Logistics Planning for Restoration of Network Connectivity After a Disaster

by

Ayşe Nur KİBAR

A Thesis Submitted to the
Graduate School of Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Industrial Engineering

Koç University

September, 2013

Koç University
Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Ayşe Nur KİBAR

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

F. Sibel Salman, Ph. D. (Advisor)

Onur Kaya, Ph. D.

Serpil Sayın, Ph. D.

Date:

To my family...

ABSTRACT

In a natural disaster; e.g., earthquake, flood, landslide, road networks can be damaged, and consequently some parts of the roads may be blocked by building or lamppost debris or may collapse due to ground liquefaction. As a result, the road network may become disconnected. In order to ensure connectivity, necessary actions should be taken after a disaster.

In the immediate disaster response phase, to facilitate emergency transportation, a critical subset of the blocked roads should be cleared to restore network connectivity as soon as possible. A fleet of machinery with required personnel and equipment which are initially positioned at various locations, e.g. depots, should be dispatched for this task. We define an optimization problem to generate a coordinated work schedule. We represent the road network by a directed graph with estimated arc traversal times for work vehicles. We assume that time to open each blocked arc will be estimated after the disaster by gathering information on road conditions. The problem is to determine which blocked arcs to open and a walk for each vehicle starting at its depot that collectively cover the selected blocked arcs. The objective is to minimize the total time of the longest walk, i.e. the makespan. We name this problem that combines network design, scheduling and arc routing aspects, *Arc Routing for Connectivity Problem with K Vehicles (K-ARCP)*. We prove that *K-ARCP* is NP-hard even when a single vehicle exists. We characterize some properties of feasible and optimal solutions. We formulate a mixed integer program using flow variables for the walks. We define two relaxations to obtain lower bounds and test their performance computationally. We generate instances using data of Istanbul highway network and analyze the effects of the number of disconnected components, the number of vehicles and depot locations on solution time and quality.

ÖZETÇE

Sel, deprem gibi doğal afetlerin etkisiyle karayolları hasara uğrayabilir ve zemin sıvılaşmasına bağlı olarak çökme ya da bina ve elektrik/lamba direklerinin enkazı yığılması sonucu bazı yollar ulaşıma kapanabilir. Sonuç olarak, bazı bölgelere ulaşım engellenebilir. Yol ağı bağlanılırlığının sağlanması için gerekli çalışmalar yerine getirilmelidir. Bu çalışmalar acil tıbbi yardım ve ulaşımı kolaylaştırmak adına büyük önem taşır.

Afet sonrasında en kısa zamanda ulaşımı hızlandırmak ve ulaşım ağında bağlantının yeniden sağlanması için, kritik olan yolların açılması gerekmektedir. Yeterli personel ve teçhizata sahip, çeşitli lokasyonlarda konuşlandırılmış araçlar bu iş için kullanılmalıdır. Bu çalışmada koordineli bir iş çizelgesi oluşturma amacıyla bir eniyileme problemi tanımlanmıştır. Yol ağı, yönlü bir çizge üzerinde, araçlara ait seyir süreleri ile birlikte temsil edilmiştir. Kapalı olan yolları açmak için gereken sürenin, afet sonrasında edinilecek yol durumu bilgisine göre tespit edileceği varsayılmıştır. Problem, hangi kapalı yolların açılacağının ve her araç için bir dağıtım noktasından (depo, acil yardım merkezinden vs.) başlayan, açılacak olan yollardan geçen turların belirlenmesidir. Amaç, en uzun tur uzunluğunu enküçüklemektir. Ayrıt rotalama, zaman çizelgeleme ve ağ tasarımı öğeleri içeren bu probleme *K Araçlı Bağlanırlık Amaçlayan Ayrıt Rotalama Problemi* adı verilmiştir ve tek araçlı durumda bile polinom zamanda çözümünün zor olduğu ispat edilmiştir. Olası ve en iyi çözümlerin özellikleri incelenmiştir. Akış karar değişkenleri kullanılarak bir karışık tamsayı programlama formulasyonu verilmiş ve iki gevşetme modelinden alt sınırlar elde edilmiştir. İstanbul karayolları ağı kullanılarak veri oluşturulmuş ve bileşen, araç ve depo sayısının çözüm süresi ve kalitesi üzerindeki etkileri analiz edilmiştir.

ACKNOWLEDGMENTS

First, I would like to thank my advisor F. Sibel Salman. I would also like to thank my thesis committee members for critical analysis of this thesis and giving great feedback on the thesis.

Thanks to TÜBİTAK (The Scientific and Technological Research Council of Turkey) for their generous financial support during my M.Sc. process.

Many people have made my graduate experience wonderful. I am grateful to my roomie Hayriye Özgül Özkan for being a friend and a family to me. I would also like to thank Buse Aras, Zehra Önen, Zeynep Göksel Özsarp, Aslı Pınar Yapıcı, İdil Arşık, Müge Tekin, Özge Tüncel, İlknur Eruçar, Abdullah Bilgin, Can Bekar, Erdal Mutlu, Hassan Salehe Matar, İsmail Kuru, Ömer Subaşi and all my friends at Koç University.

Thanks to all my family especially my parents for their support and encouragement. I am very lucky to be a part of such a wonderful family. Finally, thanks to my husband for his patience and support during my research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

TSP	:	Traveling Salesman Problem
ARP	:	Arc Routing Problem
RPP	:	Rural Postman Problem
K-RPP	:	Rural Postman Problem with K vehicles
$ARCP$	:	Arc Routing for Connectivity Problem
K-$ARCP$	:	Arc Routing for Connectivity Problem with K Vehicles
MIP	:	Mixed integer programming
k	:	Index of the vehicles
K	:	Number of vehicles
$G = (V, A)$	:	Directed graph
V	:	Set of vertices
i, j, g, h	:	Indices of the vertices
n	:	Number of vertices
$n + 1$	:	Index of the dummy sink node
A	:	Set of arcs
B	:	Set of blocked arcs
R	:	Set of required arcs
c_{ij}	:	Traversal time on arc (i, j)
b_{ij}	:	Additional time of unblocking arc (i, j)
D	:	Set of depot nodes
d	:	Index of the depots, $(d \subseteq V)$
$d(k)$	:	Index of the depot at which vehicle k is initially positioned
P_d	:	Set of vehicles which are initially positioned at depot d

Q	:	Set of disconnected components of $G_B = (V, A \setminus B)$
q	:	Index of the components
V_q	:	Set of vertices in component $q \in Q$ $(V_q \subseteq V)$
A_q	:	Set of arcs which are not blocked in component $q \in Q$, $A_q \subseteq A \backslash B$
$q(d)$	:	Index of the component at which depot d is located
Q_D	:	Set of components which include a depot
Q_N	:	Set of components which do not include a depot
$q(k)$	:	Index of the component at which vehicle k is initially positioned
F_q	:	Set of vehicles (fleet) which are initially positioned in component q
$G_q = (V_q, A_q)$	:	Graph of component q where $\bigcup_{q \in Q} V_q = V$, $A \setminus B = \bigcup_{q \in Q} A_q$
$\delta^+(q)$	:	Set of outgoing arcs in the cutset of component q
$\delta^-(q)$	:	Set of incoming arcs in the cutset of component q
C	:	Union of cutset arcs of V_q for all $q \in Q$, $C : \delta^+(q) \cup \delta^-(q)$
S	:	Set of all subsets of components within which the nodes are strongly connected
s	:	Index for elements of S
Y^+	:	Set of all subsets of components which require at least one outgoing arc but no incoming arc to be unblocked in order to be strongly connected to the remaining graph
Y^-	:	Set of all subsets of components which require at least one incoming arc but no outgoing arc to be unblocked in order to be strongly connected to the remaining graph
y	:	Index for elements of $Y^+ \cup Y^-$
p	:	Index that shows how many times an arc is traversed, $p = 1, ..., P$
M	:	A nonnegative scalar with large enough value

Chapter 1

INTRODUCTION AND PROBLEM DEFINITION

Disaster management involves taking actions before and after a disaster to minimize its destructive effects. After a disaster, it is critical to reach affected areas to provide relief operations; such as search and rescue, medical services, aid delivery and establishing temporary shelter. This study focuses on ensuring functionality of transportation networks such as highway and road networks which can be affected in a disaster; e.g., earthquake, flood, landslide. Some parts of the roads may be blocked by building, lamppost and car debris, or may be damaged due to ground liquefaction or collapse of bridges and viaducts. Damage to other infrastructure networks such as natural gas or drainage systems may also cause dysfunctionality in the roads. As a result, parts of the network may become unreachable. This will in turn disrupt emergency response and relief activities. Crippling of transportation networks has been observed in many natural disasters. For instance, in the Kobe Earthquake (1995), the Triple Disaster (2011) in Japan, the Indonesian Tsunami and Earthquake in 2004 and the Haiti Earthquake in 2010. the Kocaeli Earthquake (1999) and the Van Earthquake (2011) are two of the recent devastating natural disasters which Turkey has faced. In the Kocaeli Earthquake, some roads collapsed and were blocked by building debris. After such a situation, to ensure connectivity of the network and to provide accessibility between people in different parts of the network as fast as possible, it is crucial to unblock the roads by a fleet of machinery or vehicles, and this should be done rapidly and efficiently. Some people will want to evacuate the disaster location, while others will be coming in and for this reason strong connectivity of the network is required. Depending on the damage, some of the clearing tasks take less time; e.g. hours, while others may take a long time; e.g. days. After a disaster, the first 72 hour-time period is critical especially for response activities and ensuring connectivity in this time frame is an important operation.

Recently, several studies focused on upgrading the road network or improving accessibility after a disaster situation. However, to the best of our knowledge, the restoration of the roads after a disaster by a fleet of vehicles in order to ensure strong connectivity of a network with routing perspective has not been addressed in the Operations Research literature. We define a new network optimization problem to address road restoration with the objective of providing strong connectivity in a disconnected network in the shortest time. This problem combines arc routing, network design and scheduling concepts. Every vehicle has a walk which traverses several arcs; therefore, arc routing is one element in this study. Since connectivity of the graph is an objective, network design is another element. When more than one vehicle have to traverse a blocked arc, one vehicle unblocks it, and others have to wait until it is unblocked. Therefore, for an arc, traversal times of vehicles have to be considered and vehicles should be scheduled for their clearing tasks to minimize the makespan of the operations. Therefore, scheduling is another element which exists in this problem.

1.1 Problem Definition

Before we define our problem, some definitions may be useful to make the problem clear. A *connected graph* contains a directed path from a node i to another node j or a directed path from j to i for every pair of nodes i and j. Otherwise, the graph is disconnected. A graph is *strongly connected* if it contains a directed path from i to j and a directed path from j to i for every pair of nodes i and j. Otherwise, the graph is disconnected in the strong sense. We define *Arc Routing for Connectivity Problem with K Vehicles (K-ARCP)* on a directed, strongly connected and simple graph $G = (V, A)$ with nonnegative arc costs. After a natural disaster, roads may be difficult to traverse if they are damaged, which affects the speed of transportation that is highly dependent on road and extraordinary traffic conditions (as also stated in [27]). Therefore, costs are calculated in terms of time instead of distance. Traversal time on an unblocked (i.e. not blocked initially) or a blocked arc after it has been unblocked is equal to c_{ij}, where (i, j) represents the arc. There are K vehicles where $K > 1$. Each vehicle k is located initially at a node $d(k)$; e.g., its depot or an emergency response facility. Moreover, a subset B of arcs, which are determined to be blocked according to post-disaster information on road conditions, are given such that $G_B = (V, A \backslash B)$ is disconnected. The set B consists of all blocked arcs and the set R, a subset of B, represents the arcs which will be traversed and cleared by a vehicle in order to restore strong connectivity of the graph.

The set R is not known in advance. The solution identifies R and constructs a walk for each vehicle that starts at its depot. We want the walks in the solution to cover R collectively. In other words, the arcs in the set $A \backslash B \cup R$ induce a connected graph, G_R, on the set V. We assume that there are $|Q|$ disconnected components in G_B where Q is the set of disconnected components, in the strong sense. Each component in Q consists of strongly connected nodes. We partition Q into three classes: (i) components within which the nodes are strongly connected and which require at least one incoming and one outgoing arc in order to be strongly connected to the remaining graph, (ii) components which require at least one outgoing but no incoming arc to be unblocked in order to be strongly connected to the remaining graph, and (iii) components which require at least one incoming but no outgoing arc to be unblocked in order to be strongly connected to the remaining network. Figure 1.1 shows these classes on an example. Moreover, unblocking; i.e., passing through a blocked arc for the first time results in a longer time than the traversal time on an unblocked arc or the traversal time on a blocked arc after it has been unblocked. More formally, we define the additional time of unblocking arc (i, j) as b_{ij} where $b_{ij} \geq 0$. In a walk, c_{ij} time units elapse each time an arc is traversed, and in addition, b_{ij} units elapse once for each blocked arc which is unblocked during the walk. In other words, a blocked arc is unblocked by a vehicle in its first traversal of that arc. We assume that traffic cannot flow in both directions after a blocked road is unblocked in one direction by a vehicle. Considering that allowing traffic in the reverse direction would slow down response activities. This is a reasonable assumption.

If a vehicle arrives at a node incident to a blocked arc, it may have to wait for another vehicle to unblock that arc. This situation arises when more than one vehicle arrive at the same blocked arc within close times. In such a case, waiting time is added to total travel time of the vehicle which waits at an arc. We assume that on every blocked road, a single vehicle can operate for unblocking purpose. The case in which multiple vehicles operate on a single road at the same time can be a more difficult version of our problem and it can be investigated as a future work. The objective is to minimize the time which the graph becomes strongly connected. That is, by definition, there must be a path from each vertex to every other vertex in the network. In order to connect all the disconnected components, at least two arcs in opposite directions within the cutset of a component must be unblocked. Otherwise, the network cannot be strongly connected. If a vehicle leaves its depot, it unblocks at least one blocked arc. Since we are interested in minimizing the time when the graph becomes connected, return of the vehicles to their depots is not considered. Therefore, the walks are open.

Since there are multiple vehicles, and a separate walk for each vehicle, the aim is to minimize the travel time of the longest walk. In this way, the time at which the network becomes strongly connected is minimized and a more balanced set of walks are obtained in terms of travel time. This objective is meaningful especially for post-disaster situations [2], and it also provides a more balanced set of walks in terms of the number of roads which are unblocked by vehicles. We can define the objective function as min $\max_{k=1,\ldots,K}$ $c(W_k) + b(W_k) + w(W_k)$, where W_k is walk of the k^{th} vehicle; $c(W_k)$ is traversal time of the k^{th} vehicle and $c(W_k)$ is calculated by summing up the traversal time of arcs (in terms of c_{ij}) that are traversed by vehicle k; $b(W_k)$ is the total additional time (in terms of b_{ij}) of unblocking for vehicle k; $w(W_k)$ is total waiting time of the k^{th} vehicle at the blocked arcs. The waiting times can be possibly zero.

The motivation behind this thesis is to develop a solution method to the connectivity problem of the network that generates a solution in short time. Our goals are to formulate *K-ARCP* and to observe for which cases it can be solved in a reasonable time. Moreover, we provide an analysis of a case study of Istanbul network to generate some insights for preparedness.

The organization of this thesis is as follows. Chapter 2 reviews relevant studies in the literature. Chapter 3 gives computational complexity proof of the *K-ARCP*. In Chapter 4, Mixed Integer Programming (MIP) models for *K-ARCP* are given. Chapter 5 presents the data related to Istanbul highway network. In Chapter 6, computational results are provided. Finally, in Chapter 7, we conclude the thesis with a summary, some comments and directions for future research.

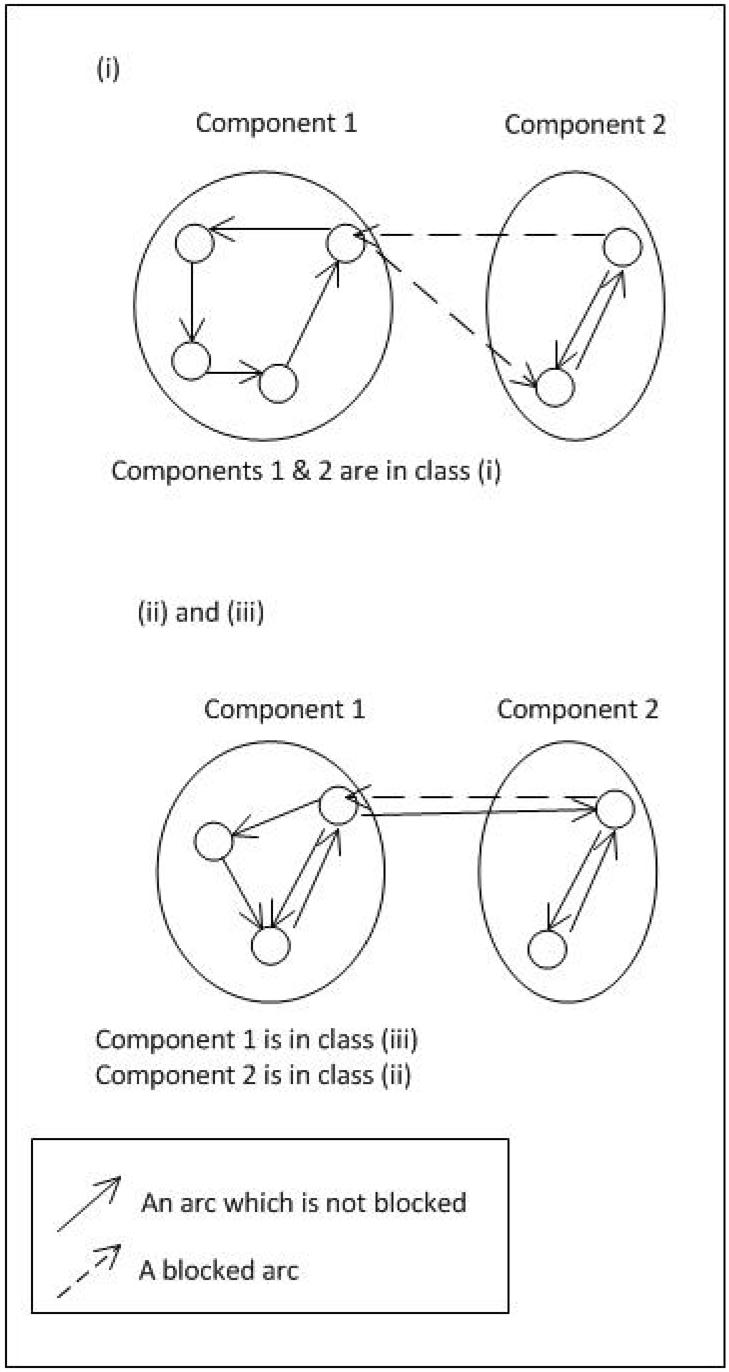

Figure 1.1: Classification of Components

Chapter 2

LITERATURE REVIEW

Arc routing and vehicle routing problems have attracted the interest of researchers and have many application areas such as network design, transportation, snow plowing. The problem addressed in this thesis falls into the class of arc routing problems. The main motivation of this section is to give an overview of arc routing problems and to introduce some problems which are related to *K-ARCP*. First, problems which are in the class of arc routing problems are presented. Next, some problems which focus on road upgrading in disaster and non-disaster cases are introduced. Finally, some related studies with their differences and similarities to this thesis are discussed.

2.1 Related Work: Arc Routing Problems

Chinese Postman Problem (CPP) is an arc routing problem which is defined as finding the minimum cost closed walk on all arcs of a given directed graph. This problem is *NP*-hard, but there are some special cases which can be solved by a polynomial time algorithm. At this point, some definitions may help to make these problems clear. A mixed graph has both directed and undirected links between nodes. A windy graph is defined on an undirected graph but the edge costs differ depending on the direction of traversal through an edge. An even graph is a graph with vertex degrees all even. If *CPP* is defined on a totally undirected, totally directed, mixed but even [15], windy but *Eulerian* networks [38], then it can be solved in polynomial time. However, if the graph is mixed, then *CPP* is *NP*-hard [28]. There exists a $\frac{4}{3}$ - approximation algorithm for *mixed CPP* [25]. There is another version of *CPP*, called the *Windy Postman Problem (WPP)*,first introduced by Minieka [26]. In this problem, there is an undirected graph $G = (V, E)$ with asymmetric nonnegative edge costs and the aim is to find the minimum traversal cost of all edges in the graph. This is an *NP*-hard problem [7]. There exists a $\frac{3}{2}$- approximation algorithm for *WPP* [34].

Hierarchical CPP is a special version of *CPP* for which the arcs are partitioned into clusters, and a precedence relation is defined on the clusters. The aim is to find the minimum tour length such that the higher prioritized arcs are traversed before the lower prioritized arcs and each arc in the network is traversed exactly once. This problem is *NP*-hard [13]. Therefore, several heuristic methods are proposed [3, 23].

