YAŞAR UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

MASTER’S THESIS

AN ARC ROUTING PROBLEM OF ELECTRIC POWERED STREET SWEEPERS WITH TIME WINDOWS AND INTERMEDIATE STOP

CANSU YURTSEVEN

THESIS ADVISOR: ASST. PROF. DR. MAHMUT ALİ GÖKÇE

INDUSTRIAL ENGINEERING

PRESENTATION DATE: 06.08.2019

BORNOVA / İZMİR
AUGUST 2019

We certify that, as the jury, we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Jury Members:	**Signature:**
Asst. Prof. Dr. Mahmut Ali GÖKÇE Yaşar University	Mahmut Ali Gökçe ……………………
Assoc. Prof. Dr. Muhittin Hakan DEMİR İzmir University of Economics	……………………
Asst. Prof. Dr. Adalet ÖNER Yaşar University	……………………

Prof. Dr. Cüneyt GÜZELİŞ
Director of the Graduate School

ABSTRACT

AN ARC ROUTING PROBLEM OF ELECTRIC POWERED STREET SWEEPERS WITH TIME WINDOWS AND INTERMEDIATE STOP

Yurtseven, Cansu
Msc, Industrial Engineering
Advisor: Asst. Prof. Dr. Mahmut Ali GÖKÇE
August 2019

Waste collection is an important public service performed by municipalities. Since waste collection problems are in real life and their effects are high, these problems have been studied more recently. Since there is no income in return, it is important that waste collection services be carried out with public funds at minimum cost. Street sweeping is an important part of municipal services in terms of waste management. Recently, electric street sweepers have become increasingly popular for their energy efficiency and environmental protection. In this study, the problem of determining the routes of electric powered street sweepers to serve a predetermined set of streets in the city is considered, taking into account realistic operational constraints such as waste disposal operation, vehicle charge planning, lunch and rest breaks. A novel mathematical model is proposed in order to determine the optimal routes of the sweepers in order to collect waste from the streets that need to be swept within the given time windows by addressing a heterogeneous fleet of electric powered street sweepers with different capacities and battery levels to perform street sweeping service. It is planned that the vehicles used will leave a depot at the beginning of the time period and return to the depot at the end of the day. The objective is to provide this service according to the constraints, while minimizing the energy consumption used in travel and waste disposal operations. In order to measure the performance of the mathematical model, a case study and experimental design was made with real life examples and the results of the experiments were examined.

Key Words: waste collection, street sweeping, electric vehicle routing problem, partial recharge, mathematical programming, optimization

ÖZ

ELEKTRİKLE ÇALIŞAN SOKAK SÜPÜRÜCÜLERİ İÇİN ZAMAN PENCERELİ VE ARA DURAKLI AYRIT ROTALAMA PROBLEMİ

Yurtseven, Cansu
Yüksek Lisans, Endüstri Mühendisliği
Danışman: Dr. Öğr. Üyesi Mahmut Ali GÖKÇE
Ağustos 2019

Atık toplama belediye hizmetleri içinde yer alır. Atık toplama problemleri gerçek hayatın içinden ve etkileri yüksek olduğundan, son zamanlarda bu problemler daha fazla çalışılmaya başlanmıştır. Atık toplama hizmetlerinin, karşılığında gelir olmadığından, kamu fonları ile minimum maliyetle gerçekleştirilmesi önemlidir. Sokak süpürme işlemi, belediye hizmetleri arasında atık yönetimi açısından önemli bir yer tutar. Son zamanlarda, elektrikli sokak süpürücüler, enerji verimlilikleri ve çevre koruma nedenleriyle artan bir oranda tercih edilmeye başlanmıştır. Bu çalışmada, atık boşaltım operasyonu, araç şarj planlaması, öğle yemeği ve dinlenme molaları gibi gerçekçi operasyonel kısıtları göz önünde bulundurarak, şehirdeki önceden belirlenmiş bir dizi sokağa hizmet vermek için elektrikle çalışan sokak süpürücülerinin rotalarını belirleme sorunu ele alınmaktadır. Sokak süpürme hizmetini gerçekleştirmek için farklı kapasitelere ve batarya seviyelerine sahip elektrikle çalışan heterojen bir araç filosunu ele alarak, verilen zaman pencereleri içerisinde süpürülme ihtiyacı olan caddelerden atıkları toplamak için, süpürücülerinin optimal rotalarını belirlemek amacıyla, yeni bir matematiksel model önerilmektedir. Kullanılan araçların zaman periyodunun başlangıcında bir depodan çıkması ve gün sonunda tekrar depoya geri dönmesi planlanmaktadır. Enazlanan amaç fonksiyonu, hizmeti kısıtlara göre sağlarken, seyahat ve atık boşaltım işlemlerinde kullanılan enerji tüketimini en aza indirmektir. Matematiksel modelin performansını ölçmek için, gerçek hayat örnekleri ile bir vaka çalışması ve deney tasarımı yapılmış, deney sonuçları incelenmiştir.

Anahtar Kelimeler: atık toplama, sokak süpürme, elektrikli araç rotalama problemi, kısmi şarj, matematiksel programlama, optimizasyon

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my thesis advisor Asst. Prof. Dr. Mahmut Ali Gökçe for his support during this thesis. His motivation, knowledge, and enthusiasm helped me in all the time of research and writing of this thesis.

Next, I would like to thank my family for supporting me spiritually throughout writing this thesis and also my life in general. In particular, I wish to thank Bilgin Tevfik Akçiçek, who gave me the strength to withstand the troubles. His positivity and everlasting trust in me was important support for me along this way.

Finally, the entire Department of Industrial Engineering at the Yaşar University was the second home for a great amount of time. I would like to thank faculty, colleagues, and particularly my officemates for patiently sharing their valuable help and experiences with me.

Thanks for all your encouragement.

Cansu Yurtseven
İzmir, 2019

TEXT OF OATH

I declare and honestly confirm that my study, titled "AN ARC ROUTING PROBLEM OF ELECTRIC POWERED STREET SWEEPERS WITH TIME WINDOWS AND INTERMEDIATE STOP" and presented as a Master's Thesis, has been written without applying to any assistance inconsistent with scientific ethics and traditions. I declare, to the best of my knowledge and belief, that all content and ideas drawn directly or indirectly from external sources are indicated in the text and listed in the list of references.

Cansu Yurtseven

Signature

……………………………………..

August 27, 2019

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGEMENTS ix

TEXT OF OATH xi

TABLE OF CONTENTS xiii

LIST OF FIGURES xvii

LIST OF TABLES xix

SYMBOLS AND ABBREVIATIONS xxv

CHAPTER 1 INTRODUCTION 1
  1.1. Street Sweeping Operations 1
  1.2. Usage of Electric Powered Vehicles 2

CHAPTER 2 PROBLEM DEFINITION 5

CHAPTER 3 LITERATURE REVIEW 7
  3.1. Waste Collection Studies 7
  3.2. Electric Vehicle Routing Problem Studies 11

CHAPTER 4 SOLUTION METHODOLOGY 17

CHAPTER 5 COMPUTATIONAL EXPERIMENTS 23
  5.1. Case Study 23
  5.2. Design of Experiments 28
    5.2.1. Full Battery Level Scenario 30

    5.2.2. Medium Battery Level Scenario 32

    5.2.3. Low Battery Level Scenario 34

CHAPTER 6 CONCLUSIONS 41
  6.1. Future Study 41

REFERENCES 43

APPENDIX 1 – Computational Results of the Mathematical Model for FBLS 45

APPENDIX 2 – Computational Results of the Mathematical Model for MBLS 55

APPENDIX 3 – Computational Results of the Mathematical Model for LBLS 65

LIST OF FIGURES

Figure 1.1. An Electric Powered Street Sweeper 3

Figure 1.2. Sweeping Operation 3

Figure 2.1. Interior View of a Street Sweeper 6

Figure 5.1. Representation of a Street Network in İzmir 23

Figure 5.2. Optimal Route for Sweeper 1 25

Figure 5.3. Optimal Route for Sweeper 2 26

Figure 5.4. Optimal Route for Sweeper 3 26

Figure 5.5. The Network Graph of the Experiment 29

Figure 5.6. Comparison Chart for the Required Arc Percentage vs. CPU Time for FBLS .. 31

Figure 5.7. Comparison Chart for the Total Number of Nodes vs. CPU Time for FBLS 32

Figure 5.8. Comparison Chart for the Required Arc Percentage vs. CPU Time for MBLS . 33

Figure 5.9. Comparison Chart for the Total Number of Nodes vs. CPU Time for MBLS ... 34

Figure 5.10. Comparison Chart for the Required Arc Percentage vs. CPU Time for LBLS 36

Figure 5.11. Comparison Chart for the Total Number of Nodes vs. CPU Time for LBLS .. 37

Figure 5.12. Comparison Chart of Battery Level Scenarios for 30 Nodes 37

Figure 5.13. Comparison Chart of Battery Level Scenarios for 40 Nodes 38

Figure 5.14. Comparison Chart of Battery Level Scenarios for 50 Nodes 38

LIST OF TABLES

Table 3.1. Summary Table for Related Studies about Waste Collection 10

Table 3.2. Summary Table for Related Studies about EVRP 14

Table 5.1. Parameter Values for the Case Study 24

Table 5.2. Results of the Case Study 27

Table 5.3. The Input for the Computational Experiments 28

Table 5.4. Average Results of the Full Battery Level Scenario Instance Sets 30

Table 5.5. Average Results of the Medium Battery Level Scenario Instance Sets 32

Table 5.6. Average Results of the Low Battery Level Scenario Instance Sets 35

Table A1.1. Results of the 1st Instance Set with 15 Nodes and Full Battery 45

Table A1.2. Results of the 2nd Instance Set with 15 Nodes and Full Battery 45

Table A1.3. Results of the 3rd Instance Set with 15 Nodes and Full Battery 46

Table A1.4. Results of the 4th Instance Set with 15 Nodes and Full Battery 46

Table A1.5. Results of the 5th Instance Set with 15 Nodes and Full Battery 47

Table A1.6. Results of the 6th Instance Set with 15 Nodes and Full Battery 47

Table A1.7. Results of the 7th Instance Set with 20 Nodes and Full Battery 47

Table A1.8. Results of the 8th Instance Set with 20 Nodes and Full Battery 48

Table A1.9. Results of the 9th Instance Set with 20 Nodes and Full Battery 48

Table A1.10. Results of the 10th Instance Set with 20 Nodes and Full Battery 49

Table A1.11. Results of the 11th Instance Set with 20 Nodes and Full Battery 49

Table A1.12. Results of the 12th Instance Set with 20 Nodes and Full Battery 50

Table A1.13. Results of the 13th Instance Set with 25 Nodes and Full Battery 50

Table A1.14. Results of the 14th Instance Set with 25 Nodes and Full Battery 51

Table A1.15. Results of the 15th Instance Set with 25 Nodes and Full Battery 51

Table A1.16. Results of the 16th Instance Set with 25 Nodes and Full Battery 51

Table A1.17. Results of the 17th Instance Set with 25 Nodes and Full Battery 52

Table A1.18. Results of the 18th Instance Set with 25 Nodes and Full Battery 52

Table A2.1. Results of the 1st Instance Set with 15 Nodes and Medium Battery.................. 55

Table A2.2. Results of the 2nd Instance Set with 15 Nodes and Medium Battery................. 55

Table A2.3. Results of the 3rd Instance Set with 15 Nodes and Medium Battery 56

Table A2.4. Results of the 4th Instance Set with 15 Nodes and Medium Battery 56

Table A2.5. Results of the 5th Instance Set with 15 Nodes and Medium Battery 57

Table A2.6. Results of the 6th Instance Set with 15 Nodes and Medium Battery 57

Table A2.7. Results of the 7th Instance Set with 20 Nodes and Medium Battery 58

Table A2.8. Results of the 8th Instance Set with 20 Nodes and Medium Battery 58

Table A2.9. Results of the 9th Instance Set with 20 Nodes and Medium Battery 58

Table A2.10. Results of the 10th Instance Set with 20 Nodes and Medium Battery 59

Table A2.11. Results of the 11th Instance Set with 20 Nodes and Medium Battery 59

Table A2.12. Results of the 12th Instance Set with 20 Nodes and Medium Battery 60

Table A2.13. Results of the 13th Instance Set with 25 Nodes and Medium Battery 60

Table A2.14. Results of the 14th Instance Set with 25 Nodes and Medium Battery 61

Table A2.15. Results of the 15th Instance Set with 25 Nodes and Medium Battery 61

Table A2.16. Results of the 16th Instance Set with 25 Nodes and Medium Battery 62

Table A2.17. Results of the 17th Instance Set with 25 Nodes and Medium Battery 62

Table A2.18. Results of the 18th Instance Set with 25 Nodes and Medium Battery 62

Table A3.1. Results of the 1st Instance Set with 15 Nodes and Low Battery........................ 65

Table A3.2. Results of the 2nd Instance Set with 15 Nodes and Low Battery....................... 65

Table A3.3. Results of the 3rd Instance Set with 15 Nodes and Low Battery 66

Table A3.4. Results of the 4th Instance Set with 15 Nodes and Low Battery 66

Table A3.5. Results of the 5th Instance Set with 15 Nodes and Low Battery 67

Table A3.6. Results of the 6th Instance Set with 15 Nodes and Low Battery 67

Table A3.7. Results of the 7th Instance Set with 20 Nodes and Low Battery 67

Table A3.8. Results of the 8th Instance Set with 20 Nodes and Low Battery 68

Table A3.9. Results of the 9th Instance Set with 20 Nodes and Low Battery 68

Table A3.10. Results of the 10th Instance Set with 20 Nodes and Low Battery 69

Table A3.11. Results of the 11th Instance Set with 20 Nodes and Low Battery 69

Table A3.12. Results of the 12th Instance Set with 20 Nodes and Low Battery 70

Table A3.13. Results of the 13th Instance Set with 25 Nodes and Low Battery 70

Table A3.14. Results of the 14th Instance Set with 25 Nodes and Low Battery 70

Table A3.15. Results of the 15th Instance Set with 25 Nodes and Low Battery 71

Table A3.16. Results of the 16th Instance Set with 25 Nodes and Low Battery 71

Table A3.17. Results of the 17th Instance Set with 25 Nodes and Low Battery 72

Table A3.18. Results of the 18th Instance Set with 25 Nodes and Low Battery 72

SYMBOLS AND ABBREVIATIONS

ABBREVIATIONS:

ALNS	Adaptive Large Neighborhood Search
API	Application Program Interface
CARP	Capacitated Arc Routing Problem
CPP	Chinese Postman Problem
ECV	Electric Commercial Vehicles
E-FSMFTW	Electric Fleet Size and Mix Vehicle Routing Problem with Time Windows and Recharging Stations
E-VRP	Electric Vehicle Routing Problem
E-VRPTW	Electric Vehicle Routing Problem with Time Windows
E-VRPTWMF	Electric Vehicle Routing Problem with Time Windows and Mixed Fleet
EVRPTW-PR	Electric Vehicle Routing Problem with Time Windows and Partial Recharges
FBLS	Full Battery Level Scenario
G-VRP	Green Vehicle Routing Problem
ICCV	Internal Combustion Commercial Vehicle
LBLS	Low Battery Level Scenario
MBLS	Medium Battery Level Scenario
MDVRPI	Multi-depot Vehicle Routing Problem with Inter-depot
NP	Nondeterministic Polynomial Time
RPP	Rural Postman Problem
SA	Simulated Annealing
TS	Tabu Search
VNS	Variable Neighborhood Search
VRP	Vehicle Routing Problem

VRPTW	Vehicle Routing Problem with Time Windows
WCVRP	Waste Collection Vehicle Routing Problem
WCVRPTW	Waste Collection Vehicle Routing Problem with Time Windows
WPP	Windy Postman Problem

SYMBOLS:

G	Graph with arcs and nodes.
N	Number of nodes.
V	Set of nodes $V = \{1, \dots, N\}$
$V_{0,N+1}$	Set of nodes including depot instances
A	Set of all arcs $A = \{(i,j) \mid i,j \in V_{0,N+1}\}$
A'	Set of required arcs which are serviced
V_D	Set of nodes which include disposal sites
V_{ND}	Set of nodes which does not include disposal sites
K	Set of vehicle types
t_{ij}	Additional time of servicing arc (i,j)
t'_{ij}	Time of traversing arc (i,j)
d_{ij}	Arc distance for (i,j)
r_k	Additional energy consumption rate when servicing by vehicle k per unit distance
r'_k	The energy consumption rate of vehicle k when traversing
c_{ijk}	Amount of energy consumption when traversing arc (i,j) by vehicle k per distance $(r'_k d_{ij})$
$engdisp_k$	The energy consumption of disposal operations for vehicle k
g_k	Recharging time of vehicle k per energy unit
B_k	Battery capacity of vehicle k

Q_k	Bin capacity of vehicle k
τ_{ij}	The expected waste amount on the arc $(i,j) \in A'$
ra_{ij}	$\{1$, if arc (i,j) is required to be swept , 0 otherwise$\}$
γ_i	$\{1$, if there is a disposal site on node , 0 otherwise$\}$
α_i	$\{1$, if there is a charging station on node i , 0 otherwise$\}$
$[e_{ik}, l_{ik}]$	Time window with the earliest and latest start of vehicle k at node i
$[a_k, b_k]$	Lunch break time window for each vehicle k
t^u	Lunch break duration
t^d	Disposal operation time
t^r	Rest break duration
x_{ijk}	Binary decision variable indicating if the arc (i,j) is serviced by vehicle k
z_{ijk}	Binary decision variable indicating if the arc (i,j) is travelled by vehicle k
w_{ijk}	Binary decision variable indicating if lunch break is taken on traveling arc (i,j) by vehicle k
δ_{ijk}	Binary decision variable indicating if a rest break is taken on traveling arc (i,j) by vehicle k
μ_{ik}	Binary decision variable indicating if vehicle k dispose of its load on node i
η_{ik}	Amount of disposal on node i by vehicle k
s_{ik}	Starting time of servicing or traversing on node i by vehicle k
p_{ik}	Amount of partial charge of vehicle k on node i
y_{ik}	Battery state of charge for vehicle k on arriving node i
q_{ik}	Current bin load of vehicle k on arriving node i

CHAPTER 1

INTRODUCTION

Waste generation is a matter of widespread concern in modern societies, not only for the increase in the amount of waste produced, but also for the complexity of certain products and components. Around 11.2 billion tons of solid waste are collected every year. Decreasing the organic rate of solid wastes contributes to about 5 percent of global greenhouse gas emissions. (UN Environment).

Waste collection is an important activity in reverse logistics system, and efficient collection of waste needs improvements. Vehicle routing problems for this specific service is an exciting area to be analyzed since it contributes to a more effective reverse logistics system. (Han and Ponce Cueto, 2015). The majority of waste collection literature categorizes waste types as commercial, residential, and roll-on-roll-off waste collection. In each case, solid wastes are collected from different locations. In a commercial waste collection problem, there are several vehicles to serve customers such as strip malls and restaurants. Vehicles involves usually specialized containers that collect wastes from 60-400 customers and dispose of waste at different disposal sites (Kim et al., 2006). In some cases, customers can be revisited in a week depending on demand. The residential waste collection problem deals with collecting waste from private homes along streets. The collection vehicles gather all waste along the streets which assigned to the vehicles. Therefore, this problem is solved as an arc routing problem. The roll-on-roll-of waste collection is based on collection of waste, transportation, unloading processes and drop off large containers on construction sites. The main difference between commercial waste and roll-on-roll-off waste is the size of the container. Generally, majority of studies for these problems aim for determining the vehicle routes while minimizing the travelled distance or travelling cost.

1.1. Street Sweeping Operations

Street sweeping operation is an important part of waste management among the municipal services. A sweeper starts its tour from a depot and traverses streets to collect small size street wastes. Street sweeping vehicles have a bin with a certain capacity that collects waste. After a while the collected waste is discharged at a disposal site. Both sides of city streets are swept with these vehicles. Most of the time particular streets can only be swept at certain times due to the parking restrictions or other municipal restrictions related to noise or business restrictions.

