Humanitarian Logistics: Optimization Techniques for Emergency Preparedness and Post-Earthquake Response

Arman Nedjati

Submitted to the
Institute of Graduate Studies and Research
in partial fulfilment of the requirements for the degree of

Doctor of Philosophy
in
Industrial Engineering

Eastern Mediterranean University
January 2017
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Mustafa Tümer
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Doctor of Philosophy in Industrial Engineering.

Assoc. Prof. Dr. Gökhan İzbırak
Chair, Department of Industrial Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Doctor of Philosophy in Industrial Engineering.

Assoc. Prof. Dr. Jamal Arkat
Co-Supervisor

Assoc. Prof. Dr. Gökhan İzbırak
Supervisor

Examining Committee

1. Prof. Dr. Ali Fuat Güneri
2. Prof. Dr. Semih Önüt
3. Prof. Dr. Béla Vizvári
4. Assoc. Prof. Dr. Gökhan İzbırak
5. Asst. Prof. Dr. Sahand Daneshvar

ABSTRACT

The most crowded cities in the world are located in high risk seismic areas. For humanitarian logistics system structure in highly populated cities we must search for fast and reliable monitoring and transportation methods with a futuristic mind. It is desirable to have a pre-planned immediate and automated post-disaster mapping, and transporting system. Due to roads blockage and time limits in the disaster response phase, Unmanned Aerial Vehicles (UAVs) can be utilized for relief distribution and rapid damage assessment. Also for medium to long-term ground response phase more complex but realistic models are needed.

In this study we present relief distribution and damage assessment systems alongside mathematical linear programming formulations and heuristics. In an applied sense the research improves the emergency preparedness and post-earthquake response activities. A relief distribution system by medium-scale UAV helicopters is investigated and the outcomes reveal that the system has efficient capability for urban areas with high population density. Moreover, a rapid damage assessment system is presented in which multiple UAVs are deployed to collect the images from the earthquake site and create a response map for extracting useful information.

Furthermore the covering tour location routing problem with replenishment at intermediate depots (CLRPR) is developed. The investigation represents a new bi-objective integer linear programming model that minimizes the total weighted waiting

time and the total amount of lost demands. Among the different applications of the problem, this study concentrates on the post-earthquake relief distribution system.

The mathematical models are coded in GAMS and solved by Cplex solver. Furthermore, some meta-heuristic algorithms are presented for CLRPR in order to find the near optimal solutions of large scale problems.

Keywords: Post-earthquake response, relief distribution system, Coverage path problem, Covering tour problem, UAV

ÖZ

Dünyadaki en kalabalık şehirler yüksek riskli sismik bölgelerde bulunmaktadır. Böyle şehirlerde insancıl lojistik sistemler geliştirirken geleceği de gözeten hızlı ve güvenilir takip ve taşıma yöntemleri gözönüne alınmalıdır. Afet sonrası önceden planlanmış anında ve otomatikleştirilmiş haritalama ve ulaştırma sistemleri kurulmalıdır. Afet sonrası yolların kapalı ve zamanın da önemli olduğu düşünüldüğünde İnsansız Hava Araçlarının (İHA) yardımların dağıtımı ve hızlı zarar tespitinde kullanılabileceği açıktır. Ayrıca orta ve uzun vadeli müdahale aşamasında karmaşık ancak gerçekçi modellere de ihtiyaç vardır.

Bu çalışmada matematiksel doğrusal programlama ve sezgisel yöntemlerle yardım dağıtımı ve hasar değerlendirme sistemleri sunulmuştur. Çalışma ayrıca acil durumlara hazırlıklılık ve deprem sonrası faaliyetleri ugulama anlamında geliştirmektedir. Orta boyutlu insansız helikopterler kullanarak yapılacak yardım dağıtım sistemi incelenmiş ve bulgular bu sistemin kalabalık kentsel yerleşkelerde etkin yeteneklere sahip olduğunu göstermiştir. Ayrıca İHA'lar marifeti ile, önemli bilgilere ulaşılmasını sağlayacak deprem bölgesinden toplanmış görüntülerle bir müdahale haritası hazırlanması ve böylece hızlı bir hasar tespit sistemi elde edilmesi de gösterilmiştir.

Çalışmada ayrıca ara yardım istasyonlarında ikmal edilebilir konum-güzergah kapsayıcı tur problemi (CLRPR) de geliştirilmiştir. Araştırma toplam ağırlıklı bekleme süresi ve toplam talep kaybını minimize eden yeni bir çift amaçlı tamsayılı doğrusal

programlama modelini göstermektedir. Problemin diğer uygulamaları mümkün olmakla birlikte, bu çalışmanın odağı, deprem sonrası yardım dağıtım sistemidir.

Matematiksel modeler GAMS ile kodlanmış ve Cplex ile çözülmüştür. Ayrıca büyük ölçekli CLRPR probleminin optimale yakın çözümlerini bulacak üst-sezgisel algoritmalar da gösterilmiştir.

Anahtar Kelimeler: Deprem sonrası müdahale, Yardım Dağıtım sistemi, Kapsayıcı Güzergah problemi, Kapsayan Tur problem, İHA

dedicated to disaster affected communities

ACKNOWLEDGMENT

I would like to express my sincere gratitude to Dr. Bela Vizvari, for his steady supports and guidance. Without his precious help it would not be possible to conduct this study. I really appreciate his brilliant ideas, vast knowledge, and mathematical skills.

A very special thanks goes out to my dear supervisor Dr. Gokhan Izbirak for his motivation, patience, and supports. He is much more than a supervisor for me. My gratitude for the valuable lessons that I have learnt from him through the past years is beyond the words.

My appreciation also extends to Dr. Jamal Arkat. His mentoring and encouragement have been really precious and his insights launched the significant parts of this study.

I would also like to thank my parents and in particular, I must acknowledge my sister, and my wife for their reassurance and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

It is estimated that every year near 75000 people are killed and 200 million others are afflicted by natural disasters [1]. Earthquake is one of the natural disasters that causes huge damage, destruction and loss of life. According to United States Geological Survey (USGS) based on observations since 1900, we had about 15 major earthquakes (7.0-7.9 magnitude) and one great earthquake (8.0 and higher magnitude) each year [2]. Comparison of the global seismic hazard map [3] with the world population density map Nemiroff & Bonnell [4] shows that the most crowded cities in the world are located in high risk seismic areas. With devastating earthquakes in recent years, the necessity of an efficient emergency management system is of paramount importance. Bam earthquake (2003) caused nearly 31,400 casualties [5], Wenchuan earthquake (2008) earthquake took the life of more than 80000 people [6] and in Haiti (2010) earthquake, between 200,000 and 300,000 people died and more than 300,000 were injured [7]. We can see the yearly increase in size and population of large cities like Tokyo, Istanbul, Tehran, *etc.* (e.g. Tehran has 8.4 million people with average annual growth rate of 1.44%). In such conditions, a disaster is expected to lead to a high degree of chaos in humanitarian logistics support. Therefore the essence of post-disaster early and long-term response systems are undeniable.

Eguchi [8] states that more rapid and coordinated response with the development and use of advanced technologies must be provided for disaster response to severely affected areas after earthquake or tsunami. He claims that there must be more government support to design, develop and test methodologies, systems, platforms and other components, so that robust disaster response tools could be developed and deployed around the world. In the case of earthquakes, damage assessment and relief distribution are vital and a critical tasks. For humanitarian logistics system structure in highly populated cities we must search for fast and reliable monitoring and transportation methods with a futuristic mind.

Earthquake will bring about several difficult situations like destruction of residential areas, buildings, structures, infrastructures, bridges, roads, railways, power lines and water supply and these damages will have considerable ripple effects on the neighboring access networks [9]. Kamp *et al.* [10] state four weeks after the Kashmir (2005) earthquake, the water could not be transported to affected areas. Aid employees reported that although enough drinking water and trucks were available, the roads were blocked due to landslides. They claim that land-slidings mostly happen along rivers and roads.

As an alternative transportation method, Kubo et al. [11] indicate that middle size hovercraft which can run on almost every surface is an appropriate vehicle for first-aid transportation in a disaster though it cannot operate on ruined field after an extreme earthquake. As a case study they did a simulation of hovercraft transportation

for the Kobe (1995) earthquake. They refer to the limited transportation capacity even in a daily situation in Japan and the great difficulty to increase the road capacities.

In the recent times, UAVs have become a popular entertaining tool globally. This study focuses on the transportation and mapping applications of small UAVs in post-earthquake situations. In UAV-mapping system, consecutive overlapping aerial images taken the UAVs are processed to obtain a complete map or to extract useful information. In earthquake situations, the information, such as building and bridge destructions, road blockages, and population relocation, aids the managers in organizing a more effective post-earthquake response system. Within the first 30 min, the post-earthquake survival rate is 91%, decreasing to 36.7% by the second day [12]. Hence, in addition to accuracy, time is also an important factor in post-earthquake mapping. Based on this fact, it can be said that an optimal solution to the UAV coverage path-planning (CPP) problem with time dependent objective function will have a great effect on the efficiency of the response system.

In many practical situations, it is not possible to serve all customers within a vehicle route. For medium to long term ground response the costumers are divided into in-tour and out-tour customers, and the out-tour customers can be allocated to those in-tour customers that are located within a predefined distance. Current and Schilling [13] investigated the concept as median tour problem (MTP) and maximal covering tour problem (MCTP), and suggested some applications of the problem for the distributed computer networks, hierarchical transportation networks such as mail

delivery, and the health-care systems for rural areas of developing countries. Recently, Allahyari et al. [14] used the concept for distribution of goods for a post disaster situation.

This study has been structured as follows: In chapter 2, we describe the emergency preparedness and early post-earthquake response system by small UAV helicopters. In chapter 3, the multi-UAV location coverage path planning problem for damage assessment of a post-disaster Situation is investigated. Finally chapter 4 contains formulation and meta-heuristics of a multi-objective location covering tour problem with the application of medium to long-term ground post-disaster response.

Chapter 2

EARLY POST-EARTHQUAKE RESPONSE PHASE

2.1 Introduction

As has been witnessed in the last decade, if relief network is damaged and commodity can no longer flow into a health care setting at the time of catastrophic disasters, operations can become seriously degraded [15]. Road, building and bridge destruction, disconnects the transportation paths, and severe traffic congestion in crowded cities interrupts the relief flow. The Loma Prieta (1989) earthquake closed nearly 142 roads in San Francisco and several of which stayed closed more than six months. Five years later Northridge (1994) earthquake trigged around the same amount of closures [16]. Anbazhagan *et al.* [17] state that the numbers of roads that have been damaged because of earthquakes are innumerable; however, limited records can be found as evidence. The other reason for road closures in crowded cities is the terrific congestion after earthquake. For instance after the Manjil (1990) and Bam (2003) earthquakes in Iran, the severe traffic jams for several hours caused long delays for relief activities [18]. Crowded cities like Tehran and Istanbul that are located on seismic belt and can potentially be stricken by earthquakes, have several traffic jams even during a normal day.

According to recent investigations related to emergency logistics, the focus is on the uncertainty of demand, travel time and capacities. Sahinidis [19] categorized optimization under uncertainty into three groups; stochastic programming, fuzzy programming, and stochastic dynamic programming. Najafi *et al.* [20] proposed a mathematical model for logistics planning in earthquake response phase and solved it by means of a robust approach for stochastic models. Najafi *et al.* [21] designed a dynamic model marked by its ability of updating the information and adjusting logistic activities during disaster response. In their model the network was predefined and they mentioned that the earthquake may completely close a road and the stochastic nature of parameters must be considered. Time is the most important factor in humanitarian logistics. The researchers try to minimize the total traveling time in uncertain environment. The transportation problem's difficulty increase with stochastic data and is still not very practical in post-earthquake response as the probability distribution function is unknown. These days the challenges for the academic community is to concentrate on designing more complex but realistic models that actually reflect the difference with the classical supply chain management approach [22].

Based on the above mentioned difficulties of transportation through the roads after earthquake, alternative methods of transportation must be taken into consideration. For long distance transportation of commodities and injured people, airplanes and helicopters are appropriate choices though these aerial vehicles require special landing conditions. Ozdamar [23] designed an emergency logistic system with

the goal of coordinating high capacity helicopter operations. The proposed helicopters deliver medical items to the affected areas and transport the injured people to the hospitals. Her model minimizes the total flight time and total load/unload time. She points out that the special restrictions on the helicopters are takeoff cargo weight limit and the fuel availability during the entire flight. Another restriction is the number of heavy lift helicopters which are estimated to be around 70 for 5 days evacuation mission in the Istanbul case.

For emergency logistic system design in crowded urban areas we must search for fast and reliable transportation methods with a futuristic mind. UAV helicopter is an appropriate vehicle for relief distribution in a multi modal network. In 1982, YAMAHA MOTOR started to develop Unmanned Helicopters. Later on, they developed an autonomous unmanned helicopter named RMAX which could play monitoring roles in disaster management.

Timothy et al. [24] categorize the civil uses of UAV systems into four groups of commercial, land management, homeland security, and earth science whereas the cargo capability of UAV helicopters in emergency cases is not accentuate. In the past by Japan's atomic and space agencies, UAVs have been used to measure levels of radiation at the contaminated site. United Nations Office for the Coordination of Humanitarian Affarirs [25] claims that one promising area of small UAV utilization is delivery of vaccines or other small medical supplies. In addition, an investigation about the use of UAVs for medical products delivery (e.g. blood derivatives and

pharmaceuticals) to hospitals or affected areas in case of emergency situation has been recently done by Thiels et al. [26]. They suggest UAVs as a viable mode of transportation for medical items.

While immediately following an earthquake aid employees are attempting to open the roads, UAV helicopters are able to transfer water and emergency supplies to the affected areas. Nowadays the necessity of an automatic system which detects the destructions and casualties, and sends the special needs to the place becomes obvious. The comprehensive process of data collection, data analyzing program, distribution center, and the UAV helicopters are the main components of our proposed system.

2.2 System Design

2.2.1 Small UAV Helicopter

In recent years big UAV helicopters are widely utilized by the U.S. army for equipment transportation. Further use of UAV aircrafts in detecting and destroying the targets in military is more common with improvements in radio controlling technology. Many experts believe that militarily advanced countries in future will make their strategies based on the use of UAVs, which tend to be fairly inexpensive [27]. Furthermore, small UAVs became more popular due to photography, filming, and entertainment activities. The size of radio controlled (RC) helicopters is much smaller and the fuel consumption and maintenance costs are much lower than the big ones. They are more maneuverable compared to big helicopters and in case of crash, the human loss is zero. These helicopters due to their low height fly ability and less

operation and maintenance cost perform better than huge ones for search and rescue activities, and quick mapping proposes. They can be used as fire fighters for tall buildings with much better maneuver capability and lower risk of crash. In case of earthquake, UAV helicopters are the best vehicles for delivering medical packages to temporary medical centers in the area. The authors believe that small UAV helicopters can be replaced by the big ones in the future for the sake of commodity transportation.

In a manner similar to Kendoul [28] we can categorize the current designs of UAV helicopters based on size and payload into 6 groups;

I. Large-scale UAV helicopters with nearly 3000 kg payload and about 15 meter length and 4 meters height. A good example is the Kaman Unmanned K-MAX, used in Afghanistan by United States army.

II. Medium-Scale UAV helicopters with nearly 25 kg payload and about 3 meters length and 1 meter height. Some good examples are Yamaha R-max, Rotomotion SR-200, and Aeroscout Scout B1-100.

III. Small-scale UAV helicopters with about 10 kg payload and near 2 meters length and less than 1 meter height like Rotomotion SR-100, Colibri I (Rotomotion SR-30), BICOPT-CH, and AF25B.

IV. Mini-scale UAV helicopters with nearly 5 kg payload like UVH-29E, Avenger-G, and ColibriIII (Rotomotion SR-20).

V. Micro-scale UAV helicopters with less than 20 centimeter length which cannot carry weights and are mostly used for entertainment. Also, some nano-sized UAVs like Black Hornet are utilized for military purposes.

VI. Multi-rotor UAV with around 1.5 kg payload. Multi-rotor is a different type of rotorcraft which does not need the complicated swash-plates and linkages, and instead uses different rotor speeds to manoeuver [29]. Recently some investigators work on payload distribution between several multi-rotor UAVs by cables. It is anticipated that the better design of multi-rotors in future can make a new generation of transportation. Some of the mentioned UAV models are shown in Figure 2.1.

Figure 2.1: Some UAV helicopter models

For small and medium scale UAV helicopters, speed varies from 40 to 60 kilometer per hour and the endurance differs between 45 minutes to 5 hours for

different types of engines. Due to several types and different priorities of commodities and also different levels of transportation, the medium-scale, small-scale, mini-scale, and multi-rotor groups of UAV helicopters can be utilized in humanitarian logistic systems. The mini-scale and multi-rotors are suitable for monitoring and search missions, the small-scale and medium-scale can carry different types of needs. Although the large-scale is proper for big bundle transportation or road opening activities, we pass up using this vehicle in our humanitarian logistic system because of expensive price and high operating and maintenance costs and prefer ground road opening activities instead. Table 2.1 expands several models' specifications.

Table 2.1: UAV models' specifications

UAV Model	**Specifications** (Engine/Length(meter)/Height(meter)/Max Payload(kg)/Maximum speed(kilometer per hour) /Endurance(minutes))	**Source**
Kaman K-MAX	Engine: gas turbine/ Length: 16/ Height: 15.8/ Max Payload: 2,722/ Speed: 148/ Endurance: 180	www.kamanaero.com
Yamaha RMAX	Engine: 2-cycle gasoline, horizontally opposed 2-cylinder/ Length: 2.75/ Height: 1.08/ Max Payload: 28/ Maximum speed: 40/ Endurance: 45-60	www.rmax.yamaha-motor.com.au
Rotomotion SR-200	Engine: 2-stroke gasoline engine/ Length: 2.79/ Height: 0.86/ Max Payload: 22.7/ Maximum speed: 60/ Endurance: 300	www.rotomotion.com
Scout B1-100	Engine: Gasoline engine/ Length: 3.3/ Height: 1/ Max Payload: 18/ Maximum speed: 80/ Endurance: 90	www.aeroscout.ch
BICOPT CH	Engine: two 26cc petrol engine/ Length: 1.95/ Height: 0.9/ Max Payload: 10/ Maximum speed: 50/ Endurance: 90	www.wecontrol.ch
AF25B	Engine: 2-stroke gasoline engine/ Length: 1.778/ Height: 0.711/ Max Payload: 11.3/ Maximum speed: 96.6/ Endurance: 50-55	www.copterworks.com
Rotomotion SR-100	Engine: 2-stroke gasoline engine/ Length: 1.47/ Height: 0.685/ Max Payload: 8/ Maximum speed: 79.2/ Endurance: 20	www.rotomotion.com

Rotomotion SR-30	Engine: 2 stroke gasoline/ Length: 1.638/ Height: 0.62/ Max Payload: 8.5/ Maximum speed: 40/ Endurance: 90	www.rotomotion.com
UVH-29E	Engine: 2-stroke gasoline engine/ Length: 1.2/ Height: 0.55/ Max Payload: 5/ Maximum speed: 120/ Endurance: 180	www.uavos.com
Avenger-G	Engine: GAS 2 PISTON/ Length: 1.47/ Height: 0.51/ Max Payload: 4.5/ Maximum speed: Not Available/ Endurance: 60-120	www.hse-uav.com
Avenger-E	Engine: Electric engine/ Length: 1.47/ Height: 0.51/ Max Payload: 4.5/ Maximum speed: Not Available/ Endurance: 25-45	www.hse-uav.com
MULTIROTOR G4 Recon One	Engine: Electric/ Length: 1.18/ Height: 0.35/ Payload (incl. camera suspension and flight battery): 5-5.3/ Maximum speed: 30-40/ Endurance: 50-70	www.service-drone.de
Zero 1600 Hexacopter	Engine: Electric/ Length: 1.6/ Payload: 6/ Maximum speed: 28.8/ Endurance: 60	www.zerouav.com

Moreover, based on the type of fuel we have electric, nitro, gas, and turbine powered UAV helicopters. The Micro-scale and multi-rotor UAVs have electric powered engines. However, discussion on the types of engines is beyond the scope of this study. There is no proper UAV helicopter engine's classification, although according to the best of our knowledge, consultation with experts, and according to some investigations ([28], [30]) we found that for imaging and reconnoitering intents we can use electric powered UAVs because of lower vibration and easy to start structure. Also, two-stroke gas engine and turbo engine or turbo-diesel engine UAV helicopters can be utilized for commodity transportation. The nitro engine's fuel is expensive compared to gas or electricity and their operational cost is high. Because of the low fuel consumption and consequently longer flying duration, the gas powered UAVs are more popular in large scales. For the electric engines, nevertheless, the charging problem reduces the electric UAV's efficiency for long-range and long-term

emergency cases. However, electric powered helicopters are going to achieve the best progress within the UAV helicopters in coming years and as Miranda et al. [31] claim, the lithium ion batteries are the most widely used and still indicate an encouraging progress potential.

2.2.2 How the System Works

The system works like this; sensors detect earthquake waves and the data collection starts. Based on the earthquake magnitude and other geographic information, systems' data and analysis of information received from other available sources, post-earthquake casualties and injuries to human beings in different locations will be estimated. Immediately after the initial estimation, the UAV helicopters will be sent to impassable places while transporting first aid commodities and requested supplies. The helicopters return to the center, reload and travel to the same or new ordered locations.

The information must be categorized as to the level of injuries and aggregation of injured people in different locations. The flow of information comes from different sources like civilian's mobile phone, rescue employees, police reports, social media, buildings' motion monitoring system, satellite monitoring and UAV helicopter monitoring system. The Data Analysis Center (DAC) must analyze the data in a way to be able to distinguish between the location and quantity of injuries, separate the level of priorities for injuries, the most useful commodities for each location, the distance to emergency centers, and the road condition like traffic congestion or road

destructions. There could be different unmanned helicopter distribution centers (UHDC) in or outside of the city. The final report of DAC to UHDCs must specify a queue, consisting of location of incidents, types of commodities required, and the amount of each commodity needed in place. Such a report also must be sent to the *general earthquake response cente*r which will send the aid employees and commodities to the affected areas and controls the whole response activities. Figure 2.2 depicts the proposed information flow system.

In UHDCs the helicopters will wait for their missions. The number and types of helicopters of each UHDC must be estimated before, using the coverage population and the probability of human loss in the area. The UHDC buildings are anti-seismic constructions located far from other tall buildings and close to the highly populated areas. After the first earthquake signal is received, the reconnoiter UAVs fly from all UHDC centers to the probable affected areas for photographing and search missions. UHDC buildings have depots of first aid commodities and relief supplies, but in case of shortage the supply transports from general inventory center to UHDCs during earthquake respond. The basic supply packs are ready to be lifted by helicopters from UHDCs to the needful locations. The packs will be released in affected areas with a flag or colored smoke sign so people will be able to spot them easily. Also for response situations in the night, a small flashing light can be prepared with aid packs. The helicopters are GPS equipped and able to find the place automatically. The helicopters return to the UHDCs for further missions.

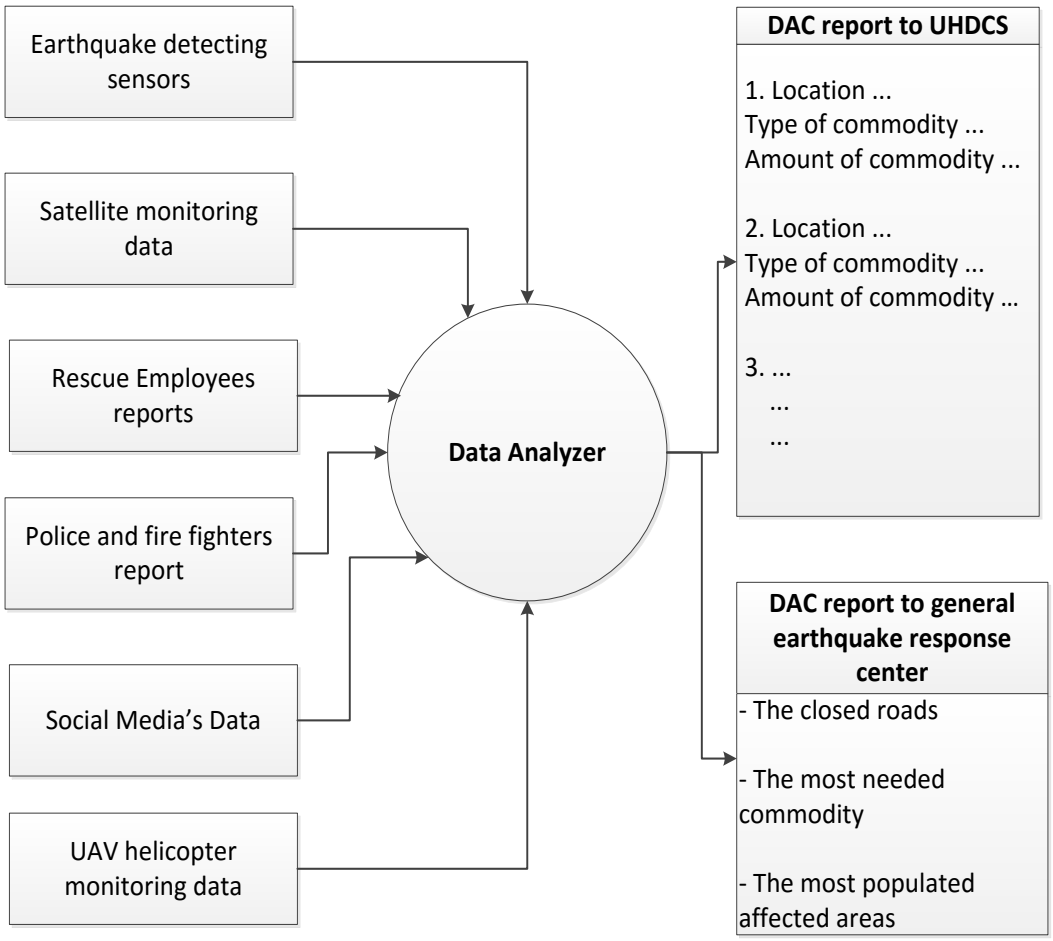


Figure 2.2: Data Analysis Center

The information flow will be updated continuously and the helicopters lift packs with more relevant commodities to the most needed areas. Special boxes for carriages are prepared and aid packs can automatically be filled due to the demand. The forecasted types of demands must be ready to fill the pack. According to the type of demand for different areas the robots in the UHDC centers will fill the packs equal to the helicopters payload. The helicopters receive the GPS address of determined area, lift the pack and release it at the appropriate place. For example in the case of dangerous situations like probable gas poisoning incidents, gas masks must be filled to the packs and distributed to the areas immediately. The information flow will be updated by

seconds and the priority of injuries or amount of injured population defines the subsequent missions. The weight lifting and release by the helicopters is automatic but yet in case of failure, human intervention can be employed. In case of heavy lifting packs, multiple helicopters may carry the weight together by cable connection. The electric helicopter batteries must be replaced after each flight and charging process must take place for further responses. Likewise, the auto gas refilling system must be provided for gas powered UAV helicopters. The UHDC buildings must definitely have the emergency power system for power outage situations.