Another class of arc routing problems is the *Rural Postman Problem (RPP)* which is a more general version of *CPP*. In this problem, all the arcs do not have to be traversed. There is a subset of arcs which are required to be traversed at least once. This problem is also *NP*-hard on an undirected or directed graph [24]. If the arc costs satisfy the triangle inequality, there exists a $\frac{3}{2}$ - approximation algorithm [16]. Some heuristic methods are applied on *RPP* [18, 20, 21].

Another type of routing problem is the *Stacker Crane Problem (SCP)*. It is defined on a mixed graph, and the aim is to determine the shortest circuit including each arc in the graph *at least once*. This problem is *NP*-hard and there exists a $\frac{9}{5}$ - approximation algorithm [17].

In the *Capacitated Arc Routing Problem (CARP)*, each arc has a nonnegative weight, and all arcs must be traversed exactly once in order to fulfill the demand of arcs by a fleet of identical capacitated vehicles based at the depot node. The aim is to minimize the total traversal time. *CARP* is *NP*-hard [19]. The *Capacitated Chinese Postman Problem (CCPP)* is a *CARP* for which all of the arcs in the graph have to be traversed by capacitated vehicles exactly once. There exists a $(\frac{3}{2} - \epsilon)$ - approximation algorithm [19]. If the arcs have the same demand values (which is equal to 1) on a linear graph and vehicles have the same capacity, this cost minimization problem can be solved to optimality by a polynomial-time algorithm [5, 8]. Moreover, if the arc demands are identical and the graph is cyclic, then the problem can be solved in $O(|V|)$ time, where V is the node set of the input graph [5].

If there is a fleet of identical vehicles, say K vehicles, then the problem of finding K tours such that all the edges are covered in a graph with minimum total cost is called *K-CPP*. If *K-CPP* is defined on totally undirected or totally directed networks, then there exist polynomial-time algorithms for these problems [5]. On mixed but even graphs *K-CPP* still can be solved in polynomial time [29]. However, if the input graph is mixed or windy, then *K-CPP* for those specifications is *NP*-hard [29]. An overview of these problems can be seen in Table 2.1.

Arc routing problems in which the timing of the arc traversals is of interest have been recently studied in the literature and there are few of them. Similar to our study, arrival time at an arc is considered in [36] as well. In [36], different types of cost functions exist depending on the purpose of traversing an arc: service time and traversal time. Service time is a linear function of time. Different than our study, multiple closed walks are considered, and there is a given required set of arcs which have to be traversed. The objective is to minimize the sum of the service and the traversal time.

Objective type of our problem is Min-Max. Therefore, it is useful to analyze problems which have the same objective type. Min-Max versions of arc routing problems are used widely for real life applications. Minimizing the longest tour length when there are K-vehicles or K-tours traversing edges of a graph is crucial for some real life situations such as responding to earthquakes and floods. *Min-Max K-CPP* on an undirected graph is NP-hard [17], and there exists a $(2 - \frac{1}{K})$ - approximation algorithm [17]. *Min-Max K-CPP* and *Min-Max K-WRPP* have been studied recently [1, 6]. For the *Min-Max K-CPP* on an undirected graph, a branch-and-cut algorithm [1] and a tabu search method are developed [2]. For the *Min-Max K-RPP* on an undirected graph, there exists a 7-approximation algorithm [4]. However, the problem in [4] is a different version of *Min-Max K-RPP* in which the routes do not start in a specified node but just any node.

2.2 Related Work: Upgrading and Road Maintenance Problems

In the disaster context, recently, there have been studies on upgrading the road network or improving accessibility after a disaster. Some of them do not have routing perspective. Instead, they focus on selection of road segments which are to be upgraded or repaired. One such study is by Duque and Sörensen [14]. They investigate the case where there is a budget constraint, and there are a number of non-operative roads which need to be repaired after a disaster situation. Moreover, they assign weights to the rural towns depending on the importance of the towns. Their objective is to minimize the weighted sum of time to travel from each rural town to its closest regional center [14]. There is no routing decision, but finding the roads to be repaired in order to have shortest paths between node pairs. Another study is by Campbell et al. [9], which focuses on determining the number of edges to be upgraded before a catastrophe while minimizing the maximum travel time between any source-terminal/origin-destination (s-t) pair. They use heuristic methods to solve the problem.

Yan and Shih [39] address the scheduling of roadway repair operations in an emergency situation. In their problem some road segments cannot be repaired after a specific time point due to the importance of different repair points. There may be a time limit for repairing some repair points; therefore those have to be scheduled to be repaired before the other repair points. Time windows should be imposed. Moreover, there are supply points from which relief distribution is started. The areas, which are defined as demand points, have to receive a minimum required amount of relief items. Relief distribution cannot continue through an arc if that arc is not repaired before. The objectives are to minimize the time needed for emergency repair at all repair points and for sending relief supplies to all demand points. They define a time-space network. A multi-objective, multi-commodity network flow model is constructed, and a heuristic method is developed.

Vishwanath and Peeta [37] study a network design problem which is named as *Multicommodity Maximal Covering Network Design Problem*. They focus on selecting critical routes to retrofit in preparation for earthquake response, defining a two-objective integer programming model. Given a budget and disconnected *O-D* pairs, the problem is to select edges to retrofit to minimize total travel time between *O-D* pairs over selected routes and to maximize total population covered by the routes. An edge can be used in a route only if it is retrofitted. Cost of retrofitting cannot exceed the budget. They solve this model with branch-and-cut module of CPLEX on a case study.

Snow removal or disposal is another research area which includes arc routing and road maintenance. There is a four part survey by Perrier et al. [30, 31, 32, 33] which presents different types of problems related to winter road maintenance. The last two parts of this survey give solution methods on vehicle routing, depot location for spreading and snow plowing operations and fleet sizing problems. Input graphs in winter road maintenance problems can be undirected, directed or mixed, and all the road segments must be serviced. The service involves snow removal or salting the roads. Different types of cost are assigned to the roads: servicing, deadheading a serviced road, deadheading a non-serviced road. The objective is minimizing the total length traveled.

There are also studies which are not in disaster context but focus on road upgrading. In [35], a rural road network design problem is presented for a disconnected graph where the objective is to maximize route efficiency and to maximize road connectivity in all seasons among villages at a region while allocating a fixed budget among a number of possible road projects. They optimize the weighted combination of traffic flow volume served by upgraded roads and weighted traveled distance associated with those traffic volumes. This problem is defined on an undirected graph, and the roads to be upgraded are not selected in advance. It finds the shortest paths between villages and ensures connectivity among them. A metaheuristic method is applied in order to solve the problem in an efficient time. Another study on upgrading arcs is [12]. It focuses on a version of the Minimum Cost Flow Problem which can also be used in telecommunication networks. A directed graph with arc costs and capacities is given. Their aim is: Given an upgrade budget and an amount of data that should be sent from supply points to demand points, find an optimum upgrading strategy and optimum routing of the flow such that flow cost on the upgraded network is as small as possible. Upgrading strategy, which is mentioned here, is to lower the cost of flow on arcs by investing on them. They propose a polynomial time approximation algorithm for this upgrading problem.

2.3 Comparison of K-ARCP with the Literature

Our problem, *K-ARCP*, differs from the problems explained above in several ways. In *K-ARCP* "connectivity" is the main concern. Most of the other studies do not aim to ensure strong connectivity of the network. It is similar to *Min-Max K-RPP*, but in our study, the set of required arcs are not known in advance and there is no requirement for the walks to be closed. *Min-Max K-WRPP* is similar to *K-ARCP* in terms of objective type. Both problems have Min-Max type of objective. However, in *K-ARCP* the graph is not windy, it is directed. Moreover, in *K-ARCP*, after one traversal of a blocked arc, the cost changes. In the *Min-Max K-WRPP*, the graph is undirected, and costs of an edge in both directions are not the same. In [14], similar to this study, accessibility of a network after a disaster situation is aimed. However they minimize the weighted sum of time to travel from each rural town to its closest regional center with a budget constraint for upgrading arcs. The objective differs from our study. Our study is similar to Campbell et al.'s study [9] in terms of its objective since both have a Min-Max type of objective. However, in our study, instead of a pre-disaster situation, a post-disaster case is examined and thus, instead of upgrading the roads we focus on unblocking them after a disaster situation. Moreover, different than Campbell et al.'s study [9], routing of multiple vehicles is implemented in this study. This study is different because the main concern is to route a set of vehicles in order to ensure connectivity of the whole network after a disaster situation. Edges which are required to be unblocked for connectivity are not known in advance. Consequently, determination of required arcs to be unblocked in a post-disaster situation requires an efficient network design.

The problem defined in [37] is similar to the *K-ARCP* in terms of disaster context, but they focus on connectivity of some *O-D* pairs and coverage of population but not on arc routing. Similar to our study, the arcs to be upgraded are not known in advance. Moreover, there is a budget limit. In [12], there is also a budget limit which makes it different than our problem and their objective is to minimize the flow cost. As we mentioned before, our problem has a Min-Max type of objective and aims strong connectivity of the network.

In [30, 31, 32, 33], similar to our study, there are different types of costs assigned to a road segment; however, different than ours, all the roads have to be serviced/traversed.

Another difference of this problem from the others given above is that, it has a scheduling perspective as well. If a vehicle arrives at a node incident to a blocked arc, it may have to wait for another vehicle to unblock that arc. This situation arises when more than one vehicle arrive at the same blocked arc in close times. As a result, we need to consider time of arrival of a vehicle at a node incident to an arc every time it is traversed. Therefore, scheduling is another concern in this study. In [39], similar to our study, scheduling aspect is discussed but they consider repairing all road segments which are damaged.

Various performance metrics have been analyzed for post-disaster rehabilitation operations. Performance measures can consist of a single metric or they can be a combination. In our study minimizing the maximum traversal and unblocking time is the performance measure. Chang and Nojima [10] evaluate the performance by considering network coverage and transport accessibility. They assume that after a disaster, network links are damaged and this disrupts the accessibility of some areas due to the closed roads. The performance measures that they describe include total length of open network, total distance-based accessibility and areal distance-based accessibility. The first one reflects the length of the network that is open to traffic at any point at a time t, and is defined as a ratio to the pre-disaster length open. The second measure is based on minimum network travel distances and takes into account the extent and the location of the damage. It attempts to measure changes in accessibility at all nodes of the network. Areal distance-based accessibility pertains to system performance or accessibility from the point of view of subareas. With these measures, they evaluate the transportation performance. They focus on rail and highway transportation systems which are affected in an earthquake. Another study which suggests performance measurements is done by Chen and Tzeng [11]. According to them, minimizing road-network reconstruction time and traffic congestion in the reconstruction period can be performance measures. Their objectives are to minimize the travel time of travelers during reconstruction period, to minimize the total working time and to minimize the idle time between work-troops which reconstruct the damaged roads. In this study they do not mention about any disconnectedness of the graph.

PROBLEM NAME	CLASS	APPROXIMATION FACTOR
CHINESE POSTMAN PROBLEM(CPP):		
CPP on Totally Undirected Graphs	P [15]	
CPP on Totally Directed Graphs	P [15]	
CPP on Mixed but Even Graphs	P [15]	
CPP on Windy Graphs (Windy Postman Problem-WPP)	NP-hard [7]	2 [38]
CPP on Windy but Eulerian Graphs	P [38]	
Mixed CPP	NP-hard [28]	$\frac{4}{3}$ [25]
CAPACITATED CPP:	NP-hard [19]	$\frac{3}{2} - \epsilon$ [19]
K-CPP:		
K-CPP on Totally Undirected Graphs	P [5]	
K-CPP on Totally Directed Graphs	P [5]	
K-CPP on Mixed but Even Graphs	P [29]	
K-CPP on Mixed Graphs	NP-hard [29]	
K-CPP on Windy Graphs	NP-hard [29]	
HIERARCHICAL CPP:	NP-hard [13]	
RURAL POSTMAN PROBLEM(RPP):		
RPP on Undirected Graphs	NP-hard [24]	
RPP on Directed Graphs	NP-hard [24]	
RPP with Costs Satisfying Triangular Inequality		$\frac{3}{2}$ [16]
CAPACITATED ARC ROUTING PROBLEM(CARP):	NP-hard [19]	
STACKER CRANE PROBLEM(SCP):	NP-hard [17]	$\frac{9}{5}$ [17]
Min-Max K-CPP on Undirected Graphs	NP-hard [17]	$2 - \frac{1}{K}$ [17]
Min-Max K-CPP on Windy Graphs	NP-hard [29]	
Min-Max K-RPP on Undirected Graphs	NP-hard	7 [4]

Table 2.1: Routing Problems and Their Computational Complexities

Chapter 3

COMPLEXITY ANALYSIS

The problem defined in this thesis, namely *K-ARCP* is new to the arc routing literature. Therefore, we first analyze the computational complexity of *K-ARCP*.

Theorem 1. *K-ARCP* is NP-hard even when a single vehicle exists.

Proof. In order to prove this theorem, we consider another NP-hard problem, *Rural Postman Problem (RPP)*. We reduce *RPP* to *K-ARCP*

Definition 1. *Undirected Rural Postman Problem (RPP)*:

Let $G = (V, E)$ be an undirected graph, where V is the vertex set, E is the edge set, c_{ij} (≥ 0) is the cost of traversing edge $(i, j) \in E$ and $R \subseteq E$ is the set of required edges. The *RPP* is to determine a least cost closed walk starting from and ending at a depot, traversing each edge of R at least once.

The *RPP* is known to be NP-hard [24]. Now let us consider *K-ARCP*:

Definition 2. *Arc Routing for Connectivity Problem with K Vehicles (K-ARCP)*:

Let $H = (N, A)$ be a directed strongly connected graph, where N is the vertex set, A is the arc set, $B \subseteq A$ is the set of blocked arcs. The graph induced by $A \backslash B$ is disconnected (in the strong sense). c_{ij} is the traversal time on an open arc $(i, j) \in A$ and b_{ij} (≥ 0) is the time of unblocking edge $(i, j) \in B$ in addition to traversal time c_{ij}. $K (\geq 1)$ is the number of vehicles, $d(k)$ is the depot of vehicle k. *K-ARCP* is to determine K walks each one starting from its depot, traversing some of the blocked arcs in B to unblock them *at the first traversal* in order to connect the network. A walk may also include open arcs. *K-ARCP* is to minimize the travel time of the longest walk among the K walks, so that the resulting graph is strongly connected.

For this proof, a special case is analyzed. We consider the case where there is one vehicle and one depot. We call this problem *Arc Routing for Connectivity Problem (ARCP)*.

We take an instance I of *RPP* and construct an instance II of *ARCP* by a polynomial transformation τ between them.

Definition 3. Transformation τ:

We define a directed and strongly connected graph H from G as follows. We replace every edge (i, j) in $E \backslash R$ with two arcs in both directions with traversal times c_{ij}. We take $G = (V, E)$, delete the edges in the set R and for each $(i, j) \in R$ add three new nodes i', j' and p. We define blocked arcs (i, i'), (i', i), (j, j'), (j', j) all with traversing and additional unblocking time of 0. Moreover, between i and j, new blocked arcs , (i, p), (p, i), (j, p), (p, j) with traversal and additional unblocking time $\frac{c_{ij}}{2}$ and 0 are defined.

In order to transform a closed walk in I to an open walk in II, we add a dummy depot d' which is connected to the original depot d of I in the *ARCP* instance with two arcs in both directions, one of them is blocked. Traversal and additional unblocking time on this blocked arc from d to d' is zero. The arc (d', d) which is not blocked has a high traversal time, say M. By assigning a high traversal time to this arc, we enforce the vehicle to visit d' last. The vehicle, located at d, first traverses other arcs in its walk, then, to ensure a strongly connected graph it visits d' as the last node in its walk. It does not visit it in the early stages of its walk because then it will continue its walk to connect the remaining nodes by traversing the arc (d', d) which increases the objective value highly.

Instances I, II and the transformation is illustrated in Figure 3.1.

Lemma 1. Transformation τ from I to II runs in polynomial time in terms of the size of the instance I of *RPP*.

Proof. For every edge in set R in I, we delete one edge and add three nodes and eight arcs. Moreover, for the depot node, one dummy depot and two arcs are added. The edges which are not required to be traversed are doubled into arcs. $\square$

Now, we need to show that we can obtain an optimal solution to I when *ARCP* is solved on II. The nodes i', j', p and d' need to be visited in order to make the graph strongly connected. No matter from which direction the vehicle comes (from i to j or from j to i), it unblocks the arcs (i, i') and (j, j') to reach i' or j'. Due to the definition of *ARCP*, for connectivity, arcs (i', i) and (j', j) have to be unblocked as well. Moreover, node p has to be connected to the network and unblocking one arc going out of node p and one arc coming into it is sufficient in order to ensure strong connectivity of p to the network. Possible routes for the arc segment which corresponds to a required edge can be $i - i' - i - p - j - j' - j$ or $i - i' - i - p - i ... - j - j' - j$ and the reverse. In all cases travel time of these route segments is c_{ij}. If the vehicle needs to pass through nodes i and j, it does not visit i' and j' not to increase its travel time unnecessarily. These route segments can be converted to the edge (i,j) in the *RPP*. Consequently, the required edge (i,j) is traversed. In each traversal of (i,p) and (j,p) (or reverse) together, the cost of the required edge, c_{ij}, is paid. Since the vehicle starts its walk in d and visit d' as the last stop for connectivity purpose, the resulting walk can be transformed to a closed walk starting and ending in the depot node d by omitting the dummy node d' and the corresponding arcs. At the end, the solution of *RPP* on I is reached by solving *ARCP* on II.

Since *RPP* is NP-hard and τ runs in polynomial time, the *ARCP with one vehicle and one depot* is at least as hard as the *RPP*. Since *K-ARCP with one vehicle and one depot* is NP-hard, a more general version *with multiple vehicles and depots* is NP-hard as well. □

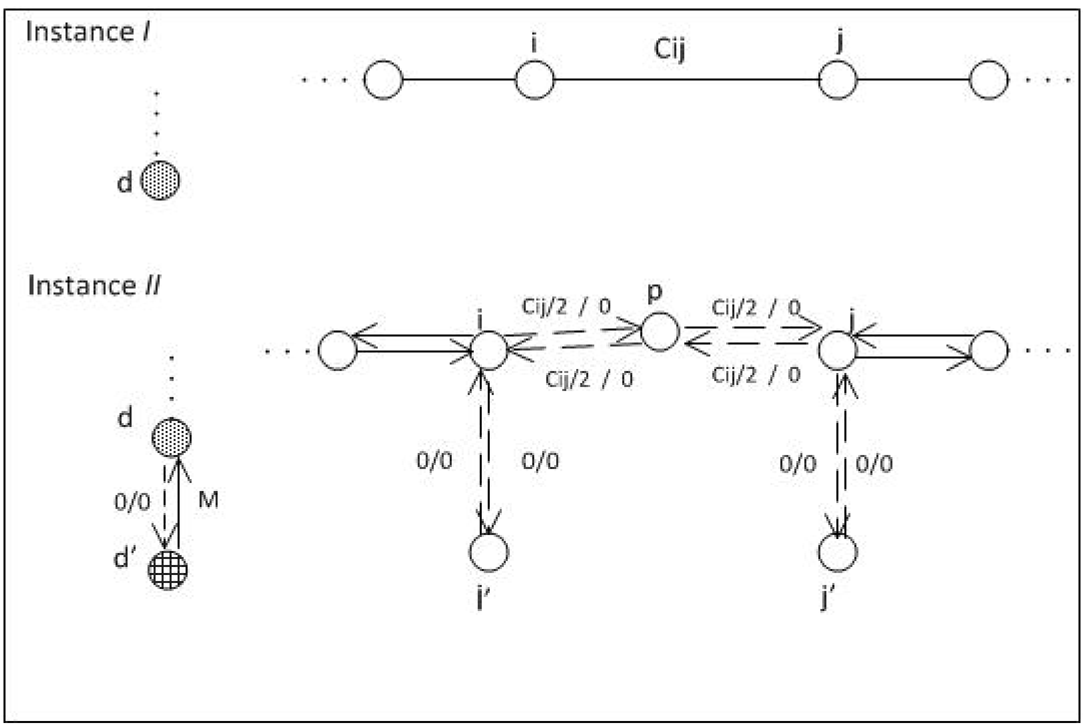

Figure 3.1: Instance I, Instance II and Transformation

Chapter 4

PROBLEM ANALYSIS AND MATHEMATICAL MODEL

This chapter presents a mathematical programming formulation of *K-ARCP* and two relaxation models. We also provide the formulation for *ARCP*.