1.2. Usage of Electric Powered Vehicles

In recent years, especially Europe has witnessed that there has been a steady increase in energy costs leading to the regulations on greenhouse gas emissions in the transportation sector. In addition, the entire transportation sector accounts for about 28% of the total greenhouse gas (GHG) emissions in countries such as the United States. One way to reduce this ratio is to include emission costs as a goal that should be minimized in routing models. The other way is using plug-in hybrid electric vehicles or electric commercial vehicles for transportation (Juan et.al., 2016). In logistic operations, electric commercial vehicles (ECVs) are becoming an alternative to traditional internal combustion commercial vehicles (ICCVs) since using electric vehicles is energy efficient and leads to less cost (Goeke and Schneider, 2015). In the beginning, electric vehicles started to be used in distribution of goods in vehicle routing problems (VRPs). Later on, electric vehicles are started to be preferred for municipal purposes including waste collection as well.

Majority is still using street sweepers as being internal combustion powered. Recent years have witnessed an increasing variety of electric powered street sweepers, which are quieter and much more energy efficient. When sweeping operations are performed with electric vehicles, different constraints for recharging operations must be considered as well.

Figure 1.1. An Electric Powered Street Sweeper

Figure 1.2. Sweeping Operation

In Figure 1.1 and Figure 1.2, electric powered street sweeper visuals are given (Tennant, 2019). With this thesis, there is a new point of view to the routing of electric vehicles used in waste collection problem of street sweeping operations.

The motivation of this thesis is that there is a lack of studies in the literature on this subject and the use of electric vehicles for sweeping operations saves a significant amount of cost and energy since these vehicles are environmentally friendly. Therefore, practically greenhouse gas emissions can be reduced in municipality activities.

In Chapter 2, a detailed problem description is given, a literature review for waste collection and electric vehicle routing problems is given in Chapter 3, the solution methodology for the problem is in Chapter 4. Computational experiment for the proposed solution method and a case study are given in Chapter 5. Finally, the conclusion and ideas for future research is mentioned in Chapter 6.

CHAPTER 2
PROBLEM DEFINITION

In this chapter, a brief information about street sweepers and the problem definition is given. Efficient and effective usage of public funds is important for the continuity of municipal services. Waste collection is one of these services, for which performing specific services at minimum cost is crucial. Because waste collection problems have significant real-life applications and implications, recently these problems have started to be studied more.

Waste collection problems can be modelled as either node routing problems or arc routing problems. There are pros and cons of both using these methods which are explained in detail in Chapter 3. Recent years has seen an increase in the use of electric vehicles for different routing problems. These electric vehicles can be either completely battery powered, i.e. plug-in electric vehicles, or hybrid that combines battery and internal combustion engines use. In literature, the great majority of electric vehicle routing problems deal with distribution of goods that has been studied as node routing problems. Street sweeping is an arc routing problem with its specific constraints. In this thesis, an arc routing problem with electric vehicles for street sweeping operations is discussed.

For this problem, we assume that there is a network of arcs to be serviced by a given heterogeneous fleet of electric powered street sweepers while minimizing total cost and the energy consumption of this service. The sweepers are constrained by the time usage of their batteries and also the capacity of their bins. These vehicles have a water tank to clean the streets while they are sweeping. A street sweeper collects the waste along the street until the capacity is full. When the capacity of the sweeper is full it needs to empty its load to one of the disposal sites. There are numerous disposal sites spread along the network so the decision of which disposal site will be used during the process must be made as well. Sweeping process inside a street sweeper is visualized in Figure 2.1. (Tennant, 2019).

Another issue is about power usage. Sweepers can be assumed that start their tour as fully charged and after some traveling, the charge level of vehicles will decrease. Any time of the route, sweepers can stop at one of many charging stations and recharge their battery, fully or partially. The question of which sweeper should be recharged how much at which station and when are other decisions that have to be made. Also routing makes a difference when vehicle uses power either just traversing or cleaning. The direction of travel is important for each arc, because the condition of the street may vary due to inclination and crowding, so the use of power may be different for each direction. We also include some real-life constraints relating to the workers. These include lunch breaks and the rest time of the sweeper drivers.

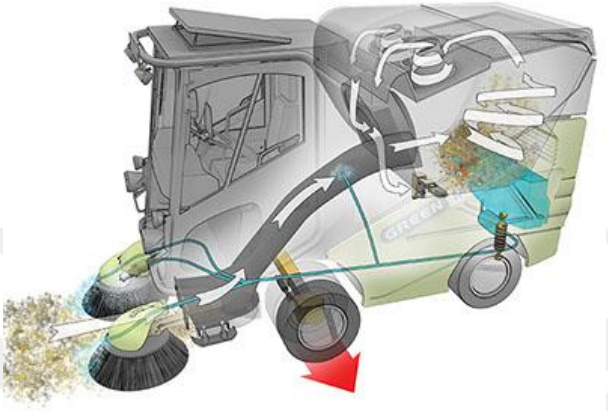

Figure 2.1. Interior View of a Street Sweeper

Certain arcs will have time windows for the sweeping service. Therefore, all sweepers start their tour from a depot and return to the depot after the tour is completed. In the problem network, the arcs may differ according to their width, slope and the length of the arc. Power consumption depends on a number of variables including working or just traversing.

In order to service all streets (arcs) obeying a predetermined schedule, sweeper may just need to traverse to get to a street for service. Another consideration is whether the arc is the main street or a side residential. Some main and crowded streets may have more demand (more frequent service) for sweeping, resulting in a higher frequency. Therefore, these differences affect the operating time of the sweepers considering the battery level and the traveling time. To the best of our knowledge, with this thesis, there is a new novel comprehensive mathematical model to the routing of electric vehicles in a waste collection problem. The main purpose is determining routing of electric powered street sweepers with different capacities and battery levels to perform a predetermined service with realistic operational constraints while the objective is minimizing energy consumption used to provide the service, travelling and disposal operations.

CHAPTER 3

LITERATURE REVIEW

In this chapter, the literature review for the waste collection problems and the electric vehicle routing problems are provided. This problem is related to both waste collection routing and electric vehicle routing. To the best of our knowledge, this is the first combining study on routing of electric powered street sweepers. For this reason, literature review is divided into two parts as waste collection problems and electric vehicle routing problems. Waste collection problems are based on how to collect waste efficiently. Garbage collection can be a part of these waste management problems while sweeping the streets is another issue of these type of problems. In literature, there are some studies about determining the vehicle route while minimizing the route cost since waste collection system is a public service related to health. Therefore performing this service with minimum cost is important. Problems in this area can be modelled as either node routing problems or arc routing problems. There are critical differences between these types of problems. Node routing have demand on the nodes of a graph that yields point-to-point transportation. In arc routing problems, demand occurs in the arcs or edges so the distribution or collection of goods is made along the arcs. Han and Ponce Cueto (2015) have a study of waste collection vehicle routing problem (WCVRP) which gives a detail literature review about previous studies. Hence, Han and Ponce Cueto made a classification and comparison of waste types mentioned in Golden’s study such that residential, commercial, and industrial (roll-on-roll-off) waste.

3.1. Waste Collection Studies

Waste collection studies can be classified according to arc or node routing problems. Eiselt et al. (1995a, b) presents algorithmic methods for arc routing problems Chinese postman problem (CPP) and rural postman problem (RPP). In the study of Han and Ponce Cueto, they also mentioned that some implementations of residential waste collection. Another type of collection is commercial waste which is a node routing problem. There are customer nodes and vehicles travel and collect commercial refuse from these customers such as restaurants and strip-malls. Generally, the solution approach for this problem is different metaheuristics and there are case studies that gain money savings to municipalities. Finally, roll-on-roll-off problems include collecting, transporting, and unloading of containers on construction sites. Han and Ponce Cueto gave information about related studies with roll-on-roll-off waste collection vehicle routing problems in their study.

Waste collection vehicle routing problems can be classified according to applied methods and solution approaches as well. In Han and Ponce Cueto's article (2015), both node routing and arc routing problem types, their objectives, and solution techniques are given. They briefly mentioned approximate algorithms used for different types of WCVRP. These algorithms consist of classical heuristics and metaheuristics. In real life, waste collection vehicle routing problems are difficult to solve considering the number of collection points and some hard constraints. Therefore heuristic algorithms are much preferred comparing the optimization methods.

Eiselt et al. (1995a, b) divided their study into two parts. First part (1995a) is about Chinese postman problem that covers the shortest walking distance for a mailman seeks for minimum cost. Chinese postman problem can be analysed as undirected CPP, directed CPP, windy postman problem (WPP), mixed CPP, and hierarchical postman problem. The undirected and directed CPPs are solvable in polynomial time however windy postman problem and mixed CPP are NP-hard. They also gave information about hierarchical postman problem such as it includes snow plowing operation which is also NP-hard.

The second part of the study of Eiselt et al. (1995b) is about rural postman problem. RPP is a small scale arc routing problem type comparing to CPP. When it is not necessary to serve all arcs in a network, it becomes a RPP. Street sweeping operation is a specific example of this type of problem, which is the main topic of this study. When some critical constraints in a sweeping problem are simplified, the problem turns into a rural postman problem. Eiselt et al. cited a study of Bodin and Kursh (1978) that call a computer assisted system for routing of street sweepers. Other applications of rural postman problem are snow plowing, garbage collection, mail delivery, school bus routing, and meter reading also explained in Eiselt's study. They enlighten RPP types and heuristics. Besides, algorithms for stacker crane problem and capacitated arc routing problem are discussed in the study. Eiselt et al. emphasize exact solution methods are not good enough to solve arc routing problems therefore instead of these methods, they recommend developing powerful heuristics.

Another study about arc routing problem is made by Bodin and Kursh (1978) who discussed a computer assisted method for routing and scheduling of street sweepers. This study is a pioneering study of street sweeping problems. They proposed an algorithm for this problem and then made a computer implementation. There is a network of streets to be swept and a sweeper can cover only one side of a street at a time, therefore, this network of arcs should be directed. The objective is covering all required streets by sweeping vehicles in order to minimize the total time of traversing. They gave a point of view for solving the one-vehicle directed branch routing problem and then discussed parking regulation constraints for street sweeping operations. In their case in New York City streets, restrictive regulation which is no parking between 8 a.m. and 9 a.m. is used that allows sweeping all street. They made a program for the street sweepers' route. In that algorithm, there are two options which are cluster first-route second approach or route first-cluster second approach.

In 2012, Buhrkal et al. made a study on waste collection vehicle routing problem with time windows (WCVRPTW) in a city logistics context. This problem is different from the general VRP problems since vehicles which collect waste also empty their loads to the disposal sites. The objective is determining the routes for waste collecting vehicles while minimizing travel cost within a time window. They constructed a mathematical model and also propose an adaptive large neighborhood search (ALNS) heuristic to solve the problem. They constructed a greedy algorithm for initial solution proposed by Benjamin and Beasley (2010). ALNS heuristic is based on destroy and repair methods. In a classic vehicle routing problem removing some of the customers from the solution is done by destroy operators and repair operators reintroduce these costumers into the solution again. There are six destroy and two repair methods in their heuristic. Besides, some customers are in the same location and have the same time window, therefore, they clustered these customers in order to reduce the solution space. Buhrkal et al. solve the WCVRPTW problem with ALNS heuristic with two different real-life data. They observed that ALNS heuristic give better results for both real-life instances and clustering customers improve the results for larger instances as well. In Table 3.1, a summary table of the mentioned studies in waste collection area is given.

Table 3.1. Summary Table for Related Studies about Waste Collection

Paper	**Problem Type**	**Objectives**	**Solution Methods**
Eiselt et al., 1995a	CPP	The shortest walking distance for a mailman with minimum cost	MIP Models
Eiselt et al., 1995b	RPP	Servicing required arcs with minimum cost	MIP Models
Bodin and Kursh, 1978	Routing and Scheduling Street Sweepers	Covering all required streets by sweeping vehicles in order to minimize the total time of traversing	Computer Assisted Method
Buhrkal et al., 2012	WCVRPTW	Determining the routes for waste collecting vehicles while minimizing travel cost within a time window	MIP, ALNS Heuristic
Han and Ponce Cueto, 2015	WCVRP	Literature Review and Classifications	Literature Paper

CPP: Chinese Postman Problem; RPP: Rural Postman Problem; WCVRPTW: Waste Collection Vehicle Routing Problem with Time Windows; WCVRP: Waste Collection Vehicle Routing Problem.

3.2. Electric Vehicle Routing Problem Studies

In this thesis, the routing of electric vehicles is taken in consideration as well. The remaining articles presented are about electric vehicle routing problem (E-VRP) with recharging constraints. In 2014, Schneider et al. have a pioneering study of electric vehicle routing problem with time windows (E-VRPTW) that incorporates recharging options. Conventional vehicles are harmful to the environment, therefore, electric commercial vehicles are used recently for transportation operations and these types of problems are studied in green logistics area as well. Schneider et al. aim that minimizes the number of vehicles used and total travelled distance while determining the routes of vehicles. Battery state of a vehicle is decreasing while traveling, so in order to complete the tour, in some cases, the vehicle must be recharged. This recharging operation requires additional time and energy consumption of this vehicle should be considered in each operation. A mixed integer programming model is constructed for this purpose and they proposed a solution method for E-VRPTW which is a combination of variable neighborhood search (VNS) and tabu search (TS). At first, they made a pre-processing to remove infeasible arcs and then constructed an initial solution to their heuristics. In hybrid VNS/TS heuristic, VNS is used to diversify the search in a structured way and TS is used for searching the solution space from a randomly generated solution of the VNS element. Besides, the acceptance criterion of the VNS is based on simulated annealing (SA) heuristic. As numerical experiments, they created E-VRPTW instances based on Solomon’s (1987) VRPTW instances and analysed the hybrid VNS/TS/SA heuristic using these instances. Schneider et al. also evaluated the performance on benchmark instances with related problems such as multi-depot VRP with inter-depot routes (MDVRPI), green vehicle routing (G-VRP), and VRPTW problems. Their method is outstanding for these problems, and the routing of electric commercial vehicles and deciding the recharging operations for that vehicles as well.

Goeke and Schneider (2015) studied electric and conventional vehicles in a routing problem. Since they consider two different types of vehicles which are electric commercial vehicles and conventional internal combustion commercial vehicles, they proposed an electric vehicle routing problem with time windows and mixed fleet (E-VRPTWMF). They have a realistic approach for the ECV problems that considering the mass of the vehicle, travel speed, and gradient of the terrain which effect the energy consumption of the vehicles. They proposed ALNS algorithm and designed test instances for their problem. Goeke and Schneider also studied the effect of different objective functions on solution. The objectives are minimizing traveled distance, minimizing cost for vehicle propulsion and labor, and lastly minimizing cost including battery replacement. The ALNS algorithm works efficiently for all cost functions except the traditional objective of minimizing total traveled distance. Finally, they proved the developed algorithm with well-known Solomon benchmark for VRPTW and also E-VRPTW benchmark instances.

In 2016, Hiermann et al. studied electric fleet size and mix vehicle routing problem with time windows and recharging stations (E-FSMFTW). This problem consists of determining the routes for different types of vehicles and the decisions of recharging times and locations while minimizing costs and total travelled distance. Vehicles vary according to their capacity, battery level, and the acquisition cost. They developed a mixed-integer programming model and presented set partitioning formulation which is solved by using branch and price approach. They also proposed a hybrid solution method using ALNS heuristic. Hiermann et al. constructed new benchmark sets for their problem and find out that the small instances can be solved optimally using ALNS heuristic, however, for larger instances, there is an optimality gap around one percent to the best known solution while branch and price can reach the optimal solution for some of these instances.

Desaulniers et.al. (2016) have a study to describe exact algorithms for EVRPTW. They considered four variants of the problem due to the single or multiple recharges in a route and type of the recharge: partially or fully. For these variants, they proposed exact branch-price-and-cut algorithms with monodirectional and bidirectional labeling algorithms to generate vehicle routes. They solved the problem up to 100 customers and 21 recharging stations. According to their computational experiments, they found that using both multiple and partial recharges provide to reduce costs.

There is a study about electric vehicle routing problem with time windows in 2016 by Keskin and Çatay. In this type of problem, while electric vehicles servicing the customers, because of the battery limitations they need to be recharged at some battery level. In previous studies, vehicles are assumed to have full battery after visiting a recharging station. In this study, that restriction is relaxed and partial recharging is allowed so the problem becomes EVRPTW-PR. They proposed a mixed integer programming model and the objective is minimizing the total distance travelled. As a solution methodology, they studied on ALNS heuristic that contains two removal and two insertion methods which are classified due to customers and recharging stations. Keskin and Çatay examined the effectiveness of the proposed solution method and compared with previous related studies' results using benchmarks of Schneider et al. (2014). The results discovered that the routes can be improved when partial recharging is allowed.

Another study belongs to Keskin and Çatay (2018) to describe a matheuristic method for electric vehicle routing problem with time windows and fast chargers. In their problem, vehicles can be partially recharged however, there are three recharging modes which are normal, fast, and super-fast recharges. The objective is minimizing recharging costs and they proposed a mathematical model for small size instances. For larger problems, they developed ALNS approach with an exact method and solved with benchmark instances from the literature. In Table 3.2, a summary table of the mentioned studies about electric vehicle routing problems is given.

Table 3.2. Summary Table for Related Studies about EVRP

Paper	**Problem Type**	**Objectives**	**Solution Methods**
Schneider et al., 2014	E-VRP	Minimizes the number of vehicles used and total traveled distance	MIP and VNS/TS/SA
Goeke and Schneider, 2015	E-VRPTWMF	Utilizing a realistic energy consumption model that incorporates speed, gradient and load distribution	ALNS Heuristic
Desaulniers et.al., 2016	EVRPTW	Minimize total cost considering single or multiple recharges in a route and type of the recharge: partially or fully	Branch-price-and-cut algorithms with monodirectional and bidirectional labeling algorithms
Hiermann et al., 2016	E-FSMFTW	Decisions of recharging times and locations while minimizing costs and total traveled distance	MIP and B&P, ALNS Heuristic
Keskin and Çatay, 2016	EVRPTW-PR	Minimizing the total distance traveled	MIP, ALNS Heuristic
Keskin and Çatay, 2018	EVRPTW-Fast Chargers	Minimizing recharging cost	Matheuristic, MILP&ALNS

E-VRP: Electric Vehicle Routing Problem; E-VRPTWMF: Electric Vehicle Routing Problem with Time Windows and Mixed Fleet; EVRPTW: Electric Vehicle Routing Problem with Time Windows; E-FSMFTW: Electric Fleet Size and Mix Vehicle Routing Problem with Time Windows and Recharging Stations; EVRPTW-PR: Electric Vehicle Routing Problem with Time Windows and Partial Recharges.

As it has been elaborated in literature review chapter, electric vehicle routing problems are commonly formulated as node routing problems and deal with distribution of goods to the customers. In some studies, only electric commercial vehicles are studied while in different studies both ECVs and ICCVs are used for the routing problems. We observed that electric vehicle routing problems differ in terms of charging types, applied solution methods, and the objectives. On the other hand, street sweeping problem is less studied but practically important problem in waste collection area. To the best of our knowledge, there is no study in literature on the problem of routing electric powered street sweepers however, in practice the use of these type of vehicles is rapidly increasing. Therefore, with this thesis we contribute to fill this gap in literature on waste collection area.