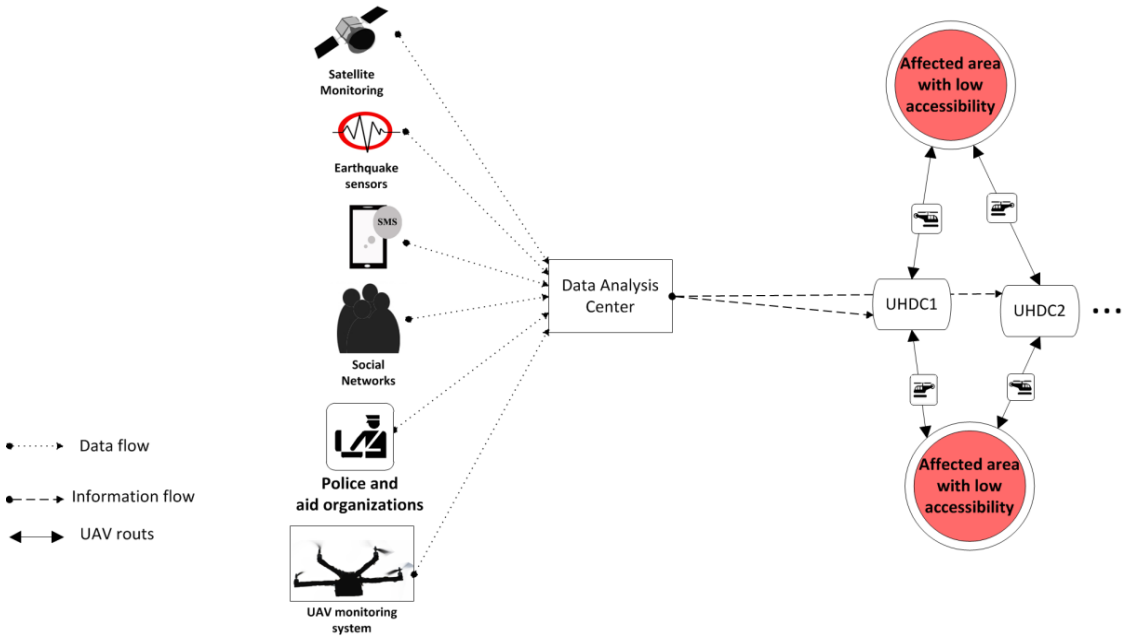


Figure 2.3: Sketch of the system

According to CDC [32] suggestion of emergency supplies for earthquake preparedness, the most useful supplies for earthquake response are water, first aid kit, first aid instruction book, medications like hydrogen peroxide and antibiotics, dressings like bandage strips and adhesive tape rolls, scissor, soap, plastic bag, non-

perishable and canned food, blankets or sleeping bags, maps of the area, flashlight, *etc*. These supplies can be easily transported by UAV helicopters, and have a good tenability to store. The sketch of the proposed system is shown in Figure 2.3.

Even the UAV helicopters can transport the supplies to the mobile medical centers. It is much faster to transport urgent supplies like blood, medicines and other medical items by small helicopters. Fast blood transportation is vital for the injuries especially after earthquake as the blood demand is high. After East Azerbaijan province (2012) earthquake in Iran, there were blood donor centers that were busy collecting different types of blood from donors all around the country to send them to the hospitals in East Azerbaijan province. Assume that an earthquake strikes a thickly populated city and severe destruction happens in some areas. Then the blood donor centers try to collect blood form volunteers of less affected areas, and the blood must be immediately transported to the hospitals. The UAV helicopter with special boxes for blood transfer which can carry the blood safely to the hospitals can handle this situation. The helicopter batteries can be replaced manually at the hospitals and sent back to the defined GPS location for further missions.

After a disaster like earthquake and thereupon vast humanitarian losses, all countries around the world send goods and support to the affected country. It happened in Haiti (2010) earthquake and major earthquakes before. The commodities received must be categorized in main distribution center which is done manually and is very time consuming. The first task related to our system is to separate emergent items and

send them to the UHDCs to dispatch. The items size and shape must be adaptable to UAV carriage boxes or carriage hooks.

2.2.3 Legal Issues

One of the important challenges to use UAVs for humanitarian purpose is the legalization. USA and OECD countries are developing authorized directions for civilian use of UAVs. US Federal Aviation Administration (FAA) categorizes the unmanned aircraft systems (UAS) into three different types of operations; Public Operations (Governmental), Civil Operations (Non-Governmental), and Model Aircraft (Hobby or Recreation only). For public operations FAA gives a certificate specifying the aircraft, purpose and area of operation that allows public organizations (governmental) to run sorties. For civil operations, FAA gives authorization with strict conditions (interested readers can refer to FAA website).

Despite implementing the UAV helicopter distribution network for thickly populated cities can be done by governments much easier, some non-governmental organizations (NGO) may interfere with specific specified duties to accelerate the response procedure. The “civilian use” must be replaced by “humanitarian use” and local and national rules must be designed properly for humanitarian situations. The national and international aerial law must be flexible for humanitarian UAVs and humanitarian NGOs must be able to make long term agreements with governments in a compliable manner. The United Nations Office for the Coordination of Humanitarian Affairs recently published a pivotal document policy [25] and mentioned that “*Many*

countries where humanitarians are working do not yet have legal frameworks, meaning that use of UAVs will probably need to be cleared on an ad hoc basis with local and national authorities.". Since flying regulations for different types of UAVs are still under investigation, it would be beneficial to ratify a particular note of humanitarian UAVs as an international law. In this way, after catastrophic events, well-equipped NGOs can interfere along with governments and facilitate the emergency response process.

2.3 Scientific Capacity and Preparedness

Scientific capacity for proposed system's implementation has been investigated by scientific studies in recent past. Many studies have been done with the concept of rapid loss assessment after an earthquake. Kubo et al. [33] designed a system with the combination of earthquake early warning system and real-time strong motion monitoring system to emergency response for high-rise buildings. Their system can immediately send emails to emergency response team and can provide information on seismic intensities at each floor. Park et al. [34] investigated microwave remote sensing by airborne or space-borne sensors for monitoring near-real-time damage over large areas. As a good reference Erdik et al. [35] summarize the investigations done over last decades with regard to earthquake quick response techniques that are performed to estimate earthquake losses in quasi real time. Some earthquake loss assessment techniques are mentioned in Table 2.2.

The proposed DAC system is completely applicable with the progress of data fusion science. Using social media, satellite monitoring, and communication systems with the help of data fusion systems can provide an integrated complex to assist the earthquake aid response program. A valuable study has been done by Jotshi et al. [36] which uses the data fusion for developing a methodology of emergency vehicles' dispatching and routing in a post disaster situation. They have also considered an earthquake scenario with large amount of casualties. Hence data science field by considering its current rapidly increasing growth can be very helpful in this issue.

Table 2.2: Earthquake loss assessment techniques

loss assessment techniques	Comment	Reference
Earthquake detection sensors	With all weather conditions synthetic aperture radar (SAR) shows ability to extract building collapse information in urban areas and provides timely remote sensing data for emergency response.	[37]
Satellite monitoring	By satellite remote sensing information, rapidly estimation of number of casualties is practical.	[38]
UAV monitoring	Unmanned Aerial Vehicles have increasingly become a common tool to deal with search and rescue missions with the ability of finding potential survivors requiring medical attention.	www.close-search-project.eu
Social network	Social networks like twitter are able to provide quick information of casualties following a disaster.	[39]

Nowadays we can see the increasing progress of UAV helicopter systems for transporting items. The AMAZON's plan for postal package delivery by UAVs called Amazon Prime Air is just a beginning. Also Heutger and Kückelhaus [40] from DHL company believe that from today's perspective, the two most promising utilities in the logistics business are Urgent express shipments and rural deliveries. Besides, scientists are designing surf lifesaving systems for automatic transfer of the floating device to the swimmer in danger, and also there are some patterns designed for helping the people in special emergency cases like heart attack. Furthermore in military we can see the new designs of armed UAV helicopters with camera that can search and fire the dangerous targets from a distance. Practicality of these plans shows the scientific capacity for the implementation of an autonomous rescue system for relief supply after an earthquake. Vast amount of investigations in recent years are focused on flight control and autonomous algorithms for UAVs (e.g. [41]–[43]).

Also community preparedness plays a very important role for after disaster response activities, especially for the described system. Earthquake preparedness lessons must be taught and trained in all levels of education. First aid training videos need to be shown on social media and local televisions. Individuals are the largest and the first aid contributors after an earthquake and they must be systematically prepared for different situations. Toyoda and Kanegae [44] recommended and verified a functional community evacuation planning design for the construction, assessment, and supplementation of community evacuation systems by citizens with local skills

who are the initial responders to catastrophe. They claim that relations among citizens are fairly weak and it leads to high social vulnerability to hazards since fewer co-operations in the disaster can be seen in cities compared to rural areas.

According to our system's design, during the first hours after an earthquake, the first aid commodities will be sent by UAVs to the affected areas. Most probably it takes time for experts to achieve all affected locations in the first hours so that the residents start rescuing injured people. According to the literature, high percentage of injured people suffers from laceration and hence the residents must be aware of using dressing materials for first aid support. They must know how to treat with the fractures or dislocations.

2.4 Case of Tehran

Tehran metropolis is the capital of Iran with latitude and the longitude of 35.70° N and 51.40° E respectively. The size of the city is 686.3 km^2 and is divided into 22 municipal districts which are shown in Figure 2.4. According to the reports of the Statistical Centre of Iran [45], Tehran city's population is estimated to be 8.3 million and more than 12 million in the wider metropolitan area with 1.44% average annual growth.

Tehran metropolis is located on multiple active seismic faults and has been struck by several high magnitude earthquakes in history. Ashtari Jafari [46] claimed that approximately once in every 10 years a large earthquake may occur around Tehran. Experts believe that a high magnitude earthquake may strike Tehran in the

future, considering that the area has not experienced a major earthquake catastrophe since 1830 and the probable return period is around 173 years [47], [48].

As Tehran Municipality [49] claims the high population density areas are districts 7, 8, 10, 11, 13, 14, 15, and 17. By same information from Tehran Municipality [49] the districts numbered 18, 21, 22 and certain districts 9 and 15 have the poorest access to the emergency services and the districts 5, 9, 21, 22, south of 18, and east of 9 has the lowest access to the fire stations. The investigation titled "The Study on Seismic Micro-zoning of the Greater Tehran Area in the Islamic Republic of Iran" has been done by Japan International Cooperation Agency (JICA) between 1998 and 2000. The main report plus "Microzoning Maps", released on November 2000 by cooperation of Centre for Earthquake and Environmental Studies of Tehran (CEST). The desired information of the mentioned study [50] is summarized as follows:

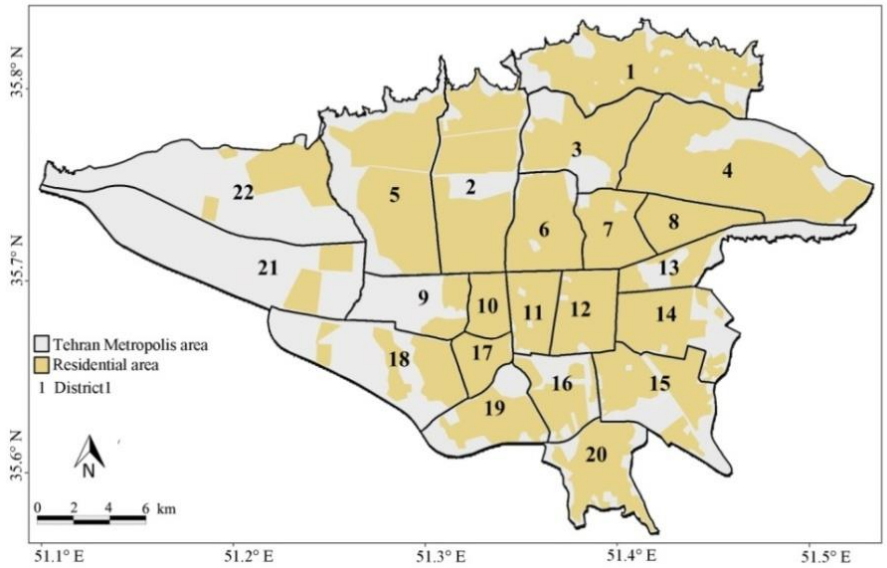


Figure 2.4: Tehran Metropolis with 22 municipal districts and residential areas

i. The main active faults of in and around Tehran can be summarized as; Mosha Fault with about 150 km long which is located on northeast of Tehran. North Tehran Fault from Kan in the west to Lashgarak in the east. South and North Ray Faults that are the most notable faults in the Tehran's southern plains. The faults are shown in Figure 2.5.

ii. Roads with three (3) and six (6) meters wide which are narrow streets are considered as weak in terms of the relief activities. Districts 10, 11, 12, 14, 15, 16 and 17 have longer length weak roads.

iii. By the result of risk evaluation by considering six items of average seismic intensity, residential building damage ratio, death ratio, population density, open space per person, and narrow road ratio, districts 9, 10, 11, 12, 14, 16, and 17 are high risk according to the Ray Fault Model, and districts 10, 11, 12, 16, and 17 are high risk districts according to Floating Model. Considering that the Mosha Fault Model created only minimum seismic hazard, it was eliminated and the North Tehran Fault Model does not result in any high risk district.

iv. Districts 10, 11, 12, and 17 have the highest disaster risk and because of the narrow roads of mentioned districts the evacuation and relief distribution will be difficult.

v. The worst earthquake scenario is Ray Fault Model which totally causes 6% and 3.1% of total Tehran Metropolis population death respectively in night-time

and day-time with no rescue operation. If first-aid relief operations are conducted in a timely manner, there will be a 25% decrease of casualties.

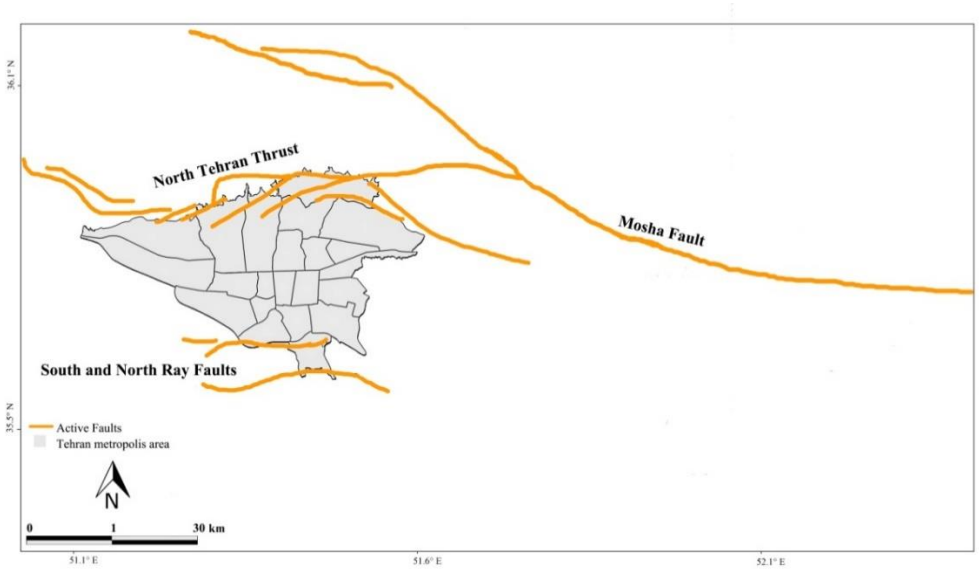


Figure 2.5: Main active faults in and around Tehran metropolis (Faults are adapted from Berberian et al. (1983))

vi. Rehabilitation work of emergency road networks (including that which would facilitate food supply) will take a few days. Meanwhile, traffic of private cars will hinder emergency operations on designated emergency roads.

vii. First aid teams must be organized by considering parks and open spaces as candidates for primary tent evacuation areas.

viii. A system for distributing emergency goods such as medicine, other first aid supplies, tableware, bedding, clothes, *etc.* Also similar systems for emergency food supply are recommended.

According to the summarized information provided, the appropriate locations for UHDCs are Ghale-Morghi (U_1) at northern part of district 19, Niroo Havayi (U_2)

at western part of district 13, and Park-e Pardisan (U_3) at central part of district 2. We consider these locations as a typical example and thereafter the number of UHDC sites and the exact locations may be optimized in future studies. All these sites have enough space for UHDC construction and proper open aerial environment for helicopters, also they are close to the high risk districts of Tehran while they cover most part of the Tehran Metropolis area within about 7 kilometers distance (see Table 2.3). These three locations have access to the highways around Tehran for re-supplying in case of shortage.

Based on analysis of JICA and CEST [50] the demand nodes are located in high risk districts and some areas in other districts which might be damaged. According to Figure 6, districts are divided into two or three residential zones or totally assumed as a single residential zone and the group of nodes in each zone summarized into one node (e.g. D1A means group of demand nodes exist in zone A of district1). The location of demand nodes and UHDCs are shown in Figure 2.6.

Table 2.3: Supposed UHDC sites characteristics

UHDC	Urban name	Covered districts within 7 kilometers approximately	Currently occupancy
U1	Ghale-Morghi	10, 11, 12, 16, 17, 18, 19, 20, and east of 9	Empty
U2	Niroo Havayi	4, 6, 7, 8, 12, 13, 14, 15, and southeast of 3	Empty
U3	Park-e Pardisan	2, 3, 5, 6, northeast of 9, and north of 10	Aerial site

Road accessibility is categorized into 5 levels with the same definition of JICA and CEST [50] in which 1 means worst and 5 means best ground accessibility after earthquake. In the worst case scenario we assumed that 90% of badly injured people in node with road accessibility of 1 must be covered with aerial help. Similarly 80% for accessibility of 2, 60% for accessibility of 3, 45% for accessibility of 4 and 30% for accessibility of 5 must be covered. The Ray fault scenario as the worst case scenario is considered for this study. The number of badly injured people for each demand node has been calculated with the ratios provided by JICA for each district.

We assumed three type missions for relief transporter helicopters. For the first 2.5 hours that injured people need immediate care, 1.5 kg packages for each person will be sent by UAV helicopters.

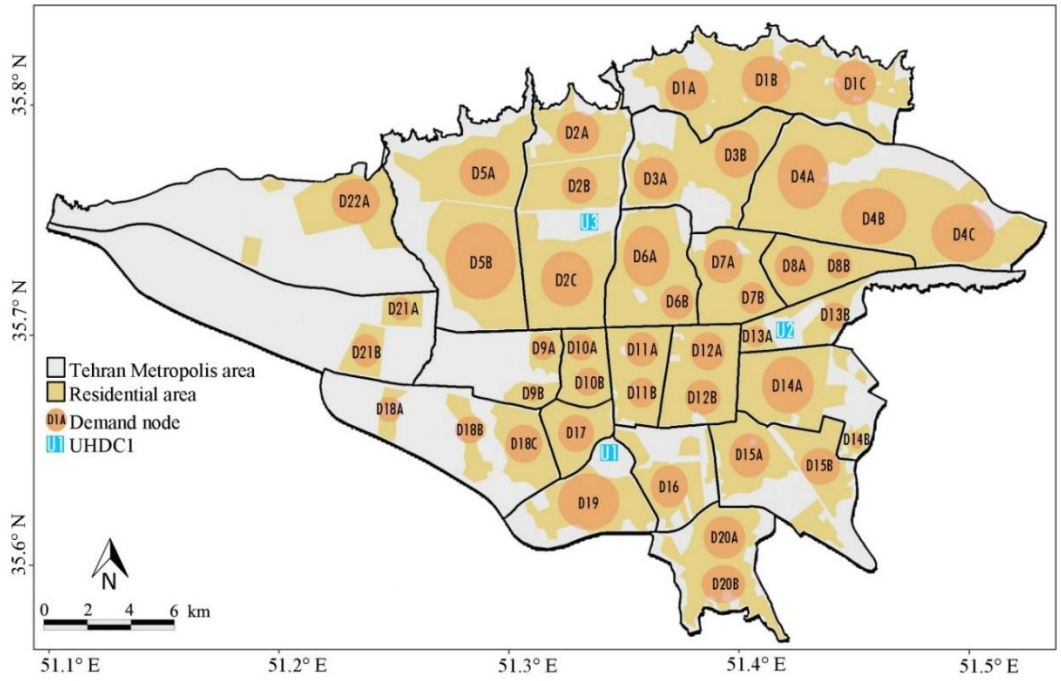


Figure 2.6: Demand nodes

Table 2.4: Presumed commodities for different mission completion times

2.5 hours mission	3 hours mission	24 hour mission
1.5 Kg per injured person	1.7 kg per resident aid-worker	4 kg per person
Water (0.35 kg), emergency low weight shovel (0.5 kg), heavy work gloves (0.02), antiseptics like normal slain and butadiene (0.2 kg), medicines like antibiotics and painkillers (0.01 kg), brown Sugar (0.03 Kg), Mini led flash light (0.05 kg), matches (0.01 Kg), and tweezers (0.04 Kg)	Water (0.6kg), infant food and high energy foods like peanut butter, nuts, milk, chocolate, and *etc.* (0.4 kg), Emergency light night sleeping bag (0.2 kg), low weight rope (0.4 kg), plastic bags and toilet papers (0.02 kg), and hand sanitizer gels (0.15 kg)	Water (2.5 kg), food (0.5 kg), blanket and cloth (0.5 kg), hygiene items like toilet paper, sanitizer, sanitary napkins, and *etc.* (0.4 kg), plastic bags, cup, fork, and knife (0.02 kg)

For the second 3 hours after the earthquake that resident aid-workers try to evacuate people from ruined construction, we proposed a 1.7 kg relief package that must be transported by UAVs. And for a 24 hour mission the helicopters must carry a 4 kg package for each person due to their needs. The three types of packages are detailed in Table 2.4.

The Ray fault scenario as the worst case scenario is considered for this study. The number of badly injured people for each demand node was calculated by the ratios provided by JICA for each district. The road accessibilities and total demand estimated for nodes are available in Table 2.5.

The distances of each UHDC to all demand nodes are known and we want to minimize the total number of missions subject to the total flight time limitation. The model assumptions are as follows:

a) The helicopter is a medium size helicopter which according to our categorization belongs to medium-scale UAVs (e.g. specification of SR-200 is: 2-stroke gasoline engine, maximum speed up to 60 kilometer per hour, endurance up to 5 hours, and the maximum payload of 22.7 kg. Yamah Rmax helicopter can be replaced by more payload weight). We assumed an average speed of 45 kilometer per hour (due to elimination of load and unload time) and payload of 18 kg for each mission (due to better stability).

b) The mission starts to takeoff from UHDC center with 18 kg commodity load and finishes by landing on the same UHDC with empty load.

c) The refuel process can take place at UHDC centers quickly and automatically in loading period in case of shortage for each 5 hours.

d) We assume that the helicopters drop the package in a standard height at the demand nodes without landing.

Table 2.5: Proposed demand nodes

Demand Nodes	Estimated population	Road accessibility	Demand estimated (kg) for 2.5 h	Demand estimated (kg) for 3 h	Demand estimated (kg) for 24 h
D1A	155713	4	315	357	2732
D1B	143735	4	291	330	2522
D1C	105212	5	142	161	1231
D2A	103741	4	420	476	3641
D2B	213967	4	867	982	7509
D2C	330677	4	1339	1518	11605
D3A	123849	5	334	379	2898
D3B	185773	4	752	853	6520
D4A	157688	4	319	362	2767
D4B	525628	3	1419	1608	12298
D4C	192730	5	260	295	2255
D5A	253137	5	456	516	3948
D5B	470112	5	846	959	7333
D6A	164265	5	591	670	5124
D6B	88450	4	478	541	4139
D7A	178386	3	1766	2001	15303
D7B	151959	2	2006	2273	17381
D8A	242005	3	2396	2715	20761
D8B	161336	4	1198	1358	10380
D9A	84809	3	1069	1211	9260

D9B	91877	3	1158	1312	10031
D10A	174789	1	4247	4814	36804
D10B	161334	1	3920	4443	33971
D11A	164085	2	5513	6247	47774
D11B	126046	2	4235	4800	36698
D12A	134727	1	5820	6596	50433
D12B	129443	1	5592	6338	48455
D13A	73274	3	725	822	6286
D13B	188421	4	1399	1586	12123
D14A	489112	2	8216	9313	71203
D14B	25742	5	162	184	1405
D15A	226424	3	4076	4619	35316
D15B	452849	3	8151	9238	70633
D16	310095	3	6140	6959	53203
D17	272663	2	7198	8158	62375
D18A	67561	5	638	724	5532
D18B	101341	5	958	1085	8298
D18C	168902	4	2394	2713	20746
D19	266022	4	3591	4070	31119
D20A	182299	3	3937	4463	34121
D20B	175150	3	3783	4288	32783
D21A	47650	5	214	243	1858
D21B	122529	5	551	625	4778
D22	115738	5	208	236	1805

e) Amount of demand for each node is reported by the DAC center to UHDCs and the information updated systematically.

f) The total amounts of available helicopters are 460 which is a reasonable amount for three UHDC centers.