Some properties of a feasible solution of *K-ARCP* are given below:

1. It is necessary that a subset R of B of arcs is unblocked.

2. A feasible solution consists of at most K walks, and each starts from a depot. Without loss of generality, each walk contains at least one required blocked arc.

3. Each walk may be open.

4. In order to ensure connectivity of the graph, total number of blocked arcs, which are unblocked in cutsets of all components *(C)*, has to be greater than or equal to $2(|Q|-1)$. Otherwise, connectivity cannot be ensured. In other words, in each component's cutset, at least two arcs which are in opposite directions must be unblocked. This property is explained in Proposition 1 in Section 4.1.18. This property is necessary for a solution to be feasible, but it is not sufficient for optimality.

5. Every component is visited by at least one vehicle at least once.

6. If a vehicle starts a walk, it must unblock at least one blocked arc in C. If it does not start a walk, there is no arc which is unblocked by that vehicle. It unblocks at least $|R'|$ arcs, where $|R'|$ is the number of times that the vehicle leaves its depot.

7. If more than one vehicle arrive at a blocked arc in close times, one of them unblocks the arc and others may have to wait until the arc becomes unblocked. Therefore, a waiting time may occur for the vehicles which do not unblock the arc.

K-ARCP does not have a simple formulation due to the following properties. These issues direct us to add time related variables and constraints.

1. Ensuring connectivity of components by cutset inequalities requires subset enumeration. For every subset of components, one incoming and one outgoing arc from that subset must be unblocked. With a large size data, enumeration makes the problem difficult to solve.

2. When a vehicle traverses an arc that was blocked before but was unblocked by another vehicle traversing that arc, we need to make sure that the timing of the traversals is correct. That is, a vehicle may traverse an arc after it is unblocked. Therefore, it may need to wait until the other vehicle completes the unblocking operation.

3. The vehicles may traverse an arc more than once, hence we need to keep track of the arrival time of a vehicle to the arc each time the arc is traversed. To calculate this value, we also need to know the preceding arc and how many times it has been traversed. Number of traversal of an arc is bounded by the number of components. Since a vehicle traverses an arc it connects a component by going through that arc.

4.1 Modeling Approach and Mathematical Models

Let $G = (V, A)$ be a directed and simple graph, where V is the vertex set, A is the arc set, $B \subseteq A$ is the set of blocked arcs such that the graph $G = (V, A \backslash B)$ is disconnected. Q where $|Q| \geq 2$ denotes the set of disconnected components of the graph G. In addition, $b_{ij} \geq 0$ is the additional time of unblocking arc $(i, j) \in B$, c_{ij} is the traversal time on an open arc $(i, j) \in A$ and $K > 1$ is the number of vehicles. When a blocked arc (i, j) is traversed for the first time, it is unblocked with additional time of unblocking b_{ij} and becomes open for later use. If a vehicle waits at an arc, its waiting time is w_{ij}^k. This waiting time depends on arrival times of vehicle k and the other vehicle which has arrived at (i, j) before k, and unblocking time of arc (i, j). A subset of vertices contains depots and each vehicle k is positioned at a depot vertex. We assume that $K > 1$ and there is at least one depot. The walks $k = 1, 2, ..., K$ start at a depot vertex and may traverse some blocked arcs to unblock it for connectivity purpose.

In order to ensure connectivity and continuity of walks, we define flow variables f_{ij}^k for each vehicle and for each arc that the vehicle passes through. For each depot, there is an amount of supply depending on the number of nodes which are visited by the vehicles of that depot. Similarly for each component there is a demand so that each component can receive flow and the graph becomes connected at the end. Then, to prevent flows on an arc which is not traversed, we relate flow variables with x_{ij}^k which shows the number of times an arc (i, j) is traversed by vehicle k. Flow variables are defined as real numbers, however due to unimodularity property, they take integer values because flow variables in the constraints have integer constants. Moreover, we add a dummy sink node and force each vehicle end its tour at this sink node $(n + 1)$. For connectivity, we define subset enumeration constraints. In cutset of each subset of components, some arcs must be unblocked. The details can be seen in the upcoming paragraphs.

Decision variables, which are described below, are the same for all of the models and thus we give them at the beginning of this section.

Sets, Indices and Input Parameters

k : Index of the vehicles

K : Number of vehicles

i, j, g, h : Indices of the vertices

$n + 1$: Index of the dummy sink node

V : Set of vertices: $1, ..., n$

A : Set of arcs

B : Set of blocked arcs

d : Index for the depots

D : Set of depot nodes, $(D \subseteq V)$

P_d : Set of vehicles which are initially positioned in depot d

q : Index of the components

Q : Set of disconnected components

Q_D : Set of components which include at least one depot

S : Set of all subsets of components within which the nodes are strongly connected

s : Index for elements of S

Y^+ : Set of all subsets of components which require at least one outgoing arc
but no incoming arc to be unblocked in order to be strongly connected
to the remaining graph

Y^- : Set of all subsets of components which require at least one incoming arc
but no outgoing arc to be unblocked in order to be strongly connected
to the remaining graph

y : Index for elements of $Y^+ \cup Y^-$

p, p', p'' : Indices that show how many times an arc is traversed, $p, p', p'' = 1, ..., P$

F_q : Set of vehicles (fleet) which are initially positioned in component q

V_q : Set of vertices in component $q \in Q$ $(V_q \subseteq V)$

C : Union of cutset arcs of V_q for all $q \in Q$, $C : \delta^+(q) \cup \delta^-(q)$

M : A nonnegative scalar with large enough value

Decision Variables

x_{ij}^k : number of times that vehicle k traverses arc $(i,j) \in A$

z_{ij}^k : binary variable indicating if blocked arc $(i,j) \in B$ is unblocked by vehicle k

f_{ij}^k : flow variable of vehicle k on arc $(i,j) \in A$

v_i^k : number of times vehicle k visits node $i \in V$

$u_{ij}^{k,p}$: binary variable indicating if arc $(i,j) \in A$ is traversed by vehicle k for the p^{th} time

$u_{ijg}^{k,p',p}$: binary variable indicating if arc $(i,j) \in A$ is traversed by vehicle k for the p'^{th} time and the next arc to be traversed is $(j,g) \in A$ and it is traversed for the p^{th} time

$t_{ijg}^{k,p',p}$: time at which vehicle k visits node j while traversing arc $(j,g) \in A$ for the p^{th} time immediately after traversing arc $(i,j) \in A$ for the p'^{th} time
($t_{ijg}^{k,p',p} = 0$, if $u_{ijg}^{k,p',p} = 0$)

$w_{ij}^{k,p}$: waiting time of vehicle k on its p^{th} traversal of arc $(i,j) \in B$ ($w_{ij}^{k,p} = 0$ for $p > 1$)

The mixed integer programming (MIP) model for *K-ARCP* determines K open walks such that the disconnected components in the network are connected after unblocking a subset of the blocked arcs. The model is formulated for the multi-depot and K vehicles-case, where $K > 1$. The K walks altogether traverse a subset of the arcs in B, say R, so that the graph $G' = (V, A \backslash B \cup R)$ is connected. We name the model that solves *K-ARCP* as *MIP1* and it gives a strongly connected graph and walks that are synchronized.

Before presenting *MIP1*, we explain the objective functions and constraints group by group.

4.1.1 Objective Function of MIP1

Constraints (4.1) represent the objective function which includes waiting times of vehicles at blocked arcs. Since, waiting can occur for a vehicle only in its first traversal of a blocked arc, p, the index for traversal time, takes the value of 1.

Minimize y

subject to

$$\sum_{(i,j)\in A} c_{ij}x_{ij}^k + \sum_{(i,j)\in B} b_{ij}z_{ij}^k + \sum_{(i,j)\in B} w_{ij}^{k,1} \leq y, \quad \forall k = 1, 2, \ldots, K \tag{4.1}$$

4.1.2 Vehicle Balance Equations

Constraints (4.2)-(4.6) are vehicle balance equations. Constraints (4.2) ensure that each vehicle starts the tour at a depot vertex where it is positioned. In constraints (4.3), for the depot nodes, the equation is written for the vehicles which are not positioned in that specific depot. (4.4) balances arrivals and departures for a node i which is not a depot. (4.5) forces every walk to end in the sink node. There is only one visit to the sink node and no return. The latter case is satisfied by the constraints (4.6). A vehicle leaves the depot and its component, and does not return there if it will not visit another disconnected component through its own component.

$$\sum_{j\in V\cup\{(n+1)\}} (x_{dj}^k - x_{jd}^k) = 1, \quad \forall d \in D, \quad \forall k \in P_d \tag{4.2}$$

$$\sum_{j\in V\cup\{(n+1)\}} (x_{dj}^k - x_{jd}^k) = 0, \quad \forall d \in D, \quad \forall k \notin P_d \tag{4.3}$$

$$\sum_{j\in V\cup\{(n+1)\}} (x_{ij}^k - x_{ji}^k) = 0, \quad \forall k = 1, 2, \ldots, K, \quad \forall i \in V \backslash D \tag{4.4}$$

$$\sum_{j\in V} x_{j(n+1)}^k = 1, \quad \forall k = 1, 2, \ldots, K \tag{4.5}$$

$$x_{(n+1)i}^k = 0, \quad \forall i \in V, \quad \forall k = 1, 2, \ldots, K \tag{4.6}$$

4.1.3 Constraints Which Relate Variables x_{ij}^k and z_{ij}^k

Constraints (4.7) show for a blocked arc that, if it is unblocked then it is also traversed. We assume a blocked arc becomes open in both directions whenever a vehicle unblocks it in one direction. This assumption can be meaningful because in disaster situations, roads have to be used in both directions in order to reach disastrous area and deliver aid. Constraints

(4.8) prevent the vehicles traversing a blocked arc if it is not unblocked. If an arc (i, j) is unblocked, it can be traversed by the same vehicle at most $2(|Q| - 1)$ times. In the worst case, if there is one vehicle to connect all graph, it connects one component each time it traverses the same arc by unblocking one arc going out of the subset of component and one arc coming into it. Therefore we multiply this value by 2. Except the component that it is deployed, there are $|Q| - 1$ components in total to be connected, thus the scalar in this constraint takes the value of $2(|Q| - 1)$. Constraints (4.9) and (4.10) show that if a vehicle starts a walk, it must unblock at least one blocked arc in C. If it does not start a walk, there is no arc which is unblocked by that vehicle. It unblocks at least $|R'|$ arcs, where $|R'|$ is the number of times that the vehicle leaves its depot. In the worst case (assuming that there is only one vehicle), the maximum number of arcs that a vehicle can unblock each time it leaves its depot is $2(|Q| - 1)$ to make the graph strongly connected.

$$x_{ij}^k \geq z_{ij}^k, \quad \forall k = 1, 2, \ldots, K, \quad \forall (i, j) \in B \tag{4.7}$$

$$x_{ij}^k \leq 2(|Q| - 1) \sum_{k'=1}^{K} z_{ij}^{k'}, \quad \forall k = 1, 2, \ldots, K, \quad \forall (i, j) \in B \tag{4.8}$$

$$\sum_{j \in V} x_{dj}^k \leq \sum_{(i,j) \in C} z_{ij}^k, \quad \forall d \in D, \quad \forall k \in P_d \tag{4.9}$$

$$\sum_{(i,j) \in C} z_{ij}^k \leq 2(|Q| - 1) \sum_{j \in V} x_{dj}^k, \quad \forall d \in D, \quad \forall k \in P_d \tag{4.10}$$

4.1.4 Flow Balance Equations

In order to ensure connectivity of the nodes in walks, we define flow variables f_{ij}^k for each vehicle and for each arc that it passes through. For depot vertices, the net flow into a depot vertex is the total number of visits to all vertices except the depot. For the other vertices, it is equal to the number of visits to the corresponding node. In other words, a vehicle leaves one unit of flow each time it visits a node. Similar to (4.3), the depot nodes are considered separately. Constraints (4.14) ensure that walks end in sink node by sending one unit of flow to the sink node. (4.15) prevent backward flow from the sink node to any other node.

$$\sum_{j:(i,j)\in A,\{i,j\}\in V\cup\{(n+1)\}} (f_{ij}^k - f_{ji}^k) = -v_i^k,$$
$$\forall k = 1, 2, \ldots, K, \quad \forall i \in V \cup \{(n+1)\} \setminus D \tag{4.11}$$
$$\sum_{j\in V\cup\{(n+1)\}} (f_{dj}^k - f_{jd}^k) = \sum_{i\in V\cup\{(n+1)\}\setminus\{d\}} v_i^k, \quad \forall k \in P_d, \quad \forall d \in D \tag{4.12}$$
$$\sum_{j:(d,j)\in A,\{i,j\}\in V\cup\{(n+1)\}} (f_{dj}^k - f_{jd}^k) = -v_d^k, \quad \forall d \in D, \quad \forall k \notin P_d \tag{4.13}$$
$$f_{(n+1)j}^k = 0, \quad \forall j \in V, \quad \forall k = 1, 2, \ldots, K \tag{4.14}$$
$$\sum_{j\in V} f_{j(n+1)}^k = 1, \quad \forall k = 1, 2, \ldots, K \tag{4.15}$$

4.1.5 Constraints Which Relate Variables f_{ij}^k and x_{ij}^k

Constraints (4.16) do not allow flow on an arc unless it is traversed. Constraints (4.17) show that if an arc is traversed, then there must be a positive amount of flow passing through it.

$$f_{ij}^k \le M x_{ij}^k, \quad \forall k = 1, 2, \ldots, K, \quad \forall (i,j) \in A, \quad \{i,j\} \in V \cup \{(n+1)\} \tag{4.16}$$
$$f_{ij}^k \ge x_{ij}^k, \quad \forall k = 1, 2, \ldots, K, \quad \forall (i,j) \in A, \quad \{i,j\} \in V \cup \{(n+1)\} \tag{4.17}$$

4.1.6 Constraints Which Relate Variables x_{ij}^k and v_i^k

Constraints (4.18) relate the variables x_{ij}^k and v_i^k so that v_i^k counts the number of visits to node i by vehicle k. A vertex is visited if and only if an arc entering that vertex is traversed.

$$\sum_{j:(i,j)\in A} x_{ji}^k = v_i^k, \quad \forall k = 1, 2, \ldots, K, \quad \forall i \in V \cup \{(n+1)\} \tag{4.18}$$

4.1.7 Component Connectivity Constraints

(4.19) and (4.20) ensure that in cutsets of every subset of components in set S, at least one arc into and one arc out of the subset is unblocked. Similarly, with constraints (4.21) and

(4.22), for connectivity of the components in sets Y^+ and Y^-, at least one arc into and one arc out of the subset is unblocked. As a result, the graph becomes strongly connected.

$$\sum_{k=1}^{K} \sum_{(i,j)\in\delta^+(s)} z_{ij}^k \geq 1, \quad \forall s \subset S \tag{4.19}$$

$$\sum_{k=1}^{K} \sum_{(i,j)\in\delta^-(s)} z_{ij}^k \geq 1, \quad \forall s \subset S \tag{4.20}$$

$$\sum_{k=1}^{K} \sum_{(i,j)\in\delta^+(y)} z_{ij}^k \geq 1, \quad \forall y \subset Y^+ \tag{4.21}$$

$$\sum_{k=1}^{K} \sum_{(i,j)\in\delta^-(y)} z_{ij}^k \geq 1, \quad \forall y \subset Y^- \tag{4.22}$$

4.1.8 Constraints Which Initialize Time Variables $t_{ijg}^{k,p',p}$

Time is 0 at the depot nodes. The following constraints initialize the time.

$$t_{dij}^{k,1,1} = 0, \quad \forall (d,i) \in A, \quad \forall (i,j) \in A, \quad \forall k = 1,2,\ldots,K, \quad \forall d \in D \tag{4.23}$$

4.1.9 Constraints Which Calculate Time of Arrival at an Arc

The following constraints calculate the arrival time of a vehicle at an arc after traversing another arc. If the vehicle is traversing a blocked arc, it may arrive at the next arc after unblocking the current arc or after waiting until it is unblocked by another vehicle. If the current arc is not a blocked arc, the vehicle traverses the arc with corresponding traversal time and arrives at the next arc.

$$\begin{aligned} & t_{ijg}^{k,1,p} \geq t_{hij}^{k,p',1} + w_{ij}^{k,1} + b_{ij} z_{ij}^k + c_{ij} u_{hij}^{k,p',1} - M(1 - u_{hij}^{k,p',1}), \\ & \quad \forall (i,j) \in B, \quad \forall (h,i) \in A, \quad \forall (j,g) \in A, \quad \forall k = 1,2,\ldots,K, \quad \forall p,p', = 1,\ldots,P \end{aligned} \tag{4.24}$$

$$\begin{aligned} & t_{ijg}^{k,p',p} \geq t_{hij}^{k,p'',p'} + c_{ij} u_{hij}^{k,p'',p'} - M(1 - u_{hij}^{k,p'',p'}), \quad \forall (i,j) \in A, \quad \forall (h,i) \in A, \\ & \quad \forall (j,g) \in A, \quad \forall k = 1,2,\ldots,K, \quad \forall p,p',p'' = 1,\ldots,P, \quad \forall p' = 1,\ldots,P, \end{aligned} \tag{4.25}$$

4.1.10 Constraints Required for Calculation of Waiting Time

Waiting time variables $w_{ij}^{k,p}$ are defined only for the blocked arcs and for the first time of traversal by a vehicle. If a blocked arc is being traversed more than once, it has been unblocked before and no waiting occurs. Constraints (4.26) state that, if two vehicles k and l arrive at a blocked arc one after another, one of them unblocks and the other may have to wait until it becomes unblocked.

$$t_{hij}^{l,p',1} + w_{ij}^{l,1} \geq t_{hij}^{k,p,1} + b_{ij} z_{ij}^{k} + c_{ij} u_{hij}^{k,p,1} - M(1 - u_{hij}^{k,p,1}),$$
$$\forall (i,j) \in B, \quad \forall (h,i) \in A, \quad \forall k, l = 1, 2, \ldots, K, \quad \forall p, p', = 1, \ldots, P \tag{4.26}$$
$$w_{ij}^{k,p} = 0 \quad \forall (i,j) \in B, \quad \forall k = 1, 2, \ldots, K, \quad \forall p = 2, \ldots, P \tag{4.27}$$

4.1.11 Constraints Which Relate Variables $t_{ij}^{k,p}$ with $u_{ij}^{k,p}$

Constraints (4.28) prevent time variables taking positive value unless an arc is traversed for the p^{th} (and p'^{th}) time.

$$t_{ijg}^{k,p',p} \leq M u_{ijg}^{k,p',p}, \quad \forall (i,j) \in A, \quad \forall (j,g) \in A,$$
$$\forall k = 1, 2, \ldots, K, \quad \forall p, p' = 1, \ldots, P \tag{4.28}$$

4.1.12 Constraints Which Identify Precedence Between Variables $t_{ijg}^{k,p',p}$

The following constraints imply that time increases as the number of traversal at an arc increases.

$$t_{ijg}^{k,p',p} \geq t_{ijg}^{k,p',(p-1)}, \quad \forall (i,j) \in A, \quad \forall (j,g) \in A,$$
$$\forall k = 1, 2, \ldots, K, \quad \forall p, p' = 1, \ldots, P \tag{4.29}$$

4.1.13 Constraints Which Relate Variables $u_{ij}^{k,p}$ with x_{ij}^{k} and z_{ij}^{k}

The following constraints define the binary variables $u_{ij}^{k,p}$. As a vehicle traverses an arc more than once, summation of variables $u_{ij}^{k,p}$ over p is equal to the total number of traversal of the arc (i, j). If a blocked arc is unblocked by a vehicle, this means, it is the first time of the vehicle traversing that arc.

$$\sum_{p=1}^{P} u_{ij}^{k,p} = x_{ij}^{k}, \quad \forall (i,j) \in A, \quad \forall k = 1,2,\ldots,K \tag{4.30}$$

$$u_{ij}^{k,1} \geq z_{ij}^{k}, \quad \forall (i,j) \in B, \quad \forall k = 1,2,\ldots,K \tag{4.31}$$

4.1.14 Constraints Which Relate Variables $u_{ijg}^{k,p',p}$ with $u_{ij}^{k,p}$

These constraints relate the variables $u_{ijg}^{k,p',p}$ with $u_{ij}^{k,p}$.

$$u_{ijg}^{k,p',p} \leq u_{ij}^{k,p'}, \quad \forall (i,j) \in A, \quad \forall (j,g) \in A, \quad \forall k = 1,2,\ldots,K, \quad \forall p,p' = 1,\ldots,P \tag{4.32}$$

$$u_{ijg}^{k,p',p} \leq u_{jg}^{k,p}, \quad \forall (i,j) \in A, \quad \forall (j,g) \in A, \quad \forall k = 1,2,\ldots,K, \quad \forall p,p' = 1,\ldots,P \tag{4.33}$$

