TWO-ECHELON VEHICLE ROUTING PROBLEMS USING UNMANNED AUTONOMOUS VEHICLES

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Md Mahbubar Rahman

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Industrial and Manufacturing Engineering

June 2017

Fargo, North Dakota

NORTH DAKOTA STATE UNIVERSITY

Graduate School

Title

TWO-ECHELON VEHICLE ROUTING PROBLEMS USING UNMANNED AUTONOMOUS VEHICLES

By

Md Mahbubar Rahman

The supervisory committee certifies that this thesis complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Chrysafis Vogiatzis

Yiwen Xu

Anne M. Denton

Approved:

June 29, 2017 — Date

Om Prakash Yadav — Department Chair

ABSTRACT

In this thesis, we investigate new multi-echelon vehicle routing problems for logistics operations using unmanned autonomous vehicles. This can provide immediate tangible outcomes, especially in high-demand areas that are otherwise difficult or costly to serve. This type of problem differs from the commonly used multi-echelon supply chain management systems in that here there exist no intermediate facilities that consolidate/separate products for delivery; instead all decisions are made on a per-vehicle basis. We describe here how we can obtain the necessary parameters (data collection) to evaluate the performance of such multi-echelon systems. We also provide three mathematical formulations based on different assumptions and case scenarios. We then study the differences between the three models in practice, as far as routing cost and duration of operations are concerned. We finally show that there are savings to be had by properly employing unmanned vehicles for logistics operations.

ACKNOWLEDGEMENTS

I would like to give my earnest thanks to my advisor, Professor Chrysafis Vogiatzis, for his endless support, advice, and encouragement during my MSc study. His continuous guidance helped me succeed in my interested area of research. He has also been a great help in organizing and writing this research.

I would like to express my special thanks to Professor Anne M. Denton for her great help in doing Java projects for my research. I could not perform these projects without her guidance and help.

I am particularly grateful to Professor Yiwen Xu for his advice and comments during my time at NDSU. I would also like to thank Professor Om Prakash Yadav, and Professor AKM Bashir Khoda for their comments and suggestions.

DEDICATION

I dedicate this research to my parents Md Ashequr Rahman and Most Momena Khatun, and to my dearest siblings.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Typically freight is transferred from its origin (suppliers/production plants) to its destination (customers) through one or more intermediate facilities (distribution centers/warehouses). This type of distribution system is called multi-echelon distribution system where origin, destination, and intermediate facilities are termed as layers/stages and each pair of layers transferring freight from one to another is termed as a level or an echelon. A special case of muti-echelon distribution system is two-echelon distribution system. This distribution system composed of three layers: origin, intermediate facilities or satellites, and destination, and two echelons: origin-satellite or first echelon, and satellite-destination or second echelon. First echelon vehicle(s) is used to transfer freight from its origin to its satellite station, and second echelon vehicle(s) is used to transfer freight from its satellite station to its destination.

To minimize the total cost of two-echelon distribution system two echelon vehicle routing problems (2E-VRPs) are formulated. In these problems freight's origin and destination(s) are fixed, a set of satellites are given, and the goal is to find a set of routes for first echelon vehicles and second echelon vehicles.

In this thesis, we investigate new multi-echelon vehicle routing problems for logistics operations in high-demand areas that are otherwise difficult to reach and serve. This extension of the classical vehicle routing problem is different than the similarly named and commonly used multi-echelon supply chain management systems. The difference lies in the fact that in this extension there exist no intermediate facilities that consolidate/separate products for delivery; instead all decisions are made on a per-vehicle basis. This is a new type of problem that has though tangible advantages in modern logistics systems, namely:

(a) it can be used to parallelize the "last-mile" of the delivery process;

(b) it enables *city logistics* where larger vehicles ensure the routing of goods from one general geographic location to another, while smaller (and potentially environmentally friendly) vehicles are tasked with the final delivery within an urban area;

(c) with the increase of the availability of unmanned vehicles, it provides us with a venue to automate the routing process, while at the same time servicing areas that are difficult to reach, or totally inaccessible

For the first advantage, let us consider the example of a traditional postal service, where a truck parks in a convenient location, and the employee/driver picks up packages of nearby customers and walks them from door to door. This happens because it is easier to serve multiple customers in the vicinity of the parked vehicle, before returning to it and leaving for the next set of suitable customers. Similarly, in a two-echelon vehicle routing problem, the first vehicle would stop and wait somewhere appropriately before releasing the other, more flexible vehicle to serve that area. In the case of multiple secondary vehicles, this can be easily parallelized, minimizing the total time to delivery.

Continuing, in many cases, larger trucks are not allowed to park and/or stop within the confines of an urban center. Very often the sheer size of the trucks makes them unfit for use within a city. This creates the need for multi-echelon vehicle routing, where again the secondary units are released to serve the demand in areas inaccessible to the original vehicle. In the case of unmanned aerial and ground vehicles, some more advantages are also the decrease in emissions and the limited traffic caused during the last-mile.

Last, and perhaps more importantly in our work, both drones and autonomous vehicles are gaining traction. They have been extensively used for data collection, surveillance and monitoring, as well as in military operations, however private initiatives are now positioning them as a major player for logistics operations. As an example, recently Amazon stated that their goal is to have products delivered to special customers within less than 30 minutes with the use of drones. While research has been ample for the technical characteristics of unmanned aerial vehicles, the same cannot be said for the logistical challenges associated with this new paradigm. We provide more details about these challenges and how our research is slated to address them in the next section.

1.1. Problem Definition

Formally, the problem we aim to address can be described as follows. Contrary to the traditional vehicle routing problem, we consider here a two echelon system where a customer can be served by a vehicle of either echelon. A first echelon vehicle can also stop and deploy any and

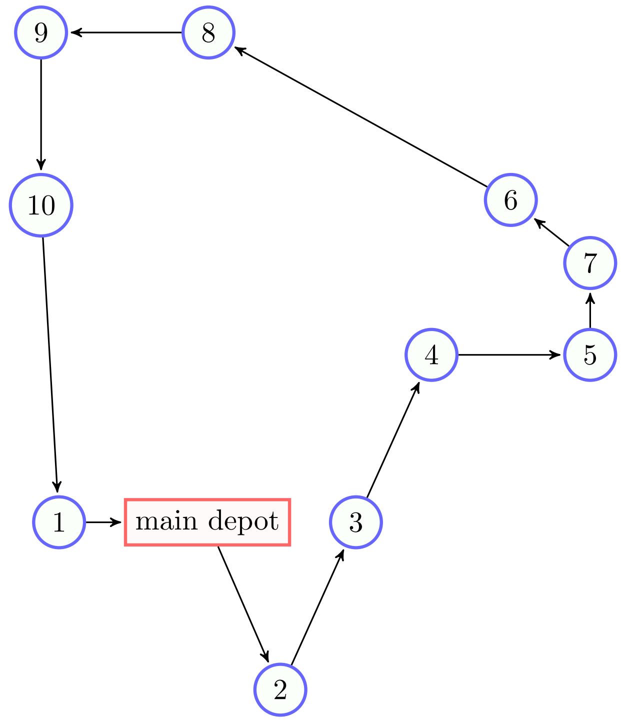

Figure 1.1. Traditional approach where a delivery truck visits all customers.

all of the second echelon vehicles it controls and then wait for them to service the area before returning. Some of the assumptions that we consider in this work follow:

A1. There exist a set of locations from where a second echelon vehicle can be deployed and/or picked up, and no deployment or pick-up is allowed en route.

A2. A second echelon vehicle deployed from a specific first echelon vehicle needs to return to that same vehicle.

A3. The first and second echelon vehicles experience different travel distances and can traverse a predefined set of arcs. This implies that some customers might be accessible by only certain vehicles and that specific modes can cover faster or slower certain routes.

A4. A second echelon vehicle has a predefined capacity in that it can only serve a limited number of customers at each round performed. Second echelon vehicles are allowed to be deployed multiple times from the same first echelon vehicle, so long as they are picked up at the end of each round.

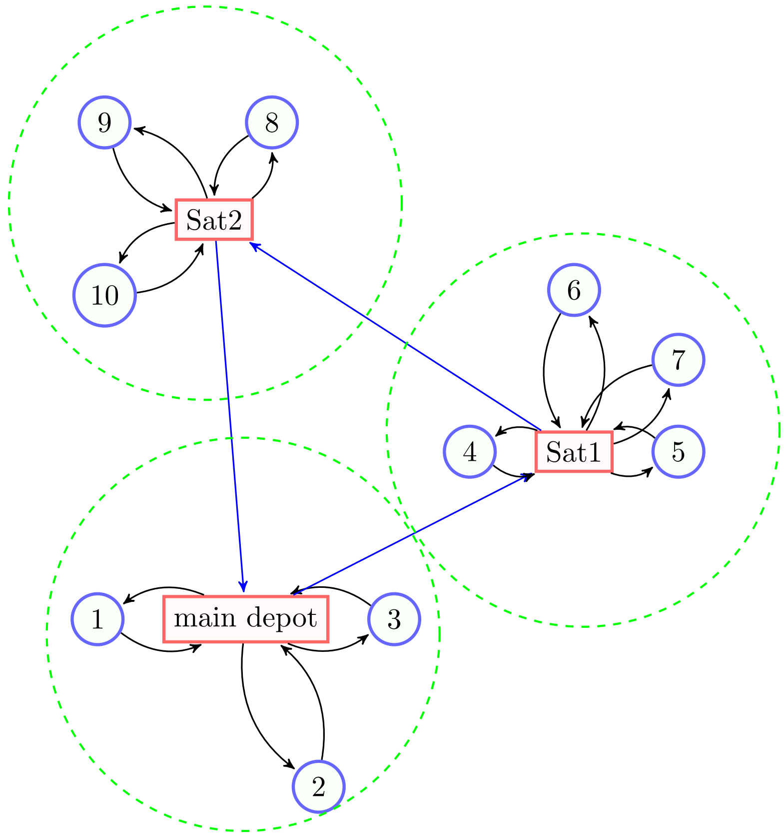

Figure 1.2. Drone delivers to all customers within flight range from main depot and satellite stations and truck moves within main depot and satellite stations.

The above problem definition is naturally fitted for applications involving autonomous vehicles, but can also describe problems where a vehicle is responsible for the transportation of many different actors in an area, where each actor can then be routed where needed. As a potentially transformative application, consider a simple humanitarian supply chain model where doctors are needed in a location after a humanitarian crisis. Employing a framework like the above to model the routing decisions, doctors can be transported from city to city within a larger, safer, and faster convoy, but can then move from patient to patient or from neighborhood to neighborhood using more convenient means of transportation, such as motorbikes or smaller cars. For a small example of how the traditional routing problems in logistics differ to the problem defined here, the interested reader is referred to Figures 1.1 and 1.2, which present the two versions of the problem (classical VRP vs. 2-echelon VRP).

1.2. Outline

This thesis is structured as follows. We proceed with an extensive literature review of classical vehicle routing problems, simultaneous facility location and vehicle routing problems, and two-echelon vehicle routing problems in Chapter 2. Then, in Chapter 3, we discuss the data collection and organization processes used in this thesis. Chapter 4 focuses on three mathematical programs designed to model and solve three specific instances of the problem defined above. We then proceed to describe our experimentation and our results in Chapter 5. This thesis concludes with our observations and remarks, along with our insight in future work in Chapter 6.