CHAPTER 4

SOLUTION METHODOLOGY

In this chapter, we present the proposed novel mixed integer programming model for the solution of electric street sweepers with time windows and intermediate stops. The problem is considered as an arc routing problem. However, two sides of the streets must be swept separately and this causes a complexity for solving the problem. Therefore, arc routing problem is solved by using nodes, which are junction points of the streets. The sweepers travel from node to node to cover the related arc. In literature, there are studies about arc routing problems, which make node-to-node transportation to cover edges (Assad and Golden, 1995). This type of formulation is mostly used for capacitated arc routing problems (Golden and Wong, 1981). The mathematical model is defined on a given network of arcs to be serviced by street sweepers. Given $G = \left(V_{0,N+1}, A\right)$, with the set of arcs $A = \{(i,j) | i,j \in V_{0,N+1}\}$ while $V_{0,N+1}$ represents the set of nodes including a single depot and dummy nodes. In order to use directed arcs for sweeping each side of the streets, each arc is replicated with each one representing a direction. From node i to node j represents both directions between these two nodes. Nodes i and j are replicated as i' and j' respectively to represent the both directions of the streets between node j and i. The distance between i and i'; j and j' is accepted as zero ($d_{ii'} = d_{jj'} = 0$). In a sweeping problem, all arcs may not be required to be serviced. Therefore, A' represents the set of required arcs to be serviced. For disposal operations, each node may not include a disposal site so V_D is the set of nodes that include a disposal site and V_{ND} represents the nodes do not include any disposal site while $V_D \cup V_{ND} = \{V\}$. There are K types of sweepers and they differ according to their bin capacity and the initial battery level. We assume different vehicles can service certain street types due to size differences. The parameters and decision variables required for the model are given in below.

Parameters:

$0, N + 1$: Depot instances

V:	Set of nodes $V = \{1, \dots, N\}$
$V_{0,N+1}$:	Set of nodes including depot instances
A:	Set of all arcs $A = \{(i,j) \vert i,j \in V_{0,N+1}\}$
A':	Set of required arcs which are serviced
V_D:	Set of nodes which include disposal sites
V_{ND}:	Set of nodes which does not include disposal sites
K:	Set of vehicle types
t_{ij}:	Additional time of servicing arc (i,j)
t'_{ij}:	Time of traversing arc (i,j)
d_{ij}:	Arc distance for (i,j)
r_k:	Additional energy consumption rate when servicing by vehicle k per unit distance
r'_k:	The energy consumption rate of vehicle k when traversing
c_{ijk}:	Amount of energy consumption when traversing arc (i,j) by vehicle k per distance $(r'_k d_{ij})$
$engdisp_k$:	The energy consumption of disposal operations for vehicle k
g_k:	Recharging time of vehicle k per energy unit
B_k:	Battery capacity of vehicle k
Q_k:	Bin capacity of vehicle k
τ_{ij}:	The expected waste amount on the arc $(i,j) \in A'$
ra_{ij}:	{1, if arc (i,j) is required to be swept 0, otherwise}
γ_i:	{1, if there is a disposal site on node i 0, otherwise}
α_i:	{1, if there is a charging station on node i 0, otherwise}

$[e_{ik}, l_{ik}]$: Time window with the earliest and latest start of vehicle k at node i

$[a_k, b_k]$: Lunch break time window for each vehicle k

t^u: Lunch break duration

t^d: Disposal operation time

t^r: Rest break duration

Decision Variables:

x_{ijk}: Binary decision variable indicating if the arc (i,j) is serviced by vehicle k

z_{ijk}: Binary decision variable indicating if the arc (i,j) is travelled by vehicle k

w_{ijk}: Binary decision variable indicating if lunch break is taken on traveling arc (i,j) by vehicle k

δ_{ijk}: Binary decision variable indicating if a rest break is taken on traveling arc (i,j) by vehicle k

μ_{ik}: Binary decision variable indicating if vehicle k dispose of its load on node i

η_{ik}: Amount of disposal on node i by vehicle k

s_{ik}: Starting time of servicing or traversing on node i by vehicle k

p_{ik}: Amount of partial charge of vehicle k on node i

y_{ik}: Battery state of charge for vehicle k on arriving node i

q_{ik}: Current bin load of vehicle k on arriving node i

The mathematical model of the problem is formulated as a mixed-integer linear program and aims to find a set of routes for sweepers while collecting wastes from required streets within a time window with the objective of minimizing the total energy consumption of vehicles.

$$Minimize\ \sum_{i\in V_{all}}\sum_{j\in V_{all}}\sum_{k\in K} c_{ijk}z_{ijk} + r_k x_{ijk} + \sum_{i\in V_{all}}\sum_{k\in K} engdisp_k * \eta_{ik} \tag{1}$$

st.

$$\sum_{j\in V_{all}} z_{0jk} = 1 \qquad \forall k \in K, \forall (0,j) \in A \tag{2}$$

$$\sum_{i\in V_{all}} z_{i,N+1k} = 1 \qquad \forall k \in K, \forall (i,N+1) \in A \tag{3}$$

$$\sum_{k\in K} x_{ijk} = ra_{ij} \qquad \forall i,j \in V_{all}, i \neq j, \forall (i,j) \in A' \tag{4}$$

$$\sum_{\substack{i=0\\ i\neq j}}^{N} z_{ijk} - \sum_{\substack{i=1\\ i\neq j}}^{N+1} z_{jik} = 0 \qquad \forall k \in K, \forall j \in V \;\; \forall (i,j),(j,i) \in A \tag{5}$$

$$z_{ijk} - x_{ijk} \geq 0 \qquad \forall i,j \in V_{all}, \forall (i,j) \in A, \forall k \in K \tag{6}$$

$$y_{ik} + \alpha_i p_{ik} - (c_{ijk} z_{ijk} + r_k x_{ijk}) - B_k\big(1 - z_{ijk}\big) \leq y_{jk}$$

$$\forall i,j \in V_{all}, \forall (i,j) \in A\ , \forall k \in K \tag{7}$$

$$y_{ik} + \alpha_i p_{ik} - (c_{ijk} z_{ijk} + r_k x_{ijk}) + B_k\big(1 - z_{ijk}\big) \geq y_{jk}$$

$$\forall i,j \in V_{all}, \forall (i,j) \in A\ , \forall k \in K \tag{8}$$

$$y_{0k} \leq B_k \qquad \forall k \in K \tag{9}$$

$$y_{ik} + \alpha_i p_{ik} \leq B_k \forall i \in V_{all}, \forall k \in K \tag{10}$$

$$q_{ik} + \tau_{ij} x_{ijk} - \gamma_i \eta_{ik} - Q_k\big(1 - z_{ijk}\big) \leq q_{jk}$$

$$\forall i,j \in V_{all}, \;\; \forall (i,j) \in A, \forall k \in K \tag{11}$$

$$q_{ik} + \tau_{ij} x_{ijk} - \gamma_i \eta_{ik} + Q_k\big(1 - z_{ijk}\big) \geq q_{jk}$$

$$\forall i,j \in V_{all}, \;\; \forall (i,j) \in A, \forall k \in K \tag{12}$$

$$\eta_{ik} \leq q_{ik} \qquad \forall i \in V_{all}, \; \forall k \in K \tag{13}$$

$$\eta_{ik} = 0 \qquad \forall i \in V_{ND}, \forall k \in K \tag{14}$$

$$M \mu_{ik} \geq \eta_{ik} \qquad \forall i \in V_D, \forall k \in K \tag{15}$$

$$q_{ik} \leq Q_k \qquad \forall i \in V_{all}, \; \forall k \in K \tag{16}$$

$$\sum_{i\in V_{all}} \sum_{j\in V_{all}} w_{ijk} + \delta_{ijk} \geq 1 \qquad \forall (i,j) \in A, \; \forall k \in K \tag{17}$$

$$w_{ijk} + \delta_{ijk} \; \leq \; z_{ijk} \qquad \forall i,j \in V_{all}, \forall (i,j) \in A, \forall k \in K \tag{18}$$

$$a_k - M\big(1 - w_{ijk}\big) \leq s_{ik} \leq b_k + M\big(1 - w_{ijk}\big)$$

$$\forall i \in V_{all} \;\; \forall (i,j) \in A, \forall k \in K \tag{19}$$

$$s_{ik} + t_{ij} x_{ijk} + t'_{ij} z_{ijk} + t^u w_{ijk} + t^r \delta_{ijk} + g_k p_{ik} + t^d \mu_{ik} - M\big(1 - z_{ijk}\big)$$

$$\leq \ s_{jk} \qquad \forall i,j \in V_{all}, \forall(i,j) \in A\,, \forall k \in K \tag{20}$$

$$e_{ik} \leq s_{ik} \leq l_{ik} \qquad \forall i \in V_{all}, \forall k \in K \tag{21}$$

$$x_{ijk}, z_{ijk}, w_{ijk}, \delta_{ijk}, \mu_{ik} \in \{0,1\} \qquad \forall i \in V_{all}, \forall(i,j) \in A\,, \forall k \tag{22}$$

$$\eta_{ik}, s_{ik}, p_{ik}, y_{ik}, q_{ik} \geq 0 \qquad \forall i,j \in V_{all}, \forall k \in K \tag{23}$$

The objective function (1) minimizes the energy consumption of vehicles while traversing, servicing and doing disposal operations. All k vehicles must leave (2) and return (3) to the depot. All required arcs must be serviced (4). Inflow and outflow must be equal for all nodes (5). If an arc is serviced by a vehicle then that vehicle must traverse on that arc as well (6).

Constraints (7) and (8) determine the current energy level y_{jk} considering the previous charging level y_{ik} and the energy consumption also after travelling arc (i,j) and also partial recharging amount p_{ik} if there is a recharging operation on that node i. Constraint (9) ensures that all vehicles start their tour with fully charged. Constraint (10) ensures that current energy level cannot exceed the maximum battery level.

Constraints (11) and (12) ensure that current bin load of the sweeper in the next node q_{jk} depends on the load of the previous node q_{ik} plus the swept wastes and considering the disposal if there is a disposal operation on that node i. Constraints (13)-(15) determine the disposal amount of a vehicle while constraint (16) restricts that load of the vehicle cannot exceed the vehicle capacity.

Constraint (17) states that all vehicles must take at least one break. A break must be taken between two connected nodes (18). A lunch break must be taken on a specific time (19). Time windows, service time and other time consuming operations are covered with constraints (20) and (21). Constraint (22) and (23) imposes binary variables and non-negativity.

The contributions of the proposed model can be listed as use of heterogeneous fleet of electric powered street sweepers with varying capacities and energy consumption rates based on route and work including the time window restrictions, lunch and rest breaks are considered.

CHAPTER 5

COMPUTATIONAL EXPERIMENTS

In this chapter, a case study and computational experiments are presented to both verify the model and evaluate the performance of the mathematical model. The complexity of the mathematical model is analyzed with different size of generated data instances on hypothetical street networks as well as on a real street network taken from using by Python, Bing Maps API.

5.1. Case Study

In order to see the problem in real life, the formulated mathematical model is demonstrated by solving a small case study with real data, using IBM ILOG CPLEX 12.8 solver on i7, 2.59 GHz with 8 GB of RAM. For this case study, a small street area is selected from İzmir, Turkey. In Figure 5.1., a representation of the selected streets is shown which taken from Bing Maps. In order to show the flow direction of traffic, replication of each node is needed. In this case study, each node represents an intersection of the streets. The street network constitutes 14 nodes and one depot which is node 0. The replicated numbers of each node are also given in the Figure 5.1.

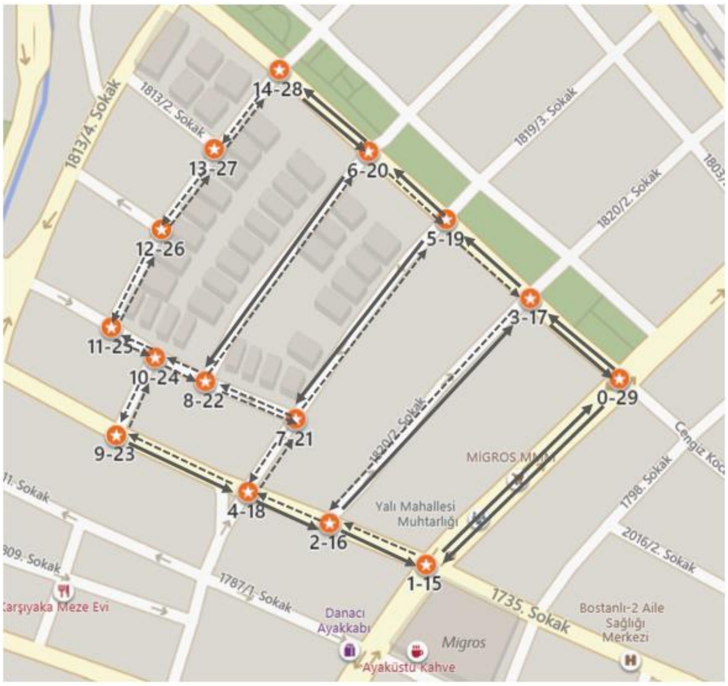

Figure 5.1. Representation of a Street Network in İzmir

The solid lines show that which streets must be swept and the dashed lines indicate that there is no need to sweep those sides of the streets but can be used to traverse. Node 9 and node 14 include a recharging station and node 2, 5, and 12 are the disposal sites in this problem.

In this case study, the real distance matrix between intersection points is taken using by Python, Bing Maps API. Time of traversing and additional time of servicing the streets are determined according to distances with maximum and average speed information of the vehicles which are 25 km/h and 10 km/h respectively. Amount of energy consumption is determined due to the travelling distances and consumption rate of the vehicles and calculated as $c_{ijk} = r'_k * d_{ij}$.

There are three electric powered sweepers used with bin capacity 2.1 m^3, 2.05 m^3 and 2 m^3 respectively. Vehicles are assumed to be not fully charged at the beginning. Energy consumption rates of these vehicles are determined 10 kWh, 9 kWh, and 8 kWh while just traversing the street and additional energy consumption rates are set to 8 kWh, 6 kWh, and 4 kWh while servicing.

Both recharging and disposal operations are time-consuming operations. The recharging time of the vehicles is different and in this problem it is determined as 2, 1.9, and 1.8 minutes per energy unit. Each disposal operation takes 5 minutes. Additional to the constraints, at least one break must be taken according to working regulations. A lunch break and one rest break duration are determined as 30 minutes. Another consideration is a lunch break must be taken in a specific time interval at noon. Finally, time windows are determined according to the availability of sweeping streets. This parameter information is shown in Table 5.1. as well.

Table 5.1. Parameter Values for the Case Study

	Sweeper 1	**Sweeper 2**	**Sweeper 3**
Capacity (Q_k)	2,1 m^3	2,05 m^3	2 m^3
Energy Cons. rate (traversing) (r'_k)	10 kWh	9 kWh	8 kWh
Energy Cons. rate (servicing) (r_k)	8 kWh	6 kWh	4 kWh

Table 5.1 (cont'd). Parameter Values for the Case Study

Energy Cons. rate (disposal) ($engdisp_k$)	1 kWh	1 kWh	1 kWh
Recharging Time (g_k)	2 min	1,9 min	1,8 min
Initial Battery Levels	50%	30%	40%

This example is solved with IBM ILOG CPLEX 12.8 solver in 94,71 seconds. After solving the case study with the above information, normal time windows and low demand, the solution representations for three sweepers are illustrated in Figure 5.2., Figure 5.3., and Figure 5.4. The solid lines show that those streets were swept and the dashed lines indicate that there is no sweeping operation for those streets but used for traversing. The objective function value is 101,192 kWh for this case study example.

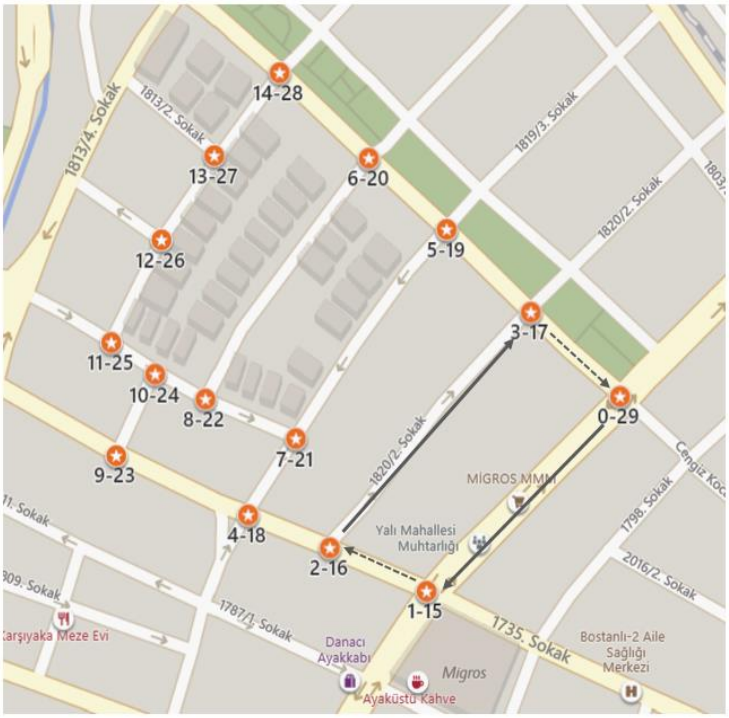

Figure 5.2. Optimal Route for Sweeper 1

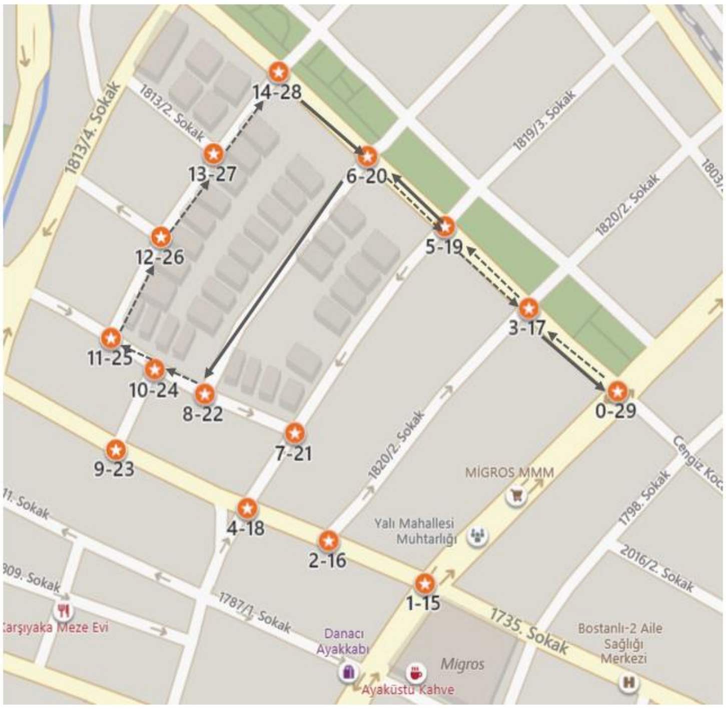

Figure 5.3. Optimal Route for Sweeper 2

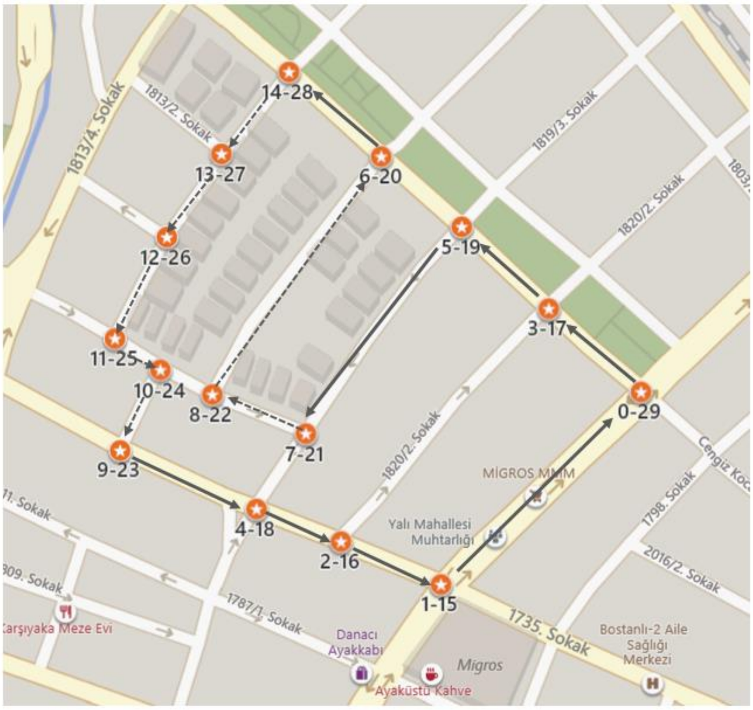

Figure 5.4. Optimal Route for Sweeper 3

According to the results, since the energy consumption rates and recharging time of the third vehicle are lower than the other vehicles, the longest route belongs to this vehicle and it travels and services more streets comparing to the others. Therefore, in order to complete its tour, disposal and recharging operations were needed for this sweeper. For this purposes in node 12 there is a disposal operation with 0,85 m^3 and in node 9, third sweeper is partially recharged as 4,17%. Besides, sweeper 2 is recharged in node 28 as 4% to complete its tour as well. At the end, final battery level of three sweepers are 28%, 0%, 0% respectively.