Parameters and variables of the IP model:

U	set of UHDC nodes
D	set of demand nodes
i	denotes the index of node in U set
j	denotes the index of node in D set
a_{ij}	flight time of a complete mission from node i to node j
X_i	the number of helicopters at node i
Y_{ij}	the number of complete missions from node i to node j
D_j	number of missions required to satisfy the demand of node j
T	the total length of all missions ($T \geq a_{ij}\ for\ all\ i \in U, j \in D$)
N	total number of helicopters

The model can therefore be stated as follows:

$$Min\ \sum_{i=1}^{U}\sum_{j=1}^{D} a_{ij}Y_{ij} \tag{2.1}$$

Subject to:

$$\sum_{i=1}^{U} Y_{ij} - D_j \geq 0 \qquad \forall j \in D \tag{2.2}$$

$$\sum_{i=1}^{U} X_i = N \tag{2.3}$$

$$\sum_{j=1}^{D} \frac{Y_{ij}}{\left\lfloor \frac{T}{a_{ij}} \right\rfloor} - X_i \leq 0 \qquad \forall i \in U \tag{2.4}$$

$$X_i, Y_{ij} \geq 0\ \&\ Integer: \qquad \forall\ i \in U, j \in D \tag{2.5}$$

In the model, Eq. (2.1) minimizes the total flight times for all UAV helicopters, constraint (2.2) guarantees that the total demand is satisfied, constraint (2.3) restricts the total number of helicopters in all the three UHDC centers to N, constraint (2.4)

ensures that the number of helicopters at node *i* is enough to handle the number of missions within time limit T, and constraint (2.5) defines the non-negativity of the variables.

The model was solved by Cplex 12.6.1.0 and the results obtained for three different mission completion times are prepared in Table 2.6:

Table 2.6: Number of helicopters for each UHDC based on the mission completion time and demand weights

	2.5 hours total completion time	3 hours total completion time	24 hours total completion time
	1.5 Kg per injured person	1.7 kg per resident aid-worker	4 kg per person
Number of UAV helicopters for UHDC1	221	233	253
Number of UAV helicopters for UHDC2	177	169	153
Number of UAV helicopters for UHDC3	62	58	54
Total number of UAV helicopters	460	460	460

For 2.5 hours of total completion time, 66730 highly injured people will be served each by 1.5 kg commodities. For 3 hours of total completion time, the same number of people who contributed in rescue activities will be served each by 1.7 kg relief packages, and for a 24 hours aid activities 216839 people will be served by 4 kg supply for each person. It is assumed that the UHDCs have enough supplies for the first 24 hours help aid. According to the UHDCs' location which is near the free roads that connect the city to neighbor provinces, the supply can be transported to them from the airport or other destinations easily. The proposed model proves that we can supply thousands of kilograms commodities following a disaster in a small amount of time.

We assumed that there are 460 helicopters available and the helicopters must return to their own UHDCs. However, by increasing the number of helicopters and changing the system in such a way that the helicopters move to the nearest UHDC after unload, the number of served people can be increased and more supply can be transported by UAVs. The system needs more investigation in technical and computational manner. The concept is applicable in large populated cities and decision makers decide the projects budget and subsequently the coverage rate.

2.5 Discussion and Conclusion

UAV helicopters can be assumed as the next generation transport vehicle for commodity transportation. Due to the difficulties of ground transportation in case of a catastrophic event, an aerial aid system is necessary and requisite. By current scientific progress in autonomous systems, designing an immediate relief distribution system by UAV helicopters is practicable. The medium size helicopters can reduce the ground load transportation and compensate the shortage in emergency response systems by low operation cost. Compared to big helicopters, they have better maneuver ability and with higher number missions they can cover widespread area in a short amount of time. The proposed complex can be considered as a complementary system in communication with ground emergency responders. Also an information system that collects and analyzes the data and guides all emergency departments is a vital need these days. Data scientists can help the engineers to build and improve such a system for higher security of residents.

In our case study we have considered a medium scale UAV helicopter for commodity transportation in first few hours after the earthquake. The case study we indicates that 460 UAV helicopters can transport near 100,000 kg of supplies in 2.5 hours from three supply centers to 44 demand nodes. The system works without disturbing the ground transportation systems immediately following a disaster to save lives. In subsequent hours the system helps to serve the demands of citizens in non-accessible areas.

The optimal design of UAV helicopters for disaster response can help the system designers to make more effective models for future response activities. This promising technology needs more investment and investigation. Cost is the most important factor for decision makers and the developed technology may result in lower cost and higher quality of the system. However for a ten million populated city the share cost of each person for our proposed system is not much, but its practicality is dependent to the governmental supports. Also the available UAVs are useful in noncritical conditions for secondary intents. They can be used for blood transportation between blood centers and hospitals, traffic monitoring systems, and agricultural purposes.

The UAV technology is improving rapidly and in future we will have the small helicopters carrying the needs even for commercial purposes due to lower cost of transportation and shorter delivery time.

Chapter 3

MULTI-UAV LOCATION COVERAGE PATH PLANNING

3.1 Introduction

Rapid damage assessment in post-earthquake situation plays an important role in the early response phase activities (i.e., evacuation of injured individuals, debris collection, and relief distribution). The ground-based post-earthquake inspection is extremely time-consuming, and unhelpful in severely damaged areas; therefore, at present, aerial systems are widely used for investigations. It is highly desirable for highly populated urban areas to have a pre-planned immediate and automated post-disaster mapping and monitoring system. In the past decade, scientists have made an attempt to improve high-resolution satellite imaging and laser scanning systems to evaluate the disaster damage and loss [51]–[58]. However, the satellite systems have many limitations for an efficient post-disaster imaging such as weather conditions (cloudy or dust whirls), time constraints for acquiring images and uplinking the acquisition plan to the satellite, delay in satellite data delivery after collection, etc. [59], [60]; therefore, much attention has been focused on utilizing small Unmanned Aerial Vehicles (UAVs) for post-disaster mapping [61], [12], [62].

UAVs of various sizes are available on the market that can carry film or photographic cameras for different applications. The classification of UAVs according to their classes, distinct applications, and characteristics is beyond the scope of this chapter, though interested readers can refer earlier studies [28], [61]. Moreover, Colomina & Molina [63] have reviewed significant applications of UAVs such as imaging and remote sensing.

CPP problem, can be considered as a variant of Vehicle Routing Problem (VRP) .In VRP, a fleet of vehicles start and end their tour at a single depot while visiting all nodes on the route. Unlike the classical VRP that considers a single main depot, in multi depot vehicle routing problem (MDVRP), there are several depots and the customers can be served from any of the depots. MDVRP has widely been studied in the literature and the interested readers may refer to the extensive survey of Montoya-Torres et al. [64]. Based on the literature the UAV routing problem is also formulated as a VRP model where additional constraints need to be added to reflect the characteristics of the problem. UAV routing problem has attracted much attention in the past decade for different applications [65], [66]. CPP, tries to find an appropriate path for robots while covering the pre-defined nodes. The problem has a wide range of applications in automated harvesting, vacuum cleaning, mapping, demining, monitoring, etc. It is worth mentioning that CPP is different from the Covering Tour Problem (CTP). In CTP there are two types of nodes, those that must be visited by the tour and those that must be covered i.e., it lies within a pre-defined distance from a

vertex of the tour [67]. Choset [68] and Galceran & Carreras [69] conducted comprehensive surveys on CPP methods, algorithms, and recent advances.

Immediately after an earthquake, the damage and location of afflicted people can be identified by imaging and then processing the obtained images. Since time factor is very important in post-earthquake response phase, UAVs are considered to be reliable for this task. Earthquake is not the only disaster that can be monitored by UAVs, but it is the most suitable one for imaging due to absence of information about the condition of each construction all over the area. The imaging application of UAV based on CPP necessitates further considerations. One important issue is the direction of the path while passing through a node. Another important issue that arises due to earthquake situation is that, unlike routing problem, the maximum mission time of vehicles must be minimized. UAVs start their route from a base and end their route at the same base in order to refueling or at the mission completion. This study extends the CPP problem for an application in the post-earthquake rapid damage assessment. The CPP features on a grid-based map are adopted on a multi-depot multi-tour vehicle routing problem for an emergency situation.

Zelinsky, Jarvis, & Byrne [62] were the first to investigate the CPP problem on a robot, starting from an origin and ending at a goal point to minimize the length, energy consumption, and travel time. Later Carvalho et al. [70] proposed an algorithm for a mobile robot in an industrial environment such that the obstacles were not pre-specified on the map. Path planning subject has attracted a great attention due to the

recent development of UAV systems. Li et al. [71] studied an exact cellular decomposition method for UAV path planning in a polygon region. For the purpose of precision agriculture mapping, Barrientos et al. [72] performed an experiment using an integrated tool. Initially they subdivided the polygon area and then conducted path planning for each subarea using a multi-UAV system. Torres et al. [73] presented a path planning algorithm for a single UAV with the aim of reducing battery usage and minimizing the number of turns, coping with both convex and non-convex regions. In a study by Wang, Sun, & Li [74], an algorithm was used that minimized the consumed energy by the UAV for covering a 3D terrain. In addition, a distinct study [75] introduced the multi robot boundary coverage problem with the application of inspection of blade surfaces inside a turbine. Furthermore, Galceran & Carreras [69] conducted a comprehensive survey on CPP.

Most CPP algorithms are based on boustrophedon path strategy. In this strategy, the back and forth motion in a sweep direction of a polygon try to cover an area with minimum number of turns. Another approach divides the decomposing area into sub-areas and then finds the visiting sequence of the sub-areas in a sweep direction base [76], describing how to find the optimal sweep direction for robots in a polygon. Huang and York claimed that the minimization of the number of turns leads to the most efficient solution (Figure 3.1). Li et al. [71] showed that the path with less number of turning motions is a more efficient coverage path for UAVs with regards to energy, distance, and travel time.

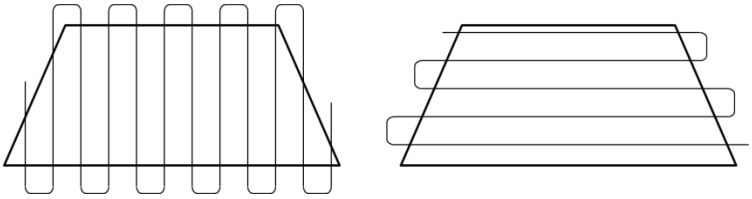


Figure 3.1: The right polygon is covered by lesser waste (lesser number of turns) than the left one.

Furthermore, the map decomposition and sweep direction method were applied for a multi-UAV system [77]. Due to lack of time in a post-earthquake situation, using a multi UAV system is potentially more beneficial and practical. Avellar et al. [78] studied a multi-UAV CPP problem in which the nodes were pre-defined correspond to the boustrophedon motion. Similar to earlier studies, they decomposed the region into several sub-regions and used a VRP model to find the optimal path. However, the travel time between the sub-areas and the base to a region as well as the possibility of turning back from the middle of a region to the base for refueling were neglected.

Although numerous studies have been conducted on the sweep direction method, path covering based on grid-based decomposition has not yet received much attention. In a study by Wang & Li [79], the desired region was divided into grids and the amount of information collected by UAV and the path-length were maximized. According to their algorithm, passing through the center of a grid is equivalent to covering the grid; however, to have a complete image of the grid, the covering is considered as entering a grid from one side and exiting from the opposite side. Therefore, the turning motions in the grids are not considered as a part of the covered

area (Figure 3.2). The grid size can vary depending on the altitude of the UAVs and focal length of the camera; however, by increasing the number of grids the complexity of the problem will increase.

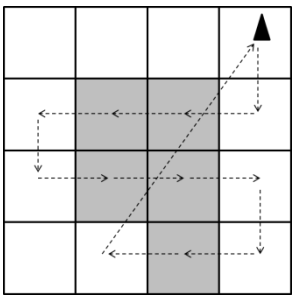


Figure 3.2: The gray cells indicate the cells that have been covered by the UAV, according to entrance and exit from the opposite sides. The picture depicts a feasible solution of grid decomposition-based CPP.

In this study, it is assumed that the number of available UAVs, potential location of UAV bases, and the possible number of open bases are based on the restrictions prescribed by the decision maker. In addition, the total mission time limit should be specified earlier according to the endurance of UAVs. We have tried to address the following questions in the proposed post-earthquake mapping problem: (1) which potential UAV base must be open; (2) How many UAVs among all the available ones should be assigned to each open base; (3) What must be the passing sequence of squares in each route; and (4) what is the minimum time required for covering all residential squares with the existing limitations. The proposed problem has a Minimax objective function that minimizes the maximum traveling time of each UAV in order to find the fastest possible mapping schedule. Each UAV belongs to a UAV base wherefrom its mission starts and ends, and battery replacement or refueling is

performed at the same base. The TD number of potential UAV bases can be opened and each UAV's mission time cannot exceed TM due to its endurance. To the best of our knowledge, this problem has not been studied thus far, and the formulations presented here are the first MILP mathematical models of the grid-based CPP problem. We call this problem multi-UAV multi-tour location coverage path planning (MUTLCPP). The application of this model is a post-earthquake mapping system due to the Minimax objective function, and a grid-based decomposition of the residential areas.

The remainder of this chapter is organized as follows: Section 3.2 represents the problem definition and two 4-index and 5-index mathematical models. Section 3.3 contains the extensions and techniques used to improve the performance of the branch and cut algorithm. Section 3.4 includes the test problems and computational results, and section 3.5 contains conclusions and future prospects of CPP.

3.2 Problem Definition and Mathematical Formulations

Associated with the MUTLCPP problem, the sets, parameters, and decision variables are defined as follows:

Sets:

S_C The set of residential squares that must be covered.

S_D The set of potential AUV base (depot) squares.

S_E The set of nonresidential squares.

U	The set of available UAVs.
L	The set of routes.
R	The set of moving directions from one square to the other.

Parameters:

t_{ij}	The travel time from square i to j
d_{ij}	The distance between square i and j
TD	Total number of UAV bases (depots) that asked to be open.
MT	The possible flight time of each UAV.

Variables:

w_{iju}^{lr}	Binary decision variable equals to 1, if UAV u exists from r side of square i to square j in route l, and 0 otherwise
w_{ij}^{ul}	Binary decision variable equals to 1, if UAV u travels from square i to square j in route l, and 0 otherwise
v_{ij}^{ul}	Positive integer variable determines how many nodes are left to cover in route l by UAV u, after visiting i and just before visiting j
y_i	Binary variable equals to 1, if potential depot i is open and 0 otherwise
x_{ij}^{lr}	Binary variable equals to 1, if a UAV travels from r side of square i to square j in route l

p_{iu}	Binary variable equals to 1, if UAV u belongs to depot i and 0 otherwise
g_{iu}^{lr}	Binary variable equals to 1, if a UAV enters from r side of i and exits from the opposite side in route l
g_{i}^{lr}	Binary variable equals to 1, if a UAV enters from r side of i and exits from the opposite side in route l
T	The maximum mission time of UAVs

The map of earthquake struck region was decomposed into equal sized squares, according to the UAV's camera footprint. The camera footprint (CF) depends on the camera sensor size (CSS), lens focal length (FL), and flight altitude (FA). Figure 3.3 shows the camera footprint, which can be calculated as follows.

$$CF = FA\frac{CSS}{FL} \tag{3.1}$$

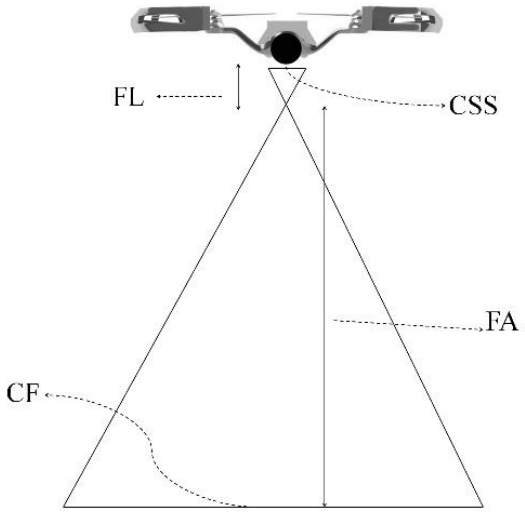


Figure 3.3: Camera footprint.

The camera plays a very important role in CPP problem especially for a post-earthquake situation. Despite the resolution demanded by the decision maker, the focal length of the lens and the sensor size of the camera have a great impact on the mapping schedule length. A smaller focal length with a larger sensor size ends up in a larger camera footprint (Equation (3.1)) and consequently less number of squares in the map and schedule length. Also, the horizontal flight disposition is preferred for covering residential areas by a sequence of overlapping images. Though, the new UAV systems allow changes in the viewing angle of the installed camera.

The Ground Sampling Distance (GSD) is a scale of resolution in aerial photogrammetry. It is the distance between two successive pixel centers on the ground. The smaller value of the GSD leads to the higher resolution of image and more differentiable details. At a fixed focal length, decreasing the altitude results in a smaller GSD and at a fixed flight altitude, by increasing the focal length the GSD value will decrease. In earthquake situation, the GSDs of several meters may suffice for damage assessment of non-urban regions. However, it is quiet insufficient when analyzing highly populated earthquake struck cities and detailed images of at least 1m GSD are essential [80].

However, the geographical mapping might not require quick scheduling process, but in post-earthquake mapping, time factor has a great importance. Since the use of UAVs for mapping is in its early days, there are potential improvements and considerable innovations related to unseen problems and technical challenges.

To ease the image processing, an overlap from each side is essential; thus, if the overlap is α% from each side, then the square edge must be equal to $(1-2\alpha)\%$ of the camera footprint length. The overlap is required to discover the common details among the images, however; the minimum and maximum overlaps depend on the post-image-processing software. Considering the fact that image-processing discussion is beyond the scope of this study, and there is no universally accepted overlap standard [81], we only mention that the overall overlap for a specified-path mapping must be at least 20% of the image width [82].

The MUTLCPP is defined on an undirected graph $G(S, A)$, where $S = S_O \cup S_D$ is a set of squares and $A = \{(i,j)|i,j \in S\}$ is the set of arcs. S_D is a set of potential depot squares and $S_O = S_C \cup S_E$, where S_C is the set of residential squares that must be covered and S_E is the set of empty squares. Non-negative travel time, t_{ij}, and travel distance, d_{ij}, are associated with each arc with a triangular inequality (i.e., $t_{ik} + t_{kj} \geq t_{ij}$). Moreover, the set of homogenous UAVs is denoted by $U = \{1,2,3,..,u\}$ and the set of routes by $L = \{1,2,3,\dots,l\}$. There is a set $R = \{Left, Right, Up, Down, Direct\}$ that specifies the leaving direction form one square to another. All the squares belonging to S_C must be covered, at least once, such that the UAV enters and exists from opposite sides (i.e., Right-Left, Left-Right, Up-Down, and Down-Up). Among the $|S_D|$ number of potential depots, TD number of them can be opened, according to the restrictions prescribed by the decision maker. The flight time limitation for each UAV is specified by MT. All UAVs start and end their tours at the same base. Because

time plays a vital role in the post-disaster response phase against cost, the proposed model must minimize the maximum mission time of all UAVs.

When M is a sufficiently large positive number, the 4-index and 5-index models of MUTLCPP are presented as follows:

3.2.1 4-index Formulation

Model:

$$Min\ T \tag{3.2}$$

The objective function (3.2) is to minimize the maximum mission time of UAVs. This objective function leads to the minimum mapping schedule length for all UAVs.

The MUCLTPP's constraints are as follows:

$$\text{S. t.} \quad \sum_{i \in S_D} y_i = TD \tag{3.3}$$

$$\sum_{i \in S_D} p_{iu} = 1 \quad \forall u \in U \tag{3.4}$$

Constraint (3.3) ensures that the total number of open bases is TD, and constraint (3.4) imposes that each UAV has to be allocated to one depot only.

$$\sum_{i \in S} \sum_{u \in U} w_{ij}^{ul} \leq 1 \quad \forall j \in S, i \neq j, l \in L \tag{3.5}$$

Constraint (3.5) ensures that each square is passed at most once in each route.

$$\sum_{j\in S} w_{ij}^{ul} - \sum_{j\in S} w_{ji}^{ul} = 0 \quad \forall i \in S, i \neq j, l \in L, u \in U \tag{3.6}$$

Constraint (3.6) guarantees that if a UAV enters a square then it must leave that square. In the other words, total number of entrance to a square is equal to the total number of existence from that square.

$$\sum_{j\in S_O} \sum_{l\in L} \sum_{u\in U} w_{ij}^{ul} - My_i \leq 0 \quad \forall i \in S_D \tag{3.7}$$

Constraint (3.7) assures that the UAVs exit from an open base only. This constraint forces the w_{ij}^{ul} variable to take the value of zero if the base is close ($y_i = 0$).

$$v_{ij}^{ul} - Mw_{ij}^{ul} \leq 0 \quad \forall i\&j \in S, i \neq j, u \in U, l \in L \tag{3.8}$$

Constraint (3.8) assigns zero to v_{ij}^{ul} variables according to the movement variable of w_{ij}^{ul}.

$$\sum_{i\in S} v_{ij}^{ul} - \sum_{i\in S} v_{ji}^{ul} - \sum_{i\in S} w_{ij}^{ul} = 0 \quad \forall j \in S_O, i \neq j, u \in U, l \in L \tag{3.9}$$

Constraint (3.9) indicates that if a UAV pass through square $i \in S_O$, then the remaining number of the nodes in route l will decrease by one.

$$\sum_{i\in S}\sum_{j\in s, j\neq i}\sum_{l\in L} t_{ij} w_{ij}^{ul} \leq MT \quad \forall l \in L \tag{3.10}$$

Constraint (3.10) ensures that the duration of each route does not exceed the UAV's flight time limitation MT.

$$\sum_{i\in S}\sum_{j\in s, j\neq i}\sum_{l\in L} t_{ij} w_{ij}^{ul} + \sum_{i\in S_D}\sum_{j\in S_O}\sum_{l\in L} ST w_{ij}^{ul} \leq T \quad \forall u \in U \tag{3.11}$$

The value of T which is the maximum mission time of UAVs is determined by constraint (3.11).

$$\sum_{j\in S_O}\sum_{l\in L} w_{ij}^{ul} - M p_{iu} \leq 0 \quad \forall i \in S_D, u \in U \tag{3.12}$$

$$\sum_{i\in S_D}\sum_{j\in S_O}\sum_{u\in U} w_{ij}^{ul} \leq 1 \quad \forall l \in L \tag{3.13}$$

Constraint (3.12) indicates that UAVs can only leave their own depot and constraint (3.13) eliminates the illegal routes.

$$\sum_{l\in L}\sum_{r\in R, r\neq Direct}\sum_{j\in S} x_{ij}^{lr} = 0 \quad \forall i \in S,\ d(i,j) > dm \tag{3.14}$$

Constraint (3.14) prevents the formation of arcs with the length more than dm for the horizontal and vertical directions. dm is the distance between the center of two consecutive squares; the length of horizontal or vertical arcs cannot exceed it.

$$\sum_{j \in S} \left(x_{ij}^{lr} + x_{ji}^{lr}\right) - 2g_i^{lr} \geq 0 \quad \forall i \in S_C, i \neq j, l \in L, r \in R | r \neq Direct \tag{3.15}$$

Constraint (3.15) specifies that the value of g_i^{lr} can be one if the entrance and exit are in the same horizontal or vertical directions.

$$\sum_{l \in L} \sum_{r \in R | r \neq Direct} g_i^{lr} \geq 1 \quad \forall i \in S_C \tag{3.16}$$

$$\sum_{r \in R} x_{ij}^{lr} - \sum_{u \in U} w_{ij}^{ul} = 0 \quad \forall i \neq i, j \in S, l \in L \tag{3.17}$$

Constraint (3.16) ensures that all residential squares must be covered by a UAV in either horizontal or vertical directions and constraint (3.17) defines the value of variable x_{ij}^{lr}.

$$x_{ij}^{l\,Up} = 0 \quad \forall i, j \in S | i - 1 \leq j, l \in L \tag{3.18}$$

$$x_{ij}^{l\,Down} = 0 \quad \forall i, j \in S | i + 1 \geq j, l \in L \tag{3.19}$$

$$x_{ij}^{l\,Left} = 0 \quad \forall i, j \in S | i \neq j + 1, l \in L \tag{3.20}$$

$$x_{ij}^{l\,Right} = 0 \quad \forall i, j \in S | i \neq j - 1, l \in L \tag{3.21}$$

Constraints (3.18)–(3.21) imposes the value of zero for the wrong horizontal and vertical directions. For example equation (3.18) ensures that the travel in the upward (up) direction from i to j is zero unless $i - 1$ is less than or equal j. The travel

from one square to its right square must be impossible for the *up*, *down*, and *left* directions and it is shown in equation (3.21).

$$w_{ij}^{ul}, x_{ij}^{lr} \in \{0,1\} \quad \forall i\&j \in S, u \in U, l \in L, r \in R \tag{3.22}$$

$$y_i, p_{iu}, g_j^{lr} \in \{0,1\} \quad \forall i \in S_D, j \in S_C, l \in L, r \in R \tag{3.23}$$

Finally, constraints (3.22) and (3.23) are used as the integrality constraints.