2. LITERATURE REVIEW

The Vehicle Routing Problem (VRP) is a very well-studied combinatorial optimization problem. It is a generalization of the infamous traveling salesman problem (TSP). Ever since the VRP was proposed as a single vehicle version [7], it has been hailed as one of the most important optimization problems. In the classical VRP, a set of vehicle routes is devised in order to satisfy the following set of "rules": (i) every customer location is visited exactly once, (ii) every vehicle route begins and ends at the same depot, and (iii) a set of other constraints is satisfied. The last set of constraints can be as diverse as vehicle capacities (i.e., no vehicle can satisfy more than an upper bound of demand at each round), priorities (e.g., a specific customer needs to be visited before or after anothe customer), time windows constraints (i.e., customers can only be served in specific times), among many others. As its practical applications are numerous, it comes as no surprise that significant research has been invested in the context of vehicle routing problems.

The VRP is well-known to be $\mathcal{NP}$-hard [13], even under the assumption that the underlying network is planar. More specifically, the problem remains $\mathcal{NP}$-hard for distances that satisfy the triangle inequality, as is the case in our work. After the first works that focused on a single vehicle version of the problem, focus shifted towards multi-vehicle counterparts since their introduction in [8]. Therein, a mathematical formulation is first introduced, to be followed by a decomposition of the original problem into smaller TSPs that are solved using linear programming. This work was immediately followed by the introduction of one of the most used heuristic approaches, the *savings method*, as described in [4]. Since then, of course, the amount of research in the topic has been exponentially increased. We refer the interested reader in the excellent surveys of VRP models, solution approaches, and challenges in [12], [10], [17], as well as the taxonomic review offered by Eksioglu et al. [9].

Our problem involves studying both vehicle routing and facility location in a simultaneous setting. An application of such a setting in *healthcare logistics* in the Netherlands is provided in [18]. In their work, a local pharmacy is wanting to set up lockers that can be used by prospective patients: if a patient is within the coverage zone (radius) of a specific locker, then they no longer need to be visited by a deliver vehicle. Instead, they an pick up their medication by a simple visit to their assigned locker. To solve the problem, a mixed integer linear program is formulated and a hybrid heuristic method is devised. The authors also show that the heuristic method consistently outperforms exact methods, such as branch-and-bound, as far as solution time is concerned.

In modern supply chain systems, we rely on an interrelated arrangement of plants, warehouses (distribution centers), and transportation networks to deliver the final goods to customers. A study of the existing complex distribution network design problem is then offered in [2]: therein, they consider a distribution network consisting of four layers. Those layers are, namely, supply points, central depots, regional facilities, and demand points. The setup of the problem then becomes the following. Where should the central depots and regional facilities be located? How are clients allocated/assigned to the open facilities? How are vehicles routed from supply points to central depots; from central depots to regional facilities; from regional facilities to end customers? Last, how much inventory should be the target in each of the central depots and regional facilities to maintain a desirable customer service level? To answer the above questions in [2], a comprehensive, large-scale mixed integer linear program is formulated and solved using commercial solvers, such as CPLEX.

A two-echelon distribution network is then a special case of multi-echelon distribution network. There are three layers in a two-echelon distribution network and these are supply points or depots, intermediate points or satellites, and customers. Freights are transferred from depots to satellites using first echelon vehicles, and from satellites to customers using second echelon vehicles. In [6], the authors address decision problems of two-echelon distribution network design as two-echelon location routing problems where decisions on the location of depots and satellites and decisions on the routing of first and second echelon vehicles are to be made simultaneously. To tackle this proble, Crainic et al. develop three mixed integer programming formulations: a three index formulation, a two index formulation, and a one index formulation for two-echelon location routing problems [6].

Two-echelon vehicle routing problems (2E-VRPs) are solved under a common objective of minimizing the global routing cost of first echelon vehicles (usually associated with freight costs of transferring goods from a depot to a satellite stations) and second echelon vehicles (associated with the final delivery of goods to customers from a satellite station). In their work, [5] have performed three sets of experiments to investigate how the objective value in 2E-VRP varies as instance parameters are varied: these parameters include the mean transportation cost, the accessibility index, the customer distribution, the satellite location rules, the depot location, the number of satellites. In the first set of experiment in their work, the mean transportation cost from the depot to the satellites, the mean accessibility index of the satellites, and the global routing costs for both 2E-VRP and the regular VRP are calculated for a set of instances. After that the ratio of the global routing cost of VRP, and 2E-VRP are calculated. In the second set of experiments, the impact of different customer and satellite location distributions, as well as varying the number of satellites are analyzed to see their effect on the cost of the 2E-VRP. At the last set of experiments, the impact of variable customer density, as well as depot locations are analyzed on global routing cost of 2E-VRP and VRP.

Two-echelon VRPs are seeing increasing interest, due in part to the advent of unmanned, autonomous vehicles. A *drone*, technically described as unmanned aerial vehicle, is a remote controlled flying robot. Because of its remote control capabilities and its autonomous flying features, it has been well-suited for military purpose; nowadays though, it is also used for a series of other purposes, including but not limited to surveying, aerial photography, monitoring and searching, and even for commercial delivery purposes. Even though drones are now quite ubiquitous and they enjoy multiple uses in everyday life, the optimal logistics and operations for deploying them in a commercial setting are still a major area of research.

In a recent contribution [15], the authors have researched a scenario in which a traditional vehicle (e.g., a truck), carries a drone which is controlled and routed to satisfy the demand of customers located in different geographic locations. The characteristics of the scenario investigated in their work are similar to the ones investigated here: the drone can be deployed from and picked up by the vehicle multiple times, it can only visit one customer before returning to the vehicle, it has a maximum range of flight, and there exist customers that (due to capacity considerations) can never be satisfied by a drone and will instead be visited by the truck. This problem, aptly named "the flying sidekick problem", is solved via the introduction of two mixed integer linear programming formulations. Larger scale problems that are of practical interest are solved via two heuristic solution procedures. To that extent, in [1] the authors investigate a variant of the TSP with drones, and they propose a series of "first route then cluster" heuristic approaches to solving the problem. These heuristics are based on local search and dynamic programming techniques.

While recent focus on routing problems with drones has peaked, there was no quantitative measure of the extent in which the solution of this new paradigm actually helps defray costs obtained by solving traditional VRPs. This gap was recently filled by Wang et al., with an excellent worst-case analysis of how much the duration of operations with and without drones can be affected [19]. To do that the authors develop two sets of problems for satisfying the same set of customers: the first problem is viewed as a traditional vehicle routing problem, while the second one is defined with similar characteristics to the work in [15], with the exception of allowing for a truck to carry more than one drones, which enables parallelization.

We finish this overview of the literature with a note on the navigation of unmanned aerial vehicles. Most of them are well equipped to be navigated by civil global positioning system (GPS) signals [14]. However, the open nature of this system makes such systems vulnerable to GPS spoofing. The University of Texas Radionavigation Laboratory indeed developed a spoofer that can be used to counterfeit civil GPS signals and hijack an unmanned aerial vehicle [11]. Although a number of civil GPS spoofing defense techniques have been developed, none of them are foolproof. This is indeed cause for careful planning, as it has been identified as one of the most important cyberthreats to the viability of autonomous and connected vehicles [16].

Another concern with the viability of using unmanned vehicles for logistics operations is their positional accuracy. However, positional accuracy of customer grade global positioning system (GPS) devices have improved significantly over the last years by differential correction i.e., incorporating wide area augmentation system (WAAS) with GPS. Horizontal positional accuracy of GPS has improved to 3-4 meters from 10-15 meters after differential correction [3].

In general, unmanned aerial vehicles are controlled by a human operator through a common data link (CDL) or video data link (VDL) from a ground station. This remote control makes UAVs vulnerable and to overcome this vulnerability [20] propose a decentralized control strategy, aptly named region-sharing strategy. Solution of decentralized time allocation problem required to implement region-sharing strategy but it is computationally intractable. In their work, the authors develop an approximate formulation which first decompose the time allocation problem and then the decomposed problem is solved by decentralized Markov decision process. This work is a great indication of how UAV vulnerabilities can be addressed, improved upon, and lead to fruitful policies through the proper use of mathematical modeling and operations research.

3. DATA COLLECTION SYSTEMS AND DATABASE

3.1. Introduction

Both distance matrices and travel-time matrices are very commonly used in location problems, vehicle routing, and, of course, logistics and multi-echelon distribution systems. Combinatorial optimization models for these fields and their problems often rely on such matrices and the solutions of the above models can vary based on the matrix contents. Even though their applications are vast, obtaining the actual distance and travel time matrices can prove time consuming and tiresome. In this chapter, we present two Java projects that have been specifically developed to obtain the road distance matrix and the aerial distance ("as the crow flies") matrix easily. The contents of these matrices is then saved in a relational database, such as MySQL, for ease of access.

3.2. Methods

Distance matrices are positively correlated with travel time matrices and, naturally, in some cases they can be and are used interchangeably. To generate the distance matrix given a range of points of interest, we can calculate the *great circle distance* (used for measuring the shortest distance between two points in a spherical surface), the *Euclidean distance* (used for measuring shortest distance between two points in a plane surface), among other metrics. In location science, vehicle routing, and distribution system design Euclidean distances are more common. Great circle distances are typically used for airplane routing.

3.2.1. Methods for Road Distance Matrix

In our case, Euclidean distances are used to calculate the road distance between two given points. However, this can prove cumbersome and unrealistic, since even in the case of "straight-line roads", the distance experiences is rarely the straight line distance. Moreover, in the case of curvilinear roads, the process of calculating this distance can become even harder. That said, an openly available resource for calculating these distances (and their traversal time counterparts) is *Google Maps*. Using the Google Maps API, one can obtain the distance and travel time between two points for a sequence of different modes, including driving, walking, biking, etc. In this work, we use the Google Maps API to obtain and store the road distance and travel time between two points for a specific travel mode.

A java application is developed to get the road distances and the travel times from Google Maps, and they are then used to store and build the distance matrix and the travel time matrix. The block diagram of this java project is given in Figure 3.1. In this application, first, the java program takes input file using Scanner class. This input file has two columns, one column is for latitudes and the other one is for longitudes. Each row of this input file represents a location. After getting the locations from input file the program sends request to Google Maps for getting distance and travel time between an origin and a destination using HttpUrlConnection class. Google Maps sends its response to the java program in JSON format. Program then separates specific array elements containing distance and travel time of the JSON object. After that the main program creates output files of distance matric and travel time matrix using PrintWriter class, and stores these values in the MySQL data base via JDBC class. Each time the program is run it creates a new distance matrix file and a new travel time matrix file but it saves all the distance and travel time data in the data base.