To measure the performance of the mathematical model in condition of varying time windows and demand rates, this case study is solved for different scenarios. 9 different scenarios are constructed considering time windows are relaxed, normal, and tight when the demands are low, medium, and high. These scenarios are solved with IBM ILOG CPLEX 12.8 solver, the objective function results and solution time are given in Table 5.2.

Table 5.2. Results of the Case Study

Scenario	**Time Windows**	**Demand**	**Objective (kWh)**	**CPU (sec)**
1	Relaxed	Low	98,156	22,44
2	Relaxed	Medium	100,156	25,53
3	Relaxed	High	100,156	32,57
4	Normal	Low	101,192	94,71
5	Normal	Medium	101,192	80,42
6	Normal	High	105,192	263,13
7	Tight	Low	101,192	50,36
8	Tight	Medium	101,192	98,45
9	Tight	High	105,192	402,25

Since this case study problem size is small, we got a solution in a reasonable time. Solving the problem in such a small size brings us to be able to make comments about the changing effect of demand and time windows. According to the results, we conclude that tight time windows and high demands together have an effect that increases the solution time. With this real life case study, the proposed mathematical model is verified and validated as well.

5.2. Design of Experiments

After the mathematical model is verified, the performance of the model is analyzed by generated data on a hypothetical street network. For this design of experiment, the inputs that are given in Table 5.3 is used. Nodes represent the intersection of the streets and each edge is called as a street. The mathematical model is solved using IBM ILOG CPLEX 12.8 solver on i7, 2.59 GHz with 8 GB of RAM.

Table 5.3. The Input for the Computational Experiments

Total Number of Nodes	Total Number of Arcs	Required Arc %
30	74	15%
40	102	20%
50	130	25%
		30%
		35%
		40%

We use 15, 20, and 25 nodes, and in order to show two way traffic flow the replication of each node is made, then total number of nodes are become 30, 40, and 50 respectively. There are 22, 31, and 40 edges and for direction transitions dummy arcs are created as well, therefore the total number of arcs are 74, 102, and 130 respectively. Since not all streets need to be swept, there are six different percentages of required arcs which are 15, 20, 25, 30, 35, and 40 percent of all arcs need to be swept. With this 3 different number of arcs and 6 different required arc percentages, there are 18 scenarios. As an example, a representation of the network for the case with 50 nodes and 130 arcs is shown in Figure 5.5. The numbers symbolize the node numbers and their replication numbers respectively.

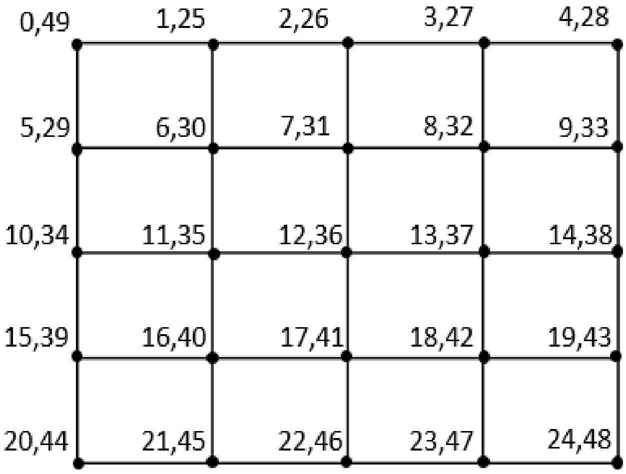

Figure 5.5. The Network Graph of the Experiment

For the experiment, the number of required arcs and corresponding demand varies in each instance while other parameters are constant. The required arcs are determined randomly and we take different percentages of total arcs (without considering dummy arcs) to have demand. Demand quantities are generated using uniform distribution between 100 and 200. It is assumed that number of vehicles for all scenarios is 3 and vehicles have different bin capacities. For the design of experiment, three battery level scenarios are identified considering the initial battery level of these vehicles, which are low battery level scenario (LBLS), medium battery level scenario (MBLS), and full battery level scenario (FBLS). With 3 battery level scenarios, 3 different number of arcs, and 6 different percentage of required arcs, we have 54 scenarios. For each scenario, 10 replications are generated and solved with different demand rates. Therefore, in total there are 540 different instances.

5.2.1. Full Battery Level Scenario

In this battery level scenario, all vehicles have 100% initial battery level. The proposed mathematical model is solved using IBM ILOG CPLEX 12.8 solver on i7, 2.59 GHz with 8 GB of RAM. The inputs and the average results for the full battery level scenario are given in Table 5.4.

Table 5.4. Average Results of the Full Battery Level Scenario Instance Sets

Scenario	# of Nodes	Total # of Nodes	# of Arcs	Total # of Arcs	Required Arc %	# of Required Arcs	Avg. Obj.	Avg. CPU (sec)	Avg. Gap %
1	15	30	22	74	15%	7	104,94	2,20	0,0%
2	15	30	22	74	20%	9	142,23	10,53	0,0%
3	15	30	22	74	25%	11	162,11	11,76	0,0%
4	15	30	22	74	30%	14	184,93	21,72	0,0%
5	15	30	22	74	35%	16	214,70	54,85	0,0%
6	15	30	22	74	40%	18	237,57	132,15	0,0%
7	20	40	31	102	15%	10	128,94	9,30	0,0%
8	20	40	31	102	20%	13	177,06	124,86	0,0%
9	20	40	31	102	25%	16	201,52	194,96	0,0%
10	20	40	31	102	30%	19	215,76	91,15	0,0%
11	20	40	31	102	35%	22	254,91	731,76	0,0%
12	20	40	31	102	40%	25	288,81	1168,45	0,0%
13	25	50	40	130	15%	12	156,69	25,57	0,0%
14	25	50	40	130	20%	16	194,88	95,05	0,0%
15	25	50	40	130	25%	20	243,23	207,71	0,0%
16	25	50	40	130	30%	24	297,5652	1462,17	0,0%
17	25	50	40	130	35%	28	329,497	3178,29	0,0%
18	25	50	40	130	40%	32	405,9114	7200*	8,7%

***Cut off point (2 hours) results**

The computational results show that as the number of required arcs increases, the solution time increases as well. Since all vehicles start their route with full battery, in most cases no recharging operation is performed. Therefore, the optimal solution was found within 2 hours for all scenarios except scenario number 18 since this scenario has high number of required arc percentage. The computational results are visualized in order to show the effect of increasing number of nodes and the required arc percentages. In Figure 5.6., it is easily seen that when total number of nodes are 50, the solution time has increased exponentially. However, solving the problem for 30 nodes the increase in solution time is linear. The detailed tables are given in the Appendix 1.

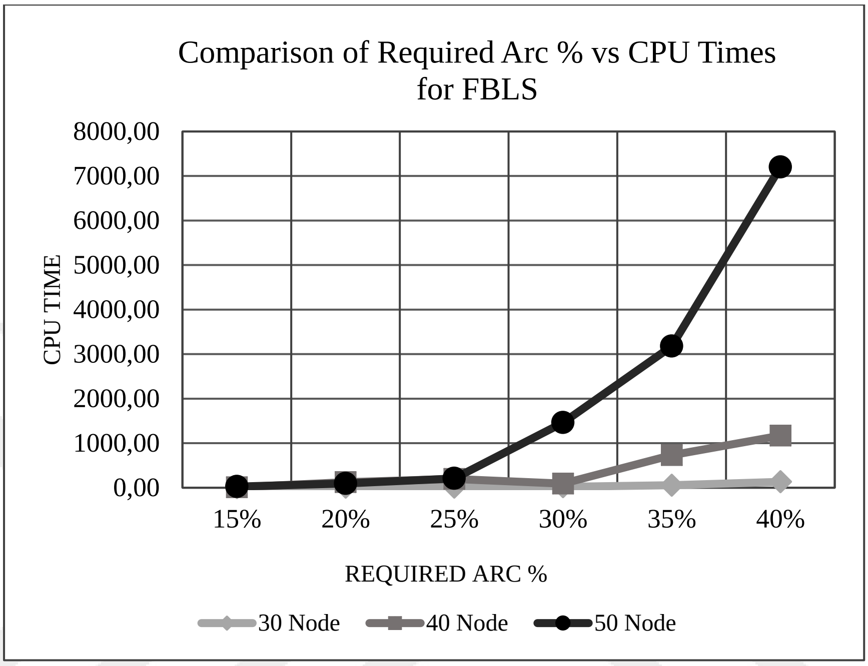

Figure 5.6. Comparison Chart for the Required Arc Percentage vs. CPU Time for FBLS

In Figure 5.7., the effect of increasing the number of required arcs on the solution time of the model is visualized. The solution time is small when we use small number of required arcs. On the other hand, when 30, 35, and 40 percent required arcs are used the solution time increased and it is related with total number of nodes as well.

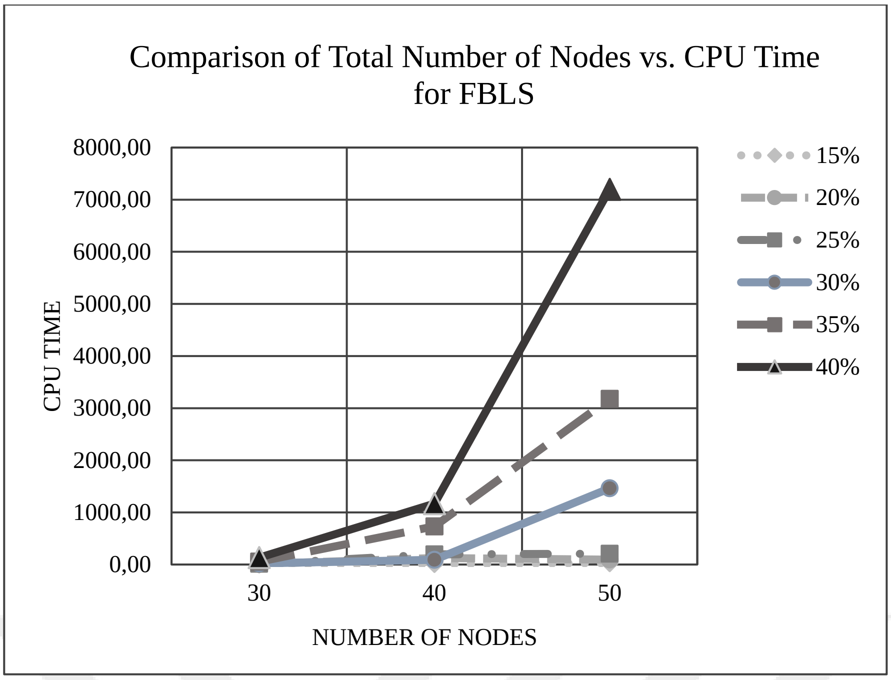

Figure 5.7. Comparison Chart for the Total Number of Nodes vs. CPU Time for FBLS

5.2.2. Medium Battery Level Scenario

For medium battery level, uniform distribution between 60% and 80% for the battery charge is used and different battery levels are determined for three vehicles as well. The inputs and the average results for the medium battery level scenario are given in Table 5.5.

Table 5.5. Average Results of the Medium Battery Level Scenario Instance Sets

Scenario	# of Nodes	Total # of Nodes	# of Arcs	Total # of Arcs	Required Arc %	# of Required Arcs	Avg. Obj.	Avg. CPU (sec)	Avg. Gap %
1	15	30	22	74	15%	7	104,90	1,96	0,0%
2	15	30	22	74	20%	9	142,23	7,67	0,0%
3	15	30	22	74	25%	11	165,70	16,29	0,0%
4	15	30	22	74	30%	14	184,93	23,12	0,0%
5	15	30	22	74	35%	16	214,70	48,86	0,0%
6	15	30	22	74	40%	18	237,57	101,92	0,0%
7	20	40	31	102	15%	10	128,94	9,86	0,0%
8	20	40	31	102	20%	13	177,06	54,11	0,0%
9	20	40	31	102	25%	16	201,76	254,50	0,0%

Table 5.5 (cont'd). Average Results of the Medium Battery Level Scenario Instance Sets

10	20	40	31	102	30%	19	215,76	111,48	0,0%
11	20	40	31	102	35%	22	254,91	962,70	0,0%
12	20	40	31	102	40%	25	292,44	1756,36	0,0%
13	25	50	40	130	15%	12	156,69	28,44	0,0%
14	25	50	40	130	20%	16	198,22	165,52	0,0%
15	25	50	40	130	25%	20	243,23	196,76	0,0%
16	25	50	40	130	30%	24	309,60	4713,64	1,7%
17	25	50	40	130	35%	28	358,08	7200*	7,1%
18	25	50	40	130	40%	32	428,56	7200*	13,4%

***Cut off point (2 hours) results**

When initial battery levels are set due to medium battery level scenario, solution time increases fast for the large number of scenarios. For the scenario number of 16, 17, and 18, the optimal solution could not be found in 2 hours and the average optimality gaps are 1.7%, 7.1%, and 13.4% respectively. For the scenario 18, average optimality gap is found as 13,4%, however, there is no solution found within 2 hours for 8 replications out of 10 replications. The detailed tables including individual results for every replication are given in Appendix 2. In Figure 5.8., the solution time is plotted considering the total number of nodes for the medium battery level scenario.

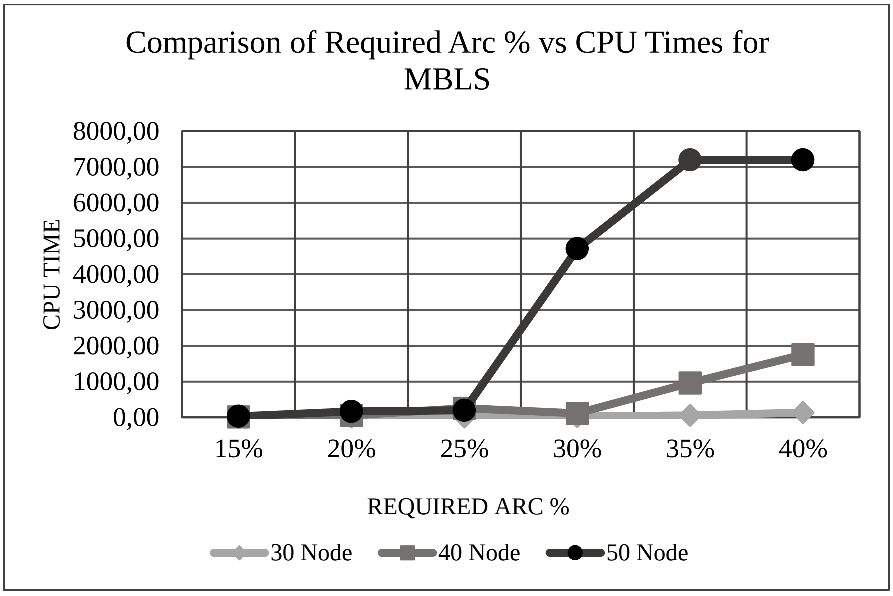

Figure 5.8. Comparison Chart for the Required Arc Percentage vs. CPU Time for MBLS

When the vehicles' battery level are set to as medium battery level, solution time of the proposed mathematical model has started to increase with 40 nodes case comparing with the first scenario. After required arc percentage is 25, solution time is increased fast for the scenarios with 50 nodes but not exponentially. Because, time limitation is 2 hours and after 35% required arcs, the optimal solution could not be found within 2 hours and this increase was stable after that point.

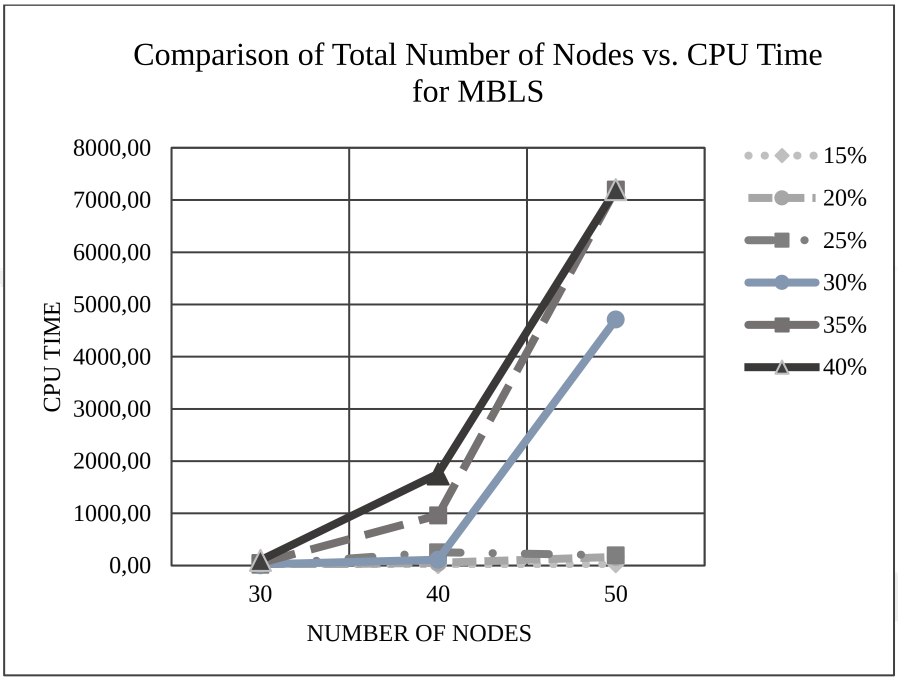

Figure 5.9. Comparison Chart for the Total Number of Nodes vs. CPU Time for MBLS

In Figure 5.9., the effect of increasing the number of required arcs on the solution time of the model is visualized for MBLS. The solution time is small when we use small number of required arcs as the previous scenario. However, this time for 30% required arcs, the increase is not exponential and the effect of required arc percentage 35% and 40% is quite close to each other considering the solution time.

5.2.3. Low Battery Level Scenario

Low battery level is determined uniform distribution between 30% and 50% and initial battery levels are different for each vehicle. The inputs, average objective function value, average solution time, and optimality gap is given in Table 5.6.

Table 5.6. Average Results of the Low Battery Level Scenario Instance Sets

Scenario	**# of Nodes**	**Total # of Nodes**	**# of Edges**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Avg. Obj.**	**Avg. CPU (sec)**	**Avg. Gap %**
1	15	30	22	74	15%	7	105,37	1,40	0,0%
2	15	30	22	74	20%	9	155,85	20,76	0,0%
3	15	30	22	74	25%	11	173,16	19,90	0,0%
4	15	30	22	74	30%	14	207,85	147,87	0,0%
5	15	30	22	74	35%	16	237,28	348,47	0,0%
6	15	30	22	74	40%	18	247,28	316,88	0,0%
7	20	40	31	102	15%	10	139,52	24,40	0,0%
8	20	40	31	102	20%	13	202,36	181,17	0,0%
9	20	40	31	102	25%	16	236,31	867,92	0,0%
10	20	40	31	102	30%	19	259,03	1265,06	0,0%
11	20	40	31	102	35%	22	296,47	5592,24*	2,9%
12	20	40	31	102	40%	25	320,16	5424,62*	3,2%
13	25	50	40	130	15%	12	161,71	122,40	0,0%
14	25	50	40	130	20%	16	251,69	7200*	12,6%
15	25	50	40	130	25%	20	326,81	7200*	18,3%
16	25	50	40	130	30%	24	NA	7200*	NA
17	25	50	40	130	35%	28	NA	7200*	NA
18	25	50	40	130	40%	32	NA	7200*	NA

***Cut off point (2 hours) results**

In the low battery level scenario, the model is solved optimally for 15 nodes cases in a short time. For the case with 20 nodes and required arc percentage is 35%, which is, scenario 11, the model was not solved in 2 hours for 6 out of 10 replications. Besides, when the scenario 12 is solved, no optimal solution was found within 2 hours for 5 out of 10 replications. For the case with 25 nodes, with smaller required arc percentage (15%) the model was solved optimally. However, for last three scenarios, there is no solution found within 2 hours for any replication therefore the objective function value and optimality gap is given as "NA" in the above table. As the result of this experiment, when the total number of nodes and arcs are increased, with the high number of required arcs the model cannot be solved optimally in a reasonable time. The detailed tables are given in the Appendix 3.

