A. and Laporte, G., 2006. Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. Transportation Science, 40(4), pp.421-438.
[48] Kang, M., Kim, S.K., Felan, J.T., Choi, H.R. and Cho, M., 2015. Development of a genetic algorithm for the school bus routing problem. International Journal of Software Engineering and Its Applications, 9(5), pp.107-126.
[49] Kim, B.I., Kim, S. and Park, J., 2012. A school bus scheduling problem. European Journal of Operational Research, 218(2), pp.577-585.
[50] Kinable, J., 2014. Ph.D. Thesis. Decomposition approaches for optimization problems.

[51] Kinable, J., Spieksma, F.C.R., and Vanden Berghe, G., 2014. School bus routing—a column generation approach. International Transactions in Operational Research, 21(3), pp.453-478.
[52] Kontoravdis, G. and Bard, J.F., 1995. A GRASP for the vehicle routing problem with time windows. ORSA Journal on Computing, 7(1), pp.10-23.
[53] Kumar, Y. and Jain, S., 2015, September. School bus routing based on branch and bound approach. In Computer, Communication, and Control (IC4), 2015 International Conference on (pp. 1-4). IEEE.
[54] Kuo, Y., 2010. Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. Computers & Industrial Engineering, 59(1), pp.157-165.
[55] Lamatsch, A., 1992. An approach to vehicle scheduling with depot capacity constraints. In Computer-Aided Transit Scheduling (pp. 181-195). Springer, Berlin, Heidelberg.
[56] Laporte, G., 1992. The vehicle routing problem: An overview of exact and approximate algorithms. European Journal of Operational Research, 59(3), pp.345-358.
[57] Laporte, G., Louveaux, F., and Mercure, H., 1992. The vehicle routing problem with stochastic travel times. Transportation Science, 26(3), pp.161-170.
[58] Laporte, G., Mercure, H., and Nobert, Y., 1986. An exact algorithm for the asymmetrical capacitated vehicle routing problem. Networks 16, 33-46.
[59] Laporte, G., Nobert, Y. and Taillefer, S., 1988. Solving a family of multi-depot vehicle routing and location-routing problems. Transportation Science, 22(3), pp.161-172.
[60] Li, L.Y.O., and Fu, Z., 2002. The school bus routing problem: a case study. Journal of the Operational Research Society, 53(5), pp.552-558.
[61] Mesquita, M. and Paixão, J., 1992. Multiple depot vehicle scheduling problem: A new heuristic based on quasi-assignment algorithms. In Computer-Aided Transit Scheduling (pp. 167-180). Springer, Berlin, Heidelberg.
[62] Miller, C., A. Tucker, and R. Zemlin. 1960. Integer Programming Formulations and Traveling Salesman Problems. J. Assoc. Computer. Mach. 7, 326-329.
[63] Mushi, A.R., Mujuni, E. and Ngonyani, B., 2015. Optimizing Schedules for School Bus Routing Problem: the case of Dar Es Salaam Schools. International Journal of Advanced Research in Computer Science, 6(1).
[64] National Association of State Directors of Pupil Transportation Services.1999. School Bus Seat Capacity
[65] National Center for Education Statistics. 2016. National Center for Education Statistics: Fast Facts. Retrieved January 6, 2019. https://nces.ed.gov/fastfacts/display.asp?id=67.
[66] Newton, R.M. and Thomas, W.H., 1974. Bus routing in a multi-school system. Computers & Operations Research, 1(2), pp.213-222.
[67] Newton, R.M. and Thomas, W.H., 1969. Design of school bus routes by computer. Socio-Economic Planning Sciences 3 (1), 75–85.
[68] Or, I., 1976. Traveling Salesman-Type Combinatorial Problems and Their Relation to the Logistics of Regional Blood Banking. Ph.D. Thesis, Department