3.2.2 5-index Formulation

The 5-index model development is presented in the same structure of the 4-index formulation as follows:

Min (3.1) S. t (3.2), (3.3), and

$$\sum_{i \in S} \sum_{r \in R} \sum_{u \in U} w_{iju}^{lr} \leq 1 \quad \forall j \in S, i \neq j, l \in L \tag{3.24}$$

$$\sum_{j \in S} \sum_{r \in R} w_{iju}^{lr} - \sum_{j \in S} \sum_{r \in R} w_{jiu}^{lr} = 0 \quad \forall i \in S, i \neq j, l \in L, u \in U \tag{3.25}$$

$$\sum_{j \in S_O} \sum_{l \in L} \sum_{u \in U} \sum_{r \in R} w_{iju}^{lr} - M y_i \leq 0 \quad \forall i \in S_D \tag{3.26}$$

$$v_{ij}^{ul} - \sum_{r \in R} M w_{iju}^{lr} \leq 0 \quad \forall i\&j \in S, i \neq j, u \in U, l \in L \tag{3.27}$$

$$\sum_{i \in S} v_{ij}^{ul} - \sum_{i \in S} v_{ji}^{ul} - \sum_{i \in S} \sum_{r \in R} w_{iju}^{lr} = 0 \quad \forall j \in S_O, i \neq j, u \in U, l \in L \tag{3.28}$$

$$\sum_{i\in S}\sum_{j\in s,j\neq i}\sum_{l\in L}\sum_{r\in R} t_{ij}\, w_{iju}^{lr} \leq MT \quad \forall l \in L \tag{3.29}$$

$$\sum_{i\in S}\sum_{j\in s,j\neq i}\sum_{l\in L}\sum_{r\in R} t_{ij} w_{iju}^{lr} + \sum_{i\in S_D}\sum_{j\in S_O}\sum_{l\in L}\sum_{r\in R} ST w_{iju}^{lr} \leq T \quad \forall u \in U \tag{3.30}$$

$$\sum_{j\in S_O}\sum_{l\in L}\sum_{r\in R} w_{iju}^{lr} - Mp_{iu} \leq 0 \quad \forall i \in S_D, u \in U \tag{3.31}$$

$$\sum_{i\in S_D}\sum_{j\in S_O}\sum_{u\in U}\sum_{r\in R} w_{iju}^{lr} \leq 1 \quad \forall l \in L \tag{3.32}$$

$$\sum_{l\in L}\sum_{u\in U}\sum_{j\in S}\sum_{r\in R, r\neq Direct} w_{iju}^{lr} = 0 \quad \forall i \in S,\ d(i,j) > dm \tag{3.33}$$

$$\sum_{j\in S}\left(w_{iju}^{lr} + w_{jiu}^{lr}\right) - 2g_{iu}^{lr} \geq 0 \quad \forall i \in S_C, i \neq j, l \in L, u \in U, r \in R | r \neq Direct \tag{3.34}$$

$$\sum_{l\in L}\sum_{r\in R|r\neq Direct}\sum_{u\in U} g_{iu}^{lr} \geq 1 \quad \forall i \in S_C \tag{3.35}$$

$$w_{iju}^{l\,Up} = 0 \quad \forall i, j \in S | i - 1 \leq j, l \in L, u \in U \tag{3.36}$$

$$w_{iju}^{l\,Down} = 0 \quad \forall i, j \in S | i + 1 \geq j, l \in L, u \in U \tag{3.37}$$

$$w_{iju}^{l\,Left} = 0 \quad \forall i, j \in S | i \neq j + 1, l \in L, u \in U \tag{3.38}$$

$$w_{iju}^{l\,Right} = 0 \quad \forall i, j \in S | i \neq j - 1, l \in L, u \in U \tag{3.39}$$

$$w_{iju}^{lr} \in \{0,1\} \quad \forall i \& j \in S, u \in U, l \in L, r \in R \tag{3.40}$$

$$y_i, p_{iu}, g_{ju}^{lr} \in \{0,1\} \quad \forall i \in S_D, j \in S_C, l \in L, r \in R, u \in U \tag{3.41}$$

Constraints (3.24)–(3.41) are the same constraints as (3.5)–(3.16) and (3.18)–(3.23) in the 4-index formulation. In this problem, the size of $|\boldsymbol{L}|$ is problem dependent, and an increase in the number of routes results in an exponential increase in the execution time of the problem. The minimum size of $|\boldsymbol{L}|$ equals to the size of $|\boldsymbol{U}|$; the maximum size can be calculated according to any feasible solution that the problem solver decides. However, it is preferable to use solutions that connect the squares (horizontally or vertically) with minimum turns.

3.3 Methodology

The 4-index and 5-index models were coded in GAMS and the Cplex solver was selected. To improve the algorithm and make it more effective, some extensions and techniques were found to be appropriate for our problem. Branch-and-Bound is the main algorithm used for solving MIP problems in Cplex. Using well-known techniques based on the model's features can effectively reduce the execution time of the problem and improve the performance of MIP for effective collaborative decision-making in disaster management. The following subsections shortly describe the techniques that were effective for the models.

3.3.1 Solver Generated Cuts and Heuristics

In the early sixties, Gomory's fractional cuts were introduced by Gomory [83]. The procedure were considered impractical until a study [84] proved that Gomory's cuts can be effectively distributed over different branches of the search tree in a branch and cut framework for a mixed 0–1 problem. In the present times, most optimizers use

Gomory's cuts for MILP problems. In the Cplex optimizer, the fraccuts option decides whether or not Gomory's fractional cuts should be generated for the problem; and the parameters 0, 1, and 2 control the cuts to be generated automatically, moderately, and aggressively, respectively. In the case of lack of progress in the best bound, one effective solution uses more aggressive levels of cut generation to further tighten the formulation [85]. As initial observations confirm the lack of progress in the best-found-bound, parameter 2 (aggressive cut generation) was found to be more effective for the proposed models. According to the structure of the mathematical formulations, cover cuts are also useful. Therefore, Gomory's fractional cuts and cover cuts were aggressively generated for the models.

GAMS (General Algebraic Modeling System) is a high level modeling language designed for programming the linear, nonlinear, and mixed integer optimization models. GAMS can support different forms of linear and nonlinear programming problems. The system has a language compiler and an integrated high-performance commercial solver, e.g., Cplex. Cplex is a famous commercial optimizer that can solve linear, mixed integer, quadratic, and quadratically constrained programming problems. GAMS provides access to Cplex solver with a variety of options to interact with it and improve the solution procedure. Cplex uses several alternative algorithms such as primal simplex, dual simplex, barrier, or sifting algorithms. For mixed integer programming problem Cplex uses branch-and-cut algorithm.

Cplex takes advantage of different heuristic techniques to avoid tree pollution, increasing the diversity of branch and bound search, achieving integer feasible solution in shorter time, and fastening the final optimality. Based on the characteristics of the proposed problem, two heuristic features from the different heuristic procedures offered by Cplex were utilized, i.e., Local Branching heuristic (LBHeur), and Node heuristic (NHeur). The LBHeur was introduced by Fischetti & Lodi [86] and has been widely used by commercial optimizers. We can infer the following facts based on the ILOG Cplex explanations:

- LBHeur attempts to enhance new incumbents found throughout an MIP search. The LBHeur algorithm will be appealed only if a new incumbent is identified. In cases where several incumbents are identified at a single node, the last one will be considered.
- NHeur implements the techniques to find a feasible solution from the present node in branch and bound algorithm. The parameter HeurFreq restricts this offer. A positive number specifies the frequency of invoking NHeur (in the number of nodes).

3.3.2 Additional Constraints

User cuts or additional constraints are the polynomial constraints that strengthen the mathematical formulation and reduce the solution space. These cuts are not supposed to affect the feasible solutions space. The following constraints eliminate unfavorable fractional solutions and expedite the execution process:

$$\sum_{i \in S_D} \sum_{j \in S_O} v_{ij}^{ul} - \sum_{i \in S} \sum_{j \in S_O} \sum_{r \in R} w_{iju}^{lr} = 0 \quad \forall u \in U, l \in L \tag{3.42}$$

$$\sum_{i \in S_O} \sum_{j \in S_D} \sum_{u \in U} \sum_{l \in L} v_{ij}^{ul} = 0 \tag{3.43}$$

Constraint (3.42) imposes that the total number of squares that are to be visited by UAV u in route l right after leaving the depot is equal to the total number of squares allocated to UAV u in route l. Constraint (3.43) ensures that the value is equal to 0 just before ending the tour at the depot. Constraints (3.9), (3.42), and (3.43) are adapted from a previous study [87].

$$p_{iu} - y_i \leq 0 \quad \forall i \in S_D, u \in U \tag{3.44}$$

Constraint (3.44) indicates that if a UAV belongs to a depot, then the depot must be open.

$$\sum_{\boldsymbol{i \in S_D}} \sum_{l \in L} \sum_{u \in U} \sum_{r \in R} \sum_{j \in S_O} w_{iju}^{lr} \leq |L| \tag{3.45}$$

Constraint (3.45) ensures that the total number of arcs leaving the depots is less than or equal to the number of the available routes.

$$\sum_{i \in S_N} \sum_{l \in L} \sum_{u \in U} \sum_{r \in R | r \neq direct} \sum_{j \in S_N} w_{iju}^{lr} = 0 \tag{3.46}$$

According to the logic of the problem and the grid-based decomposed map, the UAVs must travel directly through the nodes of the S_E set, and it is applied in constraint (3.46). As mentioned earlier, only the residential squares must be covered by horizontal or vertical movements. In other words, in favor of the objective function value, and in order to find the minimal path, the UAVs should travel in the *direct* direction while passing through the nonresidential squares.

$$\sum_{r\in R}\sum_{u\in U}\sum_{i\in S_C} g_{iu}^{lr} - \sum_{i\in S}\sum_{j\in S_C}\sum_{u\in U}\sum_{r\in R} w_{iju}^{lr} \leq 0 \qquad \forall l \in L \tag{3.47}$$

Constraint (3.47) ensures that the number of times a UAV enters a square in each route is more than or equal to the number of times the square is covered in that route.

3.3.3 Variable Branching Priority

The variable branching priority based on the mathematical formulation's specifications is one of the most efficient techniques that fastens the solution procedure and tightens the model. Without using this feature, the solver automatically decides the appropriate variable to be branch on. It is imperative to select a suitable priority among the existing variables depending on the variables' dependence, relevance, and significance in the proposed mathematical model. In this study there are 4-index and 5-index mathematical formulations of a single problem. In this study, 4-index and 5-

index mathematical formulations were used for a single problem. Based on the model's characteristics and variables' significance, the following priorities were suggested:

I. For the 4-index model, the variable branching priority order must be g_i^{lr}, x_{ij}^{lr}, w_{ij}^{ul}, v_{ij}^{ul}, p_{iu} and y_i.

II. The branching priority of the 5-index model should be g_i^{lr}, v_{ij}^{ul}, w_{iju}^{lr}, p_{iu}, and y_i.

3.4 Computational Experiments and Results

In this section, some instances are designed to test the validity of the models and features proposed in section 3. Based on the problem's characteristics, it is assumed that the number of available UAVs, number of possible open depot squares, and maximum limit for the total traveling time of each UAV are known. Each test instance was named in the X-Y-Z-L format, where X refers to the total number of squares in the map, Y refers to the total number of residential squares, Z represents the number of potential UAV bases, and L is the number of available UAVs. The test instances are shown in Figure 3.4. For instance, in problem 36-11-2-3 map, where 36 is the total number of squares, 11 squares are residential areas, 2 squares are potential depots, and 3 represents the available UAVs.

It is assumed that each small square side is one in terms of length unit and the speed of the UAVs is also one in terms of speed unit.

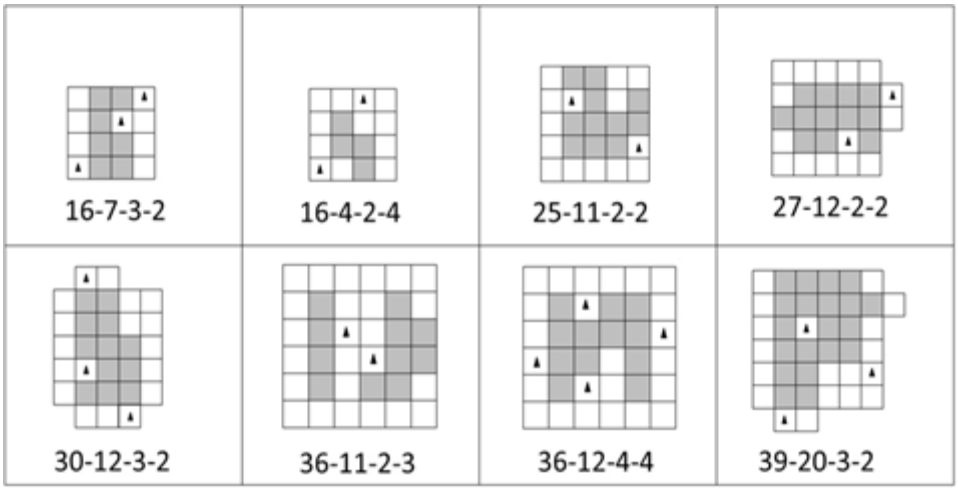


Figure 3.4: Test instance maps (the gray squares are assumed to be residential areas that should be covered by UAVs and squares with black triangles are potential UAV bases).

The mathematical models were coded in GAMS 24.7.3 and solved using the Cplex 12.6.3.0 solver. The models were run on a laptop with the configuration of Intel® Core™ i7–2620M CPU 2.70GHz, RAM 8.00 GB, and windows 10 Enterprise operating system.

To evaluate the efficiency of the techniques mentioned in section 3.3, each technique was considered alone in an individual run at the first step. Subsequently, the combination of efficient ones was used. All the problems were solved in a 5-h time limit, irrespective whether a feasible or optimal solution was found or not. The reason for the 5-h time limit is that no significant improvement would occur by increasing this limit. The calculation time increases exponentially by increase in the number of squares or the number of routes. Tables 3.1 and 3.2 show the results of the 4-index and 5-index models, respectively, in a combination of techniques mentioned in section 3.3.

Table 3.1: Effects of techniques described in section 3.3.1, 3.3.2, and 3.3.3 on the 4-index mathematical model

4-index model															
	Original model			3.3.1 ,3.3.2			3.3.1 , 3.3.3			3.3.2, 3.3.3			3.3.1 , 3.3.2 , 3.3.3		
Instance	CPU	Results	Gap (%)	CPU	Results	Gap (%)	CPU	Results	Gap (%)	CPU	Results	Gap (%)	CPU	Results	Gap (%)
16-7-3-2	254	21.3	0	136	21.3	0	281	21.3	0	216	21.3	0	284	21.3	0
16-4-2-4	18000	13.15	15.2	3850	13.15	0	4300	13.15	0	5400	13.15	0	3650	13.15	0
25-11-2-2	1400	24	0	310	24	0	343	24	0	162	24	0	84	24	0
27-12-2-2	18000	35.98	35.1	1325	28	0	2104	28	0	1206	28	0	1618	28	0
30-12-3-2	12587	26.32	0	2877	26.32	0	2754	26.32	0	3107	26.32	0	2639	26.32	0
36-11-2-3	18000	NA	-	18000	NA	-	18000	NA	-	18000	NA	-	18000	NA	-
36-12-4-4	18000	NA	-	18000	NA	-	18000	NA	-	18000	NA	-	18000	24.4	59.4
39-20-3-2	18000	NA	-	18000	NA	-	18000	NA	-	18000	NA	-	15669	36	0

Table 3.2: Effects of techniques described in section 3.3.1, 3.3.2, and 3.3.3 on the 5-index mathematical model

5-index model															
	Original model			3.3.1 ,3.3.2			3.3.1 , 3.3.3			3.3.2, 3.3.3			3.3.1 , 3.3.2 , 3.3.3		
Instance	CPU	Results	Gap (%)	CPU	Results	Gap (%)	CPU	Results	Gap (%)	CPU	Results	Gap (%)	CPU	Results	Gap (%)
16-7-3-2	888	21.3	0	467	21.3	0	782	21.3	0	159	21.3	0	**129**	21.3	0
16-4-2-4	11320	13.15	0	1359	13.15	0	5527	13.15	0	650	13.15	0	**479**	13.15	0
25-11-2-2	850	24	0	286	24	0	186	24	0	226	24	0	**48**	24	0
27-12-2-2	18000	30.82	30.9	966	28	0	18000	30.24	30.1	570	28	0	**559**	28	0
30-12-3-2	6875	26.32	0	2472	26.32	0	3830	26.32	0	2958	26.32	0	**1421**	26.32	0
36-11-2-3	18000	NA	-	18000	NA	-	18000	NA	-	18000	21.3	13.7	18000	**20.79**	6.3
36-12-4-4	18000	NA	-	18000	NA	-	18000	NA	-	18000	NA	-	18000	**22.82**	49.2
39-20-3-2	18000	NA	-	18000	42.82	27.1	18000	NA	-	18000	36	11.7	**10554**	36	0

According to the results presented in Tables 3.1 and 3.2, it can be concluded that the 5-index model shows better solutions in terms of CPU time and the absolute gap. Furthermore, as previously anticipated, the combination of the proposed LP strengthening techniques offered better solutions in a lesser amount of time. The CPU time of the 5-index problem 16-4-2-4 was 11320 seconds in the original model, which decreased to 479 seconds by adding the techniques. In addition, for problem 36-12-4-4 a feasible solution was not obtained in the original model; however, with a combination of techniques, a solution with 6.3% relative gap was acquired. Furthermore, while no feasible solution was found for original model of problem 39-20-3-2 in the 5-h time limit, the optimal solution was found under the influence of the techniques in 10554 seconds. Figure 3.5 depicts the optimal solutions for problem 39-20-3-2 and 36-11-2-3 test instances as examples.

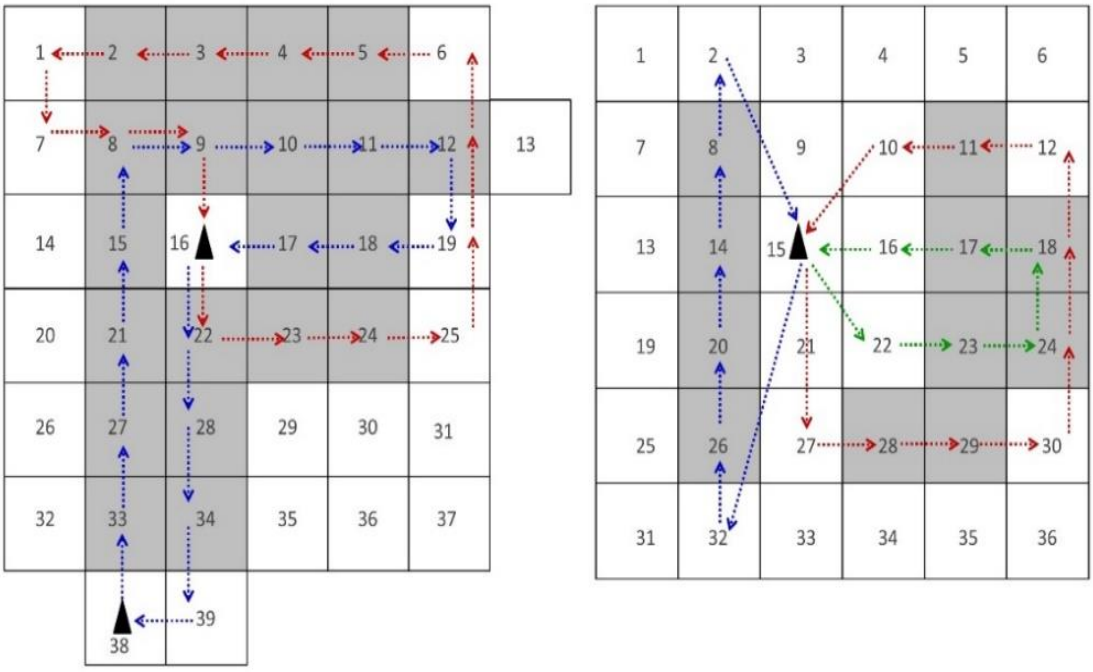


Figure 3.5: The optimal solutions of problems 39-20-3-2 and 36-11-2-3.

3.5 Conclusion and Future Prospects

A multi-UAV and multi-tour CPP problem based on the grid map decomposition was investigated. MUTLCPP supports the application of post-earthquake CPP where high accuracy is required. Moreover, due to the speedy decrease in the survival rate in post-earthquake situations, the Minimax objective function was considered for the optimization problem. Two mathematical models of MUTLCPP were presented and coded in GAMS, and the 5-index model outperformed the 4-index formulation in terms of solution quality and execution time. The models were improved by additional constraints and other techniques described in section 3.3. On the basis of the results of the test instances, it can be affirmed that the combination of the proposed techniques sped up the execution procedure and gained solutions with a lower gap in the lesser amount of time. In the small size instance of 16-7-3-2, the combination of techniques decreased the calculation time by 85% for the 5-index model but increased by 7% for the 4-index formulation. The results depicted in Tables 3.1 and 3.2 show that the linear strengthening techniques are more effective for the 5-index model. Based on the results of Tables 3.1 and 3.2, the proposed techniques decreased the CPU time by 85%, 95%, 94%, 96%, and 79% for the first five instances of the 5-index model and decreased the time by -7%, 79%, 94%, more than 91%, and 79% for the corresponding five instances of the 4-index model. For the further three instances, while the solution is not available in the original models, by the intervention of the linear strengthen techniques optimal solutions or very good feasible solutions were found. Generally, by comparing the results (Tables 3.1 and 3.2), it is clear that 5-index model performs much better that the 4-index formulation. Another conclusion based on the obtained results is that the solver generated cuts (3.3.1) have less effect,

and additional constraints (3.3.2) have the maximum effect on decreasing the CPU time.

There are several interesting directions for further investigations. Some new valid inequalities can be added to the model to strengthen the MILP formulations. In addition, exact methods such as the lagrangian relaxation can be developed to find the optimal or near-optimal solutions to large real size problems. The most interesting subject for further investigations is designing a heuristic or meta-heuristic algorithm to solve large-scale problems of MUTLCPP. Each randomly generated individual solution in a heuristic algorithm can be represented in such a way that each cell takes an integer < 6 indicating the UAV direction (set R). Finally, one can extend the problem by reflecting the map rotation possibility to test the different covering directions.

Chapter 4

BI-OBJECTIVE COVERING TOUR LOCATION ROUTING PROBLEM WITH REPLENISHMENT AT INTERMEDIATE DEPOTS

4.1 Introduction

Location routing problem (LRP) is one of the most important problems in logistics. LRP combines the two fundamental planning challenges: Facility Location Problem (FLP) and Vehicle Routing Problem (VRP). In purview of earlier studies [88], contemplating facility location and routing independently may result in sub optimal solutions. LRP refers to find the optimal location of facilities and vehicle routes in order to supply the customers under different real world constraints. In last few decades, LRP is intensively studied due to its several variants which have promising applications in areas such that product distribution, relief distribution, parcel delivery, etc. Details on comprehensive surveys of variants and solution algorithms can be found elsewhere [89], [90] .

Dantzig and Ramser [91] introduced VRP and then this has been investigated widely in the past decades owing to numerous applications in logistics and transportation. In VRP, a fleet of vehicles start and end their tour at a single depot while serving all customers on the route with an objective of minimizing the total traveling distance. Depending on the application, several single or multi objective problems have been studied. Moreover, there are different variants of VRP addressing

real world situations such as time windows for the customers, split delivery option, periodic schedule for vehicles, pick-up and delivery case, etc. The interested readers can refer to earlier studies [92]–[95]. Unlike the classical vehicle routing problem that considers a single central depot, in multi depot vehicle routing problem (MDVRP), there are several depots with available fleet of vehicles and by that the customers can be served from any of the depots. Montoya-Torres et al. [96] conducted an extensive survey of MDVRP and investigated the variants of single and multi-objective problems by employing different solution algorithms.

In the literature on LRP, it is noticed that most of the studies regard the classical MDVRP, wherein each vehicle starts and ends its tour at the same depot, in real life situations, however, a vehicle can start from one depot and ends at another. Furthermore, the vehicles can reload at any depot and they can continue serving the customers up to a defined time or distance limitation. This problem is known as vehicle routing problem with intermediate replenishment facilities (VRPIRF) which was introduced by Crevier et al. [97] such that a vehicle starts and ends the rotation at the same depot but it can reload at intermediate depots.

Nowadays waiting time for individuals in service industries and emergency situations plays an important role in customer satisfaction. Especially in emergency logistics, the delay is highly undesirable. This study presents a new model of location routing scheduling with humanitarian objectives for emergency situations. In addition, the problem has a wide range of applications in parcel delivery and other service based industries due to importance of routing and scheduling concepts. Among the several possible practical applications, we concentrate on post-earthquake relief distribution planning. Due to lack of time in emergency situations, the vehicles are not able to serve

all customers by directly visiting their nodes. Besides, depending on the limited available resources, we might lose some customers in need.

Unlike the usual routing models that focus on single trip tours, in our problem, the vehicles remain in the service up to a time limit, and they can be reloaded by any number of times during the service hours. There are some potential depot nodes and a known number of these depots can be opened for scheduling. We call the path connecting two depots (same or different) a *route*, and the sequence of the routes belonging to a vehicle a *trip*. The vehicles can start their trip from an open depot, then serve the customers, reload at any of the open depots for next set of clients, and can end their trip at any available open depot. To overcome the problem of the vehicle imbalance for further planning, the number of vehicles starting their trip at a depot will be kept equal to the number of vehicles ending their trip at that depot. There will be three types of demand nodes: *in-tour* demand nodes, *out-tour* demand nodes, and *lost* demand nodes. The in-tour demand nodes are those that are located within the vehicles' routes. Out-tour demand nodes are not located on the route of vehicles but they have an in-tour demand node in their walking distance and they must be allocated to their nearest in-tour demand node. If a demand node is neither in-tour nor out-tour, then it is a lost demand node. The vehicles satisfy the demands up to a time limit by the objectives of minimizing the total un-served demands and the total weighted waiting time of the served customers. In this study, we try to answer following questions: (1) which potential depots must be open; (2) which depot should be the start point of a vehicle; (3) which depot should be the next reloading station of a vehicle; (4) what should be the sequence of visiting the customers by the vehicles; (5) how much total demand should be assigned to each demand node; (6) what should be the load amount of each vehicle right after living the depots (the loading/reloading

amounts at depots) and subsequently between a pair of nodes ; (7) which are the in-tour, out-tour (by their allocation destination), and lost customers; (8) what is the waiting time and weighted waiting time at each demand node; and (9) which depot needs to be the final destination of each vehicle. Fig depicts an example of a feasible solution of CLRPR with 20 demand nodes, wherein 12 nodes are in-tour, seven nodes are out-tour, and two are lost demand nodes. There are four potential depot nodes and two of them have been asked to be open. Totally two vehicles are available; and each starts from one depot and ends at the other. To the best of our knowledge this problem has not been proposed in any of the earlier studies.

In this article, the bi-objective integer linear programming model of CLRPR has been coded in GAMS and solved with CPLEX solver by epsilon-constraint method. Likewise NSGAII algorithm with two improvements is operated to solve the different size test problems.

The remainder of the chapter is structured as follows: Section 4.2 surveys the related literature of the problem. The mathematical model is presented in Section 4.3 and it is solved optimally for a very small size problem. Section 4.4 contains the solution algorithms with expanded explanation. Numerical test problems and analysis of the algorithms' results are provided in Section 4.5. At the end in Section 4.6, conclusion and some suggestions for future studies are offered.