At the end of the computational experiment, the results are visualized for the lower battery level scenario in Figure 5.10., by drawing a line chart. In this chart, the CPU time (in seconds) is plotted while indicating the effect of required arc percentage on the solution time for different total number of nodes. If the required arc percentage is 15, the model can be solved in a short time. When the required arc percentage is 20, the solution time begins to increase for the cases with total number of nodes 30, 40, and 50.

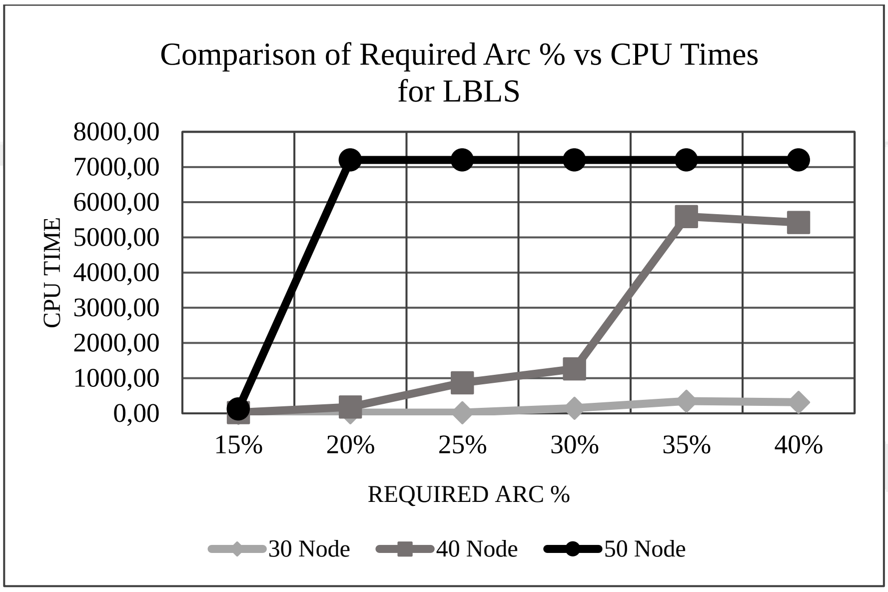

Figure 5.10. Comparison Chart for the Required Arc Percentage vs. CPU Time for LBLS

In another result graph which is indicated in Figure 5.11., the average CPU time is plotted for showing the relationship between required arc percentage and the total number of nodes in terms of the solution time as well. The effect of required arc percentage 35% and 40% is almost same with each other considering the solution time. We can understand that when the initial battery levels of the vehicles are low, the demand intensity has strong effect on the solution time.

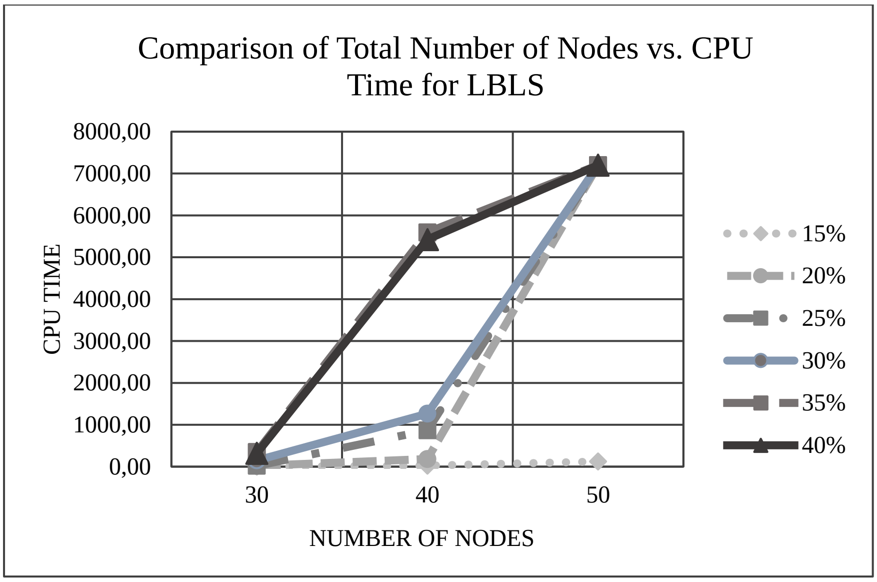

Figure 5.11. Comparison Chart for the Total Number of Nodes vs. CPU Time for LBLS

After the evaluation of different battery level scenarios, we present a comparison of three scenarios considering number of nodes and required arc percentages. In Figure 5.12., this comparison is given for 30 nodes. We can see the effect of low battery levels on the solution time is remarkable comparing the other levels. In Figure 5.13., the same comparison is given for the 40 nodes and we can make the same comment for this comparison as well, except the increase in the solution time is observable for small number of required arc percentages too.

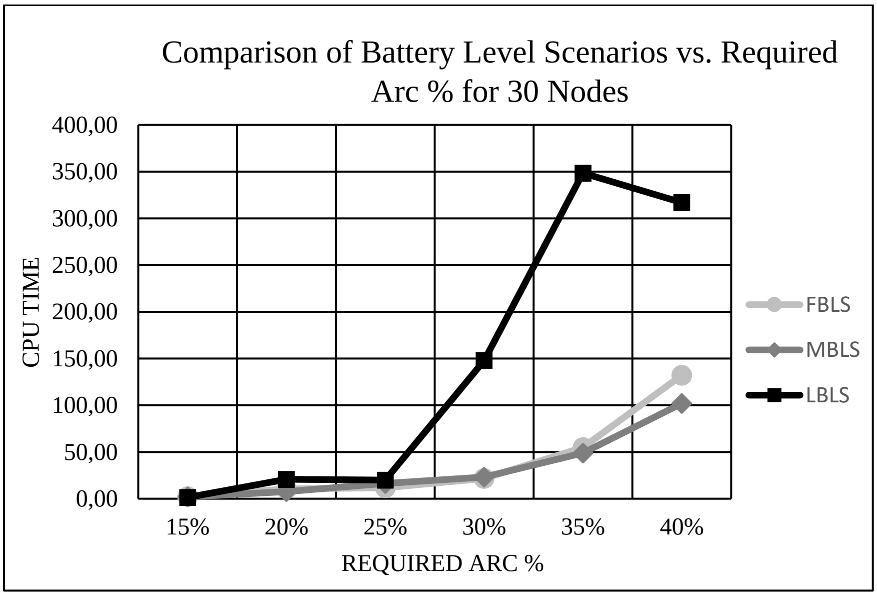

Figure 5.12. Comparison Chart of Battery Level Scenarios for 30 Nodes

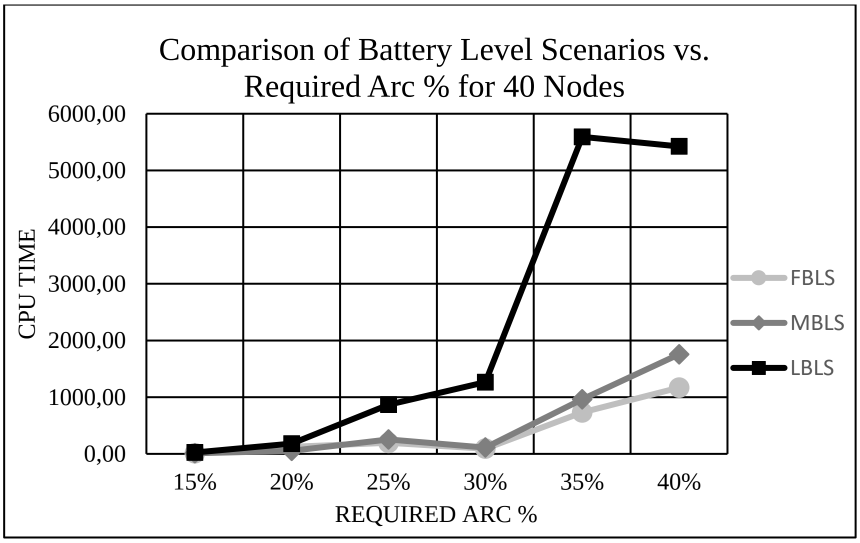

Figure 5.13. Comparison Chart of Battery Level Scenarios for 40 Nodes

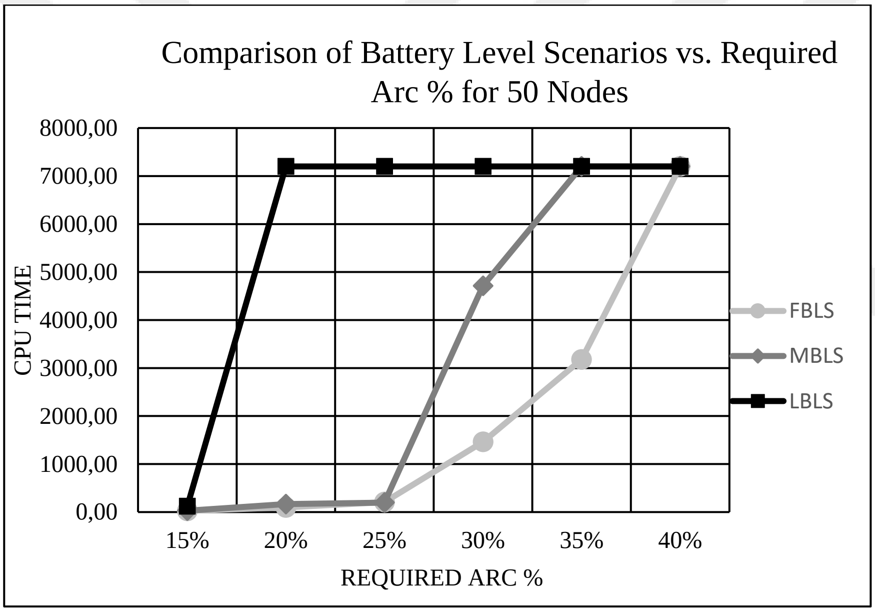

Figure 5.14. Comparison Chart of Battery Level Scenarios for 50 Nodes

In Figure 5.14., the comparison chart for 50 nodes is given. This chart shows that initial battery levels are more crucial compare to the changes in the other parameters. Together high number of nodes and demand intensity with low initial battery levels, the solution time of the model increases.

We developed this design of experiment to understand the performance of the mathematical model and the effects of changes in parameters with different combinations. Experimentation with small and large data scenarios are made to analyze complexity of the problem. The results of the computational experiments show that total number of arcs and required arc percentages have important effect on performance of the model. For small size problems and with high initial battery levels, the mathematical model was solved relatively quickly. However, when the problem size has increased, with low initial battery levels, the solution time of the model has also increased. Besides, we can conclude that the decision of recharging vehicles in their route to increase the solution time. That is why low battery level scenario is worse than the other battery level scenarios considering the solution time of the mathematical model.

According to our design of experiments, we found that solving the problem with 130 arcs (including two way sweeping) with hard constraints requires more than 2 hours. However, real-life problems include more number of arcs hence we predict that for larger problem sizes this problem is hard to solve. Our problem is a special case of the RPP and CARP in terms of modeling the problem. In literature, CARP is known to be NP-hard and since CARP is a special case of street sweeping problem, our problem is NP-hard as well (Eiselt et.al. 1995, Golden and Wong 1981).

CHAPTER 6
CONCLUSIONS

Street sweeping is an important part of municipal waste collection. In this thesis, we look at the problem of determining routing of electric powered street sweepers to service a set of predetermined arcs in city, while considering realistic operational constraints like disposal and charging planning, lunch and rest breaks etc. The detailed problem definition is given in Chapter 2. Efficient usage of public funds is crucial for this type of problems. Literature on waste collection and different versions of electric vehicle routing problems are reviewed in Chapter 3. The necessity of to fill the gap in literature and practically, minimizing the energy consumption in waste collection problems constitute the motivation of this thesis. We present a novel mixed integer programming model for the solution of this problem in Chapter 4. This is the first comprehensive mathematical model for this problem. Contributions of the proposed model can be listed as use of heterogeneous fleet of electric powered street sweepers with varying capacities and energy consumption rates based on route and work; inclusion of time window restrictions, lunch and rest breaks are considered. In Chapter 5, since street sweeping is a municipal operation, a case study is addressed and solved with real-life instances using Bing Maps API. The formulated model is demonstrated and verified with this case study. The complexity analysis of the model and experimentation with small and large data scenarios are made including different battery levels. For small size problems and with high initial battery levels the mathematical model was solved in a short time. When the problem size has increased, the solution time of the model has also increased.

6.1. Future Study

Solving the problem with 130 arcs requires more than 2 hours and in literature arc routing problems such as RPP and CARP which are simplified version of our problem called NP-hard problems, we believe there will be a need for more efficient algorithms to solve this problem (Eiselt et.al. 1995, Golden and Wong 1981). For this reason, hybrid methods which combines an exact method and heuristic approaches can be one of the solution methodology options that can be tried for this problem. In addition to this, some heuristics can be applied which have been proven useful in similar routing problems that results in some cost-efficient improvements.

REFERENCES

Assad, A. A., & Golden, B. L. (1995). Chapter 5 Arc routing methods and applications. *Handbooks in Operations Research and Management Science*, *8*(C), 375–483. https://doi.org/10.1016/S0927-0507(05)80109-4

Benjamin, A. M., & Beasley, J. E. (2010). Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities. *Computers and Operations Research*, *37*(12), 2270–2280. https://doi.org/10.1016/j.cor.2010.03.019

Bodin, L. D., & Kursh, S. J. (1978). A Computer-Assisted System for the Routing and Scheduling of Street Sweepers. *Operations Research*, *26*(4), 525–537. https://doi.org/10.1287/opre.26.4.525

Buhrkal, K., Larsen, A., & Ropke, S. (2012). The Waste Collection Vehicle Routing Problem with Time Windows in a City Logistics Context. *Procedia - Social and Behavioral Sciences*, *39*, 241–254. https://doi.org/10.1016/j.sbspro.2012.03.105

Desaulniers, G., Errico, F., Irnich, S., & Schneider, M. (2016). Exact Algorithms for Electric Vehicle-Routing Problems with Time Windows. *Operations Research*, *64*(6), 1388–1405. https://doi.org/10.1287/opre.2016.1535

Eiselt, H. A., Gendreau, M., & Laporte, G. (1995a). Arc Routing Problems, Part I: The Chinese Postman Problem. *Operations Research*, *43*(2), 231–242. https://doi.org/10.1287/opre.43.2.231

Eiselt, H. A., Gendreau, M., & Laporte, G. (1995b). Arc Routing Problems, Part II: The Rural Postman Problem. *Operations Research*, *43*(3), 399–414. https://doi.org/10.1287/opre.43.3.399

Goeke, D., & Schneider, M. (2015). Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research*, *245*(1), 81–99. https://doi.org/10.1016/j.ejor.2015.01.049

Golden, B. L., Assad, A. A., & Wasil, E. A. (2002). 10. Routing Vehicles in the Real World: Applications in the Solid Waste, Beverage, Food, Dairy, and Newspaper Industries. *The Vehicle Routing Problem*, 245–286. https://doi.org/10.1137/1.9780898718515.ch10

Golden, B. L., & Wong, R. T. (1981). Capacitated arc routing problems. *Networks*, *11*(3), 305–315. https://doi.org/10.1002/net.3230110308

Han, H., & Ponce Cueto, E. (2015). Waste collection vehicle routing problem: literature review. PROMET-Traffic&Transportation, 27(4), 345-358.

Hiermann, G., Puchinger, J., Ropke, S., & Hartl, R. F. (2016). The Electric Fleet Size and Mix Vehicle Routing Problem with Time Windows and Recharging Stations. *European Journal of Operational Research*, *252*(3), 995–1018. https://doi.org/10.1016/j.ejor.2016.01.038

Keskin, M., & Çatay, B. (2016). Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, *65*, 111–127. https://doi.org/10.1016/j.trc.2016.01.013

Keskin, M., & Çatay, B. (2018). A matheuristic method for the electric vehicle routing problem with time windows and fast chargers. *Computers and Operations Research*, *100*, 172–188. https://doi.org/10.1016/j.cor.2018.06.019

Solomon, M. M. (1987). Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research*, *35*(2), 254–265. https://doi.org/10.1287/opre.35.2.254

Schneider, M., Stenger, A., & Goeke, D. (2014). The Electric Vehicle-Routing Problem with Time Windows and Recharging Stations. *Transportation Science*, *48*(4), 500–520. https://doi.org/10.1287/trsc.2013.0490

Schneider, M., Stenger, A., & Hof, J. (2015). An adaptive VNS algorithm for vehicle routing problems with intermediate stops. *OR Spectrum*, *37*(2), 353–387. https://doi.org/10.1007/s00291-014-0376-5

Tennant. (2019). 636 Green Machines Series Air Sweeper. Retrieved 17 June 2019, from http://eu.tennantco.com/emea-en/equipment/outdoor-and-city-cleaning/green-machines-series-air-sweeper/636#OverviewTab

UN Environment, United Nations Environment Programme Solid Waste Management. Retrieved 17 July 2019, from https://www.unenvironment.org/explore-topics/resource-efficiency/what-we-do/cities/solid-waste-management

APPENDIX 1 – Computational Results of the Mathematical Model for FBLS

In below tables (Table A1.1 – Table A1.18) the computational results of the mathematical model are given. The instance set names are created according to number of nodes, the sample number, and the replication number respectively while adding after the fixed word “Arc”. For example, “Arc15_1_1” indicates that the instance with 15 nodes, sample 1, and replication 1. The last row of each table shows the results on the average for 10 replications. The model is run for 7200 seconds with CPLEX.