3.2.2. Methods for Aerial Distance Matrix

Great Circle distance is used to calculate aerial distance between two points on a sphere. The formula to get the Great Circle distance between two points is given below:

$$d = r \cdot \arccos\big(\sin\phi_1 \cdot \sin\phi_2 + \cos\phi_1 \cdot \cos\phi_2 \cdot \cos(\Delta\lambda_{12})\big)$$

where,

$$\begin{aligned} d &= \textit{aerial distance between two locations,} \\ r &= \textit{radius of the Earth,} \\ \phi_i &= \textit{latitude of a given location } i, \\ \lambda_i &= \textit{longitude of a given location } i, \\ \Delta\lambda_{ij} = |\lambda_i - \lambda_j| &= \textit{absolute difference between two longitudes } i \textit{ and } j. \end{aligned}$$

A java application is developed to calculate the aerial distance between pair of locations, to create aerial distance matrix, and to save them in the database. The block diagram of this

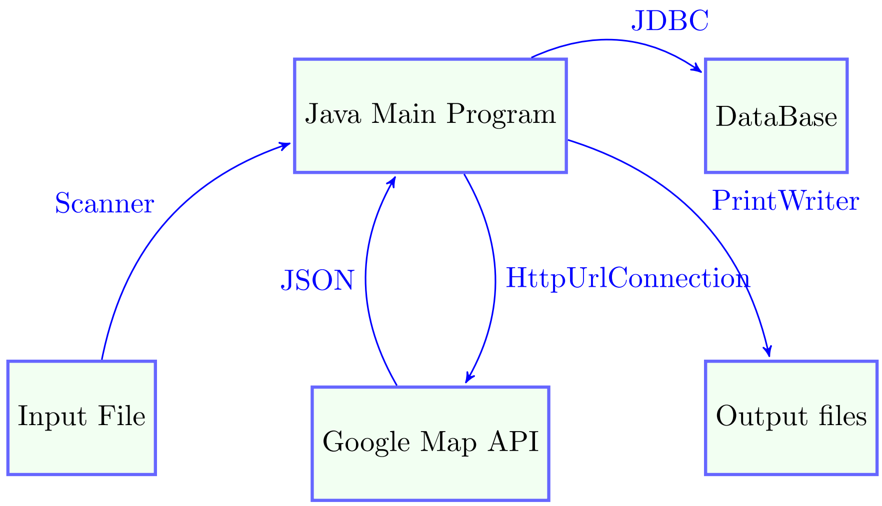

Figure 3.1. Block design of road distance and travel time extracting system from Google Maps.

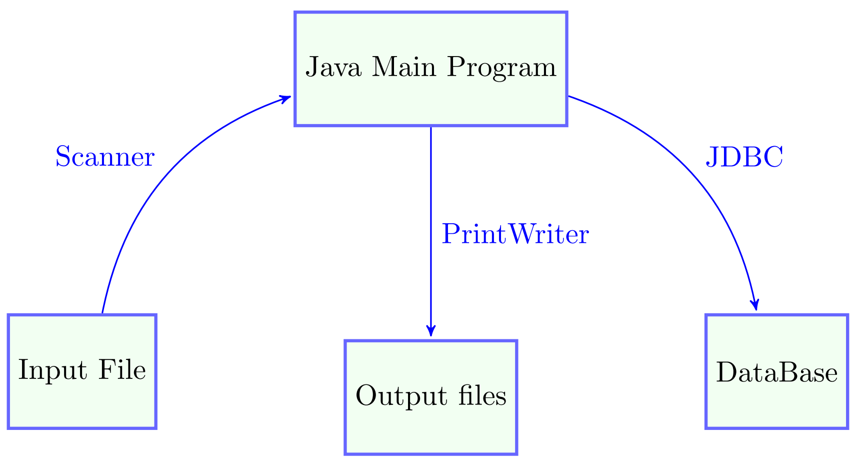

Figure 3.2. Block design of aerial distance measuring system using Great Circle Distance.

application is given in Figure 3.2. In this application, first, the java program takes input file using Scanner class. This input file has two columns, one column is for latitudes and the other one is for longitudes. Each row of this input file represents a location. After getting the locations from input file the program calculates aerial distance between every pair of locations using Great Circle distance formula and save them in the database. After that the main program creates output file of aerial distance matrix.

3.3. Conclusion

The two java applications developed to calculate, create, and save road distance matrix and aerial distance matrix are used to provide with data inputs the models presented in the following chapter.

4. MATHEMATICAL FORMULATIONS

In this chapter, we discuss the three mathematical models and their notations. We begin with a general overview of the notation, however we also provide the necessary notation for improved readability upon the beginning of each mathematical model description. The three mathematical models presented here are named *Single Vehicle Drone-Only Delivery*, *Single Vehicle Drone and Truck Delivery*, and *Multiple Vehicle Drone and Truck Delivery*, respectively.

4.1. Notation

Let G(V,E) be a graph on $n = |V|$ vertices and $m = |E|$ edges and let K_1,K_2 be the set of vehicles in the first and second echelon respectively. For simplicity, we assume that all vehicles in the set K_2 can be found within every vehicle $k \in K_1$. Node set V consists of the original depot locations (set O), the customers (set C), the satellite locations that can be used to deploy and pick up a second echelon vehicle (set S), and a set of intermediary locations on the map (set I). The edge set E can be defined as the union of a series of sets $E_k, \forall k \in (K_1, K)_2$. Each set E_k represents the edges that vehicle k can traverse; associated with every edge $(i, j) \in E_k, \forall k \in K_1, K_2$ we have the time that vehicle k needs to traverse the edge, T^k_{ij}. Similarly, for every customer node $i \in C$ and every vehicle $k \in K_1, K_2$, we have a parameter d^k_i that signals the demand of that specific customer for the unit in question, and a parameter (tied to the demand) λ^k_i to capture the time that the unit needs to spend at the location of the customer to fully serve them. Last, every satellite station is associated with a parameter μ_s denoting the set-up time, as well as deployment and picking up times for all second echelon vehicles $k \in K_2$ in that same station, denoted by V^k_s.

We can now proceed to define the decision variables of our optimization models. Our decisions are two-fold (strategic and operational) and can be summed up to the following questions:

(a) Which customers are visited by a first and which by a second echelon vehicle?

(b) In which satellite stations are the first echelon vehicles expected to stop and deploy the second echelon vehicles?

(c) How are the vehicles routed from location to location, starting from a depot/satellite station and ending in that same depot/satellite station, for first and second echelon vehicles, respectively?

(d) How much time does a unit spend at each location it stops?

We now define the following binary variables; in parentheses, we provide the question (a, b, c, and d) that each decision variable serves to address. First, define $u_s^c = 1$ if customer $c \in C$ is assigned to a second echelon vehicle deployed from satellite station $s \in S$; similarly, let $v_k^c = 1$ if customer $c \in C$ is assigned to first echelon vehicle $k \in K_1$ (a). Clearly, every customer needs to be assigned to one or the other, leading to an assignment constraint that will be described later. Continuing, let $y_s^k = 1$, if vehicle $k \in K_1$ is planned to set up and deploy its second echelon vehicles at satellite station $s \in S$ (b).

Now, for the routing decisions we define variables f_{ij}^k and $x_{ij}^{(k,s)}$ for the first and second echelon vehicles, respectively. More specifically, f_{ij}^k is equal to 1 if vehicle $k \in K_1$ is routed using $(i,j) \in E_k$ and $x_{ij}^{(k,s)}$ if vehicle $k \in K_2$ uses arc $(i,j) \in E_k$, when deployed from satellite station $s \in S$ (c). Note that the second echelon vehicles can be redeployed from several satellite stations and hence it is important to properly index the decision variable to keep track of that.

For time considerations, we are concerned with two components. The first one has to do with the time a second echelon vehicle takes to perform a round, while the second one deals with the time that a first echelon vehicle spends waiting for its second echelon counterparts to return. For the former, let t_s^k be the time that vehicle $k \in K_2$ takes to return to satellite station $s \in S$; for the latter, define w_s^k to be the time that any first echelon $k \in K_1$ vehicle spends idle at satellite station $s \in S$ (d). Finally, some auxiliary variables are necessary. We define $l_i, \forall i \in V$ as an integer variable used for subtour elimination and $z_s^k, \forall s \in S, \forall k \in K_2$ as a binary selection variable whenever second echelon vehicle k is deployed when the first echelon vehicle is set up at satellite station s: equivalently, this implies that second echelon vehicle k is required by a customer c that has been assigned to satellite station s.

4.2. Mathematical Formulations

We develop three models in an attempt to capture the intricacies of different use scenarios for unmanned aerial vehicles. In the first scenario, a single first echelon vehicle is deployed with several drones that are then used to satisfy the demand of all customers. We refer to this model as the *Single Vehicle Drone-Only Delivery* model. In the second scenario, there exist customers that cannot be served by a drone, due to customer order specifications, capacity considerations, or location. We refer to this model as the *Single Vehicle Drone and Truck Delivery* model. Last, scenario 3 is similar to the second one and serves as a generalization as far as the number of first echelon vehicles are concerned. This model is referred to as the *Multiple Vehicle Drone and Truck Delivery* model. Notation that is specific to each of these models is presented in the beginning of each subsection.

4.2.1. Single Vehicle Drone-Only Delivery

In this model a first echelon vehicle like a truck carries a set of drones to each open satellite station, deploy drones from satellite station to serve the customers assigned to that and the truck waits there until all the deployed drones come back, when all the drones are back from customer location the truck receives them and moves to another satellite station to satisfy customers assigned to that satellite station. This is how when all the open satellite stations are covered the truck goes back to the main depot or warehouse.

Let us introduce the sets, parameters, and variables of this model below:

I. Sets:

1. K_2 : set of second echelon drones for last-mile delivery/service.
2. O: depot node
3. C: customer nodes
4. S: satellite stations

II. Parameters:

1. λ_c: service time for customer $c \in C$.
2. d_c: demand for customer $c \in C$.
3. μ_s: setup time for satellite station $s \in S$.
4. V_s: deployment/pickup time for station $s \in S$.
5. T_{ij}= time/distance to traverse $(i, j) \in E_k$ for the first echelon vehicle.
6. $T_{ij}^{k_2}$= time/distance to traverse $(i, j) \in E_k$ for a second echelon drone $k_2 \in K_2$.

III. Routing Variables:

1. $x_{ij}^{ks} = 1$ if vehicle $k \in K_2$ uses arc $(i, j) \in E_k$ having deployed from station $s \in S$.
2. $f_{ij} = 1$ if the first echelon vehicle uses arc $(i, j) \in E_k$.

IV. Assignment Variables:

1. $v_c^s = 1$ if customer $c \in C$ is assigned to satellite station $s \in S$.
2. $y_s = 1$ if the first echelon vehicle sets up satellite station $s \in S$.

V. Time Variables:

1. τ_s= duration of time a first echelon vehicle spent in satellite station $s \in S$.
2. t_s^k= time vehicle $k \in K_2$ spends during a round starting at $s \in S$.
3. ρ= total time required to satisfy all customers.

VI. Other Variables:

1. l_i= subtour elimination variable for second echelon vehicles, where $i \in V$.
2. m_i= subtour elimination variable for first echelon vehicles, where $i \in V$.