of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL.
[69] Orloff, C.S., 1976. Route constrained fleet scheduling. Transportation Science, 10(2), pp.149-168.
[70] Osman, I.H., 1993. Meta-strategy simulated annealing and Tabu search algorithms for the vehicle routing problem. Annals of operations research,41(4), pp.421-451.
[71] Pacheco, J. and Martí, R., 2006. Tabu search for a multi-objective routing problem. Journal of the Operational Research Society, 57(1), pp.29-37.
[72] Park, J., and Kim, B.I., 2010. The school bus routing problem: A review. European Journal of Operational Research, 202(2), pp.311-319.
[73] Park, J., Tae, H. and Kim, B.-I., 2012. A post-improvement procedure for the mixed load school bus routing problem. European Journal of Operational Research, 217(1), pp.204-213.
[74] Potvin, J.Y. and Rousseau, J.M., 1993. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. European Journal of Operational Research, 66(3), pp.331-340.
[75] Potvin, J.-Y., T. Kervahut, and J.-M. Rousseau, 1992. A Tabu search heuristic for the vehicle routing problem with time windows. Publication 855, Centre de Recherche sur les Transports, Universite de Montreal.
[76] Prins, C., 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research, 31(12), pp.1985-2002.
[77] Renaud, J., Laporte, G., and Boctor, F.F., 1996. A Tabu search heuristic for the multi-depot vehicle routing problem. Computers & Operations Research,23(3), pp.229-235.
[78] Riera-Ledesma, J. and Salazar-González, J.J., 2013. A column generation approach for a school bus routing problem with resource constraints. Computers & Operations Research, 40(2), pp.566-583.
[79] Ripplinger, D., 2005. Rural school vehicle routing problem. Transportation Research Record: Journal of the Transportation Research Board, (1922), pp.105-110.
[80] Rochat, Y. and Semet, F., 1994. A Tabu search approach for delivering pet food and flour in Switzerland. Journal of the Operational Research Society, 45(11), pp.1233-1246.
[81] Russell, R.A., and Morrel, R.B., 1986. Routing special-education school buses. Interfaces, 16(5), pp.56-64.
[82] Ryan, D.M., and Foster, E.A., 1981. An integer programming approach to scheduling.
[83] Samadi-Dana, S., Paydar, M.M. and Jouzdani, J., 2017. A simulated annealing solution method for robust school bus routing. International Journal of Operational Research, 28(3), pp.307-326.
[84] Santana, L., Ramiro, E. and Romero Carvajal, J.D.J., 2015. A hybrid column generation and clustering approach to the school bus routing problem with time windows. Ingeniería, 20(1), pp.101-117.

[85] Schittekat, P., Kinable, J., Sörensen, K., Sevaux, M., Spieksma, F. and Springael, J., 2013. A metaheuristic for the school bus routing problem with bus stop selection. European Journal of Operational Research, 229(2), pp.518-528.
[86] Schittekat, P., Sevaux, M., Sörensen, K., 2006. A mathematical formulation for a school bus routing problem. In: Proceedings of the IEEE 2006 International Conference on Service Systems and Service Management, Troyes, France.
[87] Shafahi, A., Wang, Z. and Haghani, A., 2017. Solving the school bus routing problem by maximizing trip compatibility. Transportation Research Record: Journal of the Transportation Research Board, (2667), pp.17-27.
[88] Silva, C.M., Sarubbi, J.F., Silva, D.F., Porto, M.F. and Nunes, N.T., 2015, September. A mixed load solution for the rural school bus routing problem. In Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on (pp. 1940-1945). IEEE.
[89] Simchi-Levi, D. and Bramel, J. 1990. On the optimal solution value of the capacitated vehicle routing problem with unsplit demands. Working Paper, Department of IE & OR, Columbia University, New York.
[90] Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research, 35(2), pp.254-265.
[91] Spada, M., Bierlaire, M., Liebling, Th.M., 2005. Decision-aiding methodology for the school bus routing and scheduling problem. Transportation Science 39, 477–490.
[92] Taillard, É., Badeau, P., Gendreau, M., Guertin, F. and Potvin, J.Y., 1997. A Tabu search heuristic for the vehicle routing problem with soft time windows. Transportation Science, 31(2), pp.170-186.
[93] Thangiah, S.R., and Nygard, K.E., 1992, March. School bus routing using genetic algorithms. In Aerospace Sensing (pp. 387-398). International Society for Optics and Photonics.
[94] Tillman, F.A., 1969. The multiple terminal delivery problem with probabilistic demands. Transportation Science, 3(3), pp.192-204.
[95] Toregas, C. and ReVelle, C., 1972. Optimal location under time or distance constraints. Papers in Regional Science, 28(1), pp.133-144.
[96] Verderber, W.J., 1974. Automated pupil transportation. Computers and Operations Research 1 (2), 235–245.
[97] Wang, Z., Shafahi, A., and Haghani, A., 2017. SCDA: School Compatibility Decomposition Algorithm for Solving the Multi School Bus Routing and Scheduling Problem. arXiv preprint arXiv:1711.00532.
[98] Wren, A. and Holliday, A., 1972. Computer scheduling of vehicles from one or more depots to a number of delivery points. Journal of the Operational Research Society, 23(3), pp.333-344.
[99] Xu, H., Chen, Z.L., Rajagopal, S., and Arunapuram, S., 2003. Solving a practical pickup and delivery problem. Transportation Science, 37(3), pp.347-364.
[100] Yan, S., Hsiao, F.Y. and Chen, Y.C., 2015. Inter-school bus scheduling under stochastic travel times. Networks and Spatial Economics, 15(4), pp.1049-1074.
[101] Yao, B., Cao, Q., Wang, Z., Hu, P., Zhang, M. and Yu, B., 2016. A two-stage heuristic algorithm for the school bus routing problem with mixed load plan. Transportation Letters, 8(4), pp.205-219.