Table A1.1. Results of the 1st Instance Set with 15 Nodes and Full Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc15_1_1	15	30	22	74	15%	7	104,9	1,9	0,0%
Arc15_1_2	15	30	22	74	15%	7	105,4	1,9	0,0%
Arc15_1_3	15	30	22	74	15%	7	104,9	2,0	0,0%
Arc15_1_4	15	30	22	74	15%	7	104,9	2,2	0,0%
Arc15_1_5	15	30	22	74	15%	7	104,9	2,0	0,0%
Arc15_1_6	15	30	22	74	15%	7	104,9	2,3	0,0%
Arc15_1_7	15	30	22	74	15%	7	104,9	2,3	0,0%
Arc15_1_8	15	30	22	74	15%	7	104,9	2,2	0,0%
Arc15_1_9	15	30	22	74	15%	7	104,9	2,3	0,0%
Arc15_1_10	15	30	22	74	15%	7	104,9	2,9	0,0%
Arc15_1_Avg	15	30	22	74	15%	7	**104,94**	**2,2**	**0,0%**

Table A1.2. Results of the 2nd Instance Set with 15 Nodes and Full Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc15_2_1	15	30	22	74	20%	9	142,2	16,1	0,0%
Arc15_2_2	15	30	22	74	20%	9	142,2	8,4	0,0%
Arc15_2_3	15	30	22	74	20%	9	142,2	10,5	0,0%
Arc15_2_4	15	30	22	74	20%	9	142,2	15,5	0,0%
Arc15_2_5	15	30	22	74	20%	9	142,2	7,0	0,0%
Arc15_2_6	15	30	22	74	20%	9	142,2	10,4	0,0%
Arc15_2_7	15	30	22	74	20%	9	142,2	6,0	0,0%

Table A1.2 (cont'd). Results of the 2nd Instance Set with 15 Nodes and Full Battery

Arc15_2_8	15	30	22	74	20%	9	142,2	7,6	0,0%
Arc15_2_9	15	30	22	74	20%	9	142,2	8,0	0,0%
Arc15_2_10	15	30	22	74	20%	9	142,2	15,7	0,0%
Arc15_2_Avg	15	30	22	74	20%	9	**142,23**	**10,5**	**0,0%**

Table A1.3. Results of the 3rd Instance Set with 15 Nodes and Full Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc15_3_1	15	30	22	74	25%	11	162,1	11,1	0,0%
Arc15_3_2	15	30	22	74	25%	11	162,1	12,1	0,0%
Arc15_3_3	15	30	22	74	25%	11	162,1	24,1	0,0%
Arc15_3_4	15	30	22	74	25%	11	162,1	11,0	0,0%
Arc15_3_5	15	30	22	74	25%	11	162,1	11,5	0,0%
Arc15_3_6	15	30	22	74	25%	11	162,1	8,5	0,0%
Arc15_3_7	15	30	22	74	25%	11	162,1	10,0	0,0%
Arc15_3_8	15	30	22	74	25%	11	162,1	9,3	0,0%
Arc15_3_9	15	30	22	74	25%	11	162,1	10,9	0,0%
Arc15_3_10	15	30	22	74	25%	11	162,1	9,2	0,0%
Arc15_3_Avg	15	30	22	74	25%	11	**162,11**	**11,8**	**0,0%**

Table A1.4. Results of the 4th Instance Set with 15 Nodes and Full Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc15_4_1	15	30	22	74	30%	14	184,9	23,7	0,0%
Arc15_4_2	15	30	22	74	30%	14	184,9	23,9	0,0%
Arc15_4_3	15	30	22	74	30%	14	184,9	19,5	0,0%
Arc15_4_4	15	30	22	74	30%	14	184,9	21,8	0,0%
Arc15_4_5	15	30	22	74	30%	14	184,9	19,0	0,0%
Arc15_4_6	15	30	22	74	30%	14	184,9	42,0	0,0%
Arc15_4_7	15	30	22	74	30%	14	184,9	18,1	0,0%
Arc15_4_8	15	30	22	74	30%	14	184,9	15,8	0,0%
Arc15_4_9	15	30	22	74	30%	14	184,9	18,0	0,0%
Arc15_4_10	15	30	22	74	30%	14	184,9	15,4	0,0%
Arc15_4_Avg	15	30	22	74	30%	14	**184,93**	**21,7**	**0,0%**

Table A1.5. Results of the 5th Instance Set with 15 Nodes and Full Battery

Instances	# of Nodes	Total # of Nodes	# of Arcs	Total # of Arcs	Required Arc %	# of Required Arcs	Obj.	CPU (sec)	Gap %
Arc15_5_1	15	30	22	74	35%	16	214,7	43,4	0,0%
Arc15_5_2	15	30	22	74	35%	16	214,7	65,7	0,0%
Arc15_5_3	15	30	22	74	35%	16	214,7	120,3	0,0%
Arc15_5_4	15	30	22	74	35%	16	214,7	48,2	0,0%
Arc15_5_5	15	30	22	74	35%	16	214,7	54,2	0,0%
Arc15_5_6	15	30	22	74	35%	16	214,7	59,3	0,0%
Arc15_5_7	15	30	22	74	35%	16	214,7	34,3	0,0%
Arc15_5_8	15	30	22	74	35%	16	214,7	48,7	0,0%
Arc15_5_9	15	30	22	74	35%	16	214,7	42,1	0,0%
Arc15_5_10	15	30	22	74	35%	16	214,7	32,5	0,0%
Arc15_5_Avg	15	30	22	74	35%	16	**214,70**	**54,9**	**0,0%**

Table A1.6. Results of the 6th Instance Set with 15 Nodes and Full Battery

Instances	# of Nodes	Total # of Nodes	# of Arcs	Total # of Arcs	Required Arc %	# of Required Arcs	Obj.	CPU (sec)	Gap %
Arc15_6_1	15	30	22	74	40%	18	237,6	82,9	0,0%
Arc15_6_2	15	30	22	74	40%	18	237,6	196,7	0,0%
Arc15_6_3	15	30	22	74	40%	18	237,6	186,8	0,0%
Arc15_6_4	15	30	22	74	40%	18	237,6	165,7	0,0%
Arc15_6_5	15	30	22	74	40%	18	237,6	122,1	0,0%
Arc15_6_6	15	30	22	74	40%	18	237,6	69,0	0,0%
Arc15_6_7	15	30	22	74	40%	18	237,6	182,9	0,0%
Arc15_6_8	15	30	22	74	40%	18	237,6	90,8	0,0%
Arc15_6_9	15	30	22	74	40%	18	237,6	120,1	0,0%
Arc15_6_10	15	30	22	74	40%	18	237,6	104,4	0,0%
Arc15_6_Avg	15	30	22	74	40%	18	**237,57**	**132,2**	**0,0%**

Table A1.7. Results of the 7th Instance Set with 20 Nodes and Full Battery

Instances	# of Nodes	Total # of Nodes	# of Arcs	Total # of Arcs	Required Arc %	# of Required Arcs	Obj.	CPU (sec)	Gap %
Arc20_7_1	20	40	31	102	15%	10	128,9	10,3	0,0%
Arc20_7_2	20	40	31	102	15%	10	128,9	9,3	0,0%

Table A1.7 (cont'd). Results of the 7th Instance Set with 20 Nodes and Full Battery

Arc20_7_3	20	40	31	102	15%	10	128,9	8,8	0,0%
Arc20_7_4	20	40	31	102	15%	10	128,9	10,4	0,0%
Arc20_7_5	20	40	31	102	15%	10	128,9	7,3	0,0%
Arc20_7_6	20	40	31	102	15%	10	128,9	11,0	0,0%
Arc20_7_7	20	40	31	102	15%	10	128,9	7,8	0,0%
Arc20_7_8	20	40	31	102	15%	10	128,9	10,4	0,0%
Arc20_7_9	20	40	31	102	15%	10	128,9	8,7	0,0%
Arc20_7_10	20	40	31	102	15%	10	128,9	9,1	0,0%
Arc20_7_Avg	20	40	31	102	15%	10	**128,94**	**9,3**	**0,0%**

Table A1.8. Results of the 8th Instance Set with 20 Nodes and Full Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc20_8_1	20	40	31	102	20%	13	177,1	39,2	0,0%
Arc20_8_2	20	40	31	102	20%	13	177,1	95,4	0,0%
Arc20_8_3	20	40	31	102	20%	13	177,1	159,7	0,0%
Arc20_8_4	20	40	31	102	20%	13	177,1	97,0	0,0%
Arc20_8_5	20	40	31	102	20%	13	177,1	92,8	0,0%
Arc20_8_6	20	40	31	102	20%	13	177,1	62,2	0,0%
Arc20_8_7	20	40	31	102	20%	13	177,1	219,5	0,0%
Arc20_8_8	20	40	31	102	20%	13	177,1	100,5	0,0%
Arc20_8_9	20	40	31	102	20%	13	177,1	121,4	0,0%
Arc20_8_10	20	40	31	102	20%	13	177,1	260,9	0,0%
Arc20_8_Avg	20	40	31	102	20%	13	**177,06**	**124,9**	**0,0%**

Table A1.9. Results of the 9th Instance Set with 20 Nodes and Full Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc20_9_1	20	40	31	102	25%	16	201,5	197,2	0,0%
Arc20_9_2	20	40	31	102	25%	16	201,5	87,2	0,0%
Arc20_9_3	20	40	31	102	25%	16	201,5	225,0	0,0%
Arc20_9_4	20	40	31	102	25%	16	201,5	85,1	0,0%
Arc20_9_5	20	40	31	102	25%	16	201,5	327,0	0,0%
Arc20_9_6	20	40	31	102	25%	16	201,5	63,2	0,0%

Table A1.9 (cont'd). Results of the 9th Instance Set with 20 Nodes and Full Battery

Arc20_9_7	20	40	31	102	25%	16	201,5	145,0	0,0%
Arc20_9_8	20	40	31	102	25%	16	201,5	155,2	0,0%
Arc20_9_9	20	40	31	102	25%	16	201,5	510,7	0,0%
Arc20_9_10	20	40	31	102	25%	16	201,5	154,1	0,0%
Arc20_9_Avg	20	40	31	102	25%	16	**201,52**	**195,0**	**0,0%**

Table A1.10. Results of the 10th Instance Set with 20 Nodes and Full Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc20_10_1	20	40	31	102	30%	19	215,8	124,0	0,0%
Arc20_10_2	20	40	31	102	30%	19	215,8	66,0	0,0%
Arc20_10_3	20	40	31	102	30%	19	215,8	80,3	0,0%
Arc20_10_4	20	40	31	102	30%	19	215,8	99,3	0,0%
Arc20_10_5	20	40	31	102	30%	19	215,8	84,4	0,0%
Arc20_10_6	20	40	31	102	30%	19	215,8	83,7	0,0%
Arc20_10_7	20	40	31	102	30%	19	215,8	76,9	0,0%
Arc20_10_8	20	40	31	102	30%	19	215,8	78,6	0,0%
Arc20_10_9	20	40	31	102	30%	19	215,8	102,7	0,0%
Arc20_10_10	20	40	31	102	30%	19	215,8	115,9	0,0%
Arc20_10_Avg	20	40	31	102	30%	19	**215,76**	**91,2**	**0,0%**

Table A1.11. Results of the 11th Instance Set with 20 Nodes and Full Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc20_11_1	20	40	31	102	35%	22	254,8	1266,8	0,0%
Arc20_11_2	20	40	31	102	35%	22	254,8	1168,4	0,0%
Arc20_11_3	20	40	31	102	35%	22	254,9	573,2	0,0%
Arc20_11_4	20	40	31	102	35%	22	254,8	136,6	0,0%
Arc20_11_5	20	40	31	102	35%	22	254,8	814,4	0,0%
Arc20_11_6	20	40	31	102	35%	22	255,8	298,5	0,0%
Arc20_11_7	20	40	31	102	35%	22	254,8	1070,4	0,0%
Arc20_11_8	20	40	31	102	35%	22	254,8	417,0	0,0%
Arc20_11_9	20	40	31	102	35%	22	254,8	439,7	0,0%
Arc20_11_10	20	40	31	102	35%	22	254,8	1132,6	0,0%

Table A1.11 (cont'd). Results of the 11th Instance Set with 20 Nodes and Full Battery

Arc20_11_Avg	20	40	31	102	35%	22	**254,91**	**731,8**	**0,0%**

Table A1.12. Results of the 12th Instance Set with 20 Nodes and Full Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc20_12_1	20	40	31	102	40%	25	288,9	1427,0	0,0%
Arc20_12_2	20	40	31	102	40%	25	288	1117,1	0,0%
Arc20_12_3	20	40	31	102	40%	25	288,9	950,3	0,0%
Arc20_12_4	20	40	31	102	40%	25	288,9	1472,1	0,0%
Arc20_12_5	20	40	31	102	40%	25	288,9	1454,2	0,0%
Arc20_12_6	20	40	31	102	40%	25	288,9	1752,0	0,0%
Arc20_12_7	20	40	31	102	40%	25	288,9	816,3	0,0%
Arc20_12_8	20	40	31	102	40%	25	288,9	588,0	0,0%
Arc20_12_9	20	40	31	102	40%	25	288,9	1422,4	0,0%
Arc20_12_10	20	40	31	102	40%	25	288,9	685,0	0,0%
Arc20_12_Avg	20	40	31	102	40%	25	**288,81**	**1168,5**	**0,0%**

Table A1.13. Results of the 13th Instance Set with 25 Nodes and Full Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc25_13_1	25	50	40	130	15%	12	156,7	17,4	0,0%
Arc25_13_2	25	50	40	130	15%	12	156,7	27,5	0,0%
Arc25_13_3	25	50	40	130	15%	12	156,7	30,0	0,0%
Arc25_13_4	25	50	40	130	15%	12	156,7	65,5	0,0%
Arc25_13_5	25	50	40	130	15%	12	156,7	19,9	0,0%
Arc25_13_6	25	50	40	130	15%	12	156,7	19,7	0,0%
Arc25_13_7	25	50	40	130	15%	12	156,7	26,5	0,0%
Arc25_13_8	25	50	40	130	15%	12	156,7	15,3	0,0%
Arc25_13_9	25	50	40	130	15%	12	156,7	19,4	0,0%
Arc25_13_10	25	50	40	130	15%	12	156,7	14,6	0,0%
Arc25_13_Avg	25	50	40	130	15%	12	**156,69**	**25,6**	**0,0%**

Table A1.14. Results of the 14th Instance Set with 25 Nodes and Full Battery

Instances	# of Nodes	Total # of Nodes	# of Arcs	Total # of Arcs	Required Arc %	# of Required Arcs	Obj.	CPU (sec)	Gap %
Arc25_14_1	25	50	40	130	20%	16	194,9	99,5	0,0%
Arc25_14_2	25	50	40	130	20%	16	194,9	146,4	0,0%
Arc25_14_3	25	50	40	130	20%	16	194,9	81,9	0,0%
Arc25_14_4	25	50	40	130	20%	16	194,9	53,8	0,0%
Arc25_14_5	25	50	40	130	20%	16	194,9	70,3	0,0%
Arc25_14_6	25	50	40	130	20%	16	194,9	128,5	0,0%
Arc25_14_7	25	50	40	130	20%	16	194,9	58,1	0,0%
Arc25_14_8	25	50	40	130	20%	16	194,9	89,4	0,0%
Arc25_14_9	25	50	40	130	20%	16	194,9	105,4	0,0%
Arc25_14_10	25	50	40	130	20%	16	194,9	117,2	0,0%
Arc25_14_Avg	25	50	40	130	20%	16	**194,88**	**95,0**	**0,0%**

Table A1.15. Results of the 15th Instance Set with 25 Nodes and Full Battery

Instances	# of Nodes	Total # of Nodes	# of Arcs	Total # of Arcs	Required Arc %	# of Required Arcs	Obj.	CPU (sec)	Gap %
Arc25_15_1	25	50	40	130	25%	20	243,8	214,6	0,0%
Arc25_15_2	25	50	40	130	25%	20	243,8	210,1	0,0%
Arc25_15_3	25	50	40	130	25%	20	243,8	276,0	0,0%
Arc25_15_4	25	50	40	130	25%	20	242,8	181,2	0,0%
Arc25_15_5	25	50	40	130	25%	20	242,8	222,2	0,0%
Arc25_15_6	25	50	40	130	25%	20	242,8	140,1	0,0%
Arc25_15_7	25	50	40	130	25%	20	243,8	347,9	0,0%
Arc25_15_8	25	50	40	130	25%	20	242,8	215,1	0,0%
Arc25_15_9	25	50	40	130	25%	20	242,8	189,1	0,0%
Arc25_15_10	25	50	40	130	25%	20	242,8	80,8	0,0%
Arc25_15_Avg	25	50	40	130	25%	20	**243,23**	**207,7**	**0,0%**

Table A1.16. Results of the 16th Instance Set with 25 Nodes and Full Battery

Instances	# of Nodes	Total # of Nodes	# of Arcs	Total # of Arcs	Required Arc %	# of Required Arcs	Obj.	CPU (sec)	Gap %
Arc25_16_1	25	50	40	130	30%	24	297,6	2566,7	0,0%
Arc25_16_2	25	50	40	130	30%	24	297,6	837,0	0,0%

Table A1.16 (cont'd). Results of the 16th Instance Set with 25 Nodes and Full Battery

Arc25_16_3	25	50	40	130	30%	24	297,6	1099,0	0,0%
Arc25_16_4	25	50	40	130	30%	24	297,6	1266,8	0,0%
Arc25_16_5	25	50	40	130	30%	24	297,6	2109,8	0,0%
Arc25_16_6	25	50	40	130	30%	24	297,6	893,9	0,0%
Arc25_16_7	25	50	40	130	30%	24	297,6	1461,3	0,0%
Arc25_16_8	25	50	40	130	30%	24	297,3	1054,4	0,0%
Arc25_16_9	25	50	40	130	30%	24	297,6	1174,1	0,0%
Arc25_16_10	25	50	40	130	30%	24	297,6	2158,8	0,0%
Arc25_16_Avg	25	50	40	130	30%	24	**297,57**	**1462,2**	**0,0%**

Table A1.17. Results of the 17th Instance Set with 25 Nodes and Full Battery

Instances	# of Nodes	Total # of Nodes	# of Arcs	Total # of Arcs	Required Arc %	# of Required Arcs	Obj.	CPU (sec)	Gap %
Arc25_17_1	25	50	40	130	35%	28	329,5	3376,6	0,0%
Arc25_17_2	25	50	40	130	35%	28	329,5	4174,3	0,0%
Arc25_17_3	25	50	40	130	35%	28	329,5	2265,9	0,0%
Arc25_17_4	25	50	40	130	35%	28	329,5	2742,8	0,0%
Arc25_17_5	25	50	40	130	35%	28	329,5	3108,3	0,0%
Arc25_17_6	25	50	40	130	35%	28	329,5	4967,4	0,0%
Arc25_17_7	25	50	40	130	35%	28	329,5	2243,6	0,0%
Arc25_17_8	25	50	40	130	35%	28	329,5	1643,5	0,0%
Arc25_17_9	25	50	40	130	35%	28	329,5	1949,8	0,0%
Arc25_17_10	25	50	40	130	35%	28	329,5	5310,8	0,0%
Arc25_17_Avg	25	50	40	130	30%	24	**329,50**	**3178,3**	**0,0%**

Table A1.18. Results of the 18th Instance Set with 25 Nodes and Full Battery

Instances	# of Nodes	Total # of Nodes	# of Arcs	Total # of Arcs	Required Arc %	# of Required Arcs	Obj.	CPU (sec)	Gap %
Arc25_18_1	25	50	40	130	40%	32	408	7200,0	9,0%
Arc25_18_2	25	50	40	130	40%	32	404,1	7200,0	8,8%
Arc25_18_3	25	50	40	130	40%	32	405,7	7200,0	8,4%
Arc25_18_4	25	50	40	130	40%	32	408	7200,0	9,4%
Arc25_18_5	25	50	40	130	40%	32	408	7200,0	9,3%
Arc25_18_6	25	50	40	130	40%	32	405,8	7200,0	9,0%

Table A1.18 (cont'd). Results of the 18th Instance Set with 25 Nodes and Full Battery

Arc25_18_7	25	50	40	130	40%	32	406,44	7200,1	8,2%
Arc25_18_8	25	50	40	130	40%	32	407,81	7200,0	9,1%
Arc25_18_9	25	50	40	130	40%	32	402,54	7200,0	7,9%
Arc25_18_10	25	50	40	130	40%	32	402,54	7200,0	8,0%
Arc25_18_Avg	25	50	40	130	40%	32	**405,91**	**7200,0**	**8,7%**

APPENDIX 2 – Computational Results of the Mathematical Model for MBLS

In below tables (Table A2.1 – Table A2.18) the computational results of the mathematical model are given. The instance set names are created according to number of nodes, the sample number, and the replication number respectively while adding after the fixed word “Arc”. For example, “Arc15_1_1” indicates that the instance with 15 nodes, sample 1, and replication 1. The last row of each table shows the results on the average for 10 replications. The model is run for 7200 seconds with CPLEX. The “ * ” symbol in the tables indicates that the cut off point (2 hours) results. “NA” means that in those instances there is no feasible solution is found in 2 hours time limitation.