VII. Others:

1. C_1= maximum number of nodes the first echelon vehicle can visit in its tour.
2. C_2= maximum number of customers second echelon vehicles can visit in its tour.
3. Δ= Number of drones required to complete the operation

The formulation is the following:

$$(F) \quad minimize \qquad \sum_{i\in V}\sum_{j\in V} T_{ij} * f_{ij} + \sum_{k\in K2}\sum_{s\in S}\sum_{i\in S\cup C}\sum_{j\in S\cup C} T_{ij}^k * x_{ij}^{ks} \tag{4.1}$$

$$\sum_{s\in S} v_c^s = 1 \qquad \forall c \in C \tag{4.2}$$

$$v_c^s \leq y_s \qquad \forall c \in C, \forall s \in S \tag{4.3}$$

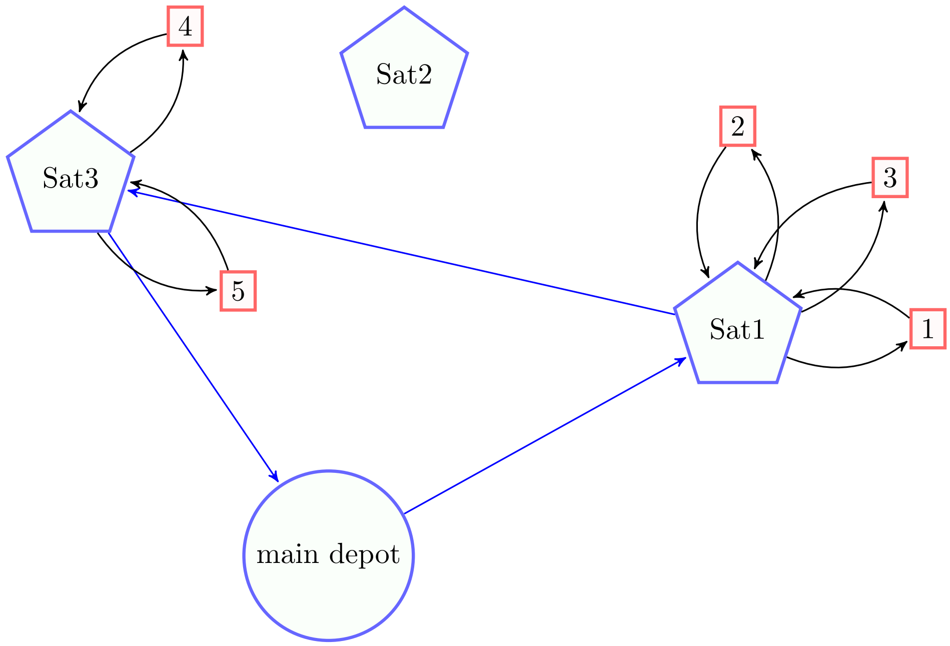

Figure 4.1. A feasible solution of the first model to satisfy 5 customers.

$$\sum_{i \in V} f_{is} = y_s \qquad \forall s \in S \tag{4.4}$$

$$\sum_{s \in S} f_{si} - \sum_{s \in S} f_{is} = 0 \qquad \forall i \in V \tag{4.5}$$

$$m_i - m_j + C_1 * f_{ij} \leq C_1 - 1 \qquad \forall \quad i, j \in S \cup C, i \neq j \tag{4.6}$$

$$\sum_{i \in O} \sum_{j \in S \cup C} f_{ji} = 1 \tag{4.7}$$

$$\sum_{s \in S} \sum_{k \in K_2} \sum_{j \in S, s=j} x_{cj}^{ks} + \sum_{s \in S} \sum_{k \in K_2} \sum_{j \in C, j \neq c} x_{cj}^{ks} = 1 \qquad \forall c \in C \tag{4.8}$$

$$\sum_{s \in S} \sum_{k \in K_2} \sum_{j \in S, s \neq j} x_{cj}^{ks} + \sum_{s \in S} \sum_{k \in K_2} \sum_{j \in S, s \neq j} x_{jc}^{ks} = 0 \qquad \forall c \in C \tag{4.9}$$

$$\sum_{j \in C} x_{ij}^{ks} - \sum_{j \in C} x_{ji}^{ks} = 0 \qquad \forall i \in S \cup C, \forall k \in K_2, \forall s \in S \tag{4.10}$$

$$l_i - l_j + C_2 * \sum_{k \in K_2} \sum_{s \in S} x_{ij}^{ks} \leq C_2 - 1 \qquad \forall \quad i, j \in C, i \neq j \tag{4.11}$$

$$\sum_{p \in S} \sum_{j \in S \cup C} x_{cj}^{kp} + \sum_{p \in S} \sum_{j \in S \cup Z} x_{sj}^{kp} - v_c^s \leq 1 \qquad \forall c \in C, \forall k \in K_2, \forall s \in S \tag{4.12}$$

$$t_s^k = \sum_{i \in S \cup C} \sum_{j \in S \cup C} x_{ij}^{ks} * T_{ij}^k + \sum_{i \in S \cup C} \sum_{j \in C} x_{ij}^{ks} * \lambda_c \qquad \forall s \in S, \forall k \in K_2 \tag{4.13}$$

$$\tau_s \geq t_s^k + (\mu_s + V_s) * y_s \qquad \forall s \in S, \forall k \in K_2 \tag{4.14}$$

$$\rho = \sum_{i \in V} \sum_{j \in V} T_{ij} * f_{ij} + \sum_{s \in S} \tau_s \tag{4.15}$$

$$\Delta \geq \sum_{c \in C} v_c^s \qquad \forall s \in S \tag{4.16}$$

The objective function (4.1) tries to minimize the global routing cost of the first echelon vehicle, and drones. Constraint (4.2) ensures that every customer is assigned to a satellite station. Constraint (4.3) makes sure that if a customer is assigned to a satellite station then that station must be open. Each open satellite station must be visited by the first echelon vehicle is confirmed by constraint (4.4). Flow preservation of the first echelon vehicle is confirmed by the the constraint (4.5). Constraint (4.6) is the subtour elimination constraint of the first echelon vehicle. Constraint (4.7) ensures the first echelon vehicle comes back to the depot at the end of its tour. Constraint (4.8) makes sure every customer is satisfied by a drone. Constraint (4.9) ensures that if a drone is launched from a satellite station then it would come back to the same station after visiting customer location(s). Constraint (4.10) is the flow preservation constraint of second echelon drones. Constraint (4.11) is the subtour elimination constraint for drones. Constraint (4.12) links between allocation and routing variables. Constraint (4.13) calculates the time each drone take in its tour from a satellite station. Constraint (4.14) calculates the waiting time of the first echelon vehicle to an open satellite station. Constraint (4.15) calculates the duration of operation in satisfying the customers. Constraint (4.16) calculates the number of drones required to run the operation. Right side of the constraint (4.15) can be used as an objective function if one wants to minimize the duration of operation to satisfy all customers but this does not guarantee that global routing cost would be optimal.

4.2.2. Single Vehicle Drone and Truck Delivery

Like first model in this model a first echelon vehicle like a truck carry a set of drones to each open satellite station, deploy drones from satellite station to serve the customers assigned to that and the truck waits there until all the deployed drones come back, when all the drones are back from customer location the truck receives them and moves to another satellite station to satisfy customers assigned to that satellite station. In addition to that the first echelon vehicle visits customer locations to satisfy them in order to optimize the global routing cost. And constraint like certain customers have to be satisfied by the first echelon vehicle is also considered in this model.

In addition to the sets, parameters, and variables of the first model, this model needs following parameters, and variables:

I. Parameters:

1. b_c= 1 if customer $c \in C$ needs to be served by a first echelon vehicle.
2. E= unit distance/time operating cost of the first echelon vehicle.
3. G= unit distance/time operating cost of the second echelon drones.

II. Variables:

1. u_c=1 if the first echelon vehicle visits the customer $c \in C$.

The formulation is the following:

$$(S) \quad minimize \qquad E * \sum_{i \in V} \sum_{j \in V} T_{ij} * f_{ij} + G * \sum_{k \in K2} \sum_{s \in S} \sum_{i \in S \cup C} \sum_{j \in S \cup C} T_{ij}^{k} * x_{ij}^{ks} \tag{4.17}$$

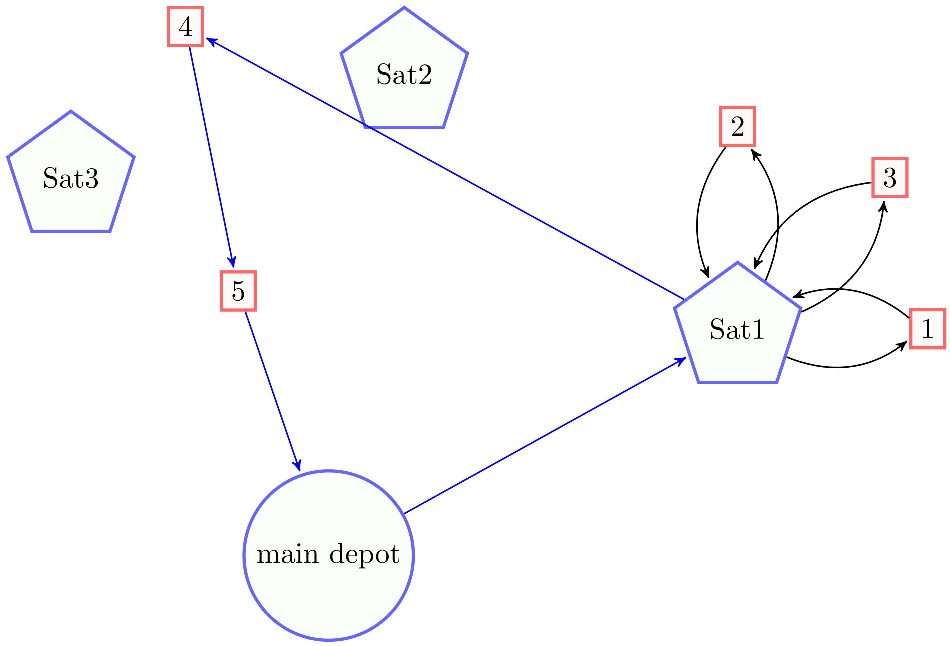

Figure 4.2. A feasible solution of the second model to satisfy 5 customers where 4th customer is initially assigned to be satisfied by the first echelon vehicle.

Subject to constraints (4.3)-(4.7), (4.9)-(4.16), and following new constraints

$$\sum_{s \in S} v_c^s + u_c = 1 \qquad \forall c \in C \tag{4.18}$$

$$u_c \geq b_c \qquad \forall c \in C \tag{4.19}$$

$$\sum_{i \in V} f_{ic} = u_c \qquad \forall c \in C \tag{4.20}$$

$$\sum_{s \in S} \sum_{k \in K_2} \sum_{j \in S, s=j} x_{cj}^{ks} + \sum_{s \in S} \sum_{k \in K_2} \sum_{j \in C, j \neq c} x_{cj}^{ks} = \sum_{s \in S} v_c^s \qquad \forall c \in C \tag{4.21}$$

The objective function (4.17) seeks to minimize the global routing cost in presence of two unit parameters. These two unit parameters represent the unit cost of operating the first echelon vehicle and drones. Unit cost of these two parameters vary based on geographic attributes, risk of lives in war zone, risk of being hijacked, road conditions, price and availability of fuel and battery, etc. If the unit cost of these two parameters are not significantly different then the solution ends up giving a single vehicle routing solution. Constraint (4.18) makes sure each customer is assigned to either a satellite station or the first echelon vehicle. Constraint (4.19) ensures that a customer can be satisfied by the first echelon vehicle even if it is not assigned to the first echelon vehicle. Constraint (4.20) ensures that the first echelon vehicle visits a customer if it is satisfied by the first echelon vehicle. And constraint (4.21) makes sure that a drone satisfies a customer if the customer is assigned to a satellite station.

4.2.3. Multiple Vehicle Drone and Truck Delivery

This is the generalization of the previous two models and considers more than one first echelon vehicles. In this model a number of first echelon vehicle equipped with drones come out from the depot to satisfy customers by either first echelon vehicles or drones launched from a satellite station.

In order to cope with the generalization of this model some sets, parameters, and variables of the previous two models are modified besides using the others unchanged. The modified or new sets, parameters, and variables are given below:

I. Sets:

1. K_1: set of first echelon vehicles.

II. Parameters:

1. T_{ij}^k: time/distance to traverse $(i,j) \in E_k$ for the first echelon vehicle $k \in K_1$.

III. Variables:

1. f_{ij}^k=1 if first echelon vehicle $k \in K_1$ visits arc $(i,j) \in E_k$.
2. y_s^k=1 if satellite station $s \in S$ is set up by a first echelon vehicle $k \in K_1$.
3. u_c^k=1 if customer $c \in C$ is satisfied by a first echelon vehicle $k \in K_1$.