4.1.15 Constraints Which Define the Variables

Constraints (4.34) - (4.36) are integrality constraints whereas constraints (4.37) are binary constraints. Constraints (4.38)-(4.40) state that flow variables and time variables are non-negative real numbers. Finally, (4.41) and (4.42) assign variables $u_{ij}^{k,p}$ and $u_{ijg}^{k,p',p}$ binary numbers.

$$x_{ij}^{k}, x_{ji}^{k} \in \mathbb{Z}_+, \quad \forall k = 1,2,\ldots,K, \quad \forall (i,j) \in A, \quad \{i,j\} \in V \tag{4.34}$$

$$x_{i(n+1)}^{k} \in \mathbb{Z}_+, \quad \forall i \in V, \quad \forall k = 1,2,\ldots,K \tag{4.35}$$

$$v_{i}^{k} \in \mathbb{Z}_+, \quad \forall k = 1,2,\ldots,K, \quad \forall i \in V \tag{4.36}$$

$$z_{ij}^{k} \in \mathbb{B}, \quad \forall k = 1,2,\ldots,K, \quad \forall (i,j) \in B \tag{4.37}$$

$$f_{ij}^{k}, f_{ji}^{k} \in \mathbb{R}_+, \quad \forall (i,j) \in A, \quad \forall k = 1,2,\ldots,K \tag{4.38}$$

$$t_{ijg}^{k,p',p} \in \mathbb{R}_+, \quad \forall (i,j) \in A, \quad \forall (j,g) \in A, \quad \forall k = 1,2,\ldots,K, \quad \forall p,p' = 1,2,\ldots,P \tag{4.39}$$

$$w_{ij}^{k,p} \in \mathbb{R}_+, \quad \forall (i,j) \in B, \quad \forall k = 1,2,\ldots,K, \quad p = 1 \tag{4.40}$$

$$u_{ijg}^{k,p',p} \in \mathbb{B}, \quad \forall (i,j) \in A, \quad \forall (j,g) \in A, \quad \forall k = 1,2,\ldots,K, \quad \forall p,p' = 1,2,\ldots,P \tag{4.41}$$

$$u_{ij}^{k,p} \in \mathbb{B}, \quad \forall (i,j) \in A, \quad \forall k = 1,2,\ldots,K, \quad \forall p = 1,2,\ldots,P \tag{4.42}$$

The following mathematical formulation finds the optimal routes for vehicles and minimizes the maximum tour length among K vehicles. As a result the graph is connected and no vehicle can traverse an arc before it is unblocked.
MIP Model of K-ARCP: MIP1

Minimize y

subject to

$$\sum_{(i,j)\in A} c_{ij}x_{ij}^k + \sum_{(i,j)\in B} b_{ij}z_{ij}^k + \sum_{(i,j)\in B} w_{ij}^{k,1} \le y, \quad \forall k = 1,2,\dots,K \tag{4.43}$$

$$\sum_{j\in V\cup\{(n+1)\}} (x_{dj}^k - x_{jd}^k) = 1, \quad \forall d \in D, \quad \forall k \in P_d \tag{4.44}$$

$$\sum_{j\in V\cup\{(n+1)\}} (x_{dj}^k - x_{jd}^k) = 0, \quad \forall d \in D, \quad \forall k \notin P_d \tag{4.45}$$

$$\sum_{j\in V\cup\{(n+1)\}} (x_{ij}^k - x_{ji}^k) = 0, \quad \forall k = 1,2,\dots,K, \quad \forall i \in V\backslash D \tag{4.46}$$

$$\sum_{j\in V} x_{j(n+1)}^k = 1, \quad \forall k = 1,2,\dots,K \tag{4.47}$$

$$x_{(n+1)i}^k = 0, \quad \forall i \in V, \quad \forall k = 1,2,\dots,K \tag{4.48}$$

$$x_{ij}^k \ge z_{ij}^k, \quad \forall k = 1,2,\dots,K, \quad \forall (i,j) \in B \tag{4.49}$$

$$x_{ij}^k \le 2(|Q|-1)\sum_{k'=1}^{K} z_{ij}^{k'}, \quad \forall k = 1,2,\dots,K, \quad \forall (i,j) \in B \tag{4.50}$$

$$\sum_{j\in V} x_{dj}^k \le \sum_{(i,j)\in C} z_{ij}^k, \quad \forall d \in D, \quad \forall k \in P_d \tag{4.51}$$

$$\sum_{(i,j)\in C} z_{ij}^k \le 2(|Q|-1)\sum_{j\in V} x_{dj}^k, \quad \forall d \in D, \quad \forall k \in P_d \tag{4.52}$$

$$\sum_{j:(i,j)\in A,\{i,j\}\in V\cup\{(n+1)\}} (f_{ij}^k - f_{ji}^k) = -v_i^k, \quad \forall k = 1,2,\dots,K, \quad \forall i \in V\cup\{(n+1)\}\setminus D \tag{4.53}$$

$$\sum_{j\in V\cup\{(n+1)\}} (f_{dj}^k - f_{jd}^k) = \sum_{i\in V\cup\{(n+1)\}\backslash\{d\}} v_i^k, \quad \forall k \in P_d, \quad \forall d \in D \tag{4.54}$$

$$\sum_{j:(d,j)\in A,\{i,j\}\in V\cup\{(n+1)\}} (f_{dj}^k - f_{jd}^k) = -v_d^k, \quad \forall d \in D, \quad \forall k \notin P_d \tag{4.55}$$

$$f_{ij}^k \le Mx_{ij}^k, \quad \forall k = 1,2,\dots,K, \quad \forall (i,j) \in A, \quad \{i,j\} \in V\cup\{(n+1)\} \tag{4.56}$$

$$f_{ij}^k \geq x_{ij}^k, \quad \forall k = 1, 2, \ldots, K, \quad \forall (i,j) \in A, \quad \{i,j\} \in V \cup \{(n+1)\} \tag{4.57}$$

$$f_{(n+1)j}^k = 0, \quad \forall j \in V, \quad \forall k = 1, 2, \ldots, K \tag{4.58}$$

$$\sum_{j \in V} f_{j(n+1)}^k = 1, \quad \forall k = 1, 2, \ldots, K \tag{4.59}$$

$$\sum_{j:(i,j) \in A} x_{ji}^k = v_i^k, \quad \forall k = 1, 2, \ldots, K, \quad \forall i \in V \cup \{(n+1)\} \tag{4.60}$$

$$\sum_{k=1}^{K} \sum_{(i,j) \in \delta^+(s)} z_{ij}^k \geq 1, \quad \forall s \subset S \tag{4.61}$$

$$\sum_{k=1}^{K} \sum_{(i,j) \in \delta^-(s)} z_{ij}^k \geq 1, \quad \forall s \subset S \tag{4.62}$$

$$\sum_{k=1}^{K} \sum_{(i,j) \in \delta^+(y)} z_{ij}^k \geq 1, \quad \forall y \subset Y^+ \tag{4.63}$$

$$\sum_{k=1}^{K} \sum_{(i,j) \in \delta^-(y)} z_{ij}^k \geq 1, \quad \forall y \subset Y^- \tag{4.64}$$

$$t_{dij}^{k,1,1} = 0, \quad \forall (d,i) \in A, \quad \forall (i,j) \in A, \quad \forall k = 1, 2, \ldots, K, \quad \forall d \in D \tag{4.65}$$

$$\begin{aligned} &t_{ijg}^{k,1,p} \geq t_{hij}^{k,p',1} + w_{ij}^{k,1} + b_{ij} z_{ij}^k + c_{ij} u_{hij}^{k,p',1} - M(1 - u_{hij}^{k,p',1}), \\ &\quad \forall (i,j) \in B, \quad \forall (h,i) \in A, \quad \forall (j,g) \in A, \quad \forall k = 1, 2, \ldots, K, \quad \forall p, p', = 1, \ldots, P \end{aligned} \tag{4.66}$$

$$\begin{aligned} &t_{ijg}^{k,p',p} \geq t_{hij}^{k,p'',p'} + c_{ij} u_{hij}^{k,p'',p'} - M(1 - u_{hij}^{k,p'',p'}), \quad \forall (i,j) \in A, \quad \forall (h,i) \in A, \\ &\quad \forall (j,g) \in A, \quad \forall k = 1, 2, \ldots, K, \quad \forall p, p', p'' = 1, \ldots, P, \quad \forall p' = 1, \ldots, P, \end{aligned} \tag{4.67}$$

$$\begin{aligned} &t_{hij}^{l,p',1} + w_{ij}^{l,1} \geq t_{hij}^{k,p,1} + b_{ij} z_{ij}^k + c_{ij} u_{hij}^{k,p,1} - M(1 - u_{hij}^{k,p,1}), \\ &\quad \forall (i,j) \in B, \quad \forall (h,i) \in A, \quad \forall k, l = 1, 2, \ldots, K, \quad \forall p, p', = 1, \ldots, P \end{aligned} \tag{4.68}$$

$$w_{ij}^{k,p} = 0 \quad \forall (i,j) \in B, \quad \forall k = 1, 2, \ldots, K, \quad \forall p = 2, \ldots, P \tag{4.69}$$

$$t_{ijg}^{k,p',p} \leq M u_{ijg}^{k,p',p}, \quad \forall (i,j) \in A, \quad \forall (j,g) \in A, \quad \forall k = 1, 2, \ldots, K, \quad \forall p, p' = 1, \ldots, P \tag{4.70}$$

$$t_{ijg}^{k,p',p} \geq t_{ijg}^{k,p',(p-1)}, \quad \forall (i,j) \in A, \quad \forall (j,g) \in A, \quad \forall k = 1, 2, \ldots, K, \quad \forall p, p' = 1, \ldots, P \tag{4.71}$$

$$\sum_{p=1}^{P} u_{ij}^{k,p} = x_{ij}^k, \quad \forall (i,j) \in A, \quad \forall k = 1, 2, \ldots, K \tag{4.72}$$

$$u_{ij}^{k,1} \geq z_{ij}^k, \quad \forall (i,j) \in B, \quad \forall k = 1, 2, \ldots, K \tag{4.73}$$

$$u_{ijg}^{k,p',p} \leq u_{ij}^{k,p'}, \quad \forall (i,j) \in A, \quad \forall (j,g) \in A, \quad \forall k = 1,2,\ldots,K, \quad \forall p,p' = 1,\ldots,P \tag{4.74}$$

$$u_{ijg}^{k,p',p} \leq u_{jg}^{k,p}, \quad \forall (i,j) \in A, \quad \forall (j,g) \in A, \quad \forall k = 1,2,\ldots,K, \quad \forall p,p' = 1,\ldots,P \tag{4.75}$$

$$x_{ij}^k, x_{ji}^k \in \mathbb{Z}_+, \quad \forall k = 1,2,\ldots,K, \quad \forall (i,j) \in A, \quad \{i,j\} \in V \tag{4.76}$$

$$x_{i(n+1)}^k \in \mathbb{Z}_+, \quad \forall i \in V, \quad \forall k = 1,2,\ldots,K \tag{4.77}$$

$$v_i^k \in \mathbb{Z}_+, \quad \forall k = 1,2,\ldots,K, \quad \forall i \in V \tag{4.78}$$

$$z_{ij}^k \in \mathbb{B}, \quad \forall k = 1,2,\ldots,K, \quad \forall (i,j) \in B \tag{4.79}$$

$$f_{ij}^k, f_{ji}^k \in \mathbb{R}_+, \quad \forall (i,j) \in A, \quad \forall k = 1,2,\ldots,K \tag{4.80}$$

$$t_{ijg}^{k,p',p} \in \mathbb{R}_+, \quad \forall (i,j) \in A, \quad \forall (j,g) \in A, \quad \forall k = 1,2,\ldots,K, \quad \forall p,p' = 1,2,\ldots,P \tag{4.81}$$

$$w_{ij}^{k,p} \in \mathbb{R}_+, \quad \forall (i,j) \in B, \quad \forall k = 1,2,\ldots,K, \quad p = 1 \tag{4.82}$$

$$u_{ijg}^{k,p',p} \in \mathbb{B}, \quad \forall (i,j) \in A, \quad \forall (j,g) \in A, \quad \forall k = 1,2,\ldots,K, \quad \forall p,p' = 1,2,\ldots,P \tag{4.83}$$

$$u_{ij}^{k,p} \in \mathbb{B}, \quad \forall (i,j) \in A, \quad \forall k = 1,2,\ldots,K, \quad \forall p = 1,2,\ldots,P \tag{4.84}$$

4.1.16 The Single Vehicle Case (ARCP) and Its Mathematical Model

In this thesis, as a special case of *K-ARCP*, we analyze the single vehicle case in which the vehicle is positioned in a depot d. The single vehicle problem, *ARCP*, is easier to solve because there is no requirement to consider the arrival time of the vehicle. Clearly, since there is only one vehicle, it will unblock necessary arcs without waiting for an other vehicle to unblock. Connectivity constraints are not necessary for every subset of components, unblocking one arc incoming and one arc outgoing in each component's cutset is sufficient for strong connectivity. No subtour or subgraph of components will be in the optimal solution, because we ensure connectivity of the walk by sending flow. Besides connectivity and time related constraints and variables, some other constraints are not necessary as well. For instance, constraints (4.9) and (4.10) are not required, since there is no other vehicle which needs to unblock a road in order to start a walk. In the single vehicle case, all the constraints which are multiplied as many as the number of vehicles are written according to the single vehicle case. The mathematical formulation for *ARCP* is given below.

Decision Variables

x_{ij} : number of times that the vehicle traverses arc (i,j)

z_{ij} : binary variable indicating if blocked arc (i,j) is unblocked

f_{ij} : flow variable on arc (i,j)

v_i : number of times the vehicle visits node i

MIP Model of ARCP

$$\text{Minimize} \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{(i,j)\in B} b_{ij}z_{ij}$$

subject to

$$\sum_{j\in V\cup\{(n+1)\}} (x_{dj} - x_{jd}) = 1, \quad d \in D \tag{4.85}$$

$$\sum_{j\in V\cup\{(n+1)\}} (x_{ij} - x_{ji}) = 0, \quad \forall i \in V\backslash D \tag{4.86}$$

$$\sum_{j\in V} x_{j(n+1)} = 1 \tag{4.87}$$

$$x_{(n+1)i} = 0, \quad \forall i \in V \tag{4.88}$$

$$x_{ij} \geq z_{ij}, \quad \forall (i,j) \in B \tag{4.89}$$

$$x_{ij} \leq 2(|Q|-1)z_{ij}, \quad \forall (i,j) \in B \tag{4.90}$$

$$\sum_{j:(i,j)\in A,\{i,j\}\in V\cup\{(n+1)\}} (f_{ij} - f_{ji}) = -v_i, \quad \forall i \in V \cup \{(n+1)\} \setminus D \tag{4.91}$$

$$\sum_{j\in V\cup\{(n+1)\}} (f_{dj} - f_{jd}) = \sum_{i\in V\cup\{(n+1)\}\setminus\{d\}} v_i, \quad d \in D \tag{4.92}$$

$$f_{ij} \leq Mx_{ij}, \quad \forall (i,j) \in A, \quad \{i,j\} \in V \cup \{(n+1)\} \tag{4.93}$$

$$f_{ij} \geq x_{ij}, \quad \forall (i,j) \in A, \quad \{i,j\} \in V \cup \{(n+1)\} \tag{4.94}$$

$$f_{(n+1)j} = 0, \quad \forall j \in V \tag{4.95}$$

$$\sum_{j\in V} f_{j(n+1)} = 1 \tag{4.96}$$

$$\sum_{(i,j)\in\delta^+(q)} z_{ij} \geq 1, \quad \forall q \subset Q \tag{4.97}$$

$$\sum_{(i,j)\in\delta^-(q)} z_{ij} \geq 1, \quad \forall q \subset Q \tag{4.98}$$

$$x_{ij}, x_{ji} \in \mathbb{Z}_+, \quad \forall (i,j) \in A, \quad \{i,j\} \in V \tag{4.99}$$

$$x_{i(n+1)} \in \mathbb{Z}_+, \quad \forall i \in V \tag{4.100}$$

$$v_i \in \mathbb{Z}_+, \quad \forall i \in V \tag{4.101}$$

$$z_{ij} \in \mathbb{B}, \quad \forall (i,j) \in B \tag{4.102}$$

$$f_{ij}, f_{ji} \in \mathbb{R}_+, \quad \forall (i,j) \in A \tag{4.103}$$

4.1.17 Relaxations of MIP1

We again consider the multi-vehicle problem and provide the formulations of *MIP2* and *MIP3* which are relaxed versions of *MIP1*.

MIP2 This is a relaxed model which results in a connected graph but walks may not have the correct timing in terms of unblocking time of the blocked arcs. In this model, we do not include time related constraints. Hence time related variables $t_{ij}^{k,p}$, $w_{ij}^{k,p}$, $u_{ij}^{k,p}$ are not defined but we add connectivity constraints.

As an example of a connected but unsynchronized solution obtained from *MIP2*, we can consider the 74 node data in Chapter 5. If we solve a three-component, three-depot and three-vehicle instance by using MIP2 we obtain the following walks for each vehicle:

23-21-20-19-16-10-16; 15-74-16-10-16; 32-29-68-65-68. In the first and the second walk there are common blocked roads (16,10) and (10,16). In the solution of *MIP2*, vehicle 1 unblocks (16,10) and vehicle 2 unblocks (10,16). Vehicle 2 arrives at (10,16) after traversing (16,10) assuming that it is unblocked before. Similarly, vehicle 1 traverses (10,16) assuming that it is unblocked before. Note that (10,16) can be unblocked after vehicle 2 traverses (16,10) which has to be unblocked by vehicle 1. As a result, vehicles traverse arcs with no information about time of unblocking processes of blocked arcs.