Table A2.1. Results of the 1st Instance Set with 15 Nodes and Medium Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc15_1_1	15	30	22	74	15%	7	104,9	1,859	0,0%
Arc15_1_2	15	30	22	74	15%	7	104,9	1,922	0,0%
Arc15_1_3	15	30	22	74	15%	7	104,9	2,171	0,0%
Arc15_1_4	15	30	22	74	15%	7	104,9	2,141	0,0%
Arc15_1_5	15	30	22	74	15%	7	104,9	1,579	0,0%
Arc15_1_6	15	30	22	74	15%	7	104,9	2,25	0,0%
Arc15_1_7	15	30	22	74	15%	7	104,9	1,813	0,0%
Arc15_1_8	15	30	22	74	15%	7	104,9	1,875	0,0%
Arc15_1_9	15	30	22	74	15%	7	104,9	1,859	0,0%
Arc15_1_10	15	30	22	74	15%	7	104,9	2,141	0,0%
Arc15_1_Avg	15	30	22	74	15%	7	**104,90**	**1,96**	**0,0%**

Table A2.2. Results of the 2nd Instance Set with 15 Nodes and Medium Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc15_2_1	15	30	22	74	20%	9	142,2	14,329	0,0%
Arc15_2_2	15	30	22	74	20%	9	142,2	4,156	0,0%
Arc15_2_3	15	30	22	74	20%	9	142,2	12,14	0,0%
Arc15_2_4	15	30	22	74	20%	9	142,2	9,671	0,0%
Arc15_2_5	15	30	22	74	20%	9	142,2	4,921	0,0%

Table A2.2 (cont'd). Results of the 2nd Instance Set with 15 Nodes and Medium Battery

Arc15_2_6	15	30	22	74	20%	9	142,2	8,25	0,0%
Arc15_2_7	15	30	22	74	20%	9	142,2	3,438	0,0%
Arc15_2_8	15	30	22	74	20%	9	142,2	5,75	0,0%
Arc15_2_9	15	30	22	74	20%	9	142,2	6,594	0,0%
Arc15_2_10	15	30	22	74	20%	9	142,2	7,422	0,0%
Arc15_2_Avg	15	30	22	74	20%	9	**142,23**	**7,67**	**0,0%**

Table A2.3. Results of the 3rd Instance Set with 15 Nodes and Medium Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc15_3_1	15	30	22	74	25%	11	165,7	12,469	0,0%
Arc15_3_2	15	30	22	74	25%	11	165,7	14,672	0,0%
Arc15_3_3	15	30	22	74	25%	11	165,7	7,469	0,0%
Arc15_3_4	15	30	22	74	25%	11	165,7	16,484	0,0%
Arc15_3_5	15	30	22	74	25%	11	165,7	16,703	0,0%
Arc15_3_6	15	30	22	74	25%	11	165,7	22,375	0,0%
Arc15_3_7	15	30	22	74	25%	11	165,7	12,093	0,0%
Arc15_3_8	15	30	22	74	25%	11	165,7	26,25	0,0%
Arc15_3_9	15	30	22	74	25%	11	165,7	12,266	0,0%
Arc15_3_10	15	30	22	74	25%	11	165,7	22,14	0,0%
Arc15_3_Avg	15	30	22	74	25%	11	**165,70**	**16,29**	**0,0%**

Table A2.4. Results of the 4th Instance Set with 15 Nodes and Medium Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc15_4_1	15	30	22	74	30%	14	184,9	19,156	0,0%
Arc15_4_2	15	30	22	74	30%	14	184,9	18,812	0,0%
Arc15_4_3	15	30	22	74	30%	14	184,9	14,407	0,0%
Arc15_4_4	15	30	22	74	30%	14	184,9	28,11	0,0%
Arc15_4_5	15	30	22	74	30%	14	184,9	17,735	0,0%
Arc15_4_6	15	30	22	74	30%	14	184,9	27,937	0,0%
Arc15_4_7	15	30	22	74	30%	14	184,9	26,672	0,0%
Arc15_4_8	15	30	22	74	30%	14	184,9	20,407	0,0%
Arc15_4_9	15	30	22	74	30%	14	184,9	35,578	0,0%

Table A2.4 (cont'd). Results of the 4th Instance Set with 15 Nodes and Medium Battery

Arc15_4_10	15	30	22	74	30%	14	184,9	22,375	0,0%
Arc15_4_Avg	15	30	22	74	30%	14	**184,93**	**23,12**	**0,0%**

Table A2.5. Results of the 5th Instance Set with 15 Nodes and Medium Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc15_5_1	15	30	22	74	35%	16	214,7	94,765	0,0%
Arc15_5_2	15	30	22	74	35%	16	214,7	33,906	0,0%
Arc15_5_3	15	30	22	74	35%	16	214,7	57	0,0%
Arc15_5_4	15	30	22	74	35%	16	214,7	50,812	0,0%
Arc15_5_5	15	30	22	74	35%	16	214,7	35,75	0,0%
Arc15_5_6	15	30	22	74	35%	16	214,7	29,625	0,0%
Arc15_5_7	15	30	22	74	35%	16	214,7	51,234	0,0%
Arc15_5_8	15	30	22	74	35%	16	214,7	62,329	0,0%
Arc15_5_9	15	30	22	74	35%	16	214,7	46,157	0,0%
Arc15_5_10	15	30	22	74	35%	16	214,7	27,031	0,0%
Arc15_5_Avg	15	30	22	74	35%	16	**214,70**	**48,86**	**0,0%**

Table A2.6. Results of the 6th Instance Set with 15 Nodes and Medium Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc15_6_1	15	30	22	74	40%	18	237,6	153,312	0,0%
Arc15_6_2	15	30	22	74	40%	18	237,6	94,875	0,0%
Arc15_6_3	15	30	22	74	40%	18	237,6	106,312	0,0%
Arc15_6_4	15	30	22	74	40%	18	237,6	86,89	0,0%
Arc15_6_5	15	30	22	74	40%	18	237,6	61,375	0,0%
Arc15_6_6	15	30	22	74	40%	18	237,6	68,609	0,0%
Arc15_6_7	15	30	22	74	40%	18	237,6	121,016	0,0%
Arc15_6_8	15	30	22	74	40%	18	237,6	158,344	0,0%
Arc15_6_9	15	30	22	74	40%	18	237,6	108,015	0,0%
Arc15_6_10	15	30	22	74	40%	18	237,6	60,485	0,0%
Arc15_6_Avg	15	30	22	74	40%	18	**237,57**	**101,92**	**0,0%**

Table A2.7. Results of the 7th Instance Set with 20 Nodes and Medium Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc20_7_1	20	40	31	102	15%	10	128,9	9,766	0,0%
Arc20_7_2	20	40	31	102	15%	10	128,9	10,063	0,0%
Arc20_7_3	20	40	31	102	15%	10	128,9	9,235	0,0%
Arc20_7_4	20	40	31	102	15%	10	128,9	10,172	0,0%
Arc20_7_5	20	40	31	102	15%	10	128,9	11,234	0,0%
Arc20_7_6	20	40	31	102	15%	10	128,9	10,344	0,0%
Arc20_7_7	20	40	31	102	15%	10	128,9	9,344	0,0%
Arc20_7_8	20	40	31	102	15%	10	128,9	11,234	0,0%
Arc20_7_9	20	40	31	102	15%	10	128,9	7,844	0,0%
Arc20_7_10	20	40	31	102	15%	10	128,9	9,375	0,0%
Arc20_7_Avg	20	40	31	102	15%	10	**128,94**	**9,86**	**0,0%**

Table A2.8. Results of the 8th Instance Set with 20 Nodes and Medium Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc20_8_1	20	40	31	102	20%	13	177,1	34,828	0,0%
Arc20_8_2	20	40	31	102	20%	13	177,1	56,032	0,0%
Arc20_8_3	20	40	31	102	20%	13	177,1	40,157	0,0%
Arc20_8_4	20	40	31	102	20%	13	177,1	51,688	0,0%
Arc20_8_5	20	40	31	102	20%	13	177,1	58,109	0,0%
Arc20_8_6	20	40	31	102	20%	13	177,1	58,063	0,0%
Arc20_8_7	20	40	31	102	20%	13	177,1	147,375	0,0%
Arc20_8_8	20	40	31	102	20%	13	177,1	28,016	0,0%
Arc20_8_9	20	40	31	102	20%	13	177,1	27,453	0,0%
Arc20_8_10	20	40	31	102	20%	13	177,1	39,391	0,0%
Arc20_8_Avg	20	40	31	102	20%	13	**177,06**	**54,11**	**0,0%**

Table A2.9. Results of the 9th Instance Set with 20 Nodes and Medium Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc20_9_1	20	40	31	102	25%	16	201,8	323,625	0,0%
Arc20_9_2	20	40	31	102	25%	16	201,8	265,219	0,0%

Table A2.9 (cont'd). Results of the 9th Instance Set with 20 Nodes and Medium Battery

Arc20_9_3	20	40	31	102	25%	16	201,8	338,282	0,0%
Arc20_9_4	20	40	31	102	25%	16	201,8	201,594	0,0%
Arc20_9_5	20	40	31	102	25%	16	201,8	289,125	0,0%
Arc20_9_6	20	40	31	102	25%	16	201,8	396,453	0,0%
Arc20_9_7	20	40	31	102	25%	16	201,8	48,14	0,0%
Arc20_9_8	20	40	31	102	25%	16	201,8	163,359	0,0%
Arc20_9_9	20	40	31	102	25%	16	201,8	338,391	0,0%
Arc20_9_10	20	40	31	102	25%	16	201,8	180,765	0,0%
Arc20_9_Avg	20	40	31	102	25%	16	**201,76**	**254,50**	**0,0%**

Table A2.10. Results of the 10th Instance Set with 20 Nodes and Medium Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc20_10_1	20	40	31	102	30%	19	215,8	78,875	0,0%
Arc20_10_2	20	40	31	102	30%	19	215,8	135,093	0,0%
Arc20_10_3	20	40	31	102	30%	19	215,8	102,75	0,0%
Arc20_10_4	20	40	31	102	30%	19	215,8	115,813	0,0%
Arc20_10_5	20	40	31	102	30%	19	215,8	9,766	0,0%
Arc20_10_6	20	40	31	102	30%	19	215,8	131,328	0,0%
Arc20_10_7	20	40	31	102	30%	19	215,8	143,781	0,0%
Arc20_10_8	20	40	31	102	30%	19	215,8	181,047	0,0%
Arc20_10_9	20	40	31	102	30%	19	215,8	127,906	0,0%
Arc20_10_10	20	40	31	102	30%	19	215,8	88,422	0,0%
Arc20_10_Avg	20	40	31	102	30%	19	**215,76**	**111,48**	**0,0%**

Table A2.11. Results of the 11th Instance Set with 20 Nodes and Medium Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc20_11_1	20	40	31	102	35%	22	254,8	410,391	0,0%
Arc20_11_2	20	40	31	102	35%	22	254,8	467,109	0,0%
Arc20_11_3	20	40	31	102	35%	22	254,9	459,985	0,0%
Arc20_11_4	20	40	31	102	35%	22	254,8	1223,922	0,0%
Arc20_11_5	20	40	31	102	35%	22	254,8	496,703	0,0%
Arc20_11_6	20	40	31	102	35%	22	255,8	488,734	0,0%

Table A2.11 (cont’d). Results of the 11th Instance Set with 20 Nodes and Medium Battery

Arc20_11_7	20	40	31	102	35%	22	254,8	615,234	0,0%
Arc20_11_8	20	40	31	102	35%	22	254,8	895,5	0,0%
Arc20_11_9	20	40	31	102	35%	22	254,8	1405,828	0,0%
Arc20_11_10	20	40	31	102	35%	22	254,8	3163,578	0,0%
Arc20_11_Avg	20	40	31	102	35%	22	**254,91**	**962,70**	**0,0%**

Table A2.12. Results of the 12th Instance Set with 20 Nodes and Medium Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc20_12_1	20	40	31	102	40%	25	292,4	959,766	0,0%
Arc20_12_2	20	40	31	102	40%	25	292,4	1880,453	0,0%
Arc20_12_3	20	40	31	102	40%	25	292,4	2138,422	0,0%
Arc20_12_4	20	40	31	102	40%	25	293	1452,25	0,0%
Arc20_12_5	20	40	31	102	40%	25	292,4	1187,015	0,0%
Arc20_12_6	20	40	31	102	40%	25	292,4	2319,047	0,0%
Arc20_12_7	20	40	31	102	40%	25	292,4	2145,219	0,0%
Arc20_12_8	20	40	31	102	40%	25	292,4	2349,125	0,0%
Arc20_12_9	20	40	31	102	40%	25	292,4	1707,265	0,0%
Arc20_12_10	20	40	31	102	40%	25	292,4	1425,062	0,0%
Arc20_12_Avg	20	40	31	102	40%	25	**292,44**	**1756,36**	**0,0%**

Table A2.13. Results of the 13th Instance Set with 25 Nodes and Medium Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc25_13_1	25	50	40	130	15%	12	156,7	17,672	0,0%
Arc25_13_2	25	50	40	130	15%	12	156,7	29,312	0,0%
Arc25_13_3	25	50	40	130	15%	12	156,7	26,89	0,0%
Arc25_13_4	25	50	40	130	15%	12	156,7	30,594	0,0%
Arc25_13_5	25	50	40	130	15%	12	156,7	17,25	0,0%
Arc25_13_6	25	50	40	130	15%	12	156,7	49,235	0,0%
Arc25_13_7	25	50	40	130	15%	12	156,7	36,547	0,0%
Arc25_13_8	25	50	40	130	15%	12	156,7	20,797	0,0%
Arc25_13_9	25	50	40	130	15%	12	156,7	18,969	0,0%
Arc25_13_10	25	50	40	130	15%	12	156,7	37,094	0,0%

Table A2.13 (cont’d). Results of the 13th Instance Set with 25 Nodes and Medium Battery

Arc25_13_Avg	25	50	40	130	15%	12	**156,69**	**28,44**	**0,0%**

Table A2.14. Results of the 14th Instance Set with 25 Nodes and Medium Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc25_14_1	25	50	40	130	20%	16	198,2	142,313	0,0%
Arc25_14_2	25	50	40	130	20%	16	198,2	371,078	0,0%
Arc25_14_3	25	50	40	130	20%	16	198,2	210,719	0,0%
Arc25_14_4	25	50	40	130	20%	16	198,2	92,719	0,0%
Arc25_14_5	25	50	40	130	20%	16	198,2	196,687	0,0%
Arc25_14_6	25	50	40	130	20%	16	198,2	93,984	0,0%
Arc25_14_7	25	50	40	130	20%	16	198,2	179,609	0,0%
Arc25_14_8	25	50	40	130	20%	16	198,2	101,281	0,0%
Arc25_14_9	25	50	40	130	20%	16	198,2	131,703	0,0%
Arc25_14_10	25	50	40	130	20%	16	198,2	135,125	0,0%
Arc25_14_Avg	25	50	40	130	20%	16	**198,22**	**165,52**	**0,0%**

Table A2.15. Results of the 15th Instance Set with 25 Nodes and Medium Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Arcs**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc25_15_1	25	50	40	130	25%	20	243,8	67,578	0,0%
Arc25_15_2	25	50	40	130	25%	20	243,8	325,562	0,0%
Arc25_15_3	25	50	40	130	25%	20	243,8	186,984	0,0%
Arc25_15_4	25	50	40	130	25%	20	242,8	168,688	0,0%
Arc25_15_5	25	50	40	130	25%	20	242,8	336,703	0,0%
Arc25_15_6	25	50	40	130	25%	20	242,8	141,156	0,0%
Arc25_15_7	25	50	40	130	25%	20	243,8	149,641	0,0%
Arc25_15_8	25	50	40	130	25%	20	242,8	133,156	0,0%
Arc25_15_9	25	50	40	130	25%	20	242,8	242,812	0,0%
Arc25_15_10	25	50	40	130	25%	20	242,8	215,359	0,0%
Arc25_15_Avg	25	50	40	130	25%	20	**243,23**	**196,76**	**0,0%**

Table A2.16. Results of the 16th Instance Set with 25 Nodes and Medium Battery

Instances	# of Nodes	Total # of Nodes	# of Arcs	Total # of Arcs	Required Arc %	# of Required Arcs	Obj.	CPU (sec)	Gap %
Arc25_16_1	25	50	40	130	30%	24	306,1	2766,703	0,0%
Arc25_16_2	25	50	40	130	30%	24	318,7	7200	7,7%
Arc25_16_3	25	50	40	130	30%	24	306,1	4180,734	0,0%
Arc25_16_4	25	50	40	130	30%	24	306,1	4278,891	0,0%
Arc25_16_5	25	50	40	130	30%	24	306,1	4096,219	0,0%
Arc25_16_6	25	50	40	130	30%	24	306,1	3560,391	0,0%
Arc25_16_7	25	50	40	130	30%	24	306,1	4019,015	0,0%
Arc25_16_8	25	50	40	130	30%	24	306,1	4606,453	0,0%
Arc25_16_9	25	50	40	130	30%	24	306,1	5227,969	0,0%
Arc25_16_10	25	50	40	130	30%	24	329	7200,015	9,4%
Arc25_16_Avg	25	50	40	130	30%	24	**309,61**	**4713,64**	**1,7%**

Table A2.17. Results of the 17th Instance Set with 25 Nodes and Medium Battery

Instances	# of Nodes	Total # of Nodes	# of Arcs	Total # of Arcs	Required Arc %	# of Required Arcs	Obj.	CPU (sec)	Gap %
Arc25_17_1	25	50	40	130	35%	28	365,9	7200,02	8,9%
Arc25_17_2	25	50	40	130	35%	28	358,5	7200,03	9,1%
Arc25_17_3	25	50	40	130	35%	28	354,7	7200,03	6,1%
Arc25_17_4	25	50	40	130	35%	28	358,5	7200,03	8,1%
Arc25_17_5	25	50	40	130	35%	28	358,5	7200,02	7,1%
Arc25_17_6	25	50	40	130	35%	28	358,5	7200,03	7,5%
Arc25_17_7	25	50	40	130	35%	28	354,7	7200,02	5,3%
Arc25_17_8	25	50	40	130	35%	28	358,5	7200,02	6,7%
Arc25_17_9	25	50	40	130	35%	28	358,5	7200,05	7,0%
Arc25_17_10	25	50	40	130	35%	28	354,7	7200,02	5,6%
Arc25_17_Avg	25	50	40	130	30%	24	**358,09**	**7200,03**	**7,1%**

Table A2.18. Results of the 18th Instance Set with 25 Nodes and Medium Battery

Instances	# of Nodes	Total # of Nodes	# of Arcs	Total # of Arcs	Required Arc %	# of Required Arcs	Obj.	CPU (sec)	Gap %
Arc25_18_1	25	50	40	130	40%	32	NA	NA	NA
Arc25_18_2	25	50	40	130	40%	32	NA	NA	NA

Table A2.18 (cont'd). Results of the 18th Instance Set with 25 Nodes and Medium Battery

Arc25_18_3	25	50	40	130	40%	32	NA	NA	NA
Arc25_18_4	25	50	40	130	40%	32	NA	NA	NA
Arc25_18_5	25	50	40	130	40%	32	NA	NA	NA
Arc25_18_6	25	50	40	130	40%	32	NA	NA	NA
Arc25_18_7	25	50	40	130	40%	32	423,18	7200,02	12,5%
Arc25_18_8	25	50	40	130	40%	32	433,95	7200,02	14,2%
Arc25_18_9	25	50	40	130	40%	32	NA	NA	NA
Arc25_18_10	25	50	40	130	40%	32	NA	NA	NA
Arc25_18_Avg	25	50	40	130	40%	32	**428,56**	**7200***	**13,4%**

APPENDIX 3 – Computational Results of the Mathematical Model for LBLS

In below tables (Table A3.1 – Table A3.18) the computational results of the mathematical model are given. The instance set names are created according to number of nodes, the sample number, and the replication number respectively while adding after the fixed word “Arc”. For example, “Arc15_1_1” indicates that the instance with 15 nodes, sample 1, and replication 1. The last row of each table shows the results on the average for 10 replications. The model is run for 7200 seconds with CPLEX. The “ * ” symbol in the tables indicates that the cut off point (2 hours) results. “NA” means that in those instances there is no feasible solution is found in 2 hours time limitation.