The formulation is the following:

$$(T) \quad minimize \qquad E * \sum_{k \in K_1} \sum_{i \in V} \sum_{j \in V} T_{ij}^k * f_{ij}^k + G * \sum_{k \in K2} \sum_{s \in S} \sum_{i \in S \cup C} \sum_{j \in S \cup C} T_{ij}^k * x_{ij}^{ks} \tag{4.22}$$

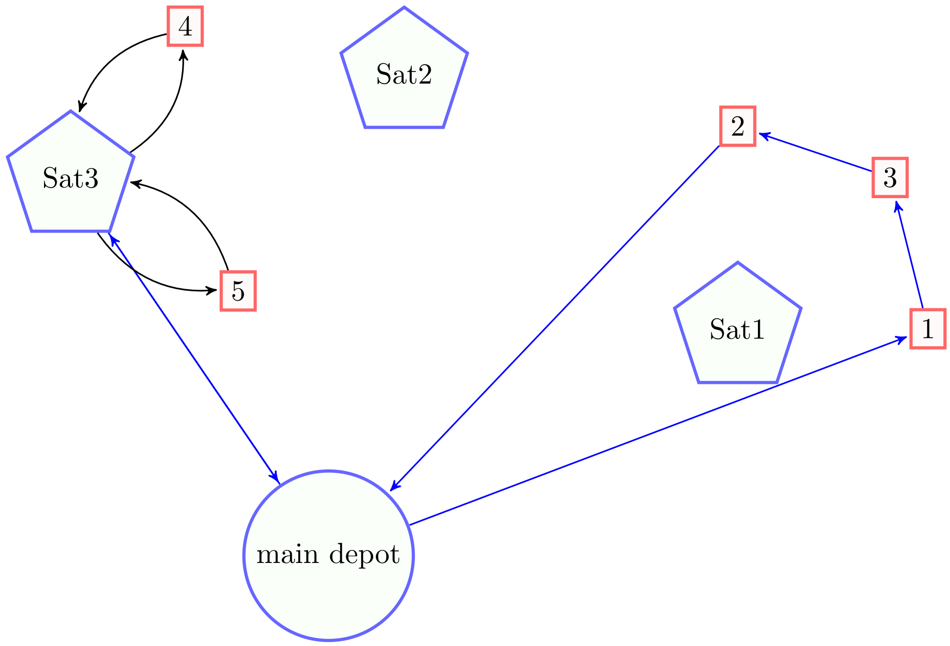

Figure 4.3. A feasible solution of the third model to satisfy 5 customers where 1st customer is initially assigned to be satisfied by the first echelon vehicle.

Subject to constraints (4.9)-(4.13), (4.21), and following new constraints

$$\sum_{s\in S} v_c^s + \sum_{k\in K_1} u_c^k = 1 \qquad \forall c \in C \tag{4.23}$$

$$\sum_{k\in K_1} u_c^k \geq b_c \qquad \forall c \in C \tag{4.24}$$

$$v_c^s \leq \sum_{k\in K_1} y_s^k \qquad \forall c \in C, \forall s \in S \tag{4.25}$$

$$\sum_{s\in S} f_{si}^k - \sum_{s\in S} f_{is}^k = 0 \qquad \forall i \in V, \forall k \in K_1 \tag{4.26}$$

$$\sum_{i\in V} f_{is}^k = y_s^k \qquad \forall s \in S, \forall k \in K_1 \tag{4.27}$$

$$m_i - m_j + C_1 * \sum_{k\in K_1} f_{ij}^k \leq C_1 - 1 \qquad \forall \quad i,j \in S \cup C, i \neq j \tag{4.28}$$

$$\tau_s \geq t_s^k + (\mu_s + V_s) * \sum_{k \in K1} y_s^k \qquad \forall s \in S, \forall k \in K_2 \tag{4.29}$$

$$\sum_{i \in O} \sum_{j \in S \cup C} f_{ji}^k = 1 \qquad \forall k \in K_1 \tag{4.30}$$

$$\sum_{i \in O \cup S} f_{ij}^k = u_j^k \qquad \forall k \in K_1, \forall j \in C \tag{4.31}$$

The objective function (4.22) seeks to minimize the global routing cost and waiting time of first echelon vehicles in satellite stations in presence of unit distance/time operating cost parameter of first echelon vehicles and drones. Constraint (4.23) ensures that each customer is satisfied by either first echelon vehicle or drone from satellite station. Constraint (4.24) indicates that a customer can be satisfied by a first echelon vehicle even if initially the customer is not assigned to satisfy by any first echelon vehicle. Constraint (4.25) indicates that a customer can be assigned to a satellite station if that station is set up by a first echelon vehicle. Constraint (4.26) is the flow preservation constraint of first echelon vehicles. Constraint (4.27) indicates every open satellite station must be visited by a first echelon vehicle. Constraint (4.28) is the subtour elimination constraint of first echelon vehicles. Constraint (4.29) calculates the waiting time of a first echelon vehicle in a satellite station. Constraint (4.30) indicates every first echelon vehicle goes back to the depot at the end of its tour. And constraint (4.31) ensures that a customer is satisfied by a first echelon vehicle if the vehicle visits the customer location.

5. COMPUTATIONAL EXPERIMENTS

We perform two distinct sets of experiments on the three models presented in the previous chapter. The numerical experiments are described based upon the distribution of customers (demand points) on the map. We opted to use the metropolitan area of Fargo-Moorhead for our experiments. In the first set of experiments, the locations of all customers are *random*, while in the second set of experiments, the locations are *bordering*. In all experiments, the locations of the main depot and the potential satellite stations to be used remain unchanged. The geographical detalis for their locations are presented in Table 5.1. For the set of customers, those are randomly generated from iteration to iteration. However, for pictorial purposes, we use the set of location as shown in Table 5.2 as an example for the numerical differences between each model.

5.1. Random Customer Distribution

In this set of experiments customers are randomly located in Fargo city. Using the random customer location generator, a java project which is given in appendix C, 10 customer locations are generated. The latitude and longitude of these customer locations are given in table 5.2. Running the java projects in appendix A, and appendix B, distance matrices of road distance and aerial distance among the depot, satellite stations, and customer locations are created. The geographic locations of each layer elements are shown in Figure 5.1.

5.1.1. Single Vehicle Drone-Only Delivery with Random Customer Locations

A two-echelon vehicle routing solution is obtained from running this model for random customers. The first echelon vehicle containing drones visits each open satellite and deploy drones to satisfy customers assigned to that satellite, and when drones are back the vehicle moves to another satellites untill all open satellites are covered, finally the vehicle goes back to the main depot.

Table 5.1. Geographical details of the depot and potential satellite stations.

Number	Type of location	Location Name	Latitude	Longitude
1	Depot	4731 13th ave s (Walmart)	46.864038	-96.865825
2	Satellite Station (sat1)	22 25th street s	46.876282	-96.819364
3	Satellite Station (sat2)	1020 19th ave n	46.904747	-96.793564
4	Satellite Station (sat3)	2520 40th ave s	46.818025	-96.819371
5	Satellite Station (sat4)	4014 45th street s	46.818074	-96.861859

Table 5.2. Random customer locations that are used for model results presentation.

Number	Customer Name	Latitude	Longitude
1	C1 (6)	46.930075225291	-96.8140615671068
2	C2 (7)	46.881646330250184	-96.80983027814386
3	C3 (8)	46.894012618262344	-96.77169210011992
4	C4 (9)	46.90943311397734	-96.81660980517788
5	C5 (10)	46.842044909863475	-96.79064743397912
6	C6 (11)	46.88610605120827	-96.81221359022248
7	C7 (12)	46.79526740327874	-96.81290639051734
8	C8 (13)	46.90817944537533	-96.82733315898396
9	C9 (14)	46.878470088135586	-96.77310811803093
10	C10 (15)	46.93230515051098	-96.77883258093172