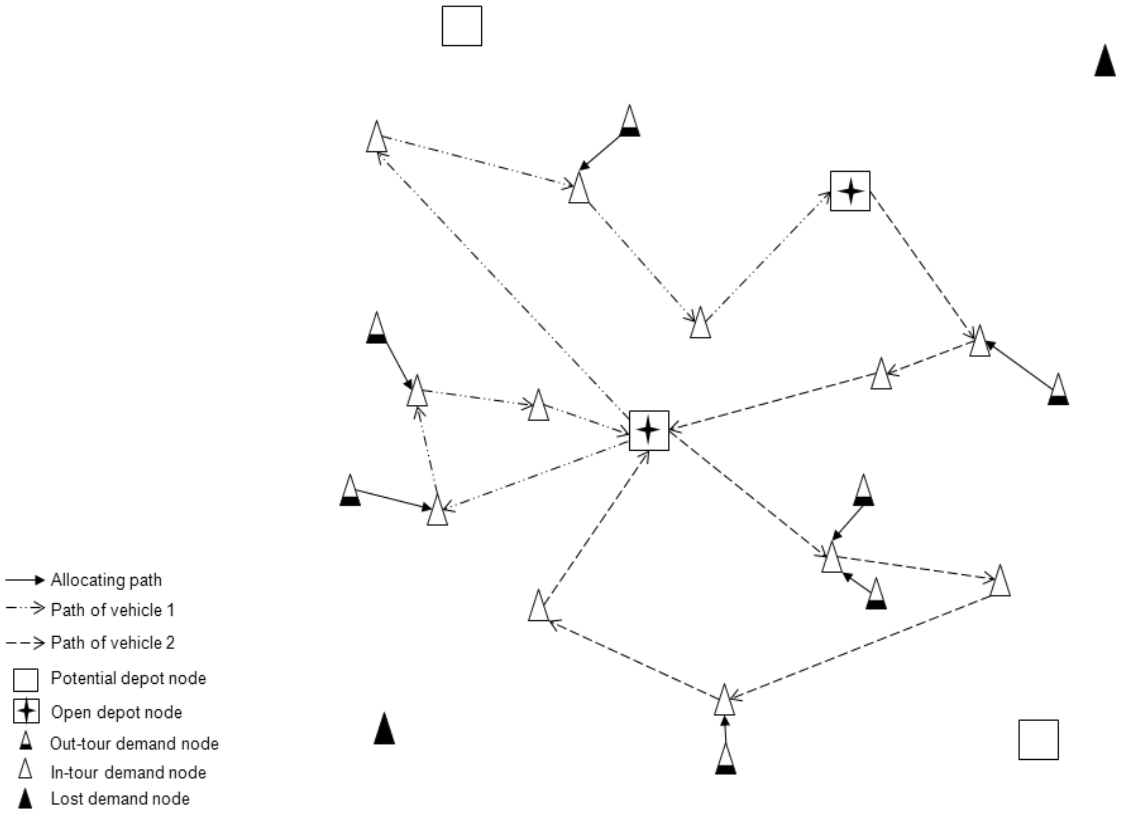


Figure 4.1: An example of CLRPR solution.

4.2 Related Literature

Vehicle routing and scheduling problem (VRSP) have several variants already investigated (i.e., VRSP with time window constraints [98], VRSP with cross docking [99] and split delivery [100], and emergency VRSP [101]). Gan et al. [102] presented a multi-trip split delivery VRSP, Yan et al. [103] developed a model with time-space network technique for cash transportation scheduling and routing problem, and Yan, Chu, Yan et al. [104] presented a multi-trip split delivery VRSP. Readers can also refer to [105] for an exhaustive review of time dependent routing problems.

Covering routing problem was introduced by Current and Schilling [13] and extended by Gendreau et al. [67]. It has been utilized by the researchers for various real life circumstances. Hodgson et al. [106] studied a multi-vehicle covering tour problem focused on supplying of the healthcare facilities with mobile units in Ghana

with an objective of minimizing the total traveling distance while serving all customers located within a predefined distance. Vogt *et al.* [107] presented a single vehicle routing allocation problem and a solution procedure of tabu search algorithm. In this problem, the start and end nodes of a vehicle are known, the customers not visited by the vehicle can be allocated to another customer on the route of the same vehicle or they can be eliminated. The objective functions are to minimize the weighted sum of routing, and allocating and eliminating costs. The decision for allocating the customers that are not located on the rout is taken by the model. The authors considered an application of medical care services in rural areas. Nolz *et al.* [109] proposed a bi-objective covering tour problem for water distribution during post disaster situations. Tricoire *et al.* [110] studied a two-stage stochastic covering tour problem for disaster relief operations and, as a case study, they applied the model to a western Senegal region. Finally, Allahyari et al. [14] developed a multi depot covering tour VRP and presented two mixed-integer linear programming model. In their model, the customers with demand can be visited directly on tour or individually be allocated to other customers on tour. The objective functions of their model are to minimize total traveling distance of vehicles and total allocating cost of individuals.

VRP with intermediate replenishment facilities gained less attention due to complexity in modeling and solution algorithms. The problem was introduced by Crevier et al. [97] for a real life grocery distribution planning in Canada. In their problem the vehicle starts at a depot and reloads at the same or another depot and eventually returns to the starting depot. The objective function was to minimize the total traveling duration time; and a tabu search heuristic was developed as the solution algorithm. Later Tarantilis *et al.* [111] developed a three-step hybrid guided local search algorithm for the proposed problem. Kek *et al.* [112] addressed the new version

of the problem with the flexible assignment of start and end depots. They developed two mathematical models for capacitated vehicle routing problem, one for fixed and other for flexible start and end depots. The objective functions were to minimize the traveling distance and fixed cost of vehicles. By providing few numerical examples, they showed that the mentioned flexible assignment has a cost saving in range of 0 to near 50%. They used the concept of dummy depot nodes to cope with the reloading formulation difficulty; however, each reload involves inclusion of a new set of dummy depot nodes that, leads to an exponential raise in complexity of the problem. Another key limitation they stated is the vehicle imbalance throughout the depots.

Humanitarian logistics as Campbell *et al.* [113] noted, usually concentrate on minimizing loss of life and suffering unlike cost minimizing problems. Therefore, they emphasized that arrival time of relief commodities at the affected areas clearly impacts the survival rate of people and degree of suffering. They also claimed that the min-sum objective (minimizing the sum of arrival times) performs better for all customers compared to the min-max objective (minimizing the latest time of receiving service). In our study, considering the humanitarian intention, the objective functions are kept to minimize the total amount of lost demand and the total weighted waiting time of affected population.

4.3 Problem Description and Mathematical Formulation

The network design of CLRPR is formally described as follows: Let $G(S, A)$ be an undirected graph in which $S = S_D \cup S_C$ is a set of nodes, where $S_D = \{1,2, \dots, n\}$ is a set of potential depot nodes with the capacity of CD, and $S_C = \{n+1, n+2, \dots, n+m\}$ is the set of demand (customer) nodes. $A = \{(i,j) | i,j \in S\}$ is the set of edges; a nonnegative distance ds_{ij} and a nonnegative traveling time t_{ij} with a triangular inequality (i.e., $ds_{ij} + ds_{jk} \geq ds_{ik}$, $t_{ij} + t_{jk} \geq t_{ik}$) are associated with

each edge. Moreover, $K = \{1,2, \dots, k\}$ represents the set of homogenous vehicles with the capacity of *CV*. Each customer i node has a known demand d_i and can be visited at most once by a vehicle. Parameter R_{ij} for all $i, j \in S_C$ will be equal to 1, if node i is located within the walking distance of node j ($R_{ii} = 1$ for all demand nodes i) and 0 otherwise. A demand node i can be allocated to demand node j if j is the nearest in-tour demand node to i, and j is located within walking distance of i. The unserved (lost) demand nodes are those that would not be served directly by a vehicle and do not have any in-tour demand node in their walking distance. It is obvious that sum of the total demands assigned to a demand node must not violate the capacity of vehicle *CV*. We assume δ to be unloading time for each unit of demand at demand nodes, and ζ is the fix loading time at depot nodes. In the proposed model, the decision maker will ask to open *TD* number of depots among *n* number of potential depot locations. The total demands assigned to the vehicles must not violate the total open depot capacities. Moreover, the total load/reload amount of each vehicle is equal to the total demand assigned to that vehicle. The waiting time at each node should be less than or equal to a constant *TT*. At the end of the schedule, the numbers of vehicles ending their trip at a depot must be equal to the numbers of vehicles started their trip at the same depot. Furthermore, each vehicle must end its trip at a nearest possible depot (to the last served demand node), and this obligation does not affect the waiting time of customers. Finally *G* is defined as all subsets of potential depot nodes in a way that $0 \leq |G| \leq TD$.

The decision variables are described as follows:

X_{ijk}^{l} Binary decision variable equals to 1 if vehicle k moves form node i to node j and its previous load/reload was in depot l, and 0 otherwise.

V_{ijp}^{k} Binary decision variable equals to 1 if $i \in S_C$, $\in S_D$, and $p \in S_C$ are visited respectively by vehicle k, and 0 otherwise.

y_l Binary decision variable equals to 1 if depot l is open, and 0 otherwise.

O_k^l Binary decision variable equals to 1 if the initial start of vehicle k is depot l, and 0 otherwise.

e_{ij}^{k} Binary decision variable equals to 1 if the demand node i is assigned to demand node j which is visited by vehicle k, and 0 otherwise.

f_{ij}^{k} The load of vehicle k passing arc (i, j).

W_{ij}^{k} The waiting time of node j that is located right after node i on the route of vehicle k.

α_i The service time of node $i \in S$.

U_i The weighted waiting time of demand node i.

4.3.1 Mathematical Model

The bi-objective integer linear programming (ILP) model of CLRPR is formulated as follows:

$$\text{Min } Z_1 = \sum_{i \in S_C} d_i - \sum_{i \in S_c} \sum_{j \in S_c} \sum_{k \in K} e_{ij}^{k} d_i \tag{4.1}$$

$$\text{Min } Z_2 = \sum_{i \in S_C} U_i \tag{4.2}$$

Subject to:

$$\sum_{i \in S} \sum_{k \in K} \sum_{l \in S_D} X_{ijk}^{l} \leq 1 \quad \forall j \in S_C \tag{4.3}$$

$$\sum_{i \in S} X_{ijk}^{l} - \sum_{i \in S} X_{jik}^{l} = 0 \quad \forall j \in S_C, k \in K, l \in S_D \tag{4.4}$$

$$\sum_{i \in Sc} \sum_{k \in K} \sum_{l \in S_D} X_{ijk}^{l} - \sum_{i \in S_C} \sum_{k \in K} \sum_{l=j} X_{jik}^{l} = 0 \quad \forall j \in S_D \tag{4.5}$$

$$\sum_{j \in S_C} \sum_{k \in K} X_{ljk}^{l} - Y_l \geq 0 \quad \forall l \in S_D \tag{4.6}$$

$$M_1 \left(\sum_{i \in S_C} \sum_{j \in G} \sum_{l \in S_D, l \notin g} X_{ijk}^{l} + \sum_{l \in G} O_k^{l} \right) - \sum_{i \in G} \sum_{j \in S_C} \sum_{l \in G} X_{ijk}^{l} \geq 0 \quad \forall k \in K \tag{4.7}$$

$$\sum_{l \in S_D} O_k^{l} = 1 \quad \forall k \in K \tag{4.8}$$

$$\sum_{l \in S_D} y_l = TD \tag{4.9}$$

$$\sum_{j \in S_C} \sum_{k \in K} e_{ij}^{k} \leq 1 \quad \forall\, i \in S_C \tag{4.10}$$

$$e_{ij}^{k} - R_{ij} \sum_{p \in S} \sum_{l \in S_D} X_{pjk}^{l} \leq 0 \quad \forall i, j \in S_C, k \in K \tag{4.11}$$

$$\sum_{k \in K} e_{ij}^{k}\, ds_{ij} - ds_{ip} - M_2 \left(1 - \sum_{q \in S_C} \sum_{k \in K} \sum_{l \in S_D} X_{qpk}^{l} \right) \leq 0 \quad \forall i, j, p \in S_C, i \neq j, i \neq p \tag{4.12}$$

$$\sum_{i \in s} \sum_{k \in K} \sum_{l \in S_D} X_{ijk}^{l} R_{pj} - \sum_{q \in S_C} \sum_{k \in K} e_{pq}^{k} \leq 0 \quad \forall p \neq j \in S_C \tag{4.13}$$

$$\sum_{l\in S_D} X_{ijk}^{l} t_{ij} + \sum_{l\in S_D} X_{pqh}^{l} t_{pq} - t_{iq} - t_{pq}$$

$$- M_2\left(\sum_{r\in S_C} V_{ijr}^{k} + \sum_{r\in S_C} V_{pqr}^{h}\right) \le 0 \quad \forall i \ne r, p \tag{4.14}$$

$$\ne r \in S_C \,\&\, j, q \in S_D | i \ne p, j \ne q \,\&\, k, h \in K | k$$

$$\ne h$$

$$\sum_{i\in S}\sum_{l\in S_D} X_{ijk}^{l} - e_{jj}^{k} = 0 \quad \forall\, j \in S_C, k \in K \tag{4.15}$$

$$\sum_{j\in S_C}\sum_{k\in K} f_{ij}^{k} - CDY_i \le 0 \quad \forall\, i \in S_D \tag{4.16}$$

$$\sum_{j\in S}\sum_{k\in K} f_{ji}^{k} - \sum_{j\in S}\sum_{k\in K} f_{ij}^{k} - \sum_{k\in K}\sum_{j\in S_C} e_{ji}^{k}\, d_j = 0 \quad \forall i \in S_C \tag{4.17}$$

$$f_{ij}^{k} - \sum_{l\in S_D} (CV - d_i) X_{ijk}^{l} \le 0 \quad \forall i, j \in S, k \in K \tag{4.18}$$

$$f_{ij}^{k} - \sum_{l\in S_D} X_{ijk}^{l} d_j \ge 0 \quad \forall i, j \in S, k \in K \tag{4.19}$$

$$\sum_{p\in S | p\ne i} W_{pi}^{k} + \alpha_i + t_{ij} - W_{ij}^{k} - M_3\left(1 - \sum_{l\in S_D} X_{ijk}^{l}\right) \le 0 \quad \forall i \in S_C, j \in S, k \in K \tag{4.20}$$

$$\sum_{p\in S_C} V_{ijp}^{k} - \sum_{l\in S_D} X_{ijk}^{l} \le 0 \quad \forall j \in S_D, i \in S_C, k \in K \tag{4.21}$$

$$\sum_{i\in S_C} V_{ijp}^{k} - X_{jpk}^{j} \le 0 \quad \forall j \in S_D, p \in S_C, k \in K \tag{4.22}$$

$$\sum_{i\in S_C}\sum_{p\in S_C} V_{ijp}^{k} - \sum_{p\in S_C} X_{jpk}^{j} + O_k^{j} = 0 \quad \forall j \in S_D, k \in K \tag{4.23}$$

$$W_{pi}^{k} + \alpha_i + t_{ij} - W_{ij}^{k} - M_3\left(1 - V_{pij}^{k}\right) \le 0 \quad \forall i \in S_D, p, j \in S_C | p \ne j, k \in K \tag{4.24}$$

$$W_{ij}^{k} - \left(t_{ij} + \zeta\right) \sum_{l \in S_D} X_{ijk}^{l} \geq 0 \quad \forall\, i \in S_D, j \in S_C, k \in K \tag{4.25}$$

$$\sum_{p \in S | p \neq i, p \neq j} \sum_{k \in K} W_{pj}^{k}\, d_i - U_i - M_4 \left(1 - \sum_{k \in K} e_{ij}^{k}\right) \leq 0 \quad \forall i, j \in S_C \tag{4.26}$$

$$\sum_{i \in S} \sum_{k \in K} W_{ij}^{k} \leq TT \quad \forall j \in S_C \tag{4.27}$$

$$\alpha_i - \sum_{j \in S_C} \sum_{k \in K} e_{ji}^{k} d_j\, \delta = 0 \quad \forall i \in S_C \tag{4.28}$$

$$\alpha_i = \zeta \qquad \forall i \in S_D \tag{4.29}$$

$$X_{ijk}^{l} \in \{0,1\} \quad \forall i, j \in S, k \in K, l \in S_D \tag{4.30}$$

$$V_{ijp}^{k} \in \{0,1\} \quad \forall i, p \in S, j \in S_D, k \in K \tag{4.31}$$

$$O_{k}^{l} \in \{0,1\} \quad \forall k \in K, l \in S_D \tag{4.32}$$

$$y_l \in \{0,1\} \quad \forall l \in S_D \tag{4.33}$$

$$f_{ij}^{k},\ W_{ij}^{k}, \alpha_i, U_p \geq 0 \quad \forall i, j \in S, p \in S_C, k \in K \tag{4.34}$$

Where M_1, M_2, M_3, and M_4 are large numbers with lower bounds, determined as follow:

$$\min(M_1) = m + 1 \tag{4.35}$$

$$\min(M_2) = 2\max\{ds_{ij}\} \tag{4.36}$$

$$\min(M_3) = m\left(\zeta + 2\max\{t_{ij}\}\right) + \sum_{p \in S_C} d_p \delta \tag{4.37}$$

$$\min(M_4) = M_2 max\{t_{ij}\} \quad (4.38)$$

In the proposed model, objective function (4.1) minimizes the amount of lost demands, while objective function (4.2) minimizes the total weighted waiting time of customers. Constraint (4.3) ensures that each demand node must be visited at most once by a vehicle and constraint (4.4) guarantees that if a vehicle enters a demand node then it must leave that node. Constraint (4.5) ensures that the total number of entering and leaving arcs to each depot is equal. This constraint makes the balance between the number of vehicles initially starting their trip from a depot and the number of vehicles ending their trip at an identical depot. Constraint (4.6) ensures that all open depots are used at least once. Constraint (4.7) is sub-tour elimination for routes; in other words, it prevents making illegal routes between depots. It imposes that for any set of $G \subseteq S_D$ either vehicle k must come from the depots not included in G or the start depot of vehicle k should be in G. Constraint (4.8) states that every vehicle must be assigned to one depot only. Constraint (4.9) limits the total number of open depots to *TD*. By constraint (4.10) each demand node, at most, can be allocated only to one of the other demand nodes or to itself. Constraint (4.11) ensures that if demand node i is allocated to demand node j, then they are within the walking distance. Constraint (4.12) indicates that the out-tour demand node has to be allocated to its nearest in-tour demand node. Constraint (4.13) guarantees that the lost customer node is not located in the walking distance of an in-tour customer node. Constraint (4.14) assures that after completing the services, the vehicles, stop at the closest depot to their final in-tour customer node location. Constraint (4.15) indicates that if we pass through a demand node, the demand of that node will be allocated to itself. Constraint (4.16) works for two purposes, first it ensures that the vehicles should leave an open depot only, and

second it prevents violation of the depot capacity. Constraints (4.17-4.19) are load flow preservation constraints and are adapted from Karaoglan and Altiparmak [114]. Constraint (4.17) ensures that the difference between loads of the vehicle before and after visiting a demand node i is equal to the total demand allocated to node i. Constraints (4.18) and (4.19) are the bounding constraints for load flow variable. Constraint (4.20) determines the W_{ij}^{k} values for all nodes that their previous node in route is a demand node. We call V_{ijp}^{k} variable *connector*, since it connects two different demand nodes before and after visiting a depot. Constraints (4.21) and (4.22) define the obtainable possible amounts for decision variable V_{ijp}^{k}. In particular, constraints (4.21) and (4.22) ensure that each demand node appears at most once in the connector, and constraint (4.23) defines the number of connections (the number of V_{ijp}^{k} with the value of 1) for each vehicle at each depot. In other words, constraint (4.23) says that total amount of connections appointed to a particular depot j and a particular vehicle k is equal to the total number of leaving arcs from depot j by vehicle k except the one that was the initial start of k. Constraint (4.24) calculates the waiting time of all demand nodes that their previous node is a depot node. Constraint (4.25) defines the lower bound for the waiting time of customers that are located next to a depot node in the route. The weighted waiting time of each demand node is designated by constraint (4.26). Constraint (4.27) ensures that the waiting time of all demand nodes (except lost demand nodes) is less than or equal to TT. Constraints (4.28) and (4.29) set values to service time variable α_i. Finally, equations (4.30)-(4.34) represent the integrality constraints.

4.3.2 Valid Inequalities

Valid inequalities are those equations that strengthen the LP formulation by reducing the solution space. These polynomial constraints just remove undesirable

fractional solutions without affecting the integer feasible solutions. We take the advantage of five additional inequalities for CLRPR model that are listed below:

$$\sum_{i\in S_D}\sum_{j\in S_D} X_{ijk}^{l} - \sum_{i\in S_D}\sum_{j\in S_D} X_{jik}^{l} + O_j^k \geq 0 \quad j \in S_D, k \in K \tag{4.39}$$

$$\sum_{i\in S_C}\sum_{j\in S_D}\sum_{p\in S_C|p\neq i}\sum_{k\in K} V_{ijp}^{k} - \sum_{i\in S_D}\sum_{j\in S_C}\sum_{k\in K} X_{ijk}^{i} + |K| = 0 \tag{4.40}$$

$$\sum_{i\in S_C}\sum_{k\in K} e_{ij}^{k}\, d_i - CV \leq 0 \quad \forall j \in S_C \tag{4.41}$$

$$\sum_{i\in S_C}\sum_{k\in K} e_{ij}^{k} - \left\lfloor \frac{CV}{min\{d_1, d_2, \dots, d_m\}} \right\rfloor \leq 0 \quad \forall j \in S_C \tag{4.42}$$

$$\sum_{k\in K} V_{ijp}^{k} \leq 1 \quad \forall i \neq p \in S_C, j \in S_D \tag{4.43}$$

Constraint (4.39) omits the illegal routes by controlling the number of outgoing arcs from each depot. Constraint (4.40) calculates the correct value of connector variables in total. Constraints (4.41) and (4.42) ensure that the total amount of assigned demand to a customer node must not violate the capacity of a vehicle. Constraint (4.43) indicates that connector variable must be at most one for a sequence of nodes.

However, several different valid inequalities can be added to the model, though the important fact is that the equations must reduce the solution space and tighten the relaxation of the model, otherwise it is not beneficial.

4.3.3 Model validation

According to Hwang and Masud [115] and based on the decision maker participation, the multi objective mathematical programming methods can be categorized into three groups: priori methods, interactive methods, and posteriori methods. Priori methods are those wherein the decision maker determines the

preference before the solution proceeds; in interactive methods, the solution algorithm searches for the best solutions interactively considering the decision maker's judgment on the most preferred solutions. While all optimal solutions are not specified in priori and interactive methods, the posteriori methods generate all the efficient solutions (Pareto optimal solutions) of the problem. Pareto optimal solutions or efficient solutions can be defined as follows: A feasible solution for a multi objective problem is called efficient, Pareto optimal or non-dominated solution, when none of the objectives is able to improve without debasing the other objectives. In other words, for i number of minimizing f objective functions, solution v is efficient if there is no other solution u such that $f_i(u) \leq f_i(v)$. A set of Pareto optimal solutions is also called non-dominated solutions when they do not dominate each other and no other solution can dominate them.

The ϵ-constraint method is a famous multi-objective posteriori method introduced by Haimes, Ladson, and Wismer, (1971). This method generates all Pareto optimal solutions by optimizing one objective function while considering the other objective functions as constraints. In case of bi-objective optimization, the solution procedure is easy. The model will be solved iteratively for one objective function (as the main objective) in order to find non-dominated solutions. Let's assume that $f_1(X)$, and $f_2(X)$ are objectives (1) and (2) respectively, while X is the decision variable vector, then, minimizing $f_1(X)$ results in lowest number of lost customers, while there might be alternative solutions for $f_2(X)$. To counter this problem, first $f_1(X)$ is minimized subject to $f_2(X) \leq \varepsilon$, and solution X_1 is obtained, and then $f_2(X)$ is minimized subject to $f_1(X) \leq f_1(X_1)$ to achieve solution X_2. The set of X_2 solutions among all iterations are non-dominated Pareto optimal solutions. The value of ε is equal to $f_2(X) - \Delta$ such that Δ is a small positive constant. So as to obtain all efficient

solutions, whereas objective function (2) is always an integer, the value of Δ equals 1. In our problem, each vehicle must serve at least one customer node; therefore the stopping criterion is when the number of lost customers exceeds $m - k$.

In order to validate the proposed model with ϵ-constraint method, we code the CLRPR mathematical formulation in GAMS and due to long execution time of the problem, a tiny sized test problem VS1 is designed. In the problem, there are three potential depot nodes and two of them must be opened (*TD*=2). The number of demand nodes is six and two vehicles with *CV*=480 are available. All characteristics of the test problems solved in this chapter are available at www.dropbox.com/l/s/p0PwXZGiqyT3MswcWOHbhr. The problem is coded in GAMS 24.4.5 and solved by Cplex solver using a PC with configuration: Pentium(R) D CPU 3.00GHz, RAM 4.00 GB, and 64 bit Windows 8.1 operating system.

Table 4.1: Computational results of test problem VS1

$\|S_D\|$	$\|S_c\|$	TD	|K|	Results ($Z_1 \| Z_2$)	CPU time (original model)	CPU time (valid inequalities added)
3	6	2	2	(268|2275277),(310|2196539),(370|2039219),(381|2027350) ,(471|1857697) ,(578|1516087) ,(638|1358767) ,(649|1346898) ,(680|1321855) ,(691|1310448) ,(739|1222389) ,(751|1166192) ,(841|1049111) ,(852|1040652) ,(948|706527) ,(959|698068) ,(1019|624028) ,(1049|599720) ,(1109|503479) ,(1120|495020) ,(1222|428106) ,(1329|426190)	2643	2089

According to the results presented in Table 4.1, the computational time of the model decreases near 20% when valid inequalities are added to the original model. Among the 22 Pareto optimal solutions of VS1, two solutions are depicted in Figure 4.2 and 4.3.

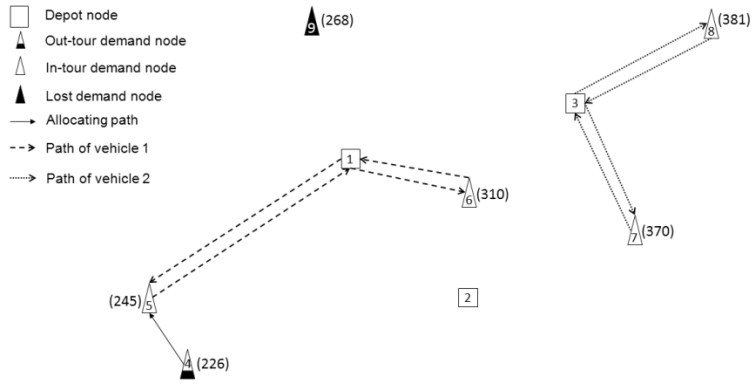


Figure 4.2: Solution (268|2275277) considering Z_1.