Relaxed Model of K-ARCP: MIP2

Minimize y

subject to

$$\sum_{(i,j)\in A} c_{ij} x_{ij}^k + \sum_{(i,j)\in B} b_{ij} z_{ij}^k \le y, \quad \forall k = 1, 2, \ldots, K \tag{4.104}$$

$$\sum_{j \in V \cup \{(n+1)\}} (x_{dj}^k - x_{jd}^k) = 1, \quad \forall d \in D, \quad \forall k \in P_d \tag{4.105}$$

$$\sum_{j \in V \cup \{(n+1)\}} (x_{dj}^k - x_{jd}^k) = 0, \quad \forall d \in D, \quad \forall k \notin P_d \tag{4.106}$$

$$\sum_{j \in V \cup \{(n+1)\}} (x_{ij}^k - x_{ji}^k) = 0, \quad \forall k = 1, 2, \dots, K, \quad \forall i \in V \backslash D \tag{4.107}$$

$$\sum_{j \in V} x_{j(n+1)}^k = 1, \quad \forall k = 1, 2, \dots, K \tag{4.108}$$

$$x_{(n+1)i}^k = 0, \quad \forall i \in V, \quad \forall k = 1, 2, \dots, K \tag{4.109}$$

$$x_{ij}^k \geq z_{ij}^k, \quad \forall k = 1, 2, \dots, K, \quad \forall (i,j) \in B \tag{4.110}$$

$$x_{ij}^k \leq 2(|Q| - 1) \sum_{k'=1}^{K} z_{ij}^{k'}, \quad \forall k = 1, 2, \dots, K, \quad \forall (i,j) \in B \tag{4.111}$$

$$\sum_{j \in V} x_{dj}^k \leq \sum_{(i,j) \in C} z_{ij}^k, \quad \forall d \in D, \quad \forall k \in P_d \tag{4.112}$$

$$\sum_{(i,j) \in C} z_{ij}^k \leq 2(|Q| - 1) \sum_{j \in V} x_{dj}^k, \quad \forall d \in D, \quad \forall k \in P_d \tag{4.113}$$

$$\sum_{j:(i,j) \in A, \{i,j\} \in V \cup \{(n+1)\}} (f_{ij}^k - f_{ji}^k) = -v_i^k, \quad \forall k = 1, 2, \dots, K, \quad \forall i \in V \cup \{(n+1)\} \setminus D \tag{4.114}$$

$$\sum_{j \in V \cup \{(n+1)\}} (f_{dj}^k - f_{jd}^k) = \sum_{i \in V \cup \{(n+1)\} \backslash \{d\}} v_i^k, \quad \forall k \in P_d, \quad \forall d \in D \tag{4.115}$$

$$\sum_{j:(d,j) \in A, \{i,j\} \in V \cup \{(n+1)\}} (f_{dj}^k - f_{jd}^k) = -v_d^k, \quad \forall d \in D, \quad \forall k \notin P_d \tag{4.116}$$

$$f_{ij}^k \leq M x_{ij}^k, \quad \forall k = 1, 2, \dots, K, \quad \forall (i,j) \in A, \quad \{i,j\} \in V \cup \{(n+1)\} \tag{4.117}$$

$$f_{ij}^k \geq x_{ij}^k, \quad \forall k = 1, 2, \dots, K, \quad \forall (i,j) \in A, \quad \{i,j\} \in V \cup \{(n+1)\} \tag{4.118}$$

$$f_{(n+1)j}^k = 0, \quad \forall j \in V, \quad \forall k = 1, 2, \dots, K \tag{4.119}$$

$$\sum_{j \in V} f_{j(n+1)}^k = 1, \quad \forall k = 1, 2, \dots, K \tag{4.120}$$

$$\sum_{j:(i,j) \in A} x_{ji}^k = v_i^k, \quad \forall k = 1, 2, \dots, K, \quad \forall i \in V \cup \{(n+1)\} \tag{4.121}$$

$$\sum_{k=1}^{K} \sum_{(i,j) \in \delta^+(s)} z_{ij}^k \geq 1, \quad \forall s \subset S \tag{4.122}$$

$$\sum_{k=1}^{K} \sum_{(i,j)\in\delta^-(s)} z_{ij}^k \geq 1, \quad \forall s \subset S \tag{4.123}$$

$$\sum_{k=1}^{K} \sum_{(i,j)\in\delta^+(y)} z_{ij}^k \geq 1, \quad \forall y \subset Y^+ \tag{4.124}$$

$$\sum_{k=1}^{K} \sum_{(i,j)\in\delta^-(y)} z_{ij}^k \geq 1, \quad \forall y \subset Y^- \tag{4.125}$$

$$x_{ij}^k, x_{ji}^k \in \mathbb{Z}_+, \quad \forall k = 1, 2, \ldots, K, \quad \forall (i,j) \in A, \quad \{i,j\} \in V \tag{4.126}$$

$$x_{i(n+1)}^k \in \mathbb{Z}_+, \quad \forall i \in V, \quad \forall k = 1, 2, \ldots, K \tag{4.127}$$

$$v_i^k \in \mathbb{Z}_+, \quad \forall k = 1, 2, \ldots, K, \quad \forall i \in V \tag{4.128}$$

$$z_{ij}^k \in \mathbb{B}, \quad \forall k = 1, 2, \ldots, K, \quad \forall (i,j) \in B \tag{4.129}$$

$$f_{ij}^k, f_{ji}^k \in \mathbb{R}_+, \quad \forall (i,j) \in A, \quad \forall k = 1, 2, \ldots, K \tag{4.130}$$

MIP3 This is a relaxation of *MIP2* which may result in a disconnected graph with walks that may not have the correct timing in terms of unblocking time of the blocked arcs. In this model, we do not include time related and connectivity constraints. Hence variables $t_{ij}^{k,p}$, $w_{ij}^{k,p}$, $u_{ij}^{k,p}$ are not defined. We add only one constraint (which is given in the following paragraphs as constraint (4.131)) to force the vehicles unblock some arcs. In order to show a disconnected and unsynchronized solution, a small example is given in Figure 4.1. In this graph, the solution of relaxed model *MIP3* is disconnected. There are two depots, at node 1 and node 5, each with one vehicle; and there are five nodes each representing a component. Objective value is 6 hours and the walks are $v_1 - v_2 - v_3$ and $v_5 - v_4$ which gives two disconnected subgraphs.

The following constraints (4.131) show that the number of arcs which are unblocked must be at least $2(|Q|-1)$. This is required for the relaxed model *MIP3*, otherwise the objective value is zero due to lack of a constraint forcing the vehicles traversing the arcs.

$$\sum_{k=1}^{K} \sum_{(i,j)\in C} z_{ij}^k \geq 2(|Q|-1) \tag{4.131}$$

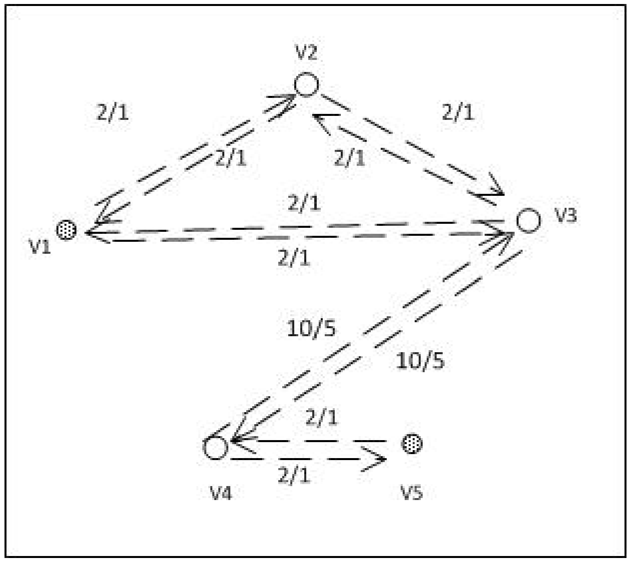

Figure 4.1: An instance which has a disconnected graph in the optimal solution of *MIP3*

Proposition 1. In any optimal solution of the *K-ARCP* when there is more than one vehicle and $|Q|$ disconnected components ($|Q| > 1$), in cutsets of the components, in total, there are at least $2(|Q| - 1)$ arcs which are unblocked. Furthermore, there are instances in which any optimal solution may unblock more than $2(|Q| - 1)$ arcs in the component cutsets.

Proof. The need for at least $2(|Q| - 1)$ arcs follows from strong connectivity. To show more than $2(|Q| - 1)$ arcs may be unblocked, we present the following instance. There are four components, one depot and two vehicles. The graph is shown in Figure 4.2. Dashed arcs show the blocked arcs. d represents the depot vertex, unblocking and traversal times are shown beside the blocked arcs. M is a large value representing a high traversal time on an arc. In the optimal solution, vehicle 1 follows $d - v_5 - v_6 - v_7 - v_8 - v_7 - v_6 - v_5$ and vehicle 2 follows $d - v_1 - v_2 - v_3 - v_4 - v_3$. Arcs $(v_5, v_6), (v_7, v_8), (v_8, v_7), (v_6, v_5), (v_1, v_2), (v_3, v_4), (v_4, v_3)$ are unblocked. If vehicle 2 does not unblock arc (v_1, v_2), in order to connect v_4 to the network, it has to go through the arc with large cost, M. Such a solution would be suboptimal. As a result, the number of arcs which are unblocked is seven and greater than $2(|Q| - 1)$ which is six.

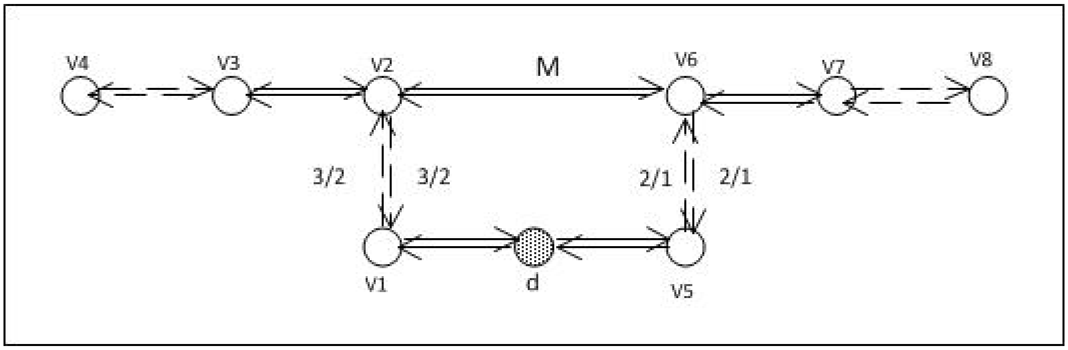

Figure 4.2: Four Components and a Single Depot Case

With this example we prove that the number of arcs which are unblocked can be larger than $2(|Q|-1)$ in a unique optimal solution. As a result of Proposition 1, we cannot write constraints (4.131) as equality.

□

The following mathematical formulation is the relaxed model *MIP3* which finds the routes for vehicles and minimizes the maximum tour length among K vehicles. The resulting graph may be disconnected and the routes are not synchronized.
Relaxed Model of K-ARCP: MIP3

Minimize y

subject to

$$\sum_{(i,j)\in A} c_{ij}x_{ij}^k + \sum_{(i,j)\in B} b_{ij}z_{ij}^k \leq y, \quad \forall k = 1, 2, \ldots, K \tag{4.132}$$

$$\sum_{j\in V\cup\{(n+1)\}} (x_{dj}^k - x_{jd}^k) = 1, \quad \forall d \in D, \quad \forall k \in P_d \tag{4.133}$$

$$\sum_{j\in V\cup\{(n+1)\}} (x_{dj}^k - x_{jd}^k) = 0, \quad \forall d \in D, \quad \forall k \notin P_d \tag{4.134}$$

$$\sum_{j\in V\cup\{(n+1)\}} (x_{ij}^k - x_{ji}^k) = 0, \quad \forall k = 1, 2, \ldots, K, \quad \forall i \in V\backslash D \tag{4.135}$$

$$\sum_{j\in V} x_{j(n+1)}^k = 1, \quad \forall k = 1, 2, \ldots, K \tag{4.136}$$

$$x_{(n+1)i}^k = 0, \quad \forall i \in V, \quad \forall k = 1, 2, \ldots, K \tag{4.137}$$

$$x_{ij}^k \geq z_{ij}^k, \quad \forall k = 1, 2, \ldots, K, \quad \forall (i, j) \in B \tag{4.138}$$

$$x_{ij}^k \leq 2(|Q| - 1) \sum_{k'=1}^{K} z_{ij}^{k'}, \quad \forall k = 1, 2, \ldots, K, \quad \forall (i, j) \in B \tag{4.139}$$

$$\sum_{j \in V} x_{dj}^k \leq \sum_{(i,j) \in C} z_{ij}^k, \quad \forall d \in D, \quad \forall k \in P_d \tag{4.140}$$

$$\sum_{(i,j) \in C} z_{ij}^k \leq 2(|Q| - 1) \sum_{j \in V} x_{dj}^k, \quad \forall d \in D, \quad \forall k \in P_d \tag{4.141}$$

$$\sum_{j:(i,j) \in A, \{i,j\} \in V \cup \{(n+1)\}} (f_{ij}^k - f_{ji}^k) = -v_i^k, \quad \forall k = 1, 2, \ldots, K, \quad \forall i \in V \cup \{(n+1)\} \setminus D \tag{4.142}$$

$$\sum_{j \in V \cup \{(n+1)\}} (f_{dj}^k - f_{jd}^k) = \sum_{i \in V \cup \{(n+1)\} \setminus \{d\}} v_i^k, \quad \forall k \in P_d, \quad \forall d \in D \tag{4.143}$$

$$\sum_{j:(d,j) \in A, \{i,j\} \in V \cup \{(n+1)\}} (f_{dj}^k - f_{jd}^k) = -v_d^k, \quad \forall d \in D, \quad \forall k \notin P_d \tag{4.144}$$

$$f_{ij}^k \leq M x_{ij}^k, \quad \forall k = 1, 2, \ldots, K, \quad \forall (i, j) \in A, \quad \{i, j\} \in V \cup \{(n+1)\} \tag{4.145}$$

$$f_{ij}^k \geq x_{ij}^k, \quad \forall k = 1, 2, \ldots, K, \quad \forall (i, j) \in A, \quad \{i, j\} \in V \cup \{(n+1)\} \tag{4.146}$$

$$f_{(n+1)j}^k = 0, \quad \forall j \in V, \quad \forall k = 1, 2, \ldots, K \tag{4.147}$$

$$\sum_{j \in V} f_{j(n+1)}^k = 1, \quad \forall k = 1, 2, \ldots, K \tag{4.148}$$

$$\sum_{j:(i,j) \in A} x_{ji}^k = v_i^k, \quad \forall k = 1, 2, \ldots, K, \quad \forall i \in V \cup \{(n+1)\} \tag{4.149}$$

$$\sum_{k=1}^{K} \sum_{(i,j) \in C} z_{ij}^k \geq 2(|Q| - 1) \tag{4.150}$$

$$x_{ij}^k, x_{ji}^k \in \mathbb{Z}_+, \quad \forall k = 1, 2, \ldots, K, \quad \forall (i, j) \in A, \quad \{i, j\} \in V \tag{4.151}$$

$$x_{i(n+1)}^k \in \mathbb{Z}_+, \quad \forall i \in V, \quad \forall k = 1, 2, \ldots, K \tag{4.152}$$

$$v_i^k \in \mathbb{Z}_+, \quad \forall k = 1, 2, \ldots, K, \quad \forall i \in V \tag{4.153}$$

$$z_{ij}^k \in \mathbb{B}, \quad \forall k = 1, 2, \ldots, K, \quad \forall (i, j) \in B \tag{4.154}$$

$$f_{ij}^k, f_{ji}^k \in \mathbb{R}_+, \quad \forall (i, j) \in A, \quad \forall k = 1, 2, \ldots, K \tag{4.155}$$

Chapter 5

DATA ACQUISITION AND GENERATION

For computational experiments, we constructed a network of Istanbul which is obtained by considering province centers and real road distances. By using Google Maps, we identified strategically important locations such as province centers, and provinces which have hospitals, disaster coordination centers, ports, airports, bus terminals, bridges. Some of these locations are used also as depots. Possible depot points are given in Table 5.1. Depot points are determined according to the locations related to highway maintenance, the locations which may have machinery e.g. cranes, trucks. There are 74 nodes including 38 province centers, 34 populated districts (See Figure 5.1). In total there are 358 links (716 arcs) (See Figure 5.2). Arcs are created according to the neighborhood relation between nodes. Since the network is for transportation purposes it is crucial to find the real distances between nodes. The road distances can be seen in Appendix A, these distances are symmetric, that is, the travel times are the same for both directions for a pair of nodes. Travel times are determined by using road distances which are calculated by using Google Maps. We assumed on the average 50km/h fixed speed for vehicles and converted road distances to time measure by dividing distance by speed.

	Province	**Node**
Disaster Coordination Center	Kağıthane	23
General Directorate of Highways (GDH) Division of Machinery Supply	Maltepe	29
GDH Division of Road Maintenance and Repair	Kartal	32
GDH Division of Road Maintenance and Repair	Edirnekapı/Eyüp	15
GDH Regional Division of Maintenance and Operations	Kavacık	27
GDH Regional Division of Maintenance and Operations	Kurtköy/Pendik	36
General Directorate of Highways (GDH)	Kağıthane	23
İstanbul Metropolitan Municipality	Fatih	19
İstanbul Metropolitan Municipality-Additional Building	Merter/Güngören	17

Table 5.1: Possible Depot Locations

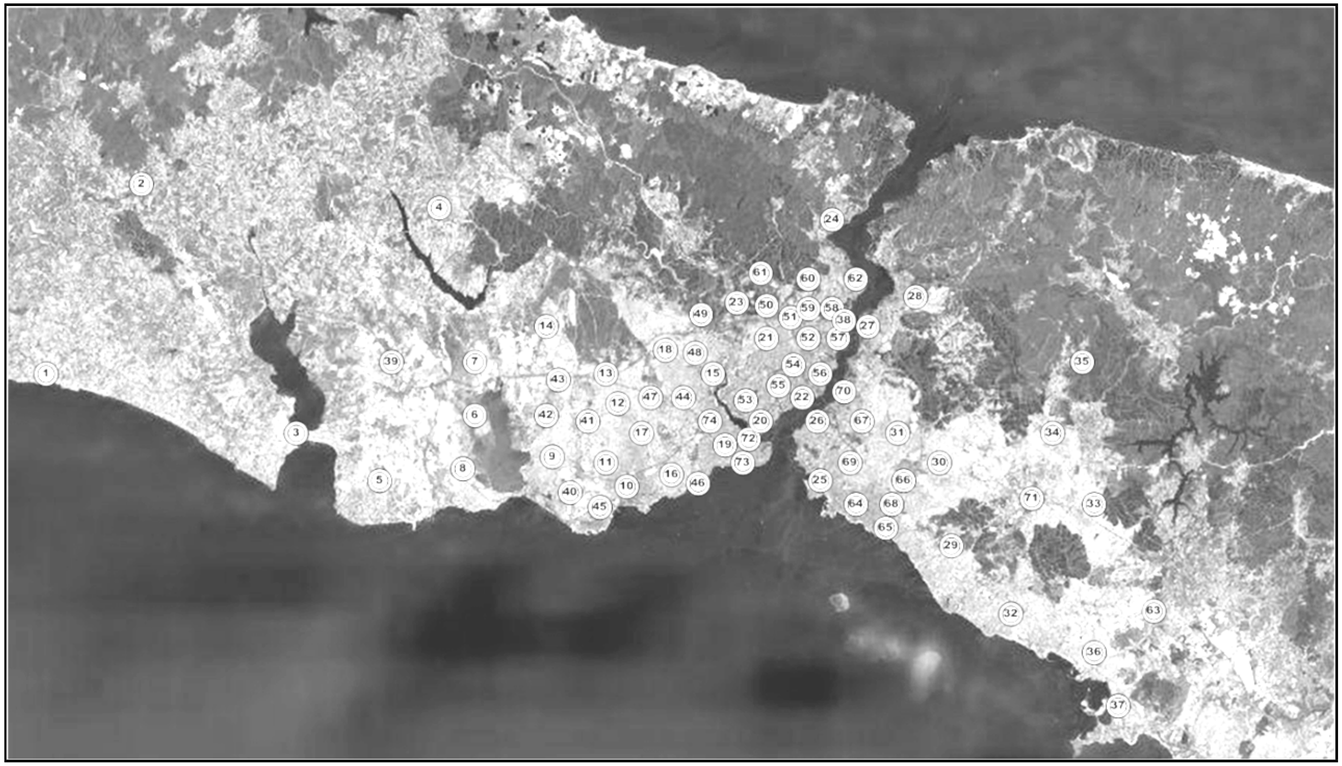

Figure 5.1: Nodes on Istanbul Map from Google Earth

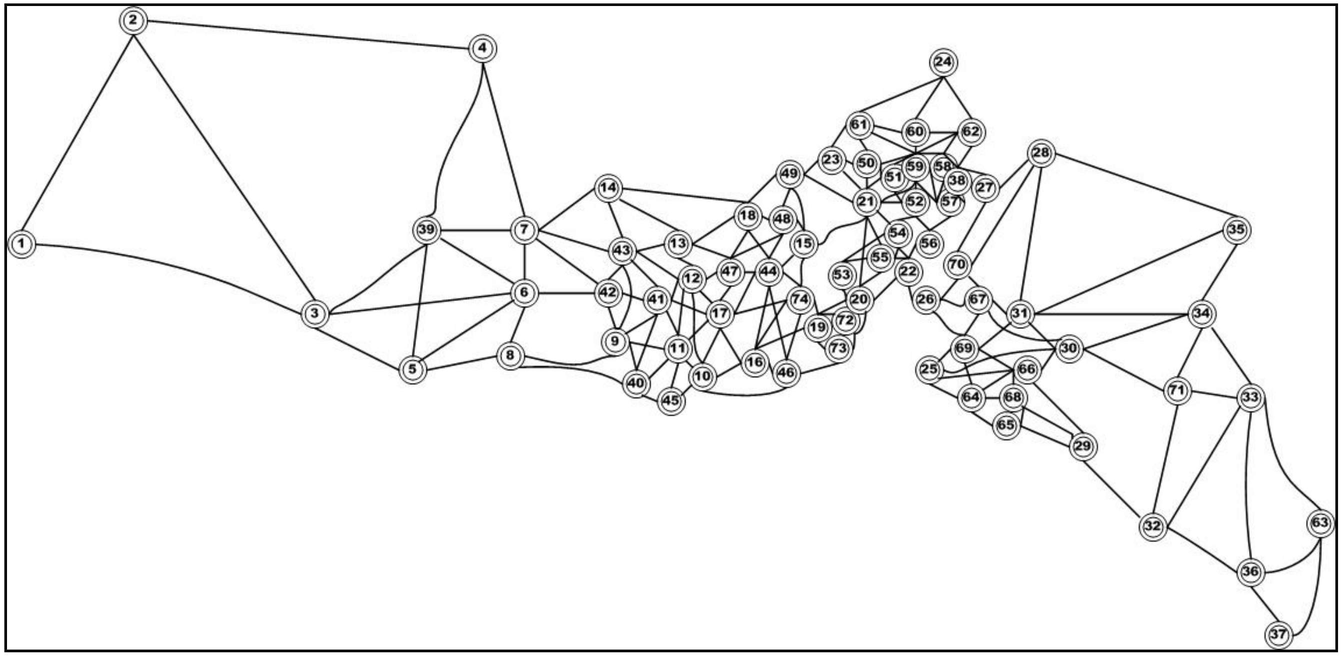

Figure 5.2: The Network Representing the Main Roads of İstanbul

The scenarios that have different sets of blocked roads are generated by referring to the latest earthquake risk map of Istanbul which is reported by The Japan International Cooperation Agency (JICA) and Istanbul Metropolitan Municipality in a study in 2002 [22] (See Figure 5.3). Roads are classified into three, based on the earthquake risk map: High Risk Roads-the bottom part on Figure 5.3 (See Table B.1 in Appendix B for high risk roads), Low Risk Roads-the second block from the bottom on Figure 5.3 (See Table B.2 in Appendix B for low risk roads) and the remaining ones-the remaining two blocks on the top on Figure 5.3.