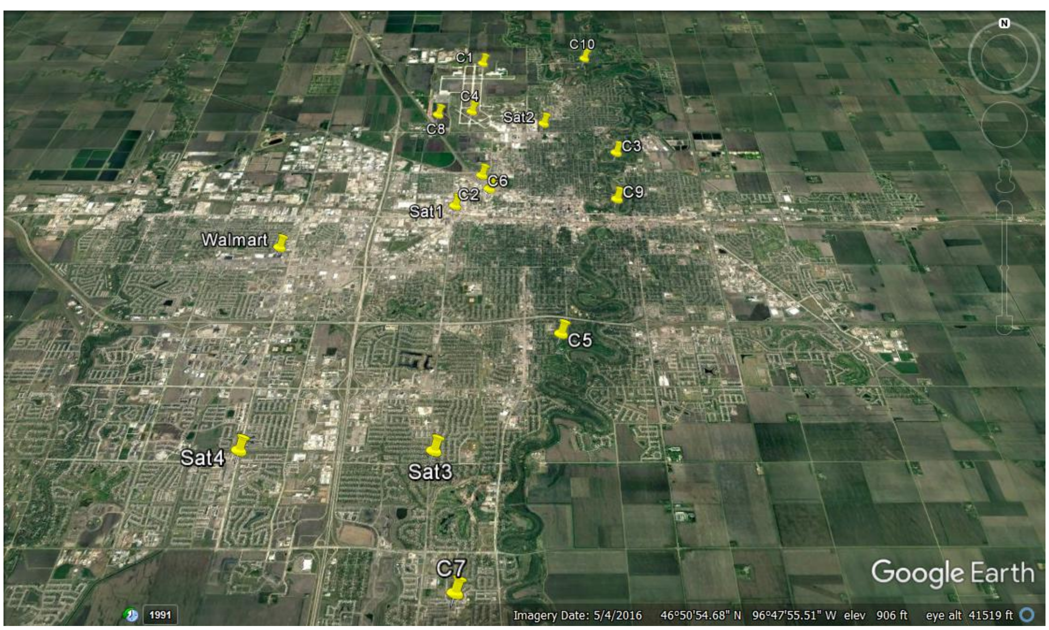

Figure 5.1. Geographic locations of the depot, satellites, and customers

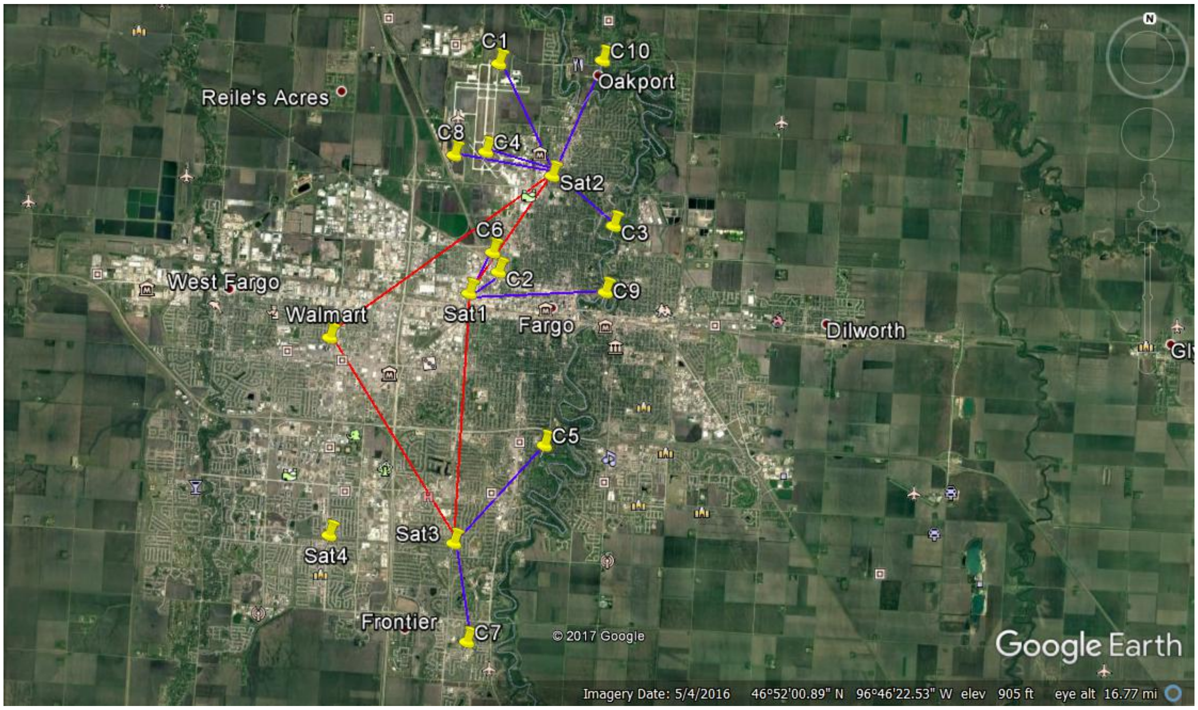

Figure 5.2. Solution from the fist model for randomly generated customers

The parameters of this model are: the first echelon vehicle can visit utmost 4 satellites, a drone can satisfy only one customer in its each tour, service time for each customer is two minutes, set up time for each satellite station is two minutes, pick up time of a drone in a satellite station is two minutes, the upper limit of the drone variable is 10.

The solution gives: 10 customers are satisfied by drones where five drones are launched from satellite sat2, three drones are launched from satellite sat1, and two drones are launched from satellite sat3; the first echelon vehicle first visits sat2, then sat1, finally sat3 before coming back to the depot; sat2 satisfies customer C1, C3, C4, C8, and C10; sat1 satisfies customer C2, C6, and C9; sat3 satisfies customer C5 and C7. The model is developed in GAMS platform, and it is solved using CPLEX solver from neos-server. The routing solution of this model is shown in Figure 5.2.

5.1.2. Single Vehicle Drone and Truck Delivery with Random Customer Locations

The second model has two unit operating cost parameters and to see how the solution varies with the change of these two parameters we considered three scenarios in this experiment. The scenarios are shown in Table 5.3.

Table 5.3. Unit operating cost combinations for different scenarios.

Scenario Number	Vehicle Operating Cost (E)	Drone Operating Cost (G)
1	1	1
2	2	1
3	3	1

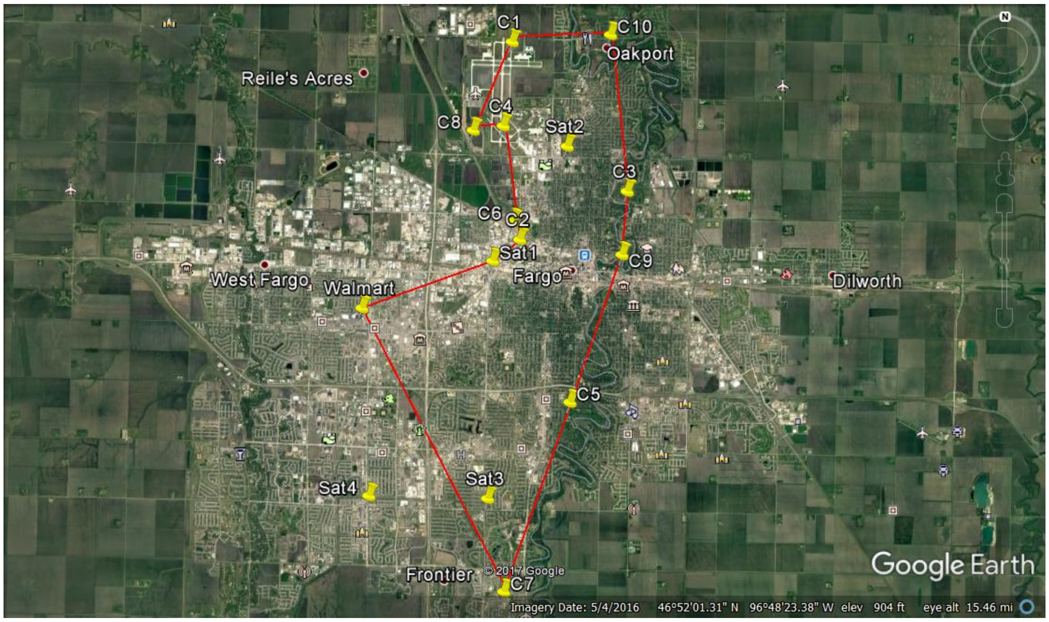

Figure 5.3. Solution of the second model for randomly generated customers when E=1, and G=1

5.1.2.1. Scenario 1 for Random Customers

A single vehicle routing solution is obtained from running this model for random customers. The parameters of this model are: unit operating cost of the vehicle (E)=1 and unit operating cost of each drone (G) =1,the first echelon vehicle can visit utmost 14 locations, a drone can satisfy only one customer in its each tour, service time for each customer is two minutes, set up time for each satellite station is two minutes, pick up time of a drone in a satellite station is two minutes, the upper limit of the drone variable is 10, and no customers are assigned initially to be satisfied by the first echelon vehicle.

The solution is all 10 customers are satisfied by the first echelon vehicle. The vehicle first leaves the depot for the sat1, then visits C2, C6, C4, C8, C1, C10, C3, C9, C5, and C7 one after another in a row before coming back to the depot. No drone is deployed in this solution. The model is developed in GAMS platform, and it is solved using CPLEX solver from neos-server. The routing solution of this model is shown in Figure 5.3.

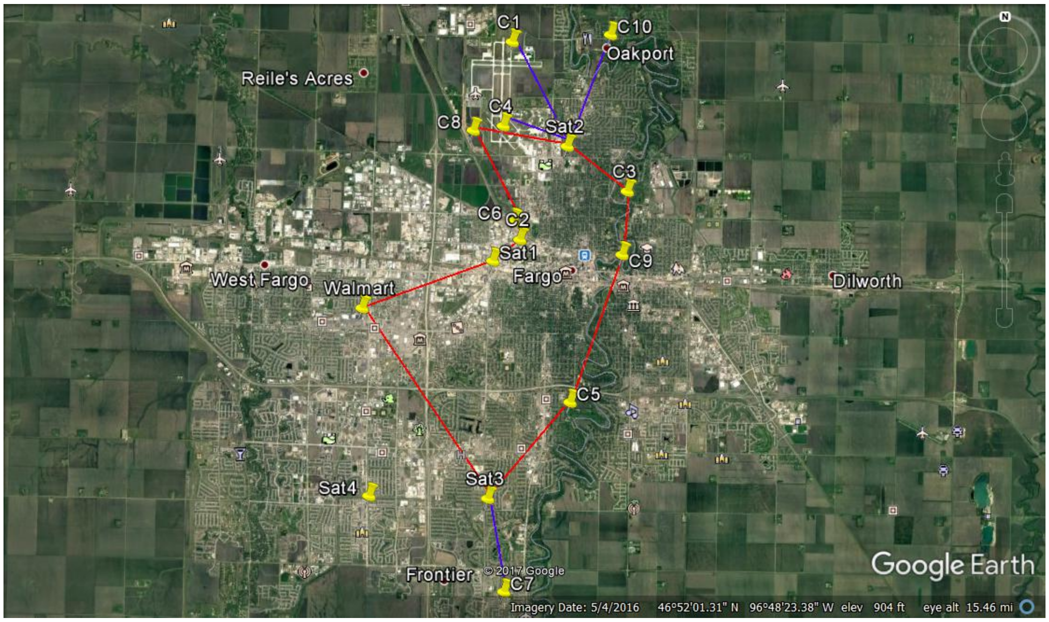

Figure 5.4. Solution of the second model for randomly generated customers when E=2, and G=1

5.1.2.2. Scenario 2 for Random Customers

A combination of first echelon vehicle and drone routing solution is obtained from running this model for random customers. The parameters of this model are: unit operating cost of the vehicle (E)=2 and unit operating cost of each drone (G) =1,the first echelon vehicle can visit utmost 14 locations, a drone can satisfy only one customer in its each tour, service time for each customer is two minutes, set up time for each satellite station is two minutes, pick up time of a drone in a satellite station is two minutess, the upper limit of the drone variable is 10, and no customers are assigned initially to be satisfied by the first echelon vehicle.

The solution is: the first echelon vehicle visits sat1, C2, C6, C8, sat2, C3, C9, C5, and sat3 one after another before coming back to the depot; three drones are launched form sat2 to satisfy C1, C4, and C10 and one drone is launched form sat3 to satisfy C7. The solution of this experiment is shown in Figure 5.4.

5.1.2.3. Scenario 3 for Random Customers

A second echelon drone routing solution is obtained from running this model while the first echelon vehicle carries drones to satellites. The parameters of this model are: unit operating cost of the vehicle (E)=3 and unit operating cost of each drone (G) =1,the first echelon vehicle can visit utmost 14 locations, a drone can satisfy only one customer in its each tour, service time for

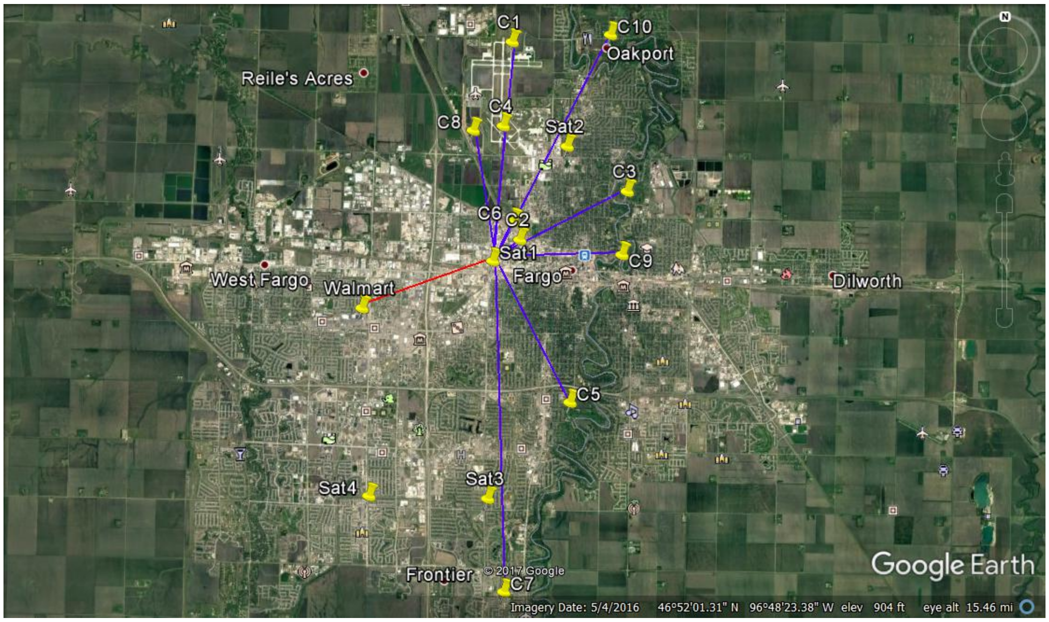

Figure 5.5. Solution of the second model for randomly generated customers when E=3, and G=1

each customer is two minutes, set up time for each satellite station is two minutes, pick up time of a drone in a satellite station is two minutes, the upper limit of the drone variable is 10, and no customers are assigned initially to be satisfied by the first echelon vehicle.