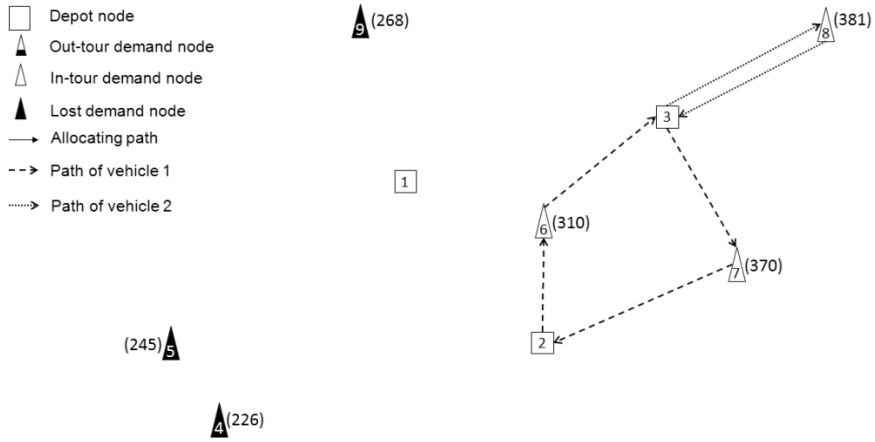


Figure 4.3: Solution (739|1222389).

Figure 4.2 shows the schedule considering only objective function 1 where total minimum value of lost customers is 268, while total weighted waiting time is 2,275,277 units. The schedule in Figure 4.3 depicts the case that the value of lost demands increases to 739 while total weighted waiting time decreases to 1,222,389.

4.4 Heuristic Solutions

Since CLRPR is a generalized LRP, it can be considered as an NP-hard problem. Therefore, the exact formulations are not appropriate to solve medium and large scale problems. Genetic algorithm (GA) is a well-known meta-heuristic

algorithm that has widely been used for combinatorial problems specially VRP. The main advantages of GA over other meta-heuristic algorithms are extensive population based search, convergence and divergence control, and short execution time. Karakatič and Podgorelec [117] conducted a survey of GA for multi depot VRP and stated that the key advantage of GA compare to other algorithms is the efficiency and final outcomes on time constraints with limited computer power that result in competitive solutions. To deal with the multi objective optimization problems, nondominated sorting genetic algorithm II (NSGAII) has been introduced by Deb *et al.* [118], which has gained a lot of attention by scientists. In contrast to most of elitist multi-objective evolutionary algorithms, NSGAII possesses elitism and divergence property together.

In this section, first NSGAII with some useful adjustments is presented for LRSIDCD problem, and then two different search mechanisms for augmenting the solution quality are added to the algorithm.

4.4.1 NSGAII and Improvements

Briefly, NSGAII has following steps:

1) Create N number of initial solutions.
2) Sort the solutions according to non-domination rank and crowding distance.
3) Apply selection procedure to form the mating pool.
4) Utilize crossover and mutation operators on mating pool solutions to get N numbers of new solutions (offspring), and add them to the population community to obtain 2N number of the solutions.
5) Sort (based on nondomination rank and crowding distance) and select N number of population community, then go to step 3).

By NSGAII, a proper solution will never get tined except if a better solution is replaced. For comprehensive illustration of the NSGAII algorithm one can refer to Deb et al. (2002).

4.4.1.1 Representation

Representation is one of the critical concerns that have deep influence on the performance of GA and NSGAII. The way a chromosome is represented directly affects the quality of the solutions and impresses the execution time. Generally different problems need different representations and choosing the best representing form depends on the characteristics of a feasible solution and structure of the operators used in GA. In order to manage all constraints of the CLRPR efficiently and to tackle the offspring feasibility obstacle, random key technique is used to obtain a feasible solution. Random key technique was introduced by Bean [119] and has been used in different evolutionary algorithms.

In this study, we consider a chromosome consisting of $n + m$ number of genes, where n is the number of potential depots and m is the number of customer nodes. Each gene has a random five digit number belonging to $[10000, 100000)$ intervals and is labeled by the number of depots and customer nodes (the random keys are non-identical). In general, the decoding procedure of a chromosome: (i) determines the TD number of open depots, (ii) assigns the customers to the open depots considering a depot's capacity, (iii) assigns the out-tour customers to their nearest in-tour demand node, (iv) generates routes according to the capacity of vehicles, (v) organizes the trips due to waiting time limit TT, and finally (vi) calculates the values of Z_1 and Z_2. The decoding procedure is detailed in Algorithm 4.1.

Algorithm 4.1: Representation procedure

WD : Walking distance
PD : removing distance
da_i : Total demand of customer i (d_i + assigned demand to i)
Ins : a constant two digit integer number
Elm : a constant four digit integer number

//Initialize
Step 1. Make Chromosome with "$n + m$" genes
Step 1.1. Initialize Elm, Ins, & $PD \leftarrow WD$
Step 1.2. Assign unique random number within [10000,100000) to all genes;
Step 1.3. Sort genes of 1 to n in ascending order of random keys; // Prioritizing of depots
Step 1.4. Sort genes of $n + 1$ to $n + m$ in ascending order of random keys; // Prioritizing of customers
// Make the $Path$ as follows:
$j \leftarrow n + 1$; //First Customer of Step 1.4.
Step 2. For ($i = 1 : TD$) do
Step 2.1. . $Path$.Insert(depot(i)); $SumofDemand \leftarrow 0$;
Step 2.2. While ($j \leq n + m$) and($SumofDemand + d_j$) < CD
Step 2.2.1. if mod($randomkey$ of customer(j), 1000) > A large three digit number (990), then Insert customer(j) to

$EliminatedList$

else $Path$.Insert(customer(j)); $SumofDemand \mathrel{+}= d_j$;
Step 2.2.2. j++; // Select Next Customer
Step 3. $EliminatedList \leftarrow$ Customers Not assigned to any depot ;
Step 4. $Eliminated1 \leftarrow EliminatedList$;
Step 5. While ($Eliminated1$.Count != 0)
Step 5.1. for each item in $Eliminated1$ do
Step 5.1.1. for each customer in $Path$ do
Step 5.1.1.1. Remove any customer from $Path$ that its $distance(item, customer) \leq PD$;
Step 5.1.1.2. Add this customer to the list named $Eliminated2$;
Step 5.2 $Deleted \leftarrow Eliminated1$;
Step 5.3 $Eliminated1 \leftarrow Eliminated2$;
Step 6. IF there is two depots next to each other or any depot with no customer back to Step 1.2
Step 7. For each customer $\{da_i \leftarrow d_i;\}$
Step 8. For each customer in path do
Step 8.1. $x \leftarrow$ customer;
// $y \leftarrow$ any customer that has the following conditions
Step 8.2. If $da_x == d_x$ then
Step 8.2.1. $y \leftarrow$ Find customer in path that has minimum WD with x;
Step 8.2.2. If (x and y has been assigned to same depot) and ($d_x + da_y \leq CV$) Then
$da_y \mathrel{+}= d_x$; Remove x from $Path$ and add it to $assignedList$;
//The inserting depots of Step9 and Step10 are same as the previous depot in $Path$
Step 9. Insert a depot before any customer with $mod(randomkey, 100) \leq Ins$;
Step10. Insert a depot before any customer that exceeded CV ;
Step11. If number of $Route$s in $Path < K$ then
Step11.1 If number of customers in $Path < K$ then {go to Step 1.2}
else { Ins *= 1.04; go to step 9 }

Step12. For each $Route$ in $Path$ make $Priority\_index$ by the following steps:

Step12.1. $element1 \leftarrow$ mod($randomkey$ of first customer of $Route$, 1000);

$element2 \leftarrow$ number of first customer of $Route$;

$Priority\_index \leftarrow (element1, element2)$;

Step13. Sort the $Route$ first based on $element1$, and in case of equality according to $element2$;

// Compute W for each $Route$

Step14. For each $Route$ do

Algorithm 4.1 (continued).

Step14.1. $W \leftarrow \zeta$

Step14.2. For (index = 1 : $|Route| - 1$)

Step14.2.1. $i \leftarrow$ number of customers in $Route$(index); $j \leftarrow$ number of customer in $Route$ (index+1);

Step14.2.2. If (index == $|Route| - 1$) then $W \mathrel{+}= t[i,j]$;

else $W \mathrel{+}= t[i,j] + \delta * da_j$;

Step15. For ($i = 1 : K$) do

Step15.1. i-th $Trip \leftarrow i$-th $Route$; W of i-th $Trip \leftarrow W$ of i-th $Route$;

Step16. For ($i = K + 1$: number of $Route$s) do

Step16.1. Insert i-th $Route$ at the end of $Trip$ with minimum (W of $Trip$ + W of i-th $Route$ + $t_{\text{last customer of Trip,first depot of i-th Route}}$);

and (W of $Trip$ += W of i -th $Route$ + $t_{\text{last customer of Trip,first depot of i-th Route}}$);

Step17. If Maximum of W of all $Trip$s $> TT$ then

Step17.1. $PD \mathrel{*}= 1.04$; $Elm \mathrel{*}= 0.98$;

Step17.2. For those $Trip$s that $W > TT$ do

Step16.2.1. Add the Customer with mod($randomkey$, 10000) >= Elm to the to $EliminatedList$;

Step17.3. go to Step 4.1

//compute waiting time for each customer in $Trip$s

$Fitness1 = Fitness2 = 0$;

Step18. for ($T = 1 : K$) do // K : number of vehicles and index of $Trip$s

Step18.1. for each item in $Trip$ do

Step18.1.1. if (item == Customer)

$i \leftarrow$ index of item in $Trip$

$W \mathrel{+}= t[i, i-1] + da_{(i-1)}$;

Step18.1.2. else if (i != 0)

$W \mathrel{+}= \zeta + da_{(i-1)} + t[i, i-1]$;

else

$W \mathrel{+}= \zeta$;

Step18.1.3. $Fitness2 \mathrel{+}= W * d_{ai}$;

Step19. $Fitness1 \leftarrow$ Total demands of customers in $Deleted$;

According to Algorithm 4.1, Step 1 makes initial feasible solutions as $m + n$ size chromosomes and sort the priorities of nodes based on increasing order of random keys. In Step 2, the path shown in part (II) of Fig. 4 is created and the depot's capacity restriction is applied. Steps 3, 4, and 5 remove customer nodes from the path to satisfy the constraints (4.12), (4.13), and (4.16). The customer allocation process is executed in Step 8. Step 9 makes random routes and Step 10 ensures that the capacity of vehicles is not violated, and to make all available vehicles to be utilized, Step11 is applied. Step 12 and 13 prioritize the routes and Step 14, 15, and 16 make the trips. Step 17 checks the time limit constraint of each trip and Step 18 and 19 calculate the fitness values. As can be seen in the algorithm, the final destinations of the vehicles are not yet appointed and will be covered in Section 4.4.2. The decoding process is designed in such a way that it rarely results in infeasible solutions; however, if it occurs another chromosome is created in initialization phase or another crossover and mutation are applied in reproduction step. It is worth emphasizing that the initial population is created randomly and all individuals are feasible solutions of the problem.

A simple decoding example is presented in Figure 4.4. The example consists of 3 potential depots and 12 customers with given demand values. In part (I), the depots and customers are arranged in increasing order of their random keys, and then depots 1 and 3 are selected as open depots. In part (II), no customer is eliminated according to step 2.2.1 and customers are assigned to the open depots according to their order and by considering the capacity of the depots. As being shown, customer 15 is eliminated due to capacity restriction of depot 3, and customer 9 is eliminated to satisfy constraint (13). Part (III) shows that customer 11 is allocated to customer 6, and customer 14 is allocated to customer 7. Part (IV) depicts the inserting process of depot 3 before customer 8, according to the last two digits of random key of the customer 8.

Vehicles' capacity restriction is applied in part (V) by inserting depot 1 right before customer 10. Routes are prioritized and assigned to the trips in part (VI).

4.4.1.2 Selection

The selection process follows the binary tournament selection to create the parent population community. N numbers of individuals (solutions) are sorted in nondominated fronts according to their dominance property, and each individual i has a label known as crowding distance $i_{distance}$. The crowding distance is an estimate of empty cuboid solution space around i. Each time a pair of solution is randomly selected and one wins the tournament that belongs to the prior non-domination rank. In case of equal non-domination ranks, the one with greater $i_{distance}$ is selected.

In every generation, N/2 number of individuals are selected and copied to the mating pool. To avoid choosing identical individuals as parents for crossover operation, the N/2 numbers of individuals in mating pool are formed as non-identical pairs of parents. Through each generation, N different pairs are drawn from mating pool in order to create N numbers of new individuals. This helps in avoiding convergence tendency due to combining identical parents or identical pairs.

4.4.1.3 Crossover and Mutation Operators

Each pair of parents can create only one offspring in the algorithm. Since N individuals are preserved for next generation, while 2N individuals are required, the crossover operator must be applied on all N pairs of parents. The crossover and mutation operators produce feasible solution since only the random keys are exchanged or reproduced.

There are various techniques to design the crossover and mutation operator, and every problem may be compatible with specific operators. Two different crossover and mutation operators are proposed for creating an offspring: (A) Uniform crossover

and swap mutation, and (B) two-point crossover and uniform mutation. Each time, by equal probability, the parents move to operators (A) or (B) to create an offspring. The reason for determining the two different pairs of operators is to avoid creating converged individuals.

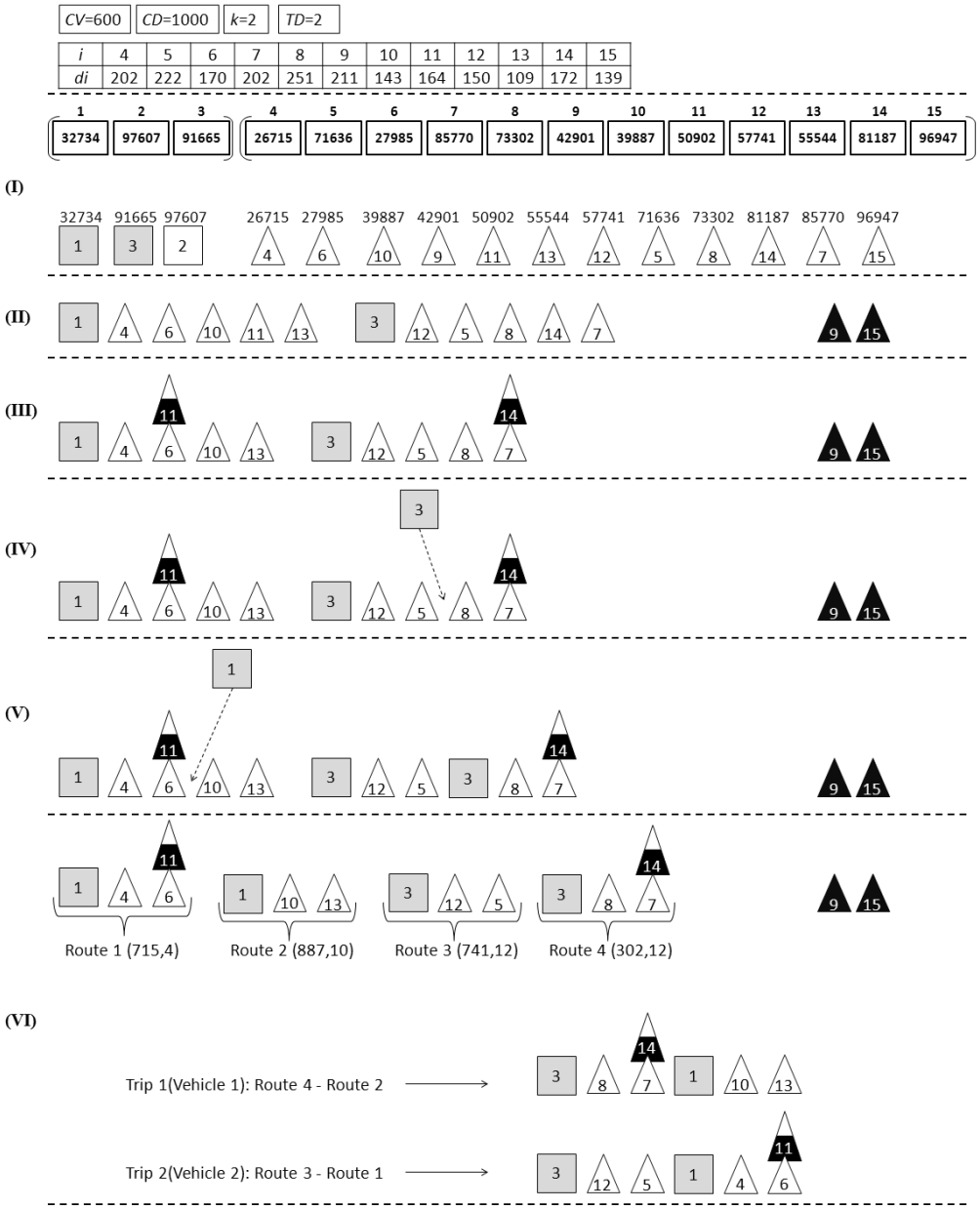


Figure 4.4: An illustrative example of chromosome decoding procedure.

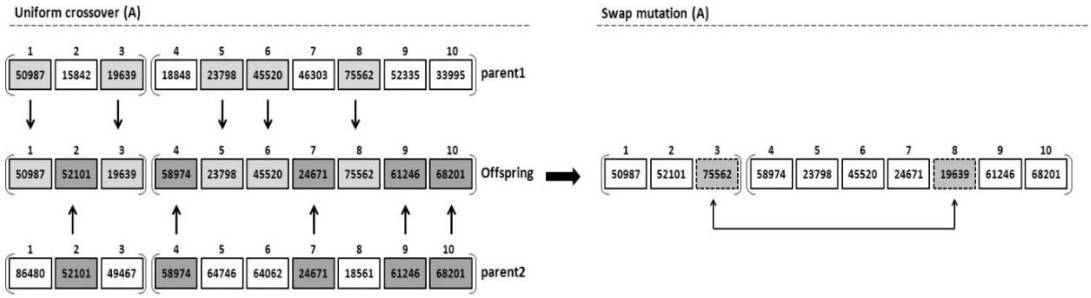


Figure 4.5: Pair of operators (A).

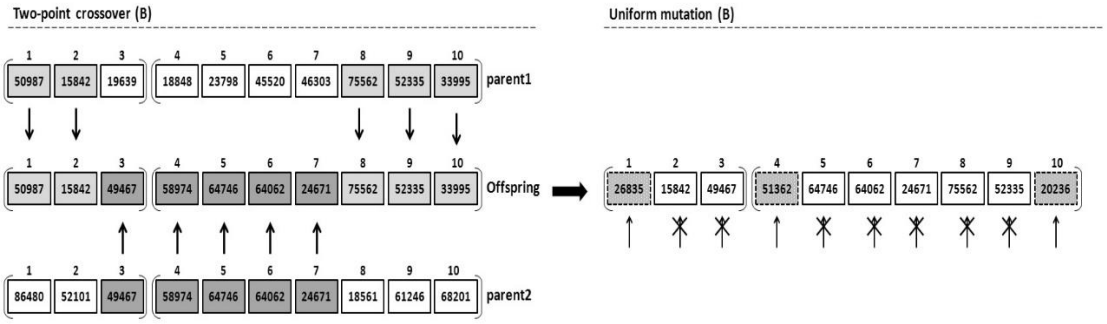


Figure 4.6: Pair of operators (B).

In case of operators (A), when uniform crossover is applied, each gene of offspring is inherited from parent 1 or parent 2 by equal probability, and then by a small probability the swap mutation might exchange two randomly selected genes of the produced offspring. When it comes to operators (B), a two-point crossover and uniform mutation is applied. In the two-point crossover, two randomly chosen crossover points are uniformly determined from 1 to $m + n$. The offspring is produced by combining the first and third part of parent 1 with the middle part of parent 2. Thereafter, the uniform mutation operator is applied, and with a low probability the random key of each gene might be changed by a new random key. To exemplify the operators (A) and (B), examples with 3 depots and 7 customers are shown in Figure 4.5 and Figure 4.6, respectively. The offspring generation is illustrated in Algorithm 4.2.

Algorithm 4.2: Generating new N number of offspring

$MakeOffsprings$
{ //binary Tournament Selection:
Step 1. $PoolSize$ = $PopSize$ $N/2$; $TourSize = 2$;
Step 2. while ($|matingPool| < poolSize$) do
Step 2.1. Get $TourSize$ non-identical Random chromosomes from population;
Step 2.2. Insert winner (based on non-domination rank and crowding distance) to mating pool;
Step 3. $PairList \leftarrow$ make ((N/2)C2) pairs from chromosomes of mating pool;
Step 4. for ($i = 1 : PopSize$) do
Step 4.1. Select and remove a random pair of parents from $PairList$;
Step 4.2. By 50% probability select " $CrossOverA$ and $MutationA$ " or "$CrossOverB$ and $MutationB$"; }
//$CrossOverA(Parent1, Parent2)$
Step A1. For ($i = 1$: Length of chromosome) do
Step A1.1. By 50% probability select the gene from $parent1$ or $parent2$ and copy it to the $Offspring$;
//$MutationA(Offspring)$
Step A2. By $P_{Swap-mutation}$ probability swap two genes of offspring;

//$CrossOverB(Parent1, Parent2)$
Step B1. Select two random points and perform two-point crossover to produce $Offspring$;
//$MutationB(Offspring)$
Step B2.For ($i = 1$: Length of chromosome) do
Step B2.1. By $P_{Uniform-mutation}$ probability assign a new random key to gene;

4.4.1.4 Replacement

The replacement phase in NSGAII is quite different than in GA, as it deals with a multi-objective problem. According to Deb et al. [118], at the beginning, N numbers of individuals are created randomly and according to selection procedure N/2 number of them is selected to form the mating pool. In each generation, N numbers of new offspring are created to establish a set of 2N individuals. Let's call the initial set of N individuals I_p and the new set of N offspring O_p. At the end of each generation to ensure elitism, $I_p \cup O_p$ must be sorted into nondominated fronts ($F_1, F_2, F_3, ...$). F_1 is the best front and contains individuals that cannot be dominated by others, F_2 contains the next best solution and so on. Now N number of individuals must be selected by the priority of fronts and if a portion of F_i is needed to complete the I_{p+1}, the individuals

with larger $i_{distance}$ must be selected. The main loop of the NSGAII is presented in Algorithm 4.3.

Algorithm 4.3: Main loop

Step 0. $I_p \longleftarrow$ Generate population with count size N;
C: Main Loop
Step 1. $O_p \longleftarrow MakeOffsprings(I_p)$;
Step 2. $T_p = I_p \; U \; O_p$
Step 3. $T_p = Remove\_Duplicate\_Chrom(T_p)$ and replace new chromosomes instead;
Step 4. $F = Fast\_NonDominated\_Sort(T_p)$;
Step 5. $I_{p+1} = 0$ and $i = 1$;
Step 6. while $|I_{p+1}| + |F_i| \leq N$
  Step 6.1. $Crowdingdistance\_Assignment(F_i)$;
  Step 6.2. $I_{p+1} = I_{p+1} \; U \; F_i$;
  Step 6.3. $i = i + 1$;
Step 7. $Crowdingdistance\_Assignment(F_i)$;
Step7. $I_{p+1} = I_{p+1} \; U \; F_i[\,1 : (N - |I_{p+1}|)]$;
Step9. $p = p + 1$;

According to Figure 4.7 that depicts the loop, F_1 and F_2 move to I_{p+1}, based on their nondomination rank and individuals in F_3 those have a larger $i_{distance}$ are selected.

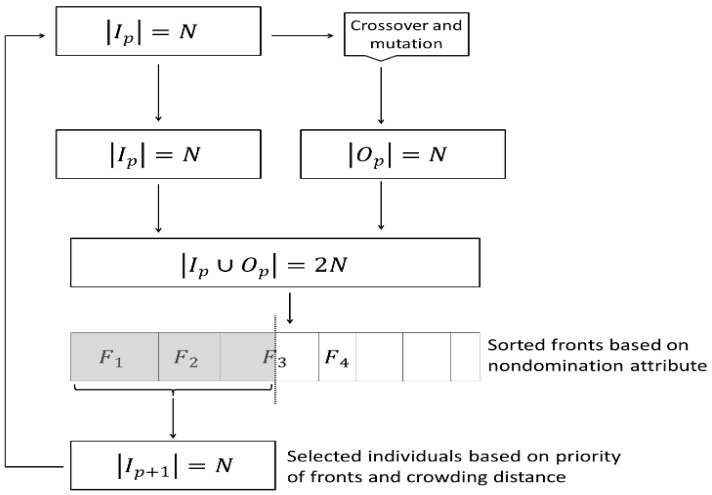


Figure 4.7: Main loop of NSGAII algorithm

4.4.1.5 2N Improvement

The 2N improvement (2NI) procedure is applied on a fraction of solutions existing in $|I_p \cup O_p|$. In the main loop of NSGAII before $Fast\_NonDominated\_Sort$ process, R numbers of chromosomes are selected from T_p. For each selected chromosome, a reverse mechanism is applied to improve the solution. By reverse mechanism, the random keys of randomly V numbers of genes are reversed. This improvement applies only in one direction of objective functions, which means in case of an improvement in at least one objective function; the new chromosome is accepted and added to T_p. If no improvement occurs the algorithm shifts to the next chromosome. The procedure is presented in Algorithm 4. 2NI improves the 2N solutions and adds the new improved individuals to T_p; therefore, the size of T_p exceeds 2N, though it does not interrupt the main loop of NSGAII. According to Step 6 of Algorithm 4.4, eventually N numbers of individuals are selected as I_{p+1} for next generation.

Algorithm 4.4: 2N improvement

Step 1. $SelectedChrom \leftarrow$select R number of chromosomes from T_p;
Step 2. For each item in $SelectedChrom$ do
    Step 2.1. Select V random number of genes and reverse their $randomkeys$;
    // Evaluate New Generated Chromosome
    Step 2.2. if (($NewChrom.Objective1 < item.Objective1$) || ($NewChrom.Objective2 < item.Objective2$))
        Step 2.2.1. Add $NewChrom$ to $newList$;
Step 3. Add $newList$ to T_p ;

4.4.1.6 First Front Improvement

In order to improve the first front's solutions, the first front improvement (FFI) procedure is taken into consideration. In FFI, the reverse operator is applied on all chromosomes belonging to F_1 for last 5% of generations. The procedure tries to find a

new solution that dominates the selected chromosome and continues to find better solution for G times. Algorithm 4.5 demonstrates FFI. However, the execution time increases due to applying $Fast\_NonDominated\_Sort$ procedure twice in each iteration, the procedure guarantees to improve the quality of objective functions. Unlike, 2NI improvement, the size of T_p does not exceed 2N, while the new preferred solutions are replaced by the old ones. It is worth mentioning that different search mechanisms (i.e., shift, permutation) have been tested for both improvements and the reverse mechanism has been found to perform best.