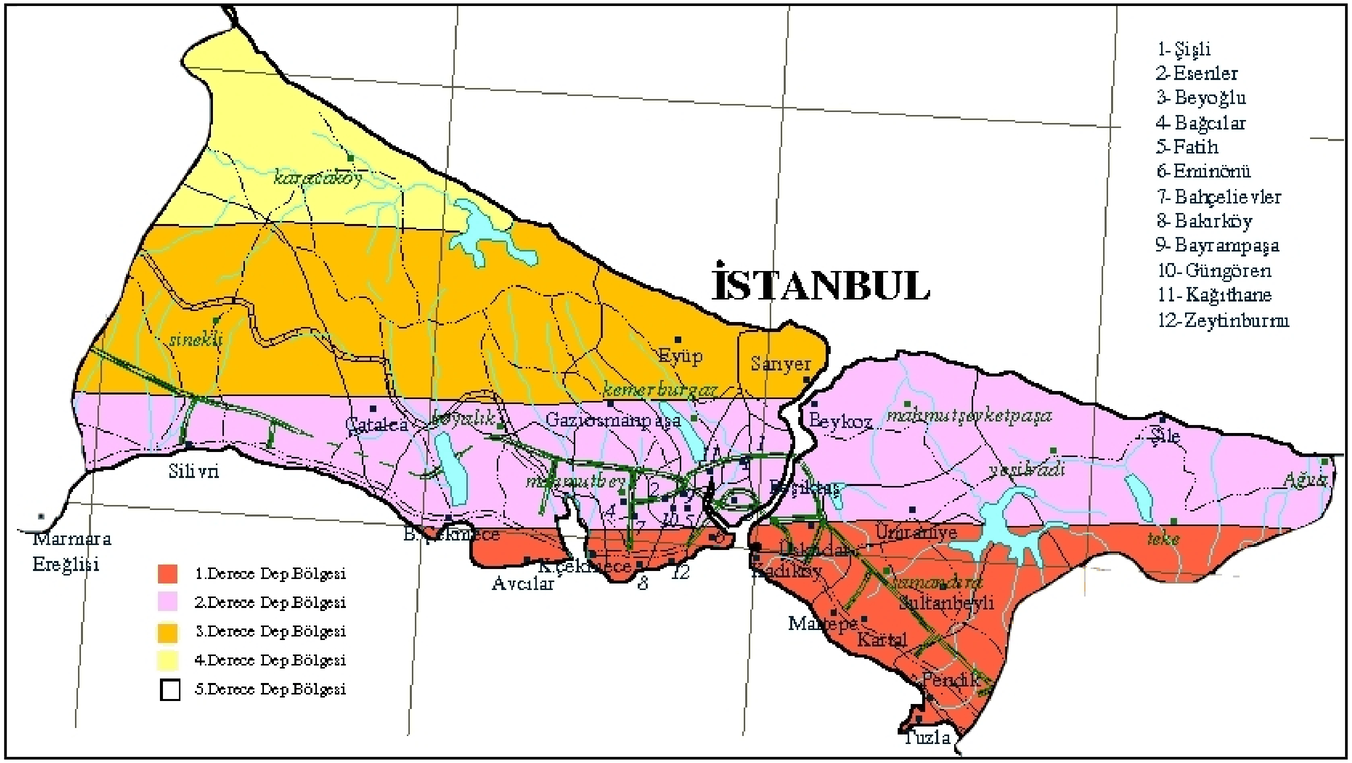

Figure 5.3: Earthquake Risk Map of Istanbul

Ten different scenarios are created on the whole network. More roads are picked to be blocked in high risk area than in low risk area, but within each risk area blocked roads are selected randomly. As a result some disconnected components are formed. In order to see the effect of the number of components on the computational performance, number of disconnected components varies between 3-6. Number of disconnected components and number of blocked roads in each scenario are given in Table 5.2.

Scenarios (S)	**Number of Blocked Arcs** ($\|B\|$)	**Number of Components** ($\|Q\|$)
1	30	3
2	32	3
3	40	4
4	44	3
5	54	4
6	60	4
7	74	5
8	80	6
9	80	6
10	82	5

Table 5.2: Scenarios

After a disaster, every road segment gets affected in a different degree. For each scenario, two time levels are considered here for unblocking time (b_{ij}), which represents a high and low level of damage. All unblocking times are proportional to the length of the arcs, thereby are proportional to traversal time. In high unblocking time case, unblocking operations take longer time than those in low unblocking time case, due to a higher level of damage. For these two cases, with different ratios, blocked roads are classified into three: High damage, medium damage, low damage. In two cases, distribution of these roads differ. A road can be either a high damage, a medium damage or a low damage road. In high unblocking time case, there are more number of high damage roads, and less number of low damage roads.

Similarly, in low unblocking time case, there are more number of low damage roads and less number of high damage roads. For example, in a scenario with high unblocking time input, 60% of the blocked arcs have high damage, while 10% of the blocked arcs have low damage. We set the ratios of roads in each cases as those listed in Table 5.3. Then, for each type of road, a parameter α is defined and it is multiplied by the traversal time. As a result the unblocking times (b_{ij}) are formed. Parameter α which has a uniform distribution, takes values between $(10, 50), (5, 10)$ and $(2, 5)$ depending on the degree of damage on the roads. Value of α is high for the high damage roads, low for the low damage roads and moderate for the remaining. Table 5.3 shows the classification of roads according to damage level and values of α.

	High Unblocking Time			**Low Unblocking Time**		
	High damage	**Medium damage**	**Low damage**	**High damage**	**Medium damage**	**Low damage**
Prob.	0.6	0.3	0.1	0.1	0.3	0.6
α	Uniform (10,50)	Uniform (5,10)	Uniform (2,5)	Uniform (10,50)	Uniform (5,10)	Uniform (2,5)

Table 5.3: Damage Level, Probabilities, α, Classification of Blocked Roads

Chapter 6

COMPUTATIONAL EXPERIMENTS AND RESULTS

MIP1 contains time constraints and connectivity constraints which require enumeration of all subsets of components. Due to size of the data, solving the model *MIP1* takes a long time. Even solving *MIP2* is hard, but since this is a more relaxed model, runtime performance of the model is expected to be better. The main purpose of this study is to find the action routes for vehicles quickly, so that after a disaster the blocked roads can be opened as fast as possible. For this reason, instead of analyzing computational results of *MIP1*, this thesis focuses on the relaxed but computationally faster model *MIP2*.

Moreover, the model for single vehicle case gives feasible solution. There are no waiting time and synchronization issues between the vehicles, thus optimal solution can be obtained by solving single vehicle single depot formulation. Therefore, we investigate the single vehicle case, then relaxations are investigated for multi vehicle case.

In order to evaluate computational performance, we used GAMS 24.0 and a 64-bit, 2 processors, Intel Xeon CPU E5-2643, 3.30 GHz, 32 GB RAM operating system for the experiments. CPLEX 12.5 was employed to solve the models with options threads 0, parallelmode 1 turned on. By this setting, CPLEX is run as a multi-threaded application. One hour (3600 s) of time limit is set.

The objectives of the models in this study are to minimize the total travel time for single vehicle case, to minimize the longest travel time for multi vehicle case and the computational objective is to maximize the lower bound and runtime of the model.

In this thesis, effect of the following issues on computational performance is analyzed for single vehicle case: Degree of Damage, Location of the Depot. Moreover,for multi vehicle case, we analyze the effect of the following issues on computational performance of the models *MIP2* and *MIP3*: Degree of Damage, Number of Vehicles and Number of Depots.

6.1 Single Vehicle Case (ARCP)

For a single vehicle, single depot case, first, effect of damage level; then effect of location of depot on the computational performance are analyzed.

6.1.1 Effect of Degree of Damage (Effect of b_{ij} Values on the Solution)

Two damage levels are used when the depot node is 23, and there is only one vehicle deployed in the depot. The results can be seen in Table 6.1. All scenarios can be solved to optimality in a short time (at most 114 seconds). On average, low unblocking time case gives 23% higher runtime.

			High Unblocking Time			**Low Unblocking Time**		
S	**|B|**	**|Q|**	**Obj. (hr)**	**Gap (%)**	**Time (s)**	**Obj. (hr)**	**Gap (%)**	**Time (s)**
1	30	3	4.46	0	39	2.51	0	30
2	32	3	2.96	0	8	2.4	0	10
3	40	4	8.5	0	8	5.13	0	9
4	44	3	6.4	0	6	3.42	0	9
5	54	4	3.34	0	8	2.66	0	8
6	60	4	6.09	0	13	3.43	0	9
7	74	5	7.3	0	50	5.49	0	114
8	80	6	11.16	0	27	5.93	0	27
9	80	6	9.05	0	23	5.87	0	17
10	82	5	8.2	0	14	3.51	0	21
AVERAGE			**6.75**	**0**	**20**	**4.04**	**0**	**26**

Table 6.1: Effect of Degree of Damage and Computational Results - Single Vehicle Case

6.1.2 Effect of Location of the Depot on the Solution

In order to evaluate the effect of the location of the depot on the computational performance, we picked different nodes as the depot and solved the model with high unblocking time. Nodes 15, 23, 27, 29 and 32 are picked randomly as depots among the potential depots. Nodes 15 and 23 are located in European side of Istanbul, whereas nodes 27, 19 and 32 are in Asian side.

Table 6.2 shows the result for all scenarios. All scenarios are solved to optimality within the time limit. Objective value does not change much as the depot changes. For the single vehicle case, choosing node 29 as the depot seems rational since it gives a better solution in terms of objective value and runtime. The reason of this performance may be due to the location of the depot and layout of the network. The network has a rectangular shape and node 29 resides on bottom-right of it. Node 32 has a similar setting as well. The other depots are in a more central position with respect to layout of the network. Starting from the edges and connecting the components may result in a shorter travel time. However, starting from a central location may result traversals back and forth to the component in the center. Therefore, the travel time may be longer.

6.2 Multiple Vehicle Case (K-ARCP)

6.2.1 Effect of Degree of Damage (Effect of b_{ij} Values on the Solution)

As explained in the previous chapter, two different damage levels are considered: high and low unblocking time. In order to see the effect of damage level on the gap between lower bound and objective values, and on runtime we solve the *MIP2* and *MIP3* on 10 different scenarios. One of the nodes, that is node 23, is picked as the depot according to its strategic importance on road clearance and maintenance operations. There are two vehicles positioned in the depot. For each scenario, both cases with high and low unblocking time are examined. The results can be seen in Tables 6.3 and 6.4. As the number of components and blocked arcs increase, effect of damage level on runtime can be seen more clearly. As the level of damage decreases, solution time increases. On average, lower unblocking time results in a lower objective value, but the decision of arcs to unblock gets more difficult due to low unblocking time and runtime increases. There are instances which cannot be solved without gap by using model *MIP2* within time limit of 3600 s. When we increase the time limit to 21600 s. the gap becomes zero for the instances which cannot be solved within 3600 s. Only one instance (Scenario 9, low unblocking time) has 3% gap even after 21600 s. time limit. On the other hand, the more relaxed model *MIP3* solves all in one hour with no gap.

If we check the solutions found by model *MIP2* with both high and low unblocking time settings, we observe that except five instances out of 20 instances are solved to optimality. The walks of these solutions can be seen in Appendix C. The graphs in these solutions

	Depot ID: 15			Depot ID: 23		
S	**Obj. (hr)**	**Gap (%)**	**Time (s)**	**Obj. (hr)**	**Gap (%)**	**Time (s)**
1	4.3	0	10	4.46	0	39
2	2.8	0	6	2.96	0	8
3	8.59	0	8	8.5	0	8
4	6.22	0	6	6.4	0	6
5	3.15	0	6	3.34	0	8
6	5.91	0	9	6.09	0	13
7	7.16	0	22	7.3	0	50
8	11.08	0	29	11.16	0	27
9	8.86	0	13	9.05	0	23
10	8.15	0	11	8.2	0	14
AVR.	**6.62**	**0**	**12**	**6.75**	**0**	**20**
	Depot ID: 27			**Depot ID: 29**		
S	**Obj. (hr)**	**Gap (%)**	**Time (s)**	**Obj. (hr)**	**Gap (%)**	**Time (s)**
1	4.5	0	15	4.01	0	6
2	3.15	0	9	2.74	0	5
3	8.57	0	8	8.37	0	8
4	7.69	0	12	7.01	0	6
5	3.51	0	8	3.01	0	6
6	6.25	0	15	6.05	0	8
7	7.38	0	32	6.46	0	8
8	11.16	0	39	10.41	0	11
9	9.28	0	13	8.82	0	11
10	8.2	0	14	7.81	0	9
AVR.	**6.97**	**0**	**17**	**6.47**	**0**	**8**
	Depot ID: 32					
S	**Obj. (hr)**	**Gap (%)**	**Time (s)**			
1	4.14	0	7			
2	2.87	0	6			
3	8.5	0	8			
4	7.14	0	6			
5	3.14	0	6			
6	5.92	0	19			
7	6.33	0	9			
8	10.28	0	12			
9	9.01	0	18			
10	8.41	0	11			
AVR.	**6.58**	**0**	**11**			

Table 6.2: Effect of Location of the Depot on the Solution-Single Vehicle Case

are strongly connected and there is no situation which requires waiting time of any vehicle. When we check one of the infeasible solutions, we can see that the infeasible solution can be fixed by adding waiting times but the objective value increases. Some blocked arcs should be unblocked by a vehicle before another vehicle traverses it. In an infeasible solution of *MIP2*, this point is ignored and an arc is traversed by a vehicle although it is not unblocked by another vehicle. We can describe how we fix an infeasible solution on Scenario 4 with high damage level which can be seen below. Equality sign (=) shows the arcs which are unblocked by the corresponding vehicle. The others are the ones which are not blocked in the disaster situation. Both vehicles arrive at node 41 at the same time but the infeasible solutions shows that the arc $(41 - 11)$ is unblocked by Vehicle 1 and Vehicle 2 traverses it without any waiting although the arc is not unblocked yet. Therefore, Vehicle 2 should wait for 0.5 hour until $(41-11)$ is unblocked by Vehicle 1. Similarly, when the traversal, unblocking and waiting times are calculated, it is seen that Vehicle 1 should wait for Vehicle 2 until Vehicle 2 unblocks $(11 - 41)$. Vehicle 1 also should wait at node 6 for Vehicle 2 unblocking $(6 - 5)$. As a result, the objective increases from 3.91 to 5.7 but the solution becomes feasible. The other infeasible solutions can be fixed as well in a similar way. These results will give us an upper bound for the objective value. Feasible solutions can be seen in Appendix C. Note that in Scenario 3, low unblocking time feasible solution does not have any waiting time. By allowing the first arriving vehicle to unblock a blocked arc sometimes eliminates waiting times, as it can be seen in this instance. Moreover, Table 6.5 shows the lower bounds (LB) obtained from *MIP2* for each scenario with a single depot and two vehicles. The upper bound (UB) is the solution of *MIP2* in cases where the gap is zero and in the other cases, a feasible solution is obtained from *MIP2* solution by inserting waiting times. The upper bound refers to these solutions. For these solution the gap between lower bound and upper bound is also given.

Vehicle 1 $23 - 49 - 18 - 13 - 12 - 41 = 11 - 41 - 9 - 8 - 6 - 5 = 3$

Vehicle 2 $23 - 49 - 18 - 13 - 12 - 41 - 11 = 41 - 9 - 8 - 6 = 5$

To compare two relaxed models in high and low unblocking time cases, see Table 6.6 and 6.7. *MIP3* is better in runtime but it may result in disconnected and unsynchronized walks. On average, it is around 40% less than the lower bound which is obtained by solving *MIP2* (41% less for high unblocking time, 45% for low unblocking time). We calculate this value by using the following formula:

			High Unblocking Time				Low Unblocking Time			
S	**|B|**	**|Q|**	**Obj. (hr)**	**LB**	**Gap (%)**	**Time (s)**	**Obj. (hr)**	**LB**	**Gap (%)**	**Time (s)**
1	30	3	2.49	2.49	0	866	1.09	1.09	0	37
2	32	3	1.47	1.47	0	9	1.37	1.37	0	34
3	40	4	4.93	4.93	0	9192	3.01	3.01	0	349
4	44	3	3.91	3.91	0	40	2.39	2.39	0	5
5	54	4	1.71	1.71	0	9	1.5	1.5	0	15
6	60	4	3.42	3.42	0	128	1.82	1.82	0	131
7	74	5	3.46	3.46	0	79	2.72	2.72	0	6673
8	80	6	5.49	5.49	0	590	3.19	3.19	0	16863
9	80	6	4.81	4.81	0	1684	3.42	3.32	2.9	21600
10	82	5	4.25	4.25	0	89	1.93	1.93	0	118
AVERAGE			**3.59**	**3.59**	**0**	**1269**	**2.24**	**2.23**	**0.29**	**4583**

Table 6.3: Effect of Degree of Damage and Computational Results - MIP2 (two vehicles, single depot)

			High Unblocking Time				Low Unblocking Time			
S	**|B|**	**|Q|**	**Obj. (hr)**	**LB**	**Gap (%)**	**Time (s)**	**Obj. (hr)**	**LB**	**Gap (%)**	**Time (s)**
1	30	3	1.64	1.64	0	7	1.07	1.07	0	27
2	32	3	1.37	1.37	0	7	0.91	0.91	0	15
3	40	4	3.37	3.37	0	7	1.67	1.67	0	18
4	44	3	1.52	1.52	0	6	1	1	0	36
5	54	4	1	1	0	7	0.42	0.42	0	7
6	60	4	1.74	1.74	0	9	1.47	1.47	0	35
7	74	5	2.34	2.34	0	16	1.54	1.54	0	62
8	80	6	2.74	2.74	0	9	1.06	1.06	0	9
9	80	6	2.63	2.63	0	57	2.14	2.14	0	19
10	82	5	2.94	2.94	0	63	0.99	0.99	0	6
AVERAGE			**2.13**	**2.13**	**0**	**18.8**	**1.23**	**1.23**	**0**	**24**

Table 6.4: Effect of Degree of Damage and Computational Results - MIP3 (two vehicles, single depot)

S	**Unblocking Time**	**LB**	**UB**	**Gap (%)**
1	**High**	2.49	2.49	0
	Low	1.09	1.09	0
2	**High**	1.47	1.47	0
	Low	1.37	1.37	0
3	**High**	4.93	5.54	12
	Low	3.01	3.86	28
4	**High**	3.91	5.7	46
	Low	2.39	3.61	51
5	**High**	1.71	1.71	0
	Low	1.5	1.5	0
6	**High**	3.42	3.42	0
	Low	1.82	1.82	0
7	**High**	3.46	3.46	0
	Low	2.72	3	10
8	**High**	5.49	5.49	0
	Low	3.19	3.19	0
9	**High**	4.81	4.81	0
	Low	2,9	2,9	0
10	**High**	4.25	4.25	0
	Low	1.93	1.93	0
AVERAGE		**2.89**	**3.13**	**7,39**

Table 6.5: Lower Bound obtained from MIP2 and Upper Bound obtained by making MIP2 solution feasible in Scenarios 3, 4, 7 (two vehicles, single depot)

$$\frac{(\overline{LB}_{MIP2} - \overline{LB}_{MIP3})}{\overline{LB}_{MIP2}} \times 100$$

	MIP2				**MIP3**			
S	**Obj. (hr)**	**LB**	**Gap (%)**	**Time (s)**	**Obj. (hr)**	**LB**	**Gap (%)**	**Time (s)**
1	2.49	2.49	0	866	1.64	1.64	0	7
2	1.47	1.47	0	9	1.37	1.37	0	7
3	4.93	4.93	0	9192	3.37	3.37	0	7
4	3.91	3.91	0	40	1.52	1.52	0	6
5	1.71	1.71	0	9	1	1	0	7
6	3.42	3.42	0	128	1.74	1.74	0	9
7	3.46	3.46	0	79	2.34	2.34	0	16
8	5.49	5.49	0	590	2.74	2.74	0	9
9	4.81	4.81	0	1684	2.63	2.63	0	57
10	4.25	4.25	0	89	2.58	2.58	0	63
AVR.	**3.59**	**3.59**	**0**	**1269**	**2.13**	**2.13**	**0**	**19**

Table 6.6: Comparison of Relaxed Models MIP2 and MIP3 - High Unblocking Time

Since on average, high unblocking time cases result in lower runtime, in order to have solvable cases for the difficult instances, in the upcoming sections we use high unblocking time.