The solution is: the first echelon vehicle equipped with drones visits sat1 and deploys all 10 drones to satisfy C1, C2, C3, C4, C5, C6, C7, C8, C9, and C10. When all drones are back the vehicle comes back to the depot. The solution of this experiment is shown in Figure 5.5.

5.1.3. Multiple Vehicle Drone and Truck Delivery with Random Customer Locations

The specialty of this model is it allows more than one first echelon vehicles. We have considered three first echelon vehicles here, and similarly to the second model three scenarios are experimented with this model to see the routing solutions.

5.1.3.1. Scenario 1 for Random Customers

Like the previous scenario 1, a single vehicle routing solution is obtained from running this model but other two first echelon vehicles are routed to a satellite station for no purpose. The parameters of this model are: unit operating cost of the vehicle (E)=1 and unit operating cost of each drone (G) =1, a first echelon vehicle can visit utmost 14 locations, a drone can satisfy only one customer in its each tour, service time for each customer is two minutes, set up time for each satellite station is two minutes, pick up time of a drone in a satellite station is two minutes, the

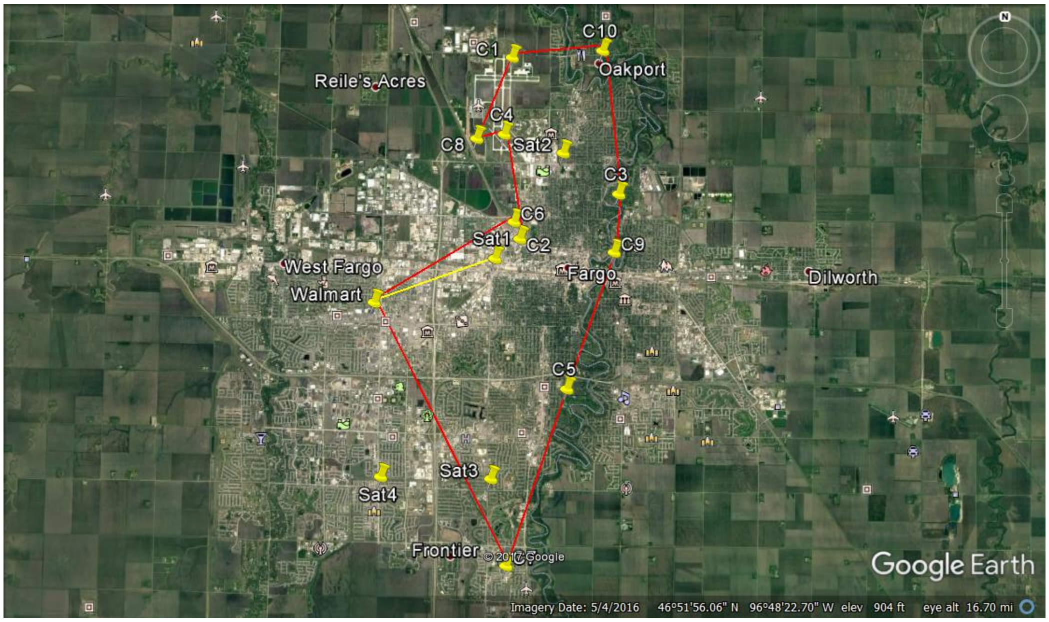

Figure 5.6. Solution of the third model for randomly generated customers when E=1, G=1, and three first echelon vehicles

upper limit of the drone variable is 10, and no customers are assigned initially to be satisfied by the first echelon vehicle.

The solution is: a first echelon vehicle visits C6, C2, C4, C8,C1, C10, C3, C9, C5, and C7 one after another before coming back to the depot; two other first echelon vehicle visits sat1 and come back to the depot for no purpose. No drone is deployed in this solution. The model is developed in GAMS platform, and it is solved using CPLEX solver from neos-server. The routing solution of this model is shown in Figure 5.6.

5.1.3.2. Scenario 2 for Random Customers

A combination of first echelon vehicles and drone routing solution is obtained from running this model for random customers. The parameters of this model are: unit operating cost of each first echelon vehicle (E)=2 and unit operating cost of each drone (G) =1,the first echelon vehicle can visit utmost 14 locations, a drone can satisfy only one customer in its each tour, service time for each customer is two minutes, set up time for each satellite station is two minutes, pick up time of a drone in a satellite station is two minutes, the upper limit of the drone variable is 10, and no customers are assigned initially to be satisfied by the first echelon vehicle.

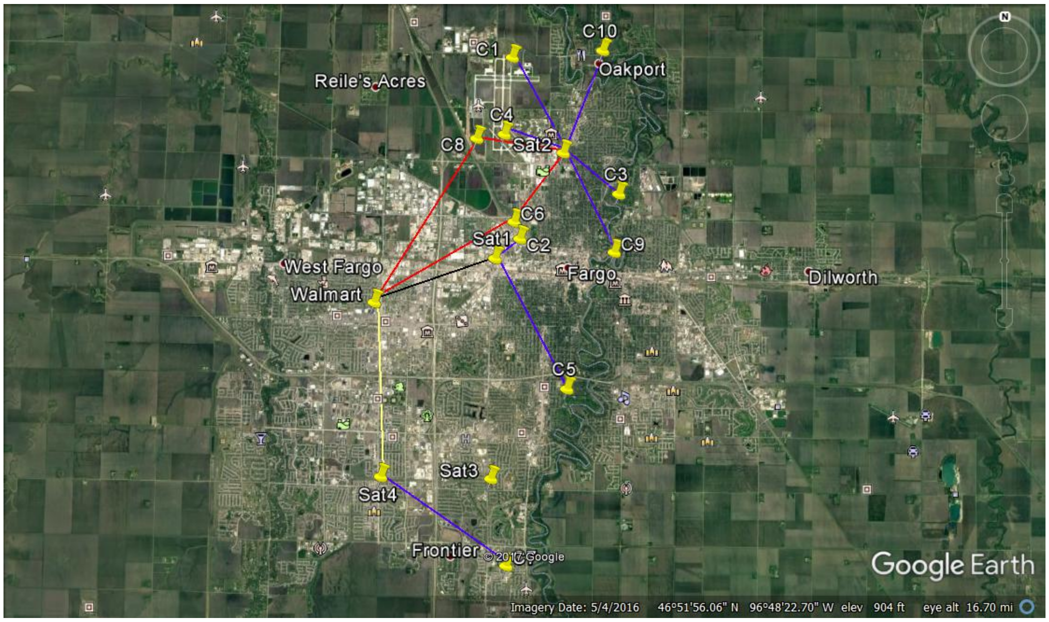

Figure 5.7. Solution of the third model for randomly generated customers when E=2, G=1, and three first echelon vehicles

The solution is: one first echelon vehicle first visits C8, then sat2 and deploys five drones to satisfy C1, C3, C4, C9, and C10 and waits for drones coming back, then visits C6 before coming back to the depot; another first echelon vehicle visits sat1 and deploys two drones to satisfy C2, and C5 and when drones are back it moves to the depot; the other first echelon vehicle moves to sat4 and deploys a drone to satisfy C7, when the drone comes back the vehicle moves to the depot. The solution of this experiment is shown in Figure 5.7.

5.1.3.3. Scenario 3 for Random Customers

A second echelon drone routing solution is obtained from running this model. The parameters of this model are: unit operating cost of the vehicle (E)=3 and unit operating cost of each drone (G) =1,the first echelon vehicle can visit utmost 14 locations, a drone can satisfy only one customer in its each tour, service time for each customer is two minutes, set up time for each satellite station is two minutes, pick up time of a drone in a satellite station is two minutes, the upper limit of the drone variable is 10, and no customers are assigned initially to be satisfied by the first echelon vehicle.

The solution is one first echelon vehicle equipped with drones visits sat1 and deploys all 10 drones to satisfy C1, C2, C3, C4, C5, C6, C7, C8, C9, and C10 and when all drones are back the

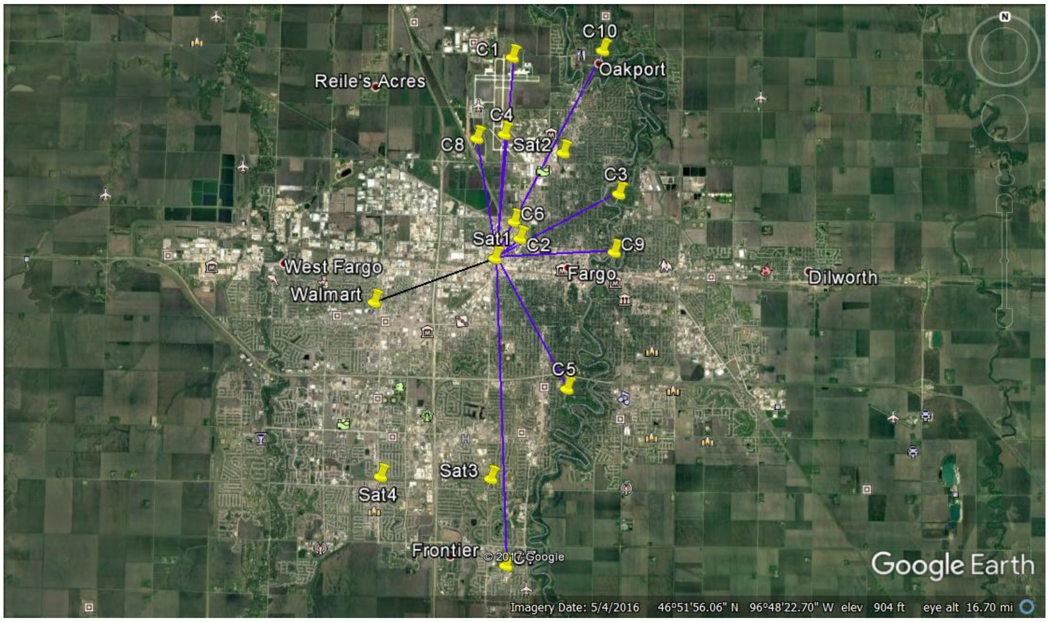

Figure 5.8. Solution of the third model for randomly generated customers when E=3, G=1, and three first echelon vehicles

vehicle comes back to the depot, but other two first echelon vehicle visits sat1 for no reason and come back to the depot. The solution of this experiment is shown in Figure 5.8.

5.2. Bordering Customer Distribution

In this set up customers are distributed to the border region of Fargo City. The set of bordering customers are generated arbitrarily and their locations are shown in Table 5.4. For ease of presentation, the geographic locations of each layer of this distribution network on a map are shown in Figure 5.9.

Table 5.4. Bordering customer locations.