Algorithm 4.5: First front improvement

Step 1. $SelectedChrom$ ⟵All chromosomes of the first front (F_1);
Step 2. For each $item$ in $SelectedChrom$ do
    Step 2.1. for ($i = 1 : G$)
        Step 2.1.1 $NewChrom$ ⟵ Reverse random r $randomkeys$;
        Step 2.1.2 If ($NewChrom$ dominates $item$) then
            Step 2.1.2.1. Replace $item$ with $NewChrom$;
Step 3. Return $SelectedChrom$;

4.4.2 Final Destination Path

According to the chromosome encoding procedure in Section 4.4.1.1, the final destination (end depot node) of the vehicles was not specified as it did not affect the fitness functions. Constraints (5) and (14) state that the number of vehicles that start their trip from a depot must be equal to the number of vehicles that end their trip at the same depot, and the vehicles must be assigned to the nearest depot in the final step of their mission. To satisfy these conditions and in order to find an optimal solution, a simple LP model is presented.

S_E is a subset of S_C that contains the last customer node of each trip, and decision variable L_{ij} is 1 if the vehicle passes through arc $(i,j) \in \{(i,j)|i \in S_E, j \in S_D\}$ and 0 otherwise. It must be noted that O_i^k is a parameter for this

problem since the start points of the trips are known from the heuristics. The mathematical formulation for final destination problem (FDP) is given as follows:

$$Min\ Z_3 = \sum_{i \in S_E} \sum_{j \in S_D} L_{ij} t_{ij} \tag{4.44}$$

$$\sum_{j \in S_D} L_{ij} = 1 \quad \forall i \in S_E \tag{4.45}$$

$$\sum_{k \in K} O_j^k = \sum_{i \in S_E} L_{ij} \quad \forall j \in S_D \tag{4.46}$$

$$L_{ij} \in \{0,1\} \quad \forall i \in S_E, j \in S_D \tag{4.47}$$

Equation (44) minimizes the total final destination traveling times. Constraint (4.45) ensures that all vehicles are assigned to a depot as the last node. Constraint (4.46) guarantees that the number of vehicles starting and ending their trips at a depot is equal, and equation (4.47) defines the decision variable. The experiments with 200 numbers of vehicles and 20 depot nodes show that FDP model can be solved within 0.13 seconds optimally by GAMS using Cplex solver, thus, the computation time of FDP is negligible. Once the trips are attained by heuristic algorithms, the results are used to obtain the final destination of vehicles within a concise time period.

4.5 Computational Results

In this section, the performance of proposed NSGAII algorithm and improved versions are investigated. The test instance characteristics and discussions about parameters setting are provided in Section 5.1. In order to evaluate the performance of heuristics four comparison metrics are given in Section 5.2, and the analysis of the results is presented in Section 5.3. The algorithms are coded in C# programming

language and run on a PC having configuration of a Pentium(R) D CPU 3.00 GHz, RAM 4.00 GB, and Windows 8.1 operating system.

4.5.1 Data Generation and Parameters Setting

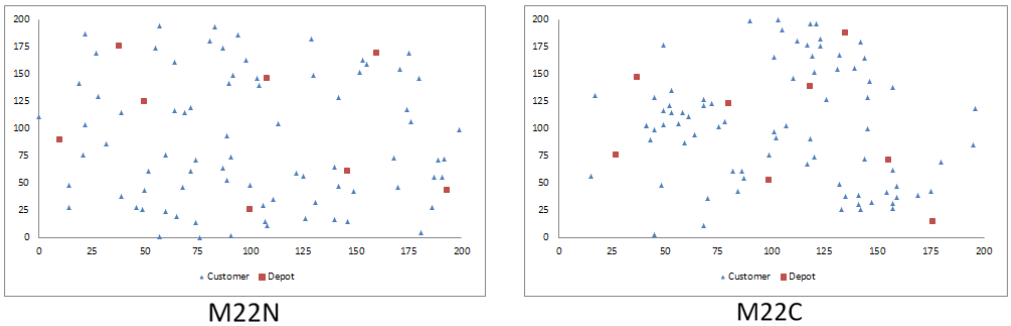
Several randomly generated instances are designed for CLRPR problem. The instances are categorized into three groups of small-sized, medium-sized and large-sized problems. Generally the number of customer nodes $|S_C|$, the number of potential depot nodes $|S_D|$, and the number of vehicles $|K|$, differ between 25 and 160, 4 and 14, 2 and 16, respectively. The coordinates of vertices are generated randomly in two forms: normally distributed vertices and clustered distributed vertices in $[0,200]$ intervals. In normal distributed pattern, the vertices are generated uniformly through the intervals and in the clustered pattern, some areas have higher density of nodes. An average speed of vehicles is considered to calculate the travel times between the nodes. The Euclidean distance and consequently the travel time matrix are rounded to a 4-digits integer numbers. The demand amounts are generated as uniformly distributed in $[100,500]$, and the depots' capacity CD, which is same for all depots, is calculated such that at least 80% of the demands is covered. The capacity of vehicles is divided into two categories 1 and 2. The walking distance, δ, and ζ are assumed to be 12, 1, 400, respectively, for all test problems.

General characteristics of the test problems and the ranges are shown in Table 4.2. Each problem instance has a name in format of XYZL, where X refers to the size of the problem (i.e., Small, Medium, and Large), Y refers to the number of the problem from 1 to 4 corresponding to the problem size, Z refers to the capacity of the vehicle (i.e., 1500 as 1, and 3000 as 2), and L represents the form of distributed vertices such that C means clustered based and N means uniform based distributed vertices. As an example M21C is the medium sized problem number 2, with vehicle's capacity of

1100 with clustered pattern of vertices. An example of clustered based and uniform based pattern is shown in Fig. 4.8. The test instances are all available at www.dropbox.com/l/s/p0PwXZGiqyT3MswcWOHbhr.

Table 4.2: Characteristics of the instances

	Number	$\|S_C\|$	$\|S_D\|$	TD	CD	CV		$\|K\|$	Coordinates	
						1	2		C	N
Small	1	25	4	2	2500	800	-	2	-	N
	2	35	5	3	2500	800	-	3	-	N
	3	40	6	4	3000	800	-	5	-	N
	4	50	6	4	3000	800	-	5	-	N
Medium	1	65	7	4	3500	1100	2300	6	C	N
	2	80	8	5	3500	1100	2300	7	C	N
	3	90	8	6	4000	1100	2300	9	C	N
	4	100	9	6	4000	1100	2300	9	C	N
Large	1	120	10	7	4500	1500	3000	11	C	N
	2	135	11	8	4500	1500	3000	13	C	N
	3	150	13	8	5000	1500	3000	14	C	N
	4	160	14	8	5000	1500	3000	16	C	N



M22N M22C

Figure 4.8: Two medium sized test problems with different patterns

The parameters of the algorithms are set by trial and error method, and by searching for initial ranges from related literature. However, we do not assert that the parameters are ideal, though with this arrangement, the algorithms perform better than the other arrangements. The number of evaluations was similar for all algorithms and all metrics mentioned in Section 5.2 were utilized as guidelines to set the parameters.

Due to similarity of NSGAII, NSGAII-2NI, and NSGAII-FFI the crossover and mutation probabilities are same for all algorithms as $P_{Crossover} = 1$, $P_{Swap-mutation} = 0.5$, and $P_{Uniform-mutation} = 0.08$. For NSGAII-2NI the parameter R is equal to $0.25N$ and for NSGAII-FFI the parameter G is equal to 6. The best achieved values of numbers of generations and population size for different test instances are shown in Table 4.3.

Table 4.3: Generation and population size parameters

Problem	Generation number		Population size
	NSGAII	NSGAII-2NI & NSGAII-FFI	
S11N,S21N	500	400	130
S31N,S41N	700	600	130
M1**,M2**	1000	900	160
M3**,M4**	1300	1150	175
L1**,L2**	1500	1350	180
L3**,L4**	1700	1500	200

4.5.2 Performance Metrics

In order to compare the multi-objective heuristics some performance metrics are utilized. The four different metrics applied in this study are described as follows:

- *Quantity metric (QNM)*: To compare the size of nondominated solutions obtained by an algorithm with others the QNM metric is utilized. The metric compares the power of the algorithm in generating higher number of solutions; therefore, more solutions provide more options for the decision maker. The QNM metric is defined as follows:

$$QNM_i = \frac{|F_{1i}|}{\sum_{i=1}^{l}|F_{1i}|} \tag{4.48}$$

If there are l numbers of algorithms and F_{1i} is the first front set of algorithm i. A higher value for QNM indicates the better first front set in terms of number of solutions.

- *Quality metric (QLM)*: This metric determines the number of nondominated solutions belonging to an algorithm that can dominate the nondominated solutions obtained by other algorithms and is based on C metric introduced by Zitzler *et al.* [120]. Let X be the first front set of algorithm A, and X' be the set of nondominated solutions belonging to all other algorithms. The QLM metric is defined as follow:

$$QLM(A) = \frac{|\{a' \in X'; \exists a \in X: a \succcurlyeq a'\}|}{|X'|} \tag{4.49}$$

Where $a \succcurlyeq a'$ indicates that solution a' is dominated by or equal to solution a. A higher value of QLM shows the better optimal solutions of an algorithm.

- *Diversity metric (DM)*: The spread of solutions obtained by an algorithm is calculated by DM that was proposed by Nekooghadirli *et al.* [121]. In contrast to DM the higher value indicates the better solutions.

$$DM = \sqrt{\left(\frac{maxf_{1i} - minf_{1i}}{f_{1,total}^{max} - f_{1,total}^{min}}\right)^2 + \left(\frac{maxf_{2i} - minf_{2i}}{f_{2,total}^{max} - f_{2,total}^{min}}\right)^2} \tag{4.50}$$

In equation (57), $(maxf_{1i}, minf_{1i})$ is the maximum and minimum of the objective function 1 within the nondominated solutions of i, and $(f_{1,total}^{max}, f_{1,total}^{min})$ refer to maximum and minimum amount of objective 1, respectively, among all algorithms (same for other objective).

- *Spread metric (SM)*: Let $\theta_i = arctan\left(\frac{f_{2i}}{f1i}\right) \forall\, i \in F_1$ and θ_i' be the difference of two consecutive solutions, then SM is presented as follows:

$$SM = \frac{\sum_{i=2}^{n}\left|\theta_i' - \bar{\theta}'\right|}{2(max\theta_i - min\theta_i)} \tag{4.51}$$

SM measures the spread uniformity of obtained solutions. The lower value in this metric shows the better spread of points.

4.5.3 Result Analysis

In order to assess the performance of suggested algorithms, 36 test instance problems are utilized. Table 4.4 reports the average values of performance metrics that have been obtained by 10 runs of algorithms for each test problem. The graphical comparisons of algorithms in terms of four different metrics are shown in Figure 4.9. Note that in terms of quality, quantity, and diversity metrics the larger values indicate better performance, and in terms of spread metric the smaller values imply better performance. In general, it is clear that with regard to QNM and QLM metrics NSGAII-FFI outperforms the others. Out of total 36 results, NSGAII-FFI algorithm obtained 22 best and 5 second best results in terms of DM, and 23 best and 5 second best results in terms of SM.

Table 4.4 :The obtained metrics values for algorithms

	NSGAII					NSGAII-2NI					NSGAII-FFI				
Problem	QNM	QLM	DM	SM	CPU	QNM	QLM	DM	SM	CPU	QNM	QLM	DM	SM	CPU
S11N	0.32	0.33	1.46	**0.46**	36.8	0.33	0.80	1.49	0.69	32.7	**0.33**	**0.76**	**1.95**	0.58	**23.45**
S21N	0.33	**0.76**	1.76	0.32	30.3	0.33	0.25	1.75	**0.31**	29.8	**0.33**	0.74	**1.99**	0.35	**26.6**
S31N	0.33	0.80	1.79	0.37	55.8	0.32	0.32	1.91	0.38	50.1	**0.35**	**4.44**	**1.99**	**0.34**	**47.6**
S41N	0.32	0.15	0.18	0.38	55.1	**0.33**	0.48	1.86	0.37	58.6	**0.33**	**2.78**	**1.98**	**0.36**	**54.4**
M11N	**0.33**	1.39	**1.92**	**0.42**	115	0.32	0.24	1.82	0.49	122.	**0.33**	**1.44**	1.86	0.49	**115.2**
M12N	0.33	1.48	**1.95**	0.49	123	0.33	0.25	1.88	0.50	129	**0.33**	**1.53**	1.87	**0.48**	**122.1**
M11C	0.32	0.31	1.72	0.45	**110**	0.33	0.48	1.86	0.44	112	**0.34**	**2.76**	**1.94**	**0.41**	113.8

M12C	0.33	0.36	1.91	0.38	**112**	0.33	0.38	1.64	0.39	116	**0.33**	**1.54**	**1.93**	**0.38**	120.2
M21N	0.33	0.11	**1.96**	0.46	**145**	**0.33**	1.62	1.90	0.47	170	**0.33**	**2.48**	1.80	**0.44**	183.2
M22N	**0.33**	0.42	1.72	0.41	**140**	0.33	1.12	1.76	0.40	163.	0.33	**3.10**	**1.78**	**0.37**	166.3
M21C	0.34	1.00	1.75	0.40	**147**	0.30	0.67	**1.93**	0.43	152	**0.35**	**1.71**	1.62	**0.39**	170.1
M22C	0.33	1.65	1.90	0.59	**146**	0.33	0.25	**1.92**	0.61	150	0.33	**3.28**	1.81	**0.58**	165.4
M31N	0.32	0.35	1.77	0.51	**222**	0.33	0.66	1.91	0.45	237	**0.34**	**4.32**	**1.93**	**0.43**	224.4
M32N	**0.34**	1.84	1.82	**0.38**	238.	0.33	0.24	**1.93**	0.44	240.	0.33	**2.34**	1.91	0.44	**234.17**
M31C	0.31	0.23	1.84	0.39	246	0.29	1.31	1.89	0.41	**243**	**0.36**	**1.65**	**1.85**	**0.37**	262.6
M32C	**0.33**	0.38	1.86	0.43	244.	0.33	0.41	**1.93**	**0.41**	**240**	0.33	**0.89**	1.74	0.42	256.5
M41N	0.33	1.21	**1.94**	0.42	**294**	**0.33**	0.52	1.83	**0.38**	313.	0.33	**3.45**	1.87	0.42	315
M42N	0.30	1.01	1.84	0.54	**303.**	0.34	2.10	1.85	0.52	298.	**0.35**	**5.39**	**1.89**	**0.44**	317.7
M41C	0.33	0.37	1.81	0.38	315.	0.33	1.37	1.88	**0.38**	**297**	**0.32**	**1.74**	**1.94**	0.39	337.2
M42C	0.33	0.68	1.87	0.56	296.	0.33	0.99	1.76	0.54	**288**	**0.33**	**3.18**	**1.99**	**0.52**	317.7
L11N	0.32	0.34	1.78	**0.46**	480.	0.33	1.12	1.83	0.49	**434**	**0.34**	**2.04**	**1.91**	0.51	478
L12N	0.31	0.46	1.85	0.44	471.	0.33	1.02	1.80	0.43	**428**	**0.35**	**3.41**	**1.92**	**0.41**	463.9
L11C	0.32	0.55	**1.96**	0.33	447.	**0.34**	0.61	1.83	**0.31**	**398**	0.33	**1.63**	1.76	0.34	453.2
L12C	0.33	0.57	1.76	0.42	449.	0.33	0.42	**1.98**	0.45	**417**	**0.33**	**1.95**	1.6	**0.40**	459.7
L21N	**0.33**	0.67	**1.90**	0.45	**459**	0.33	0.37	1.87	0.46	470	**0.33**	**4.85**	1.90	**0.45**	509.1
L22N	0.31	1.06	1.76	0.42	458	0.30	0.71	1.90	**0.37**	**455**	**0.37**	**1.82**	**1.90**	0.43	502.3
L21C	0.32	0.50	1.65	0.38	**453**	0.32	0.89	1.76	0.34	485	**0.34**	**4.34**	**1.97**	**0.31**	462.9
L22C	0.32	0.81	1.83	0.46	**462**	0.32	1.09	1.95	0.44	499	**0.34**	**3.27**	**1.54**	**0.37**	478
L31N	0.34	0.34	1.89	0.47	690	0.31	1.43	1.82	0.43	**686**	**0.34**	**1.41**	**1.98**	**0.43**	792.3
L32N	0.31	0.31	1.75	0.37	**633**	0.32	2.23	**1.79**	**0.35**	676	**0.36**	**6.71**	1.74	0.38	751.7
L31C	0.33	0.16	1.89	0.45	685	0.33	0.49	1.74	0.45	**678**	**0.33**	**2.76**	**1.90**	**0.42**	741
L32C	0.33	0.51	1.76	0.55	710	0.33	1.34	**1.85**	0.55	**702**	**0.33**	**3.11**	1.77	**0.54**	772.5
L41N	**0.34**	1.19	1.83	**0.52**	699	0.32	0.45	1.71	0.56	**681**	**0.34**	**3.50**	**1.89**	0.55	801.6
L42N	0.33	0.80	1.64	0.43	719	0.33	1.84	1.89	0.46	**705**	**0.33**	**6.92**	**1.97**	**0.43**	803.5
L41C	0.33	0.00	1.64	**0.37**	758	0.32	7.57	1.79	0.38	**746**	**0.34**	**8.45**	**2**	0.44	822.8
L42C	0.33	1.25	**1.93**	0.39	1084	0.32	1.27	1.85	0.41	**101**	**0.34**	**3.45**	1.79	**0.38**	1292
Average	0.33	0.68	1.77	0.44	362.3	0.33	1.04	1.84	0.45	**359**	**0.34**	**3.04**	**1.88**	**0.43**	393.3

Figure 4.10 represents the CPU time comparisons for all test problems and indicates that CPU time is not significantly different for the small and medium size problems while in large size problems NSGAII-2NI has the lowest runtime and NSGAII-FFI has the highest runtime. However, the runtime of NSGAII-FFI is more than others in medium and large size instances, the runtime difference (with maximum 120 seconds) can be negligible for decision maker due to superior performance of the algorithm in terms of four performance metrics. It must be mentioned that the runtime

of GAMS for FDP model is almost identical for different results of same test problem and, therefore, it has not been added to the heuristics CPU time.

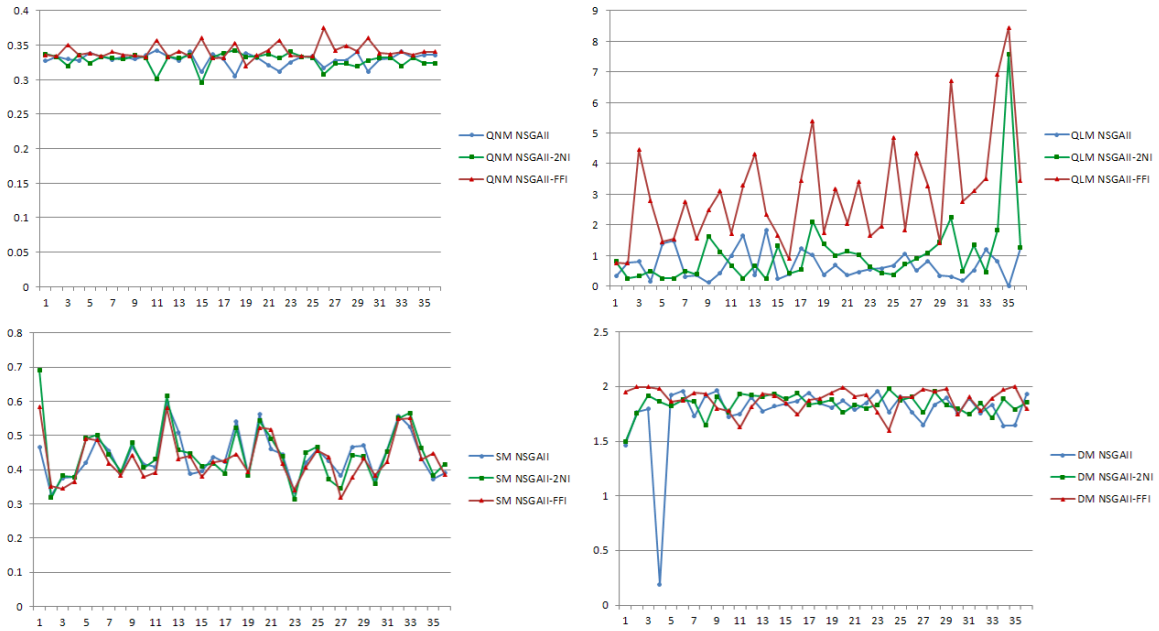


Figure 4.9: Graphical comparisons of the algorithms for 36 test instances

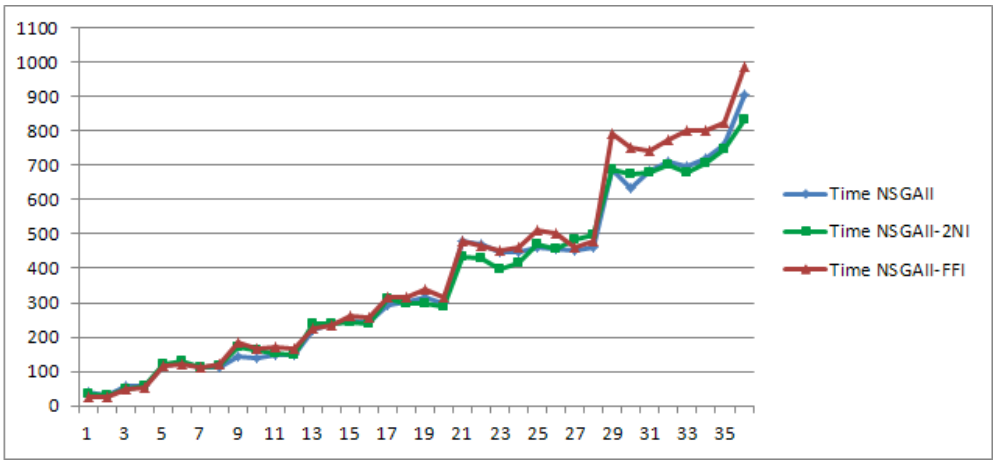


Figure 4.10: Algorithms comparison of CPU time on all test instances

It is important to note that for VS1 problem which is solved optimally in Section 4.3.3, all three algorithms gained exact Pareto optimal fronts in less than 7 seconds, and it shows at great capability of the algorithm in detecting the Pareto front solutions. Another remarkable fact worth mentioning is that in NSGAII-FFI by

improving the quality of first front solution in terms of QLM, the consequential improvement of quantity, diversity, and spread metrics are also observed. However, one direction search operator of NSGAII-2NI slightly improved the results in lower CPU time, while overall NSGAII-FFI performed best. In order to exemplify the results' analysis, the nondominated solutions of the proposed algorithms for a test problem are schemed in Figure 4.11.

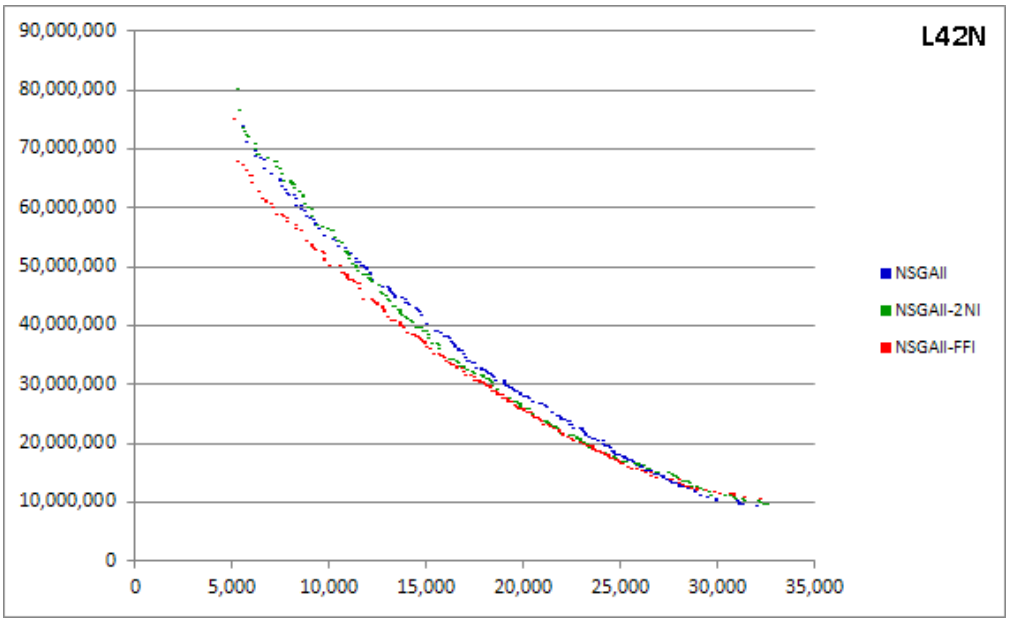


Figure 4.11: Obtained results of algorithms on a test instance of L42N (Horizontal axis is objective1 and vertical axis is obective 2). NSGAII and NSGAII-2NI have 196, and NSGAII-FFI has 198 Pareto front solutions.

4.6 Conclusion

In this study a bi-objective covering tour location routing problem with intermediate depots under service time limit and flexible final destination of vehicles was investigated. The mathematical model was presented with valid inequalities in order to decrease the computation time. Since the problem is NP-Hard, the NSGAII meta-heuristic with some modifications and two extra different search mechanisms was utilized to solve the problem. The NSGAII, NSGAII-2NI, and NSGAII-FFI

algorithms were applied on 36 randomly generated different sized test instances. Eventually based on the obtained results, it was expressed that:

- NSGAII-FFI is best in terms of QNM, QLM, DM, and SM;
- NSGAII-2NI is best in terms of runtime;

For future study one can consider the uncertain situations like road blockage, cost and humanitarian objective functions together, and time windows to develop the model. Moreover, different newly found multi-objective algorithms can also be applied and compared with the proposed ones in this chapter.