Table A3.1. Results of the 1st Instance Set with 15 Nodes and Low Battery

Instances	# of Nodes	Total # of Nodes	# of Edges	Total # of Arcs	Required Arc %	# of Required Arcs	Obj.	CPU (sec)	Gap %
Arc15_1_1	15	30	22	74	15%	7	105,37	1,24	0,0%
Arc15_1_2	15	30	22	74	15%	7	105,37	1,31	0,0%
Arc15_1_3	15	30	22	74	15%	7	105,37	1,28	0,0%
Arc15_1_4	15	30	22	74	15%	7	105,37	1,44	0,0%
Arc15_1_5	15	30	22	74	15%	7	105,37	1,23	0,0%
Arc15_1_6	15	30	22	74	15%	7	105,37	1,49	0,0%
Arc15_1_7	15	30	22	74	15%	7	105,37	1,69	0,0%
Arc15_1_8	15	30	22	74	15%	7	105,37	1,27	0,0%
Arc15_1_9	15	30	22	74	15%	7	105,37	1,72	0,0%
Arc15_1_10	15	30	22	74	15%	7	105,37	1,34	0,0%
Arc15_1_Avg	15	30	22	74	15%	7	**105,37**	**1,40**	**0,0%**

Table A3.2. Results of the 2nd Instance Set with 15 Nodes and Low Battery

Instances	# of Nodes	Total # of Nodes	# of Edges	Total # of Arcs	Required Arc %	# of Required Arcs	Obj.	CPU (sec)	Gap %
Arc15_2_1	15	30	22	74	20%	9	155,85	18,16	0,0%
Arc15_2_2	15	30	22	74	20%	9	155,85	20,30	0,0%
Arc15_2_3	15	30	22	74	20%	9	155,85	23,61	0,0%
Arc15_2_4	15	30	22	74	20%	9	155,85	20,98	0,0%
Arc15_2_5	15	30	22	74	20%	9	155,85	18,88	0,0%
Arc15_2_6	15	30	22	74	20%	9	155,85	17,84	0,0%

Table A3.2 (cont'd). Results of the 2nd Instance Set with 15 Nodes and Low Battery

Arc15_2_7	15	30	22	74	20%	9	155,85	34,55	0,0%
Arc15_2_8	15	30	22	74	20%	9	155,85	19,75	0,0%
Arc15_2_9	15	30	22	74	20%	9	155,85	17,81	0,0%
Arc15_2_10	15	30	22	74	20%	9	155,85	15,73	0,0%
Arc15_2_Avg	15	30	22	74	20%	9	**155,85**	**20,76**	**0,0%**

Table A3.3. Results of the 3rd Instance Set with 15 Nodes and Low Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Edges**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc15_3_1	15	30	22	74	25%	11	173,16	15,28	0,0%
Arc15_3_2	15	30	22	74	25%	11	173,16	24,67	0,0%
Arc15_3_3	15	30	22	74	25%	11	173,16	10,97	0,0%
Arc15_3_4	15	30	22	74	25%	11	173,16	11,36	0,0%
Arc15_3_5	15	30	22	74	25%	11	173,16	20,41	0,0%
Arc15_3_6	15	30	22	74	25%	11	173,16	14,75	0,0%
Arc15_3_7	15	30	22	74	25%	11	173,16	25,49	0,0%
Arc15_3_8	15	30	22	74	25%	11	173,16	22,83	0,0%
Arc15_3_9	15	30	22	74	25%	11	173,16	25,33	0,0%
Arc15_3_10	15	30	22	74	25%	11	173,16	27,95	0,0%
Arc15_3_Avg	15	30	22	74	25%	11	**173,16**	**19,90**	**0,0%**

Table A3.4. Results of the 4th Instance Set with 15 Nodes and Low Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Edges**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc15_4_1	15	30	22	74	30%	14	207,85	102,80	0,0%
Arc15_4_2	15	30	22	74	30%	14	207,85	90,48	0,0%
Arc15_4_3	15	30	22	74	30%	14	207,85	118,48	0,0%
Arc15_4_4	15	30	22	74	30%	14	207,85	129,78	0,0%
Arc15_4_5	15	30	22	74	30%	14	207,85	163,48	0,0%
Arc15_4_6	15	30	22	74	30%	14	207,85	240,84	0,0%
Arc15_4_7	15	30	22	74	30%	14	207,85	79,81	0,0%
Arc15_4_8	15	30	22	74	30%	14	207,85	91,72	0,0%
Arc15_4_9	15	30	22	74	30%	14	207,85	119,75	0,0%
Arc15_4_10	15	30	22	74	30%	14	207,85	341,55	0,0%
Arc15_4_Avg	15	30	22	74	30%	14	**207,85**	**147,87**	**0,0%**

Table A3.5. Results of the 5th Instance Set with 15 Nodes and Low Battery

Instances	# of Nodes	Total # of Nodes	# of Edges	Total # of Arcs	Required Arc %	# of Required Arcs	Obj.	CPU (sec)	Gap %
Arc15_5_1	15	30	22	74	35%	16	237,28	576,38	0,0%
Arc15_5_2	15	30	22	74	35%	16	237,28	390,06	0,0%
Arc15_5_3	15	30	22	74	35%	16	237,28	400,28	0,0%
Arc15_5_4	15	30	22	74	35%	16	237,28	266,31	0,0%
Arc15_5_5	15	30	22	74	35%	16	237,28	157,31	0,0%
Arc15_5_6	15	30	22	74	35%	16	237,28	435,28	0,0%
Arc15_5_7	15	30	22	74	35%	16	237,28	448,75	0,0%
Arc15_5_8	15	30	22	74	35%	16	237,28	255,72	0,0%
Arc15_5_9	15	30	22	74	35%	16	237,28	372,77	0,0%
Arc15_5_10	15	30	22	74	35%	16	237,28	181,81	0,0%
Arc15_5_Avg	15	30	22	74	35%	16	**237,28**	**348,47**	**0,0%**

Table A3.6. Results of the 6th Instance Set with 15 Nodes and Low Battery

Instances	# of Nodes	Total # of Nodes	# of Edges	Total # of Arcs	Required Arc %	# of Required Arcs	Obj.	CPU (sec)	Gap %
Arc15_6_1	15	30	22	74	40%	18	247,28	337,45	0,0%
Arc15_6_2	15	30	22	74	40%	18	247,28	314,91	0,0%
Arc15_6_3	15	30	22	74	40%	18	247,28	548,36	0,0%
Arc15_6_4	15	30	22	74	40%	18	247,28	76,16	0,0%
Arc15_6_5	15	30	22	74	40%	18	247,28	133,63	0,0%
Arc15_6_6	15	30	22	74	40%	18	247,28	204,11	0,0%
Arc15_6_7	15	30	22	74	40%	18	247,28	889,70	0,0%
Arc15_6_8	15	30	22	74	40%	18	247,28	96,83	0,0%
Arc15_6_9	15	30	22	74	40%	18	247,28	252,98	0,0%
Arc15_6_10	15	30	22	74	40%	18	247,28	314,64	0,0%
Arc15_6_Avg	15	30	22	74	40%	18	**247,28**	**316,88**	**0,0%**

Table A3.7. Results of the 7th Instance Set with 20 Nodes and Low Battery

Instances	# of Nodes	Total # of Nodes	# of Edges	Total # of Arcs	Required Arc %	# of Required Arcs	Obj.	CPU (sec)	Gap %
Arc20_7_1	20	40	31	102	15%	10	139,52	19,20	0,0%
Arc20_7_2	20	40	31	102	15%	10	139,52	25,41	0,0%
Arc20_7_3	20	40	31	102	15%	10	139,52	22,61	0,0%

Table A3.7 (cont'd). Results of the 7th Instance Set with 20 Nodes and Low Battery

Arc20_7_4	20	40	31	102	15%	10	139,52	35,80	0,0%
Arc20_7_5	20	40	31	102	15%	10	139,52	19,92	0,0%
Arc20_7_6	20	40	31	102	15%	10	139,52	24,45	0,0%
Arc20_7_7	20	40	31	102	15%	10	139,52	22,67	0,0%
Arc20_7_8	20	40	31	102	15%	10	139,52	22,59	0,0%
Arc20_7_9	20	40	31	102	15%	10	139,52	22,52	0,0%
Arc20_7_10	20	40	31	102	15%	10	139,52	28,86	0,0%
Arc20_7_Avg	20	40	31	102	15%	10	**139,52**	**24,40**	**0,0%**

Table A3.8. Results of the 8th Instance Set with 20 Nodes and Low Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Edges**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc20_8_1	20	40	31	102	20%	13	202,36	185,56	0,0%
Arc20_8_2	20	40	31	102	20%	13	202,36	150,39	0,0%
Arc20_8_3	20	40	31	102	20%	13	202,36	191,88	0,0%
Arc20_8_4	20	40	31	102	20%	13	202,36	110,89	0,0%
Arc20_8_5	20	40	31	102	20%	13	202,36	157,49	0,0%
Arc20_8_6	20	40	31	102	20%	13	202,36	298,34	0,0%
Arc20_8_7	20	40	31	102	20%	13	202,36	139,84	0,0%
Arc20_8_8	20	40	31	102	20%	13	202,36	218,89	0,0%
Arc20_8_9	20	40	31	102	20%	13	202,36	142,66	0,0%
Arc20_8_10	20	40	31	102	20%	13	202,36	215,81	0,0%
Arc20_8_Avg	20	40	31	102	20%	13	**202,36**	**181,17**	**0,0%**

Table A3.9. Results of the 9th Instance Set with 20 Nodes and Low Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Edges**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc20_9_1	20	40	31	102	25%	16	236,31	710,70	0,0%
Arc20_9_2	20	40	31	102	25%	16	236,31	571,13	0,0%
Arc20_9_3	20	40	31	102	25%	16	236,31	884,30	0,0%
Arc20_9_4	20	40	31	102	25%	16	236,31	833,41	0,0%
Arc20_9_5	20	40	31	102	25%	16	236,31	708,70	0,0%
Arc20_9_6	20	40	31	102	25%	16	236,31	793,00	0,0%
Arc20_9_7	20	40	31	102	25%	16	236,31	948,70	0,0%
Arc20_9_8	20	40	31	102	25%	16	236,31	1005,63	0,0%

Table A3.9 (cont'd). Results of the 9th Instance Set with 20 Nodes and Low Battery

Arc20_9_9	20	40	31	102	25%	16	236,31	1016,95	0,0%
Arc20_9_10	20	40	31	102	25%	16	236,31	1206,72	0,0%
Arc20_9_Avg	20	40	31	102	25%	16	**236,31**	**867,92**	**0,0%**

Table A3.10. Results of the 10th Instance Set with 20 Nodes and Low Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Edges**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc20_10_1	20	40	31	102	30%	19	259,03	1182,63	0,0%
Arc20_10_2	20	40	31	102	30%	19	259,03	510,77	0,0%
Arc20_10_3	20	40	31	102	30%	19	259,03	1004,41	0,0%
Arc20_10_4	20	40	31	102	30%	19	259,03	1197,45	0,0%
Arc20_10_5	20	40	31	102	30%	19	259,03	961,86	0,0%
Arc20_10_6	20	40	31	102	30%	19	259,03	2824,19	0,0%
Arc20_10_7	20	40	31	102	30%	19	259,03	930,94	0,0%
Arc20_10_8	20	40	31	102	30%	19	259,03	2228,19	0,0%
Arc20_10_9	20	40	31	102	30%	19	259,03	877,28	0,0%
Arc20_10_10	20	40	31	102	30%	19	259,03	932,94	0,0%
Arc20_10_Avg	20	40	31	102	30%	19	**259,03**	**1265,06**	**0,0%**

Table A3.11. Results of the 11th Instance Set with 20 Nodes and Low Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Edges**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc20_11_1	20	40	31	102	35%	22	299,05	7200*	5,9%
Arc20_11_2	20	40	31	102	35%	22	296,47	7200*	4,1%
Arc20_11_3	20	40	31	102	35%	22	296,47	2620,00	0,0%
Arc20_11_4	20	40	31	102	35%	22	296,47	3546,70	0,0%
Arc20_11_5	20	40	31	102	35%	22	296,47	3851,30	0,0%
Arc20_11_6	20	40	31	102	35%	22	296,47	2704,27	0,0%
Arc20_11_7	20	40	31	102	35%	22	296,47	7200*	3,7%
Arc20_11_8	20	40	31	102	35%	22	296,47	7200*	6,8%
Arc20_11_9	20	40	31	102	35%	22	296,47	7200*	5,0%
Arc20_11_10	20	40	31	102	35%	22	296,47	7200*	3,8%
Arc20_11_Avg	20	40	31	102	35%	22	**296,73**	**5592,24**	**2,9%**

Table A3.12. Results of the 12th Instance Set with 20 Nodes and Low Battery

Instances	**# of Nodes**	**Tot. # of Nodes**	**# of Edges**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc20_12_1	20	40	31	102	40%	25	320,2	7200*	1,2%
Arc20_12_2	20	40	31	102	40%	25	320,2	4757,64	0,0%
Arc20_12_3	20	40	31	102	40%	25	320,2	4823,49	0,0%
Arc20_12_4	20	40	31	102	40%	25	320,2	2859,13	0,0%
Arc20_12_5	20	40	31	102	40%	25	320,2	7200*	6,2%
Arc20_12_6	20	40	31	102	40%	25	320,2	7200*	5,9%
Arc20_12_7	20	40	31	102	40%	25	320,2	7200*	7,5%
Arc20_12_8	20	40	31	102	40%	25	320,2	3302,59	0,0%
Arc20_12_9	20	40	31	102	40%	25	320,2	2503,25	0,0%
Arc20_12_10	20	40	31	102	40%	25	327,2	7200*	11,7%
Arc20_12_Avg	20	40	31	102	40%	25	**320,8**	**5424,62**	**3,2%**

Table A3.13. Results of the 13th Instance Set with 25 Nodes and Low Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Edges**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc25_13_1	25	50	40	130	15%	12	161,71	93,14	0,0%
Arc25_13_2	25	50	40	130	15%	12	161,71	53,58	0,0%
Arc25_13_3	25	50	40	130	15%	12	161,71	151,59	0,0%
Arc25_13_4	25	50	40	130	15%	12	161,71	49,09	0,0%
Arc25_13_5	25	50	40	130	15%	12	161,71	175,98	0,0%
Arc25_13_6	25	50	40	130	15%	12	161,71	124,84	0,0%
Arc25_13_7	25	50	40	130	15%	12	161,71	106,02	0,0%
Arc25_13_8	25	50	40	130	15%	12	161,71	187,33	0,0%
Arc25_13_9	25	50	40	130	15%	12	161,71	101,63	0,0%
Arc25_13_10	25	50	40	130	15%	12	161,71	180,83	0,0%
Arc25_13_Avg	25	50	40	130	15%	12	**161,71**	**122,40**	**0,0%**

Table A3.14. Results of the 14th Instance Set with 25 Nodes and Low Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Edges**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc25_14_1	25	50	40	130	20%	16	246,19	7200*	11,2%
Arc25_14_2	25	50	40	130	20%	16	250,04	7200*	9,4%
Arc25_14_3	25	50	40	130	20%	16	NA	NA	NA

Table A3.14 (cont'd). Results of the 14th Instance Set with 25 Nodes and Low Battery

Arc25_14_4	25	50	40	130	20%	16	246,19	7200*	11,9%
Arc25_14_5	25	50	40	130	20%	16	264,21	7200*	17,2%
Arc25_14_6	25	50	40	130	20%	16	248,01	7200*	9,0%
Arc25_14_7	25	50	40	130	20%	16	249,31	7200*	12,5%
Arc25_14_8	25	50	40	130	20%	16	NA	NA	NA
Arc25_14_9	25	50	40	130	20%	16	260,28	7200*	16,9%
Arc25_14_10	25	50	40	130	20%	16	249,31	7200*	13,2%
Arc25_14_Avg	25	50	40	130	20%	16	**251,69**	**7200***	**12,6%**

Table A3.15. Results of the 15th Instance Set with 25 Nodes and Low Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Edges**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc25_15_1	25	50	40	130	25%	20	NA	NA	NA
Arc25_15_2	25	50	40	130	25%	20	NA	NA	NA
Arc25_15_3	25	50	40	130	25%	20	NA	NA	NA
Arc25_15_4	25	50	40	130	25%	20	NA	NA	NA
Arc25_15_5	25	50	40	130	25%	20	326,81	7200*	18,3%
Arc25_15_6	25	50	40	130	25%	20	NA	NA	NA
Arc25_15_7	25	50	40	130	25%	20	NA	NA	NA
Arc25_15_8	25	50	40	130	25%	20	NA	NA	NA
Arc25_15_9	25	50	40	130	25%	20	NA	NA	NA
Arc25_15_10	25	50	40	130	25%	20	NA	NA	NA
Arc25_15_Avg	25	50	40	130	25%	20	**326,81**	**7200***	**18,3%**

Table A3.16. Results of the 16th Instance Set with 25 Nodes and Low Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Edges**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc25_16_1	25	50	40	130	30%	24	NA	NA	NA
Arc25_16_2	25	50	40	130	30%	24	NA	NA	NA
Arc25_16_3	25	50	40	130	30%	24	NA	NA	NA
Arc25_16_4	25	50	40	130	30%	24	NA	NA	NA
Arc25_16_5	25	50	40	130	30%	24	NA	NA	NA
Arc25_16_6	25	50	40	130	30%	24	NA	NA	NA
Arc25_16_7	25	50	40	130	30%	24	NA	NA	NA

Table A3.16 (cont'd). Results of the 16th Instance Set with 25 Nodes and Low Battery

Arc25_16_8	25	50	40	130	30%	24	NA	NA	NA
Arc25_16_9	25	50	40	130	30%	24	NA	NA	NA
Arc25_16_10	25	50	40	130	30%	24	NA	NA	NA
Arc25_16_Avg	25	50	40	130	30%	24	**NA**	**NA**	**NA**

Table A3.17. Results of the 17th Instance Set with 25 Nodes and Low Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Edges**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc25_17_1	25	50	40	130	35%	28	NA	NA	NA
Arc25_17_2	25	50	40	130	35%	28	NA	NA	NA
Arc25_17_3	25	50	40	130	35%	28	NA	NA	NA
Arc25_17_4	25	50	40	130	35%	28	NA	NA	NA
Arc25_17_5	25	50	40	130	35%	28	NA	NA	NA
Arc25_17_6	25	50	40	130	35%	28	NA	NA	NA
Arc25_17_7	25	50	40	130	35%	28	NA	NA	NA
Arc25_17_8	25	50	40	130	35%	28	NA	NA	NA
Arc25_17_9	25	50	40	130	35%	28	NA	NA	NA
Arc25_17_10	25	50	40	130	35%	28	NA	NA	NA
Arc25_17_Avg	25	50	40	130	30%	24	**NA**	**NA**	**NA**

Table A3.18. Results of the 18th Instance Set with 25 Nodes and Low Battery

Instances	**# of Nodes**	**Total # of Nodes**	**# of Edges**	**Total # of Arcs**	**Required Arc %**	**# of Required Arcs**	**Obj.**	**CPU (sec)**	**Gap %**
Arc25_18_1	25	50	40	130	40%	32	NA	NA	NA
Arc25_18_2	25	50	40	130	40%	32	NA	NA	NA
Arc25_18_3	25	50	40	130	40%	32	NA	NA	NA
Arc25_18_4	25	50	40	130	40%	32	NA	NA	NA
Arc25_18_5	25	50	40	130	40%	32	NA	NA	NA
Arc25_18_6	25	50	40	130	40%	32	NA	NA	NA
Arc25_18_7	25	50	40	130	40%	32	NA	NA	NA
Arc25_18_8	25	50	40	130	40%	32	NA	NA	NA
Arc25_18_9	25	50	40	130	40%	32	NA	NA	NA
Arc25_18_10	25	50	40	130	40%	32	NA	NA	NA
Arc25_18_Avg	25	50	40	130	40%	32	**NA**	**NA**	**NA**