6.2.2 Effect of the Number of Vehicles on Computational Performance

In order to investigate the effect of the number of vehicles (K) on computational performance, we select a distinguished vertex as the depot and increase the number of vehicles which are deployed at this depot one by one. For the relaxed model *MIP2*, as the number of vehicles increases, runtime and gap between lower bound and the objective usually increase. Although there are scenarios which can be solved with zero gap, many of the scenarios cannot reach the zero gap within the time limit of one hour. As the number of vehicles increases, objective value decreases. Doubling the number of vehicles when there is one vehicle, reduces the value of the objective function by approximately 50% at each scenario. However, the improvement in the objective value stops at some point. For some

	MIP2				MIP3			
S	**Obj. (hr)**	**LB**	**Gap (%)**	**Time (s)**	**Obj. (hr)**	**LB**	**Gap (%)**	**Time (s)**
1	1.09	1.09	0	37	1.07	1.07	0	27
2	1.37	1.37	0	34	0.91	0.91	0	15
3	3.01	3.01	0	349	1.67	1.67	0	18
4	2.39	2.39	0	5	1	1	0	36
5	1.5	1.5	0	15	0.42	0.42	0	7
6	1.82	1.82	0	131	1.47	1.47	0	35
7	2.72	2.72	0	6673	1.54	1.54	0	62
8	3.19	3.19	0	16863	1.06	1.06	0	9
9	3.42	3.18	2.9	21600	2.14	2.14	0	19
10	1.93	1.93	0	118	0.99	0.99	0	6
AVR.	**2.24**	**2.23**	**0.29**	**4583**	**1.22**	**1.22**	**0**	**24**

Table 6.7: Comparison of Relaxed Models MIP2 and MIP3 - Low Unblocking Time

scenarios, after $|Q|-1$ vehicles there is no improvement on the objective, whereas for some scenarios improvement stops after $|Q|+1$ vehicles. We can suggest that, minimum $|Q|-1$ vehicles can be utilized for unblocking operations. We can comment that, approximately one vehicle is assigned for connecting one component. Upper limit for this number is $|Q|+1$. Computational results of all scenarios *until the improvement in the objective stops* can be seen in Tables 6.8 and 6.9.

We solve the second relaxed model *MIP3* for the same scenarios as well. Since *MIP3* is a more relaxed model than *MIP2*, the average values for runtime, lower bound and gap are smaller. Similar to the result of *MIP2*, minimum $|Q|-1$ vehicles can be sufficient. Upper limit for the number of vehicles for model *MIP3* is Q. After Q vehicles, for some scenarios, there is no improvement in the objective value. The results can be seen in Tables 6.10 and 6.11.

6.2.3 Effect of the Number of Depots on Computational Performance

In this part, we investigate the effect of the number of depots ($|D|$) on the computational performance. For each scenario, we start with one depot with one vehicle, then increase the number of depots till $|D|=|Q|$. For multi-depot cases, one vehicle is deployed in each depot. In one-depot case, the depot is located in Europe side of Istanbul. Additional depot

S	|Q|	K	Obj. (hr)	LB	Gap (%)	Time (s)
1	**3**	**1**	4.46	4.46	0	39
		2	2.49	2.49	0	866
		3	1.69	1.69	0	449
		4	1.69	1.39	17.75	3600
2	**3**	**1**	2.96	2.96	0	8
		2	1.47	1.47	0	9
		3	1.37	1.37	0	921
		4	1.05	1.05	0	119
		5	1.05	0.98	6.29	3600
3	**4**	**1**	8.5	8.5	0	8
		2	4.93	4.83	1.97	3600
		3	3.78	3.78	0	828
		4	3.16	2.46	22.15	3600
		5	2.68	1.87	30.15	3600
		6	2.68	1.54	42.54	3600
4	**3**	**1**	6.4	6.4	0	6
		2	3.91	3.91	0	40
		3	3.29	3.29	0	290
		4	3.29	2.44	25.74	3600
5	**4**	**1**	3.34	3.34	0	8
		2	1.71	1.71	0	9
		3	1.49	1.49	0	514
		4	1.28	1.28	0	206
		5	1.28	1.19	7.27	3600

Table 6.8: Effect of the Number of Vehicles on Computational Performance - MIP2 (cont'd)

S	|Q|	K	Obj. (hr)	LB	Gap (%)	Time (s)
6	**4**	**1**	6.09	6.09	0	13
		2	3.42	3.42	0	128
		3	2.28	2.28	0	909
		4	2.22	2.12	4.5	3600
		5	2.22	1.3	41.44	3600
7	**5**	**1**	7.3	7.3	0	50
		2	3.46	3.46	0	79
		3	2.62	2.62	0	3385
		4	2.34	1.93	17.73	3600
		5	2.11	1.67	20.66	3600
		6	1.99	1.99	0	548
		7	1.99	0.96	51.78	3600
8	**6**	**1**	11.16	11.16	0	27
		2	5.49	5.49	0	590
		3	4.07	3.66	10.18	3600
		4	3.38	2.59	23.51	3600
		5	2.75	2.12	22.89	3600
		6	2.56	2.32	9.38	3600
		7	2.42	1.35	44.21	3600
		8	2.42	1.38	43.03	3600
9	**6**	**1**	9.05	9.05	0	23
		2	4.81	4.81	0	1684
		3	3.61	3.16	12.4	3600
		4	3.04	2.23	26.49	3600
		5	2.41	2.26	6.22	3600
		6	2.41	1.48	38.65	3600
10	**5**	**1**	8.2	8.2	0	14
		2	4.25	4.25	0	89
		3	3.25	2.91	10.46	3600
		4	2.54	2.24	11.64	3600
		5	2.31	2.31	0	2540
		6	2.31	2.31	0	1572
AVERAGE			**3.4**	**3.15**	**9.8**	**1893**

Table 6.9: Effect of the Number of Vehicles on Computational Performance - MIP2

S	|Q|	K	Obj. (hr)	LB	Gap (%)	Time (s)
1	3	1	2.9	2.9	0	5
		2	1.64	1.64	0	7
		3	1.64	1.64	0	238
2	3	1	2.6	2.6	0	5
		2	1.37	1.37	0	7
		3	1.37	1.22	11.08	3600
3	4	1	8.39	8.39	0	24
		2	3.37	3.37	0	7
		3	1.7	1.7	0	7
		4	1.7	1.7	0	14
4	3	1	3.5	3.5	0	5
		2	1.52	1.52	0	6
		3	1.52	1.48	2.63	3600
5	4	1	2.14	2.14	0	6
		2	1	1	0	7
		3	0.74	0.74	0	9
		4	0.72	0.72	0	1699
		5	0.72	0.51	28.77	3600

Table 6.10: Effect of the Number of Vehicles on Computational Performance - MIP3 (cont'd)

S	|Q|	K	Obj. (hr)	LB	Gap (%)	Time (s)
6	**4**	**1**	4.32	4.32	0	5
		2	1.74	1.74	0	9
		3	1	1	0	15
		4	1	0.96	4	3600
7	**5**	**1**	5.01	5.01	0	11
		2	2.34	2.34	0	16
		3	1.84	1.73	5.97	3600
		4	1.19	1.19	0	158
		5	1.19	0.87	26.74	3600
8	**6**	**1**	5.68	5.68	0	7
		2	2.74	2.74	0	9
		3	2.2	1.99	10.04	3600
		4	1.66	1.58	5.07	3600
		5	1.18	1.18	0	98
		6	1.18	0.9	23.31	3600
9	**6**	**1**	5.13	5.13	0	5
		2	2.63	2.63	0	57
		3	2.06	1.74	15.69	3600
		4	1.55	1.28	17.33	3600
		5	1.18	1.01	14.58	3600
		6	1.01	0.77	23.53	3600
		7	1.01	0.68	32.81	3600
10	**5**	**1**	6.22	6.22	0	8
		2	2.94	2.94	0	63
		3	2.18	2.18	0	3281
		4	1.14	1.14	0	19
		5	1.14	1.14	0	276
AVERAGE			**2.24**	**2.18**	**4.92**	**1256**

Table 6.11: Effect of the Number of Vehicles on Computational Performance - MIP3

in the two-depot case is located in the Asian side of Istanbul. As we increase the number of depots, we pick them from different sides of the city in order to reduce the effect of the depots' locations. For example, for the four depot case, we pick two depots from the Asian and the European sides. Moreover, in each scenario we try to choose the depots from those listed in Table 5.1 to be located in different components. Table 6.12 shows the depots which are used in each scenario.

S	**1 depot**	**2 depots**	**3 depots**	**4 depots**	**5 depots**	**6 depots**
1	23	15,32	15,32,23	-	-	-
2	23	15,32	15,32,23	-	-	-
3	23	15,32	15,32,17	15,32,17,36	-	-
4	23	15,32	15,32,23	-	-	-
5	23	15,32	15,32,17	15,32,17,36	-	-
6	23	15,32	15,32,19	15,32,19,36	-	-
7	23	15,32	15,32,23	15,32,23,27	15,32,23,27,36	-
8	23	15,32	15,32,17	15,32,17,36	15,32,17,36,19	15,32,17,36,19,23
9	23	15,32	15,32,17	15,32,17,36	15,32,17,36,19	15,32,17,36,19,23
10	23	15,32	15,32,19	15,32,19,29	15,32,19,29,23	-

Table 6.12: Depot Nodes Which Are Used In Each Scenario

If we solve the relaxed model *MIP2* for all scenarios, we see that objective value decreases as the number of depots increases. At the same time, usually, the gap between lower bound and the objective value and run time increases. However, if we analyze in terms of runtime, we observe that less number of depots does not always give better runtime. Location of depots which are utilized is also important. A reasonable approach for selecting depots can be choosing the ones which has more blocked edges or disconnected components around. Table 6.13 shows, for each scenario, the effect of the number of depots on computational performance.

When we solve *MIP3*, we observe a similar trend in the results. Objective value decreases as the number of depots increases. Since it is a more relaxed model than *MIP2* and easier to solve, we do not observe any gap between lower bound and the objective value except some scenarios. Moreover, on the average, *MIP3* gives lower values for the measures than *MIP2*. Table 6.14 shows the results for *MIP3*.

Furthermore, if we compare effect of the number of vehicles and effect of the number of

S	|Q|	|D|	Obj.(hr)	LB	Gap(%)	Time(s)
1	**3**	**1**	4.46	4.46	0	39
		2	2.32	2.32	0	20
		3	1.53	1.53	0	37
2	**3**	**1**	2.96	2.96	0	8
		2	1.31	1.31	0	7
		3	0.98	0.98	0	8
3	**4**	**1**	8.5	8.5	0	8
		2	4.89	4.89	0	135
		3	3.49	3.49	0	199
		4	3.14	3.06	2.51	3600
4	**3**	**1**	6.4	6.4	0	6
		2	4.06	4.06	0	33
		3	3.22	3.13	2.8	3600
5	**4**	**1**	3.34	3.34	0	8
		2	1.43	1.43	0	8
		3	1.37	1.37	0	29
		4	0.89	0.89	0	8
6	**4**	**1**	6.09	6.09	0	13
		2	2.96	2.96	0	9
		3	2.11	2.11	0	24
		4	1.76	1.66	5.57	3600
7	**5**	**1**	7.3	7.3	0	50
		2	3.12	3.12	0	15
		3	2.49	2.38	4.28	3600
		4	1.97	1.61	18.22	3600
		5	1.68	1.44	14.29	3600
8	**6**	**1**	11.16	11.16	0	27
		2	5.27	5.27	0	39
		3	3.65	3.65	0	410
		4	3.04	2.73	10.33	3600
		5	2.57	2.17	15.46	3600
		6	2.09	1.84	11.9	3600
9	**6**	**1**	9.05	9.05	0	23
		2	4.44	4.44	0	141
		3	3.37	3.26	3.32	3600
		4	2.35	2.35	0	232
		5	2.16	1.91	11.54	3600
		6	2.12	1.58	25.47	3600
10	**5**	**1**	8.2	8.2	0	14
		2	4.63	4.63	0	255
		3	3.2	3.2	0	1365
		4	2.56	2.41	5.9	3600
		5	2.11	2.11	0	242
AVERAGE			**3.62**	**3.55**	**3.06**	**1168**

Table 6.13: Effect of the Number of Depots on Computational Performance($K = |D|$, one vehicle at each depot) - MIP2

S	|Q|	|D|	Obj. (hr)	LB	Gap (%)	Time (s)
1	**3**	**1**	2.9	2.9	0	5
		2	1.73	1.73	0	11
		3	1.32	1.32	0	9
2	**3**	**1**	2.6	2.6	0	5
		2	1.31	1.31	0	6
		3	0.98	0.98	0	17
3	**4**	**1**	8.39	8.39	0	24
		2	3.46	3.46	0	9
		3	1.92	1.92	0	6
		4	1.79	1.79	0	88
4	**3**	**1**	3.5	3.5	0	5
		2	2.17	2.17	0	22
		3	1.52	1.52	0	34
5	**4**	**1**	2.14	2.14	0	6
		2	1.37	1.37	0	11
		3	1	1	0	23
		4	0.86	0.86	0	26
6	**4**	**1**	4.32	4.32	0	5
		2	2.1	2.1	0	26
		3	1.56	1.56	0	35
		4	1.56	1.56	0	3251
7	**5**	**1**	5.01	5.01	0	11
		2	2.68	2.68	0	13
		3	1.84	1.84	0	71
		4	1.44	1.44	0	681
		5	1.38	1.18	14.49	3600
8	**6**	**1**	5.68	5.68	0	7
		2	3.05	3.05	0	15
		3	2.09	2.09	0	26
		4	1.94	1.94	0	156
		5	1.5	1.49	0.4	3600
		6	1.18	1.18	0	182
9	**6**	**1**	5.13	5.13	0	5
		2	2.97	2.97	0	316
		3	1.92	1.92	0	511
		4	1.84	1.84	0	3064
		5	1.37	1.37	0	546
		6	1.08	1.08	0	2533
10	**5**	**1**	6.22	6.22	0	8
		2	3.55	3.55	0	1729
		3	2.09	2.09	0	993
		4	1.98	1.98	0	2041
		5	1.69	1.69	0	2456
AVERAGE			**2.47**	**2.46**	**0.35**	**610**

Table 6.14: Effect of the Number of Depots on Computational Performance($K = |D|$, one vehicle at each depot) – MIP3

depots on the relaxed model *MIP3*, we see that utilizing more vehicles at a single depot is more difficult to solve than using more depots each with a single vehicle. Consider Scenario 9 with 6 components and 80 initially blocked edges. If we solve *MIP3* with single depot and increase the number of vehicles, 5 of 7 instances cannot be solved to optimality within the time limit. When there are multiple depots, only one of 6 instances cannot be solved in one hour. These results can be seen in Table 6.15 and Table 6.16.

S	**|Q|**	**K**	**Obj. (hr)**	**LB**	**Gap (%)**	**Time (s)**
		1	5.13	5.13	0	5
		2	2.63	2.63	0	57
		3	2.06	1.74	15.69	3600
9	**6**	**4**	1.55	1.28	17.33	3600
		5	1.18	1.01	14.58	3600
		6	1.01	0.77	23.53	3600
		7	1.01	0.68	32.81	3600

Table 6.15: Effect of the Number of Vehicles on Computational Performance - MIP3, Scenario 9

S	**|Q|**	**|D|**	**Obj. (hr)**	**Gap (%)**	**Time (s)**
		1	5.13	0	5
		2	2.97	0	316
		3	1.92	0	511
9	**6**	**4**	1.84	0	3064
		5	1.37	0	546
		6	1.08	0	2533

Table 6.16: Effect of the Number of Depots on Computational Performance($K = |D|$, one vehicle at each depot) - MIP3, Scenario 9

Moreover, for a further analysis, we consider *MIP2* for Scenario 9 with different number of vehicles and depots and increase the time limit from 3600 s. (1 hour) to 21600 s. (6 hours) for the instances which cannot be solved without gap in one hour. Table 6.17 and Table 6.18 show the results. Among these instances, only five out of 13 instances give a feasible solution. As the number of vehicles and depots increase solutions become infeasible due to the blocked arcs which are traversed by more than one vehicle. The model *MIP2* cannot make the walks synchronized with the correct timing. If we consider the gap between lower bound and the objective, although gap values decrease, some instances still cannot be solved with no gap within six hours. For example, for a larger number of vehicles, increasing the time limit to 21600 s. is not enough to solve the model without any gap. Those instances with 3, 4, 5 and 6 vehicles still have gap. For a larger number of depots, increasing the time limit works for zero gap. Only one instance which has six depots cannot be solved in 21600 s.

S	**|Q|**	**K**	**Obj. (hr)**	**LB**	**Gap (%)**	**Time (s)**
9	**6**	**1**	9.05	9.05	0	23
		2	4.81	4.81	0	1684
		3	3.61	3.16	9.4	21600
		4	3.04	2.23	23.1	21600
		5	2.41	2.26	6.22	21600
		6	2.41	1.48	35.41	21600

Table 6.17: Effect of the Number of Vehicles on Computational Performance - MIP2, Scenario 9

S	**|Q|**	**K**	**Obj. (hr)**	**LB**	**Gap (%)**	**Time (s)**
9	**6**	**1**	9.05	9.05	0	23
		2	4.44	4.44	0	141
		3	3.37	3.37	0	8348
		4	2.35	2.35	0	232
		5	2.16	2.16	0	14692
		6	2.12	1.96	7.68	21600

Table 6.18: Effect of the Number of Depots on Computational Performance($K = |D|$, one vehicle at each depot) - MIP2, Scenario 9

Chapter 7

CONCLUSIONS

In this thesis, we study the *Arc Routing for Connectivity Problem* which is new to the Arc Routing Literature. Considering a post-disaster situation, we use a Min-Max type of objective with multiple vehicles and depots. As a special case, we investigate the single vehicle single depot case. Unblocking some of the blocked roads by vehicles, the aim is to make the disconnected graph strongly connected in shortest time. The vehicles leave their depots and unblock some roads with an unblocking time which is spent only for the first time the blocked road is traversed. We show that the problem is NP-hard even for single vehicle-single depot case. We develop an *MIP* formulation and solve it for the single vehicle case in computational tests. The arc routing part of the problem is handled by sending flows. To the best of our knowledge, this is new in the arc routing literature. For the multi-vehicle cases, due to the difficulty of solving the model on large instances, two relaxed models are presented and are solved with various number of vehicles and depots.

As the testing data, Istanbul Highway Map is used. Ten different scenarios are constructed. Two levels of damage is defined and unblocking times are calculated accordingly. Some illustrative numerical examples are presented. For the valid and relaxed models, we observe that high unblocking time cases are easier to solve than low unblocking time cases. For a special case with a single vehicle, effect of location of depot on the objective value, the gap and runtime is investigated. Changing the location of depot does not affect the objective value and runtime much. For the relaxed models, the effects of number of vehicles and number of depots on solution quality are observed. More vehicles mean longer runtime and improvement on the objective value is limited by the number of vehicles. Desired number of vehicles for a low objective value is between $|Q|-1$ and $|Q|+1$ for *MIP2*, between $|Q|-1$ and $|Q|$ for *MIP3*. Another observation of this study is that increasing the number of depots always improves the objective value, however it usually worsens the runtime. Moreover, we see that assigning more vehicles to a single depot is more difficult to solve than assigning a single vehicle to multiple depots.

This line of research can be extended in several directions in future research. Undirected version of this problem can be analyzed. Moreover, the model *MIP1* can be improved in order to make it a more efficient model. A heuristic method which uses the result of *MIP2* and transforms it to synchronized walks can be developed. However, when the number of vehicles increases *MIP2* (even *MIP3*) cannot be solved to optimality. Another way to solve *K-ARCP* can be a 2-phase method which first solves the *Minimum Spanning Tree Problem* on the induced graph which has the components represented by single nodes. By doing so, the arcs which have to be unblocked in cutsets of components can be identified. Then, on the induced graph, *Min-Max RPP* where the set R consists of the arcs in the solution of Minimum Spanning Tree Problem can be solved. Moreover, partitioning the graph and solving the *K-ARCP* with one vehicle for each partition can be another extension of this thesis, because it can be solved in very short time. Note that the partitioning should be done carefully. Moreover, we assumed that the vehicles are homogeneous, but in some cases, vehicles can be non-homogeneous, i.e. different types of damage may require a different method to clear the road. In such a case, for each vehicle, a set of blocked arcs it can unblock can be defined. In the formulation, this can be handled with an easy modification.

BIBLIOGRAPHY

[1] D. Ahr. *Contributions to Multiple Postmen Problems.* PhD thesis, University of Heidelberg, 2004.