BIBLOGRAPHY

[1] L. N. Van Wassenhove, “Humanitarian aid logistics: supply chain management in high gear,” *J. Oper. Res. Soc.*, vol. 57, no. 5, pp. 475–489, May 2006.

[2] USGS, “USGS Website,” 2012. [Online]. Available: http://earthquake.usgs.gov/earthquakes/eqarchives/year/eqstats.php.

[3] D. Giardini, G. Grunthal, K. M. Shedlock, and P. Zhang, “The GSHAP Global Seismic Hazard Map,” *Ann. di Geofis.*, vol. 42, no. 6, pp. 1225–1230, 1999.

[4] R. Nemiroff and J. Bonnell, “Where People Live on Planet Earth,” 2003. [Online]. Available: http://apod.nasa.gov/apod/ap030305.html.

[5] M. Ghafory-Ashtiany and M. Hosseini, “Post-Bam earthquake: recovery and reconstruction,” *Nat. Hazards*, vol. 44, no. 2, pp. 229–241, Apr. 2007.

[6] J. Xu and Y. Lu, “A comparative study on the national counterpart aid model for post-disaster recovery and reconstruction: 2008 Wenchuan earthquake as a case,” *Disaster Prev. Manag.*, vol. 22, no. 1, pp. 75–93, 2013.

[7] A. Versluis and G. Adolphus, “Formal and informal material aid following the 2010 Haiti earthquake as reported by camp dwellers,” *Disasters*, vol. 38, pp. 94–109, 2014.

[8] R. T. Eguchi, “We have come a long way, yet we still have far to go,” *Nat. Hazards*, vol. 68, no. 1, pp. 201–202, Jan. 2013.

[9] E. Shieh, K. Habibi, K. Torabi, and H. E. Masoumi, “Earthquake risk in urban street network: an example from region 6 of Tehran, Iran,” *Int. J. Disaster Resil. Built Environ.*, vol. 5, no. 4, pp. 413–426, Nov. 2014.

[10] U. Kamp, L. a. Owen, B. J. Growley, and G. a. Khattak, “Back analysis of landslide susceptibility zonation mapping for the 2005 Kashmir earthquake: an

assessment of the reliability of susceptibility zoning maps," *Nat. Hazards*, vol. 54, no. 1, pp. 1–25, Sep. 2009.

[11] S. Kubo, H. Akimoto, and T. Moriwake, "First-Aid Transportation by Hovercraft in a Disaster," *Nat. Hazards*, vol. 29, no. 3, pp. 553–566, 2003.

[12] J. Qi *et al.*, "Search and Rescue Rotary-Wing UAV and Its Application to the Lushan Ms 7.0 Earthquake," *J. F. Robot.*, vol. 33, no. 3, pp. 290–321, May 2016.

[13] J. R. Current and D. A. Schilling, "The median tour and maximal covering tour problems: Formulations and heuristics," *Eur. J. Oper. Res.*, vol. 73, no. 1, pp. 114–126, Feb. 1994.

[14] S. Allahyari, M. Salari, and D. Vigo, "A hybrid metaheuristic algorithm for the multi-depot covering tour vehicle routing problem," *Eur. J. Oper. Res.*, vol. 242, no. 3, pp. 756–768, May 2015.

[15] J. D. VanVactor, "Strategic health care logistics planning in emergency management," *Disaster Prev. Manag.*, vol. 21, no. 3, pp. 299–309, 2012.

[16] A. Kiremidjian, E. Stergiou, and R. Lee, *Earthquake Geotechnical Engineering*, vol. 6. Dordrecht: Springer Netherlands, 2007.

[17] P. Anbazhagan, S. Srinivas, and D. Chandran, "Classification of road damage due to earthquakes," *Nat. Hazards*, vol. 60, no. 2, pp. 425–460, Nov. 2011.

[18] K. A. Hosseini, M. K. Jafari, M. Hosseini, B. Mansouri, and S. Hosseinioon, "Development of urban planning guidelines for improving emergency response capacities in seismic areas of Iran.," *Disasters*, vol. 33, no. 4, pp. 645–64, Oct. 2009.

[19] N. V. Sahinidis, "Optimization under uncertainty: state-of-the-art and opportunities," *Comput. Chem. Eng.*, vol. 28, no. 6–7, pp. 971–983, Jun. 2004.

[20] M. Najafi, K. Eshghi, and W. Dullaert, “A multi-objective robust optimization model for logistics planning in the earthquake response phase,” *Transp. Res. Part E Logist. Transp. Rev.*, vol. 49, no. 1, pp. 217–249, 2013.

[21] M. Najafi, K. Eshghi, and S. de Leeuw, “A dynamic dispatching and routing model to plan/ re-plan logistics activities in response to an earthquake,” *OR Spectr.*, vol. 36, no. 2, pp. 323–356, Jan. 2013.

[22] a. M. Anaya-Arenas, J. Renaud, and a. Ruiz, “Relief distribution networks: a systematic review,” *Ann. Oper. Res.*, vol. 223, no. 1, pp. 53–79, Apr. 2014.

[23] L. Ozdamar, “Planning helicopter logistics in disaster relief,” *OR Spectr.*, vol. 33, no. 3, pp. 655–672, Jun. 2011.

[24] H. C. Timothy, Christopher J. Nagy, M. A. Skoog, and I. A. Somers, “Civil UAV Capability Assessment,” *NASA Intern. Tech. report*, no. December, 2004.

[25] United Nations Office for the Coordination of Humanitarian Affarirs, “Unmanned Aerial Vehicles in Humanitarian Response,” 2014.

[26] C. A. Thiels, J. M. Aho, S. P. Zietlow, and D. H. Jenkins, “Use of Unmanned Aerial Vehicles for Medical Product Transport.,” *Air Med. J.*, vol. 34, no. 2, pp. 104–108, Jan. 2015.

[27] E. Buzhinsky, “the Outlook for Uav Research and Development,” *Secur. Index A Russ. J. Int. Secur.*, vol. 19, no. 3, pp. 59–65, Sep. 2013.

[28] F. Kendoul, “Survey of Advances in Guidance , Navigation , and Control of Unmanned Rotorcraft Systems,” vol. 29, no. 2, pp. 315–378, 2012.

[29] P. Pounds, R. Mahony, and P. Corke, “Modelling and control of a large quadrotor robot,” *Control Eng. Pract.*, vol. 18, no. 7, pp. 691–699, Jul. 2010.

[30] J. Watkinson, *Art of the Helicopter*. Elsevier, 2004.

[31] D. Miranda, C. M. Costa, and S. Lanceros-Mendez, “Lithium ion rechargeable

batteries: state of the art and future needs of microscopic theoretical models and simulations," *J. Electroanal. Chem.*, vol. 739, pp. 97–110, Dec. 2014.

[32] CDC, "Emergency Supplies for Earthquake Preparedness," 2014. [Online]. Available: http://www.bt.cdc.gov/disasters/earthquakes/supplies.asp.

[33] T. Kubo, Y. Hisada, M. Murakami, F. Kosuge, and K. Hamano, "Application of an earthquake early warning system and a real-time strong motion monitoring system in emergency response in a high-rise building," *Soil Dyn. Earthq. Eng.*, vol. 31, no. 2, pp. 231–239, Feb. 2011.

[34] S.-E. Park, Y. Yamaguchi, and D. Kim, "Polarimetric SAR remote sensing of the 2011 Tohoku earthquake using ALOS/PALSAR," *Remote Sens. Environ.*, vol. 132, pp. 212–220, May 2013.

[35] M. Erdik *et al.*, *Perspectives on European Earthquake Engineering and Seismology*, vol. 34. Cham: Springer International Publishing, 2014.

[36] A. Jotshi, Q. Gong, and R. Batta, "Dispatching and routing of emergency vehicles in disaster mitigation using data fusion," *Socioecon. Plann. Sci.*, vol. 43, no. 1, pp. 1–24, Mar. 2009.

[37] Y. Dong, Q. Li, A. Dou, and X. Wang, "Extracting damages caused by the 2008 Ms 8.0 Wenchuan earthquake from SAR remote sensing data," *J. Asian Earth Sci.*, vol. 40, no. 4, pp. 907–914, Mar. 2011.

[38] T. Feng *et al.*, "Estimation of earthquake casualties using high-resolution remote sensing: a case study of Dujiangyan city in the May 2008 Wenchuan earthquake," *Nat. Hazards*, vol. 69, no. 3, pp. 1577–1595, Jun. 2013.

[39] Y. Liang, J. Caverlee, and J. Mander, "Text vs . Images : On the Viability of Social Media to Assess Earthquake Damage," in *WWW '13 Companion Proceedings of the 22nd International Conference on World Wide Web*, 2013,

pp. 1003–1006.

[40] M. Heutger and M. Kückelhaus, “UNMANNED AERIAL VEHICLES IN LOGISTICS A DHL perspective on implications and use cases for the logistics industry,” 2014.

[41] F. P. Kemper, K. a. O. Suzuki, and J. R. Morrison, “UAV Consumable Replenishment: Design Concepts for Automated Service Stations,” *J. Intell. Robot. Syst.*, vol. 61, no. 1–4, pp. 369–397, Nov. 2010.

[42] M. Bernard and K. Kondak, “Autonomous Transportation and Deployment with Aerial Robots for Search and Rescue Missions,” vol. 28, no. 6, pp. 914–931, 2011.

[43] O. Karas, “UAV Simulation Environment for Autonomous Flight Control Algorithms,” 2012.

[44] Y. Toyoda and H. Kanegae, “A community evacuation planning model against urban earthquakes,” *Reg. Sci. Policy Pract.*, vol. 6, no. 3, pp. 231–249, Aug. 2014.

[45] Statistical Centre of Iran, “Statistical Center of Iran,” 2011. [Online]. Available: http://www.amar.org.ir/Default.aspx?tabid=1603.

[46] M. Ashtari Jafari, “Statistical prediction of the next great earthquake around Tehran, Iran,” *J. Geodyn.*, vol. 49, no. 1, pp. 14–18, Jan. 2010.

[47] K. Amini Hosseini, M. Hosseini, S. Hosseinioon, Y. O. Izadkhah, T. Shaw, and R. Takahashi, “A survey on evacuation planning and its challenges for potential earthquake in Tehran,” *Int. J. Disaster Resil. Built Environ.*, vol. 5, no. 1, pp. 38–52, Mar. 2014.

[48] S. Yaghmaei-Sabegh and N. T. K. Lam, “Ground motion modelling in Tehran based on the stochastic method,” *Soil Dyn. Earthq. Eng.*, vol. 30, no. 7, pp. 525–

535, Jul. 2010.

[49] Tehran Municipality, “ATLAS OF TEHRAN METROPOLIS,” 2006. [Online]. Available: http://atlas.tehran.ir/.

[50] JICA and CEST, “‘The study on seismic microzoning of the greater Tehran area in the Islamic Republic of Iran’. Japan International Cooperation Agency (JICA), Centre for Earthquake and Environmental Studies of Tehran (CEST), Tehran Municipality, Pacific Consultants Intern,” *Final Rep.*, no. Report No.: SSF/ J R/00–186, pp. 1–379, 2000.

[51] M. Pesaresi, A. Gerhardinger, and F. Haag, “Rapid damage assessment of built-up structures using VHR satellite data in tsunami-affected areas,” *Int. J. Remote Sens.*, vol. 28, no. 13–14, pp. 3013–3036, Jul. 2007.

[52] K. T. Chang and E. H. Wang, “Developing procedures for post-earthquake structural evaluation by laser scanning techniques,” *Insight - Non-Destructive Test. Cond. Monit.*, vol. 54, no. 10, pp. 562–567, Oct. 2012.

[53] Z. Szantoi, S. Malone, F. Escobedo, O. Misas, S. Smith, and B. Dewitt, “A tool for rapid post-hurricane urban tree debris estimates using high resolution aerial imagery,” *Int. J. Appl. Earth Obs. Geoinf.*, vol. 18, pp. 548–556, Aug. 2012.

[54] M. J. Olsen, Z. Chen, T. Hutchinson, and F. Kuester, “Optical techniques for multiscale damage assessment,” *Geomatics, Nat. Hazards Risk*, vol. 4, no. 1, pp. 49–70, Mar. 2013.

[55] A. G. Kashani, P. S. Crawford, S. K. Biswas, A. J. Graettinger, and D. Grau, “Automated Tornado Damage Assessment and Wind Speed Estimation Based on Terrestrial Laser Scanning,” *J. Comput. Civ. Eng.*, vol. 29, no. 3, p. 4014051, May 2015.

[56] S. Jiang and C. J. Friedland, “Automatic urban debris zone extraction from post-

hurricane very high-resolution satellite and aerial imagery," *Geomatics, Nat. Hazards Risk*, vol. 7, no. 3, pp. 933–952, May 2016.

[57] M. Janalipour and A. Mohammadzadeh, "Building Damage Detection Using Object-Based Image Analysis and ANFIS From High-Resolution Image (Case Study: BAM Earthquake, Iran)," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 9, no. 5, pp. 1937–1945, May 2016.

[58] J. Fernandez Galarreta, N. Kerle, and M. Gerke, "UAV-based urban structural damage assessment using object-based image analysis and semantic reasoning," *Nat. Hazards Earth Syst. Sci.*, vol. 15, no. 6, pp. 1087–1101, 2015.

[59] F. A. Al-Wassai and N. V. Kalyankar, "Major Limitations of Satellite images," vol. 4, pp. 51–59, Jul. 2013.

[60] D. Grandoni, "Advantages and limitations of using satellite images for flood mapping," *Workshop on the Use of the Copernicus Emergency Service for Floods*. e-GEOS, Brussel, 2013.

[61] A. Nedjati, B. Vizvari, and G. Izbirak, "Post-earthquake response by small UAV helicopters," *Nat. Hazards*, vol. 80, no. 3, pp. 1669–1688, Oct. 2015.

[62] S. Y. A. Zelinsky, R.A. Jarvis, J. C. Byrne, "Planning Paths of Complete Coverage of an Unstructured Environment by a Mobile Robot," in *Proceedings of International Conference on Advanced Robotics*, 1993, pp. 533–538.

[63] I. Colomina and P. Molina, "Unmanned aerial systems for photogrammetry and remote sensing: A review," *ISPRS J. Photogramm. Remote Sens.*, vol. 92, pp. 79–97, 2014.

[64] J. R. Montoya-Torres, J. López Franco, S. Nieto Isaza, H. Felizzola Jiménez, and N. Herazo-Padilla, "A literature review on the vehicle routing problem with multiple depots," *Comput. Ind. Eng.*, vol. 79, pp. 115–129, Jan. 2015.

[65] P. Oberlin, S. Rathinam, and S. Darbha, “Today’s Traveling Salesman Problem,” *IEEE Robot. Autom. Mag.*, vol. 17, no. 4, pp. 70–77, Dec. 2010.

[66] H.-M. Ho and J. Ouaknine, “The Cyclic-Routing UAV Problem is PSPACE-Complete,” in *Foundations of Software Science and Computation Structures*, Springer-Verlag Berlin Heidelberg, 2015, pp. 328–342.

[67] M. Gendreau, G. Laporte, and F. Semet, “The Covering Tour Problem,” *Oper. Res.*, vol. 45, no. 4, pp. 568–576, Aug. 1997.

[68] H. Choset, “Coverage for robotics – A survey of recent results,” *Ann. Math. Artif. Intell.*, vol. 31, no. 1/4, pp. 113–126, 2001.

[69] E. Galceran and M. Carreras, “A survey on coverage path planning for robotics,” *Rob. Auton. Syst.*, vol. 61, no. 12, pp. 1258–1276, Dec. 2013.

[70] R. N. De Carvalho, H. A. Vidal, P. Vieira, and M. I. Ribeiro, “Complete coverage path planning and guidance for cleaning robots,” in *ISIE ’97 Proceeding of the IEEE International Symposium on Industrial Electronics*, 1997, vol. 2, pp. 677–682.

[71] Y. Li, H. Chen, M. Joo Er, and X. Wang, “Coverage path planning for UAVs based on enhanced exact cellular decomposition method,” *Mechatronics*, vol. 21, no. 5, pp. 876–885, 2011.

[72] A. Barrientos *et al.*, “Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots,” *J. F. Robot.*, vol. 28, no. 5, pp. 667–689, Sep. 2011.

[73] M. Torres, D. A. Pelta, J. L. Verdegay, and J. C. Torres, “Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction,” *Expert Syst. Appl.*, vol. 55, pp. 441–451, 2016.

[74] X. Wang, T. Sun, and D. Li, “Energy-optimal coverage path planning on

topographic map for environment survey with unmanned aerial vehicles," *Electron. Lett.*, vol. 52, no. 9, pp. 699–701, Apr. 2016.

[75] K. Easton and J. Burdick, "A Coverage Algorithm for Multi-robot Boundary Inspection," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 727–734.

[76] W. H. Huang and N. York, "Optimal Line-sweep-based Decompositions for Coverage Algorithms," in *Proceedings of the 2001 IEEE International Conference on Robotics & Automation Seoul, Korea*, 2001, pp. 27–32.

[77] I. Maza and A. Ollero, "Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms," in *Distributed Autonomous Robotic Systems 6*, Tokyo: Springer Japan, 2007, pp. 221–230.

[78] G. Avellar, G. Pereira, L. Pimenta, and P. Iscold, "Multi-UAV Routing for Area Coverage and Remote Sensing with Minimum Time," *Sensors*, vol. 15, no. 11, pp. 27783–27803, Nov. 2015.

[79] X. Wang and D. Li, "Coverage path planning for UAVs in unknown directional regions," *Int. J. Wirel. Mob. Comput.*, vol. 8, no. 3, pp. 285–293, 2015.

[80] M. Lemmens, "Geo-information Technology – What It Is, How It Was and Where It Is Heading to," in *Geo-information*, Dordrecht: Springer Netherlands, 2011, pp. 1–22.

[81] F. Greenwood, "HOW TO MAKE MAPS WITH DRONES," in *DRONES AND AERIAL OBSERVATION: NEW TECHNOLOGIES FOR PROPERTY RIGHTS, HUMAN RIGHTS, AND GLOBAL DEVELOPMENT A PRIMER*, Washington D.C.: New America, 2015, pp. 35–48.

[82] E. Falkner and D. Morgan, *Aerial mapping : methods and applications*. Boca Raton, Fla.: Lewis Publishers, 2002.

[83] R. E. Gomory, “An algorithm for the mixed integer problem,” RAND Corporation, Santa Monica, 1960.

[84] E. Balas, S. Ceria, G. Cornuéjols, and N. Natraj, “Gomory cuts revisited,” *Oper. Res. Lett.*, vol. 19, no. 1, pp. 1–9, Jul. 1996.

[85] E. Klotz and A. M. Newman, “Practical guidelines for solving difficult mixed integer linear programs,” *Surv. Oper. Res. Manag. Sci.*, vol. 18, no. 1, pp. 18–32, 2013.

[86] M. Fischetti and A. Lodi, “Local branching,” *Math. Program.*, vol. 98, no. 1–3, pp. 23–47, Sep. 2003.

[87] I. Karaoglan and F. Altiparmak, “A memetic algorithm for the capacitated location-routing problem with mixed backhauls,” *Comput. Oper. Res.*, vol. 55, pp. 200–216, Mar. 2015.

[88] S. Salhi and G. K. Rand, “The effect of ignoring routes when locating depots,” *Eur. J. Oper. Res.*, vol. 39, no. 2, pp. 150–156, Mar. 1989.

[89] G. Nagy and S. Salhi, “Location-routing: Issues, models and methods,” *Eur. J. Oper. Res.*, vol. 177, no. 2, pp. 649–672, Mar. 2007.

[90] M. Drexl and M. Schneider, “A survey of variants and extensions of the location-routing problem,” *Eur. J. Oper. Res.*, vol. 241, no. 2, pp. 283–308, Mar. 2015.

[91] G. B. Dantzig and J. H. Ramser, “The Truck Dispatching Problem,” *Manage. Sci.*, vol. 6, no. 1, pp. 80–91, Oct. 1959.

[92] G. Laporte, “What you should know about the vehicle routing problem,” *Nav. Res. Logist.*, vol. 54, no. 8, pp. 811–819, Dec. 2007.

[93] N. Jozefowiez, F. Semet, and E.-G. Talbi, “The bi-objective covering tour problem,” *Comput. Oper. Res.*, vol. 34, no. 7, pp. 1929–1942, Jul. 2007.

[94] J. Faulin, “The Vehicle Routing Problem: Latest Advances and New Challenges,” *Interfaces (Providence).*, vol. 39, no. 6, pp. 555–556, Nov. 2009.

[95] J. Caceres-Cruz, P. Arias, D. Guimarans, D. Riera, and A. A. Juan, “Rich Vehicle Routing Problem,” *ACM Comput. Surv.*, vol. 47, no. 2, pp. 1–28, Dec. 2014.

[96] J. R. Montoya-Torres, J. López Franco, S. Nieto Isaza, H. Felizzola Jiménez, and N. Herazo-Padilla, “A literature review on the vehicle routing problem with multiple depots,” *Comput. Ind. Eng.*, vol. 79, pp. 115–129, Jan. 2015.

[97] B. Crevier, J.-F. Cordeau, and G. Laporte, “The multi-depot vehicle routing problem with inter-depot routes,” *Eur. J. Oper. Res.*, vol. 176, no. 2, pp. 756–773, Jan. 2007.

[98] M. M. Solomon, “On the worst-case performance of some heuristics for the vehicle routing and scheduling problem with time window constraints,” *Networks*, vol. 16, no. 2, pp. 161–174, 1986.

[99] Y. H. Lee, J. W. Jung, and K. M. Lee, “Vehicle routing scheduling for cross-docking in the supply chain,” *Comput. Ind. Eng.*, vol. 51, no. 2, pp. 247–256, Oct. 2006.

[100] S. Shahin Moghadam, S. M. T. Fatemi Ghomi, and B. Karimi, “Vehicle routing scheduling problem with cross docking and split deliveries,” *Comput. Chem. Eng.*, vol. 69, pp. 98–107, Oct. 2014.

[101] X.-X. Rong, Y. Lu, R.-R. Yin, and J.-H. Zhang, “A Robust Optimization Approach to Emergency Vehicle Scheduling,” *Math. Probl. Eng.*, vol. 2013, pp. 1–8, 2013.

[102] X. Gan, Y. Wang, J. Kuang, Y. Yu, and B. Niu, “Emergency Vehicle Scheduling Problem with Time Utility in Disasters,” *Math. Probl. Eng.*, vol.

2015, pp. 1–7, 2015.

[103] S. Yan, S.-S. Wang, and M.-W. Wu, “A model with a solution algorithm for the cash transportation vehicle routing and scheduling problem,” *Comput. Ind. Eng.*, vol. 63, no. 2, pp. 464–473, Sep. 2012.

[104] S. Yan, J. C. Chu, F.-Y. Hsiao, and H.-J. Huang, “A planning model and solution algorithm for multi-trip split-delivery vehicle routing and scheduling problems with time windows,” *Comput. Ind. Eng.*, vol. 87, pp. 383–393, Sep. 2015.

[105] M. Gendreau, G. Ghiani, and E. Guerriero, “Time-Dependent Routing Problems: a review,” *Comput. Oper. Res.*, vol. 64, pp. 189–197, Jun. 2015.

[106] M. J. Hodgson, G. Laporte, and F. Semet, “A Covering Tour Model for Planning Mobile Health Care Facilities in SuhumDistrict, Ghama,” *J. Reg. Sci.*, vol. 38, no. 4, pp. 621–638, Nov. 1998.

[107] L. Vogt, C. A. Poojari, and J. E. Beasley, “A tabu search algorithm for the single vehicle routing allocation problem,” *J. Oper. Res. Soc.*, vol. 58, no. 4, pp. 467–480, 2006.

[108] P. C. Nolz, K. F. Doerner, and R. F. Hartl, “Water distribution in disaster relief,” *Int. J. Phys. Distrib. Logist. Manag.*, vol. 40, no. 8/9, pp. 693–708, Sep. 2010.

[109] P. C. Nolz, K. F. Doerner, W. J. Gutjahr, and R. F. Hartl, *A Bi-objective Metaheuristic for Disaster Relief Operation Planning*. Springer Berlin Heidelberg, 2010.

[110] F. Tricoire, A. Graf, and W. J. Gutjahr, “The bi-objective stochastic covering tour problem.,” *Comput. Oper. Res.*, vol. 39, no. 7, pp. 1582–1592, Jul. 2012.

[111] C. D. Tarantilis, E. E. Zachariadis, and C. T. Kiranoudis, “A Hybrid Guided Local Search for the Vehicle-Routing Problem with Intermediate

Replenishment Facilities," *INFORMS J. Comput.*, vol. 20, no. 1, pp. 154–168, 2008.

[112] A. G. H. Kek, R. L. Cheu, and Q. Meng, "Distance-constrained capacitated vehicle routing problems with flexible assignment of start and end depots," *Math. Comput. Model.*, vol. 47, no. 1–2, pp. 140–152, Jan. 2008.

[113] A. M. Campbell, D. Vandenbussche, and W. Hermann, "Routing for Relief Efforts," *Transp. Sci.*, vol. 42, no. 2, pp. 127–145, May 2008.

[114] I. Karaoglan and F. Altiparmak, "A memetic algorithm for the capacitated location-routing problem with mixed backhauls," *Comput. Oper. Res.*, vol. 55, pp. 200–216, Mar. 2015.

[115] C.-L. Hwang and A. S. M. Masud, *Multiple Objective Decision Making — Methods and Applications*, vol. 164. Berlin, Heidelberg: Springer Berlin Heidelberg, 1979.

[116] Y. . Haimes, L. . Ladson, and D. . Wismer, "On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization," *IEEE Trans. Syst. Man. Cybern.*, vol. 1, no. 3, pp. 296–297, 1971.

[117] S. Karakatič and V. Podgorelec, "A survey of genetic algorithms for solving multi depot vehicle routing problem," *Appl. Soft Comput.*, vol. 27, pp. 519–532, Feb. 2015.

[118] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[119] J. C. Bean, "Genetic Algorithms and Random Keys for Sequencing and Optimization," *ORSA J. Comput.*, vol. 6, no. 2, pp. 154–160, May 1994.

[120] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary

algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.

[121] N. Nekooghadirli, R. Tavakkoli Moghaddam, V. R. Ghezavati, and S. Javanmard, "Solving a new bi-objective location-routing-inventory problem in a distribution network by meta-heuristics," *Comput. Ind. Eng.*, vol. 76, pp. 204–221, 2014.