[2] D. Ahr and G. Reinelt. A tabu search for the min-max k-Chinese postman problem. *Computers and Operations Research*, 33:3403–3422, 2006.

[3] A.S. Alfa and D.Q. Liu. Postman routing problem in a hierarchical network. *Engineering Optimization*, 14:127–138, 1988.

[4] E.M. Arkin, R. Hassin, and A. Levin. Approximations for minimum and min-max vehicle routing problems. *Journal of Algorithms*, 59:1–18, 2006.

[5] A. Assad, W. Pearn, and B. Golden. The capacitated Chinese postman problem: Lower bounds and solvable cases. *American Journal of Mathematical and Management Science*, 7:63–88, 1987.

[6] E. Benavent, A. Corberán, I. Plana, and J.M. Sanchis. Min-max k-vehicles windy rural postman problem. *Networks*, 54:216–226, 2009.

[7] P. Brucker. The Chinese postman problem for mixed graphs. *Graph Theoretic Concepts in Computer Science*, 100:354–366, 1981.

[8] I.K. Busch. *Vehicle Routing on Acyclic Networks.* PhD thesis, Johns Hopkins University, 1991.

[9] A. Campbell, T. Lowe, and L. Zhang. Upgrading arcs to minimize the maximum travel time in a network. *Networks*, 47:72–80, 2006.

[10] S.E. Chang and N. Nojima. Measuring post-disaster transportation system performance: the 1995 kobe earthquake in comparative perspective. *Transportation Research*, 35:475–494, 2001.

[11] W.Y. Chen and G.H. Tzeng. A fuzzy multi-objective model for reconstructing post-earthquake road-network by genetic algorithm. *International Journal of Fuzzy Systems*, 1:85–95, 1999.

[12] I. Demgensky, H. Noltemeier, and H.-C Wirth. On the flow cost lowering problem. *European Journal of Operational Research*, 137:265–271, 2002.

[13] M. Dror, H. Stern, and P. Trudeau. Postman tour on a graph with precedence relation on arcs. *Networks*, 17:283–294, 1987.

[14] P.M. Duque and K. Sörensen. A GRASP metaheuristic to improve accessibility after a disaster. *OR Spectrum*, 33:525–542, 2011.

[15] J. Edmonds and E.L. Johnson. Matching, Euler tours and the Chinese postman problems. *Mathematical Programming*, 5:88–124, 1973.

[16] G.N. Frederickson. Approximation algorithms for some postman problems. *Journal of ACM*, 26:538–554, 1979.

[17] G.N. Frederickson, M.S. Hecht, and C.E. Kim. Approximation algorithms for some postman problems. *SIAM Journal of Computing*, 7:178–193, 1978.

[18] G. Ghiani, D. Laganà, and R. Musmanno. A constructive heuristic for the undirected rural postman problem. *Computers and Operations Research*, 33:3450–3457, 2006.

[19] B.L. Golden and R.T. Wong. Capacitated arc routing problems. *Networks*, 11:305–315, 1981.

[20] G.V. Groves and J.H. van Vuuren. Efficient heuristics for the rural postman problem. *ORiON*, 21:33–51, 2005.

[21] K. Holmberg. Heuristics for the rural postman problem. *Computers and Operations Research*, 37:981–990, 2010.

[22] The Japan International Cooperation Agency (JICA) and Istanbul Metropolitan Municipality. The study on a disaster prevention / mitigation basic plan in Istanbul including microzonation in The Republic of Turkey. Technical report, Istanbul Metropolitan Municipality, September 2002.

[23] M. Krishnamurthi and P. Damodaran. A modified postman tour heuristic for efficient snow removal planning. In *Proceedings of 7^{th} Industrial Engineering Research Conference*, 1998.

[24] J.K. Lenstra and A.H.G. Rinnooy Kan. On general routing problems. *Networks*, 6: 273–280, 1976.

[25] F.J.Z. Martinez. *Postman Problems on Mixed Graphs*. PhD thesis, University of Waterloo, 2003.

[26] E. Minieka. The Chinese postman problem for mixed networks. *Management Science*, 25:643–648, 1979.

[27] P.C. Nolz, F. Semet, and K.F. Doerner. Risk approaches for delivering disaster relief supplies. *OR Spectrum*, 33:543–569, 2011.

[28] C.H. Papadimitriou. On the complexity of edge traversing. *Journal of ACM*, 23: 544–554, 1976.

[29] W.L. Pearn. Solvable cases of the k-person Chinese postman problem. *Operations Research Letters*, 16:241–244, 1994.

[30] N. Perrier, A. Langevin, and J.F. Campbell. A survey of models and algorithms for winter road maintenance. part i: System design for spreading and plowing. *Computers and Operations Research*, 33:209–238, 2006.

[31] N. Perrier, A. Langevin, and J.F. Campbell. A survey of models and algorithms for winter road maintenance. part ii: System design for snow disposal. *Computers and Operations Research*, 33:239–262, 2006.

[32] N. Perrier, A. Langevin, and J.F. Campbell. A survey of models and algorithms for winter road maintenance. part iii: Vehicle routing and depot location for spreading. *Computers and Operations Research*, 34:211–257, 2007.

[33] N. Perrier, A. Langevin, and J.F. Campbell. A survey of models and algorithms for winter road maintenance. part iv: Vehicle routing and fleet sizing for plowing and snow disposal. *Computers and Operations Research*, 34:258–294, 2007.

[34] B. Raghavachari and J. Veerasamy. Approximation algorithms for asymmetric postman problem. *Proceedings of SODA*, 1999.

[35] M. C. Scaparra and R. L. Church. A GRASP and path relinking heuristic for rural network development. *Journal of Heuristic*, 11:89–108, 2005.

[36] M. Tagmouti, M. Gendreau, and J.-Y. Potvin. Arc routing problems with time-dependent service costs. *European Journal of Operational Research*, 181:30–39, 2007.

[37] K. Vishwanath and S. Peeta. Multicommodity maximal covering network design problem for planning critical routes for earthquake response. *Transportation Research Record: Journal of the Transportation Research Board*, 1857:1–10, 2003.

[38] Z. Win. *Contributions to Routing Problems.* PhD thesis, University of Augsburg, 1987.

[39] S. Yan and Y.-L. Shih. Optimal scheduling of emergency roadway repair and subsequent relief distribution. *Computers and Operations Research*, 36:2049–2065, 2009.

Appendix A

ROAD DISTANCES

Table A.1: Road Distances

Origin Node	*Destination Node*	*Distance (km)*
1	2	39.0
1	3	39.0
2	1	39.0
2	3	18.0
2	4	45.0
3	1	39.0
3	2	18.0
3	5	12.0
3	6	12.0
3	39	11.0
4	2	45.0
4	7	25.0
4	39	30.0
5	3	12.0
5	6	9.5
5	8	11.0
5	39	9.5
6	3	12.0
6	5	9.5
6	7	5.5
6	8	10.0
6	39	4.0
6	42	18.0

Origin Node	*Destination Node*	*Distance (km)*
7	4	25.0
7	6	5.5
7	14	26.0
7	39	7.0
7	42	10.0
7	43	19.0
8	5	11.0
8	6	10.0
8	9	9.0
8	40	12.0
9	8	9.0
9	11	6.0
9	40	8.0
9	41	3.0
9	42	14.0
9	43	11.0
10	11	5.0
10	12	7.0
10	16	3.5
10	17	5.0
10	45	3.5
10	46	4.5
11	9	6.0
11	10	5.0
11	12	6.5
11	17	5.0
11	40	8.0
11	41	5.0
11	45	3.5
12	10	7.0
12	11	6.5
12	13	3.0
12	17	4.5
12	41	5.0
12	43	10.0
12	47	12.0
13	12	3.0
13	14	11.0
13	18	8.5

Origin Node	*Destination Node*	*Distance (km)*
13	43	11.0
13	47	8.0
14	7	26.0
14	13	11.0
14	18	10.0
14	43	13.0
15	18	2.5
15	21	7.5
15	44	3.0
15	48	1.6
15	49	4.5
15	74	2.0
16	10	3.5
16	17	4.5
16	19	6.5
16	44	9.0
16	46	1.9
16	74	7.5
17	10	5.0
17	11	5.0
17	12	4.5
17	16	4.5
17	41	11.0
17	44	4.5
17	47	7.0
17	74	7.0
18	13	8.5
18	14	10.0
18	15	2.5
18	44	3.0
18	47	8.0
18	48	1.9
18	49	4.5
19	16	6.5
19	20	3.0
19	72	3.0
19	73	3.0
19	74	3.0
20	19	3.0

Origin Node	*Destination Node*	*Distance (km)*
20	21	5.0
20	22	4.0
20	53	1.8
20	55	1.6
20	72	1.8
21	15	7.5
21	20	5.0
21	23	3.0
21	49	5.5
21	50	2.0
21	51	2.5
21	52	2.5
21	54	2.5
21	55	3.5
21	59	3.5
22	20	4.0
22	26	9.5
22	54	2.0
22	55	4.0
22	56	3.0
23	21	3.0
23	49	5.0
23	50	1.2
23	61	5.5
24	60	8.0
24	61	11.0
24	62	10.0
25	30	7.5
25	64	6.5
25	66	10.0
25	68	10.0
25	69	3.5
26	22	9.5
26	67	3.0
26	69	5.0
26	70	2.5
27	28	6.5
27	38	6.5
27	70	10.0

Origin Node	*Destination Node*	*Distance (km)*
28	27	6.5
28	31	17.0
28	35	21.0
28	70	17.0
29	32	6.5
29	65	5.0
29	66	8.0
29	68	6.5
30	25	7.5
30	31	5.0
30	34	28.0
30	66	4.0
30	67	5.5
30	71	15.0
31	28	17.0
31	30	5.0
31	34	16.0
31	35	9.5
31	67	4.5
31	69	5.0
32	29	6.5
32	33	16.0
32	36	4.5
32	71	13.0
33	32	16.0
33	34	5.0
33	36	13.0
33	63	7.5
33	71	5.0
34	30	28.0
34	31	16.0
34	33	5.0
34	35	7.0
34	71	5.0
35	28	21.0
35	31	9.5
35	34	7.0
36	32	4.5
36	33	13.0

Origin Node	*Destination Node*	*Distance (km)*
36	37	11.0
36	63	8.0
37	36	11.0
37	63	13.0
38	27	6.5
38	57	0.5
38	58	1.7
38	62	4.0
39	3	11.0
39	4	30.0
39	5	9.5
39	6	4.0
39	7	7.0
40	8	12.0
40	9	8.0
40	11	8.0
40	41	8.5
40	45	4.0
41	9	3.0
41	11	5.0
41	12	5.0
41	17	11.0
41	40	8.5
41	42	8.0
41	43	5.5
42	6	18.0
42	7	10.0
42	9	14.0
42	41	8.0
42	43	2.5
43	7	19.0
43	9	11.0
43	12	10.0
43	13	11.0
43	14	13.0
43	41	5.5
43	42	2.5
44	15	3.0
44	16	9.0

Origin Node	*Destination Node*	*Distance (km)*
44	17	4.5
44	18	3.0
44	46	7.0
44	47	5.0
44	48	2.5
44	74	3.5
45	10	3.5
45	11	3.5
45	40	4.0
46	10	4.5
46	16	1.9
46	44	7.0
46	73	7.0
46	74	7.0
47	12	12.0
47	13	8.0
47	17	7.0
47	18	8.0
47	44	5.0
48	15	1.6
48	18	1.9
48	44	2.5
48	49	4.5
49	15	4.5
49	18	4.5
49	21	5.5
49	23	5.0
49	48	4.5
50	21	2.0
50	23	1.2
50	51	2.5
50	59	3.5
50	61	6.0
51	21	2.5
51	50	2.5
51	52	1.5
51	59	2.5
52	21	2.5
52	51	1.5

Origin Node	*Destination Node*	*Distance (km)*
52	54	3.0
52	56	2.5
52	59	1.9
53	20	1.8
53	54	4.0
53	55	2.5
54	21	2.5
54	22	2.0
54	52	3.0
54	53	4.0
54	55	3.0
55	20	1.6
55	21	3.5
55	22	4.0
55	53	2.5
55	54	3.0
56	22	3.0
56	52	2.5
56	57	3.5
57	38	0.5
57	56	3.5
57	58	2.5
57	59	3.0
58	38	1.7
58	57	2.5
58	59	1.5
58	62	6.0
59	21	3.5
59	50	3.5
59	51	2.5
59	52	1.9
59	57	3.0
59	58	1.5
59	60	6.0
59	61	7.0
59	62	8.0
60	24	8.0
60	59	6.0
60	61	3.0

Origin Node	*Destination Node*	*Distance (km)*
60	62	4.0
61	23	5.5
61	24	11.0
61	50	6.0
61	59	7.0
61	60	3.0
62	24	10.0
62	38	4.0
62	58	6.0
62	59	8.0
62	60	4.0
63	33	7.5
63	36	8.0
63	37	13.0
64	25	6.5
64	65	5.5
64	66	6.0
64	68	4.5
64	69	6.0
65	29	5.0
65	64	5.5
65	68	2.0
66	25	10.0
66	29	8.0
66	30	4.0
66	64	6.0
66	68	3.5
66	69	7.5
67	26	3.0
67	30	5.5
67	31	4.5
67	69	3.5
67	70	2.0
68	29	6.5
68	64	4.5
68	65	2.0
68	66	3.5
69	25	3.5
69	26	5.0

Origin Node	*Destination Node*	*Distance (km)*
69	31	5.0
69	64	6.0
69	66	7.5
69	67	3.5
70	26	2.5
70	27	10.0
70	28	17.0
70	67	2.0
71	30	15.0
71	32	13.0
71	33	5.0
71	34	5.0
72	19	3.0
72	20	1.8
72	73	1.3
73	19	3.0
73	46	7.0
73	72	1.3
74	15	2.0
74	16	7.5
74	17	7.0
74	19	3.0
74	44	3.5
74	46	7.0

Appendix B

HIGH RISK AND LOW RISK ROADS

EUROPE			**ASIA**	
i	**j**		**i**	**j**
3	5		25	30
5	8		25	69
8	9		25	64
8	40		25	66
9	11		29	32
9	40		30	71
10	11		32	33
10	17		32	36
10	16		32	71
10	46		33	36
16	46		33	63
16	19		36	63
16	74		36	37
19	73		37	63
19	72		64	68
19	20		64	65
20	22		64	66
40	45		64	69
40	11		65	68
45	10		65	29
72	20		66	29
72	73		66	30
			66	69
			68	66
			68	29
			71	33

Table B.1: High Risk Roads

EUROPE													**ASIA**	
i	**j**	**i**	**j**	**i**	**j**	**i**	**j**	**i**	**j**	**i**	**j**		**i**	**j**
1	3	13	47	21	50	43	41	52	51	61	24		26	67
1	2	13	18	23	61	43	9	52	56	61	60		26	69
2	4	14	13	23	50	44	16	53	55	61	59		27	28
2	3	14	18	23	21	44	74	53	54	62	24		27	70
3	39	15	74	38	27	44	15	53	20	62	58		28	31
4	7	15	21	38	57	44	48	54	52	62	38		28	35
4	39	15	49	39	7	47	44	54	22	74	19		31	35
6	7	17	16	39	6	47	18	55	22	74	46		31	30
6	42	17	74	39	5	47	48	55	54				31	69
6	5	17	44	41	40	47	12	55	20				31	34
6	8	17	47	41	11	48	15	56	22				34	33
6	3	17	16	41	12	48	49	57	56				34	71
7	42	18	49	41	17	48	18	58	57				34	30
7	43	18	44	41	9	49	23	58	38				35	34
7	14	20	55	42	41	49	21	59	57				67	31
11	17	21	20	42	43	50	51	59	60				67	30
12	11	21	55	42	9	50	59	59	62				67	69
12	10	21	54	43	14	50	61	59	58				70	28
12	17	21	52	43	12	51	59	60	24				70	67
13	12	21	51	43	13	52	59	60	62				70	26

Table B.2: Low Risk Roads

Appendix C

MIP2 SOLUTIONS OF ALL SCENARIOS WITH HIGH AND LOW UNBLOCKING TIMES

High Unblocking Time (1depot, 2vehicles)

Scenario 1:

$23-21-20-19-16-10-45-40=9=40$

$23-21-54-22-26-69-25-64=68=64-69-25$

Scenario 2:

$23-21-20-19-16=10=16$

$23-21-55-22-26-69-66-68=65=68$

Scenario 3:

Infeasible Solution:

$23-49-48-44-17-12-41-9-40-8=5=8$

$23-21-54-22-26-69=25=69-26-22-55-20-19-74-17-12-41-9-40=8$

Feasible Solution (with waiting times):

$23-49-48-44-17-12-41-9-40=8=5=8$

$23-21-54-22-26-69=25=69-26-22-55-20-19-74-17-12-41-9-40-(0.13hr)-8$

Scenario 4:

Infeasible Solution:

$23-49-18-13-12-41=11-41-9-8-6-5=3$

$23-49-18-13-12-41-11=41-9-8-6=5$

Feasible Solution (with waiting times):

$23-49-18-13-12-41=11-(0.6hr)-41-9-8-6-(2.27hr)-5=3$

$23-49-18-13-12-41-(0.5hr)-11=41-9-8-6=5$

Scenario 5:

$23-21-54-22-26-69-25=66=25$

$23-21-20-19-73=72=73-46=16=46$

Scenario 6:

$23-21-54-22-26-67-31-34=33=34$

$23-49-48-44-17-11-10=46=10-12-41=9=41$

Scenario 7:

$23-21-54-22-26=69=31-34-71-32=36=32$

$23-21-20-19-16=10=16-17-12-43-42-7-39=5=39$

Scenario 8:

$23-49-48-44-17-10-45=40-8=6=3=6$

$23-21-20=19=20-22-26-69-64-65=68=65-29-32=36=32$

Scenario 9:

$23-21-54-22-26-69-66-29=32=36-63=33$

$23-21-20=19-72=73-46-16-17=11=17-44-15-21-54-22-26-69-25=64=25$

Scenario 10:

$23-21-54-22-26=69-25=66=30$

$23-21-15-74=19=74-17-12-43-9-40-45=10=45-40$

Low Unblocking Time (1depot, 2vehicles)

Scenario 1:

$23-21-54-22-26-69-64-65=68=65$

$23-49-48-44-17-11=9=41$

Scenario 2:

$23-21-15-74-16=10=16$

$23-21-54-22-26-69-64-68=65=68$

Scenario 3:

Infeasible Solution:

$23-50-21-54-22-26-69=25=69-26-22-55-20-19-16-10-45=40=8$

$23-21-20-19-16-10-45-40-8=5=8$

Feasible Solution:

$23-50-21-54-22-26-69=25=69-26-22-55-20-19-16-10-45-40-8$

$23-21-20-19-16-10-45=40=8=5=8$

Scenario 4:

Infeasible Solution:

$23-21-20-72-73-46-10=45-40-8-6-5=6$

$23-21-20-72-73-46=10-45-40-8-6=5$

Feasible Solution (with waiting times):

$23-21-20-72-73-46-(0.36)-10=45-40-8-6-(0.79hr)-5=6$

$23-21-20-72-73-46=10-45-40-8-6=5$

Scenario 5:

$23-21-54-22-26-67-31=30=31$

$23-21-54-55-20=72=19-74-15-18-48-44-46=16=46$

Scenario 6:

$23-21-54-22-26-67-30-71=33=71-33$

$23-21-55-20-19=16-46=44-17-11=45=11$

Scenario 7:

Infeasible Solution:

$23-21-20-19-16-10=16-19-20-22-26=69-66=68-29-32=36=32$

$23-21-20-19-16=10-16-19-16-17-12-43-42-7-39=3=39$

Feasible Solution (with waiting times):

$23-21-20-19-16-(0.21hr)-10-(0.14)-16-19-20-22-26=69-66=68-29-32=36=32$

$23-21-20-19-16=10=16-19-16-17-12-43-42-7-39=3=39$

Scenario 8:

$23-49-15=21-20=19=74-15-21-54-22-26-69-64-65=68=65-29-32=36=32$

$23-49-48-44-17-11=40-8=6=3=6$

Scenario 9:

$23-21-20-19=20-22-26-67-30-71=32=36=33$

$23-49-15-44-17=11=17-44-15-21-20=19-20-22-26-69-25=64=25$

Scenario 10:

$23-21-20-72=73=72-20-21-15-44-17=11=17$

$23-21-54-22-26=69=67-30=66=68$

VITA

Ayşe Nur KİBAR was born in Aydın on June 9, 1987. She received her B.Sc. degree in Industrial Engineering from Koç University, İstanbul, in 2010. In September 2010, she started her M.Sc. degree and worked as a teaching and research assistant at Industrial Engineering Department of Koç University, İstanbul.