Number	Customer Name	Latitude	Longitude
1	C1 (6)	46.915580	-96.776561
2	C2 (7)	46.906722	-96.777582
3	C3 (8)	46.880026	-96.790176
4	C4 (9)	46.873380	-96.787027
5	C5 (10)	46.870060	-96.790245
6	C6 (11)	46.835552	-96.803165
7	C7 (12)	46.928149	-96.838797
8	C8 (13)	46.922293	-96.796837
9	C9 (14)	46.814643	-96.818757
10	C10 (15)	46.806209	-96.845361

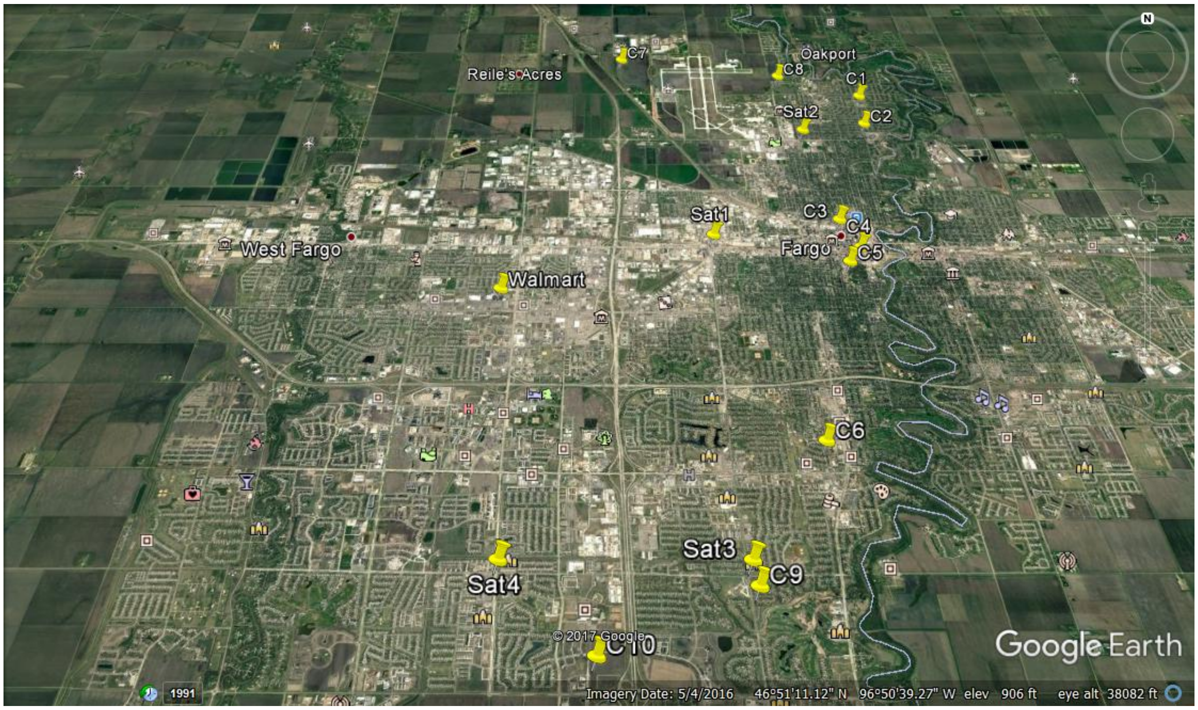

Figure 5.9. Geographic locations of depot, satellites, and bordering customers

5.2.1. Single Vehicle Drone-Only Delivery with Bordering Customer Locations

Like randomly distributed customers for the first model of bordering customers a two-echelon vehicle routing solution is obtained. The first echelon vehicle containing drones visits each open satellite and deploy drones to satisfy customers assigned to that satellite, and when drones are back the vehicle moves to another satellites until all open satellites are covered, finally the vehicle goes back to the main depot.

The parameters of this model are: the first echelon vehicle can visit utmost 4 satellites, a drone can satisfy only one customer in its each tour, service time for each customer is two minutes, set up time for each satellite station is two minutes, pick up time of a drone in a satellite station is two minutes, the upper limit of the drone variable is 10.

The solution gives: 10 customers are satisfied by drones where four drones are launched from satellite sat2, three drones are launched from satellite sat1, two drones are launched from satellite sat3, and one drone is launched from sat4; the first echelon vehicle first visits sat2, then sat1, sat3, and sat4 in a row before coming back to the depot; sat2 satisfies customer C1, C2, C7, and C8; sat1 satisfies customer C3, C4, and C5; sat3 satisfies customer C6 and C9; sat4 satisfies

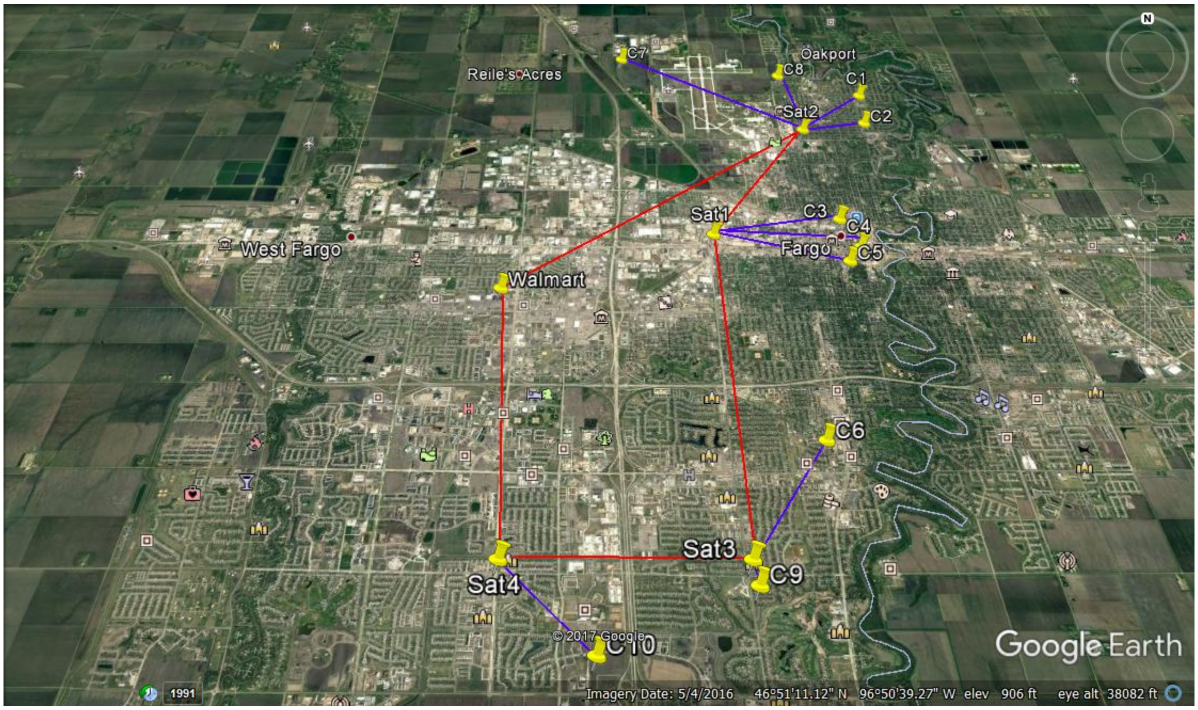

Figure 5.10. Solution of the first model for bordering customers

customer C10. The model is developed in GAMS platform, and it is solved using CPLEX solver from neos-server. The routing solution of this model is shown in Figure 5.10.

5.2.2. Single Vehicle Drone and Truck Delivery with Bordering Customer Locations

Like the second model with randomly distributed customers with this model three scenarios are experimented to see the routing results.

5.2.2.1. Scenario 1 for Bordering Customer Locations

A combination first echelon vehicle and second echelon drone routing solution is obtained from running this model for bordering customers. The parameters of this model are: unit operating cost of the vehicle (E)=1 and unit operating cost of each drone (G) =1,the first echelon vehicle can visit utmost 14 locations, a drone can satisfy only one customer in its each tour, service time for each customer is two minutes, set up time for each satellite station is two minutes, pick up time of a drone in a satellite station is two minutes, the upper limit of the drone variable is 10, and no customers are assigned initially to be satisfied by the first echelon vehicle.

The solution is 9 customers are satisfied by the first echelon vehicle and the other one by a drone. The vehicle first leaves the depot for C7, then visits C8, C1, C2, C3, C4, C5, C6, sat3, and C10, C5, and C7 one after another in a row before coming back to the depot. At sat3 the vehicle

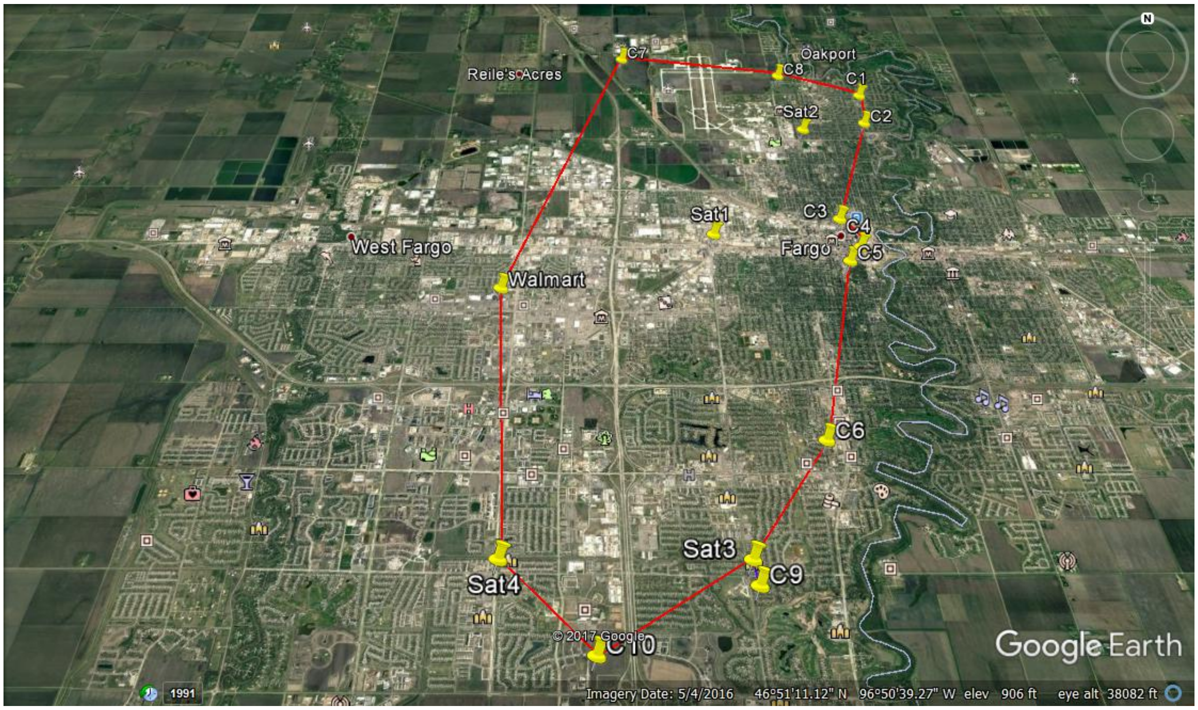

Figure 5.11. Solution of the second model for bordering customers when E=1, and G=1

launches a drone to satisfy C9 and wait for the drone before moving to C10. The routing solution of this model is shown in Figure 5.11.

5.2.2.2. Scenario 2 for Bordering Customer Locations

A combination of the first echelon vehicle and drone routing solution is obtained from running this model for bordering customers. The parameters of this model are: unit operating cost of the vehicle (E)=2 and unit operating cost of each drone (G) =1,the first echelon vehicle can visit utmost 14 locations, a drone can satisfy only one customer in its each tour, service time for each customer is two minutes, set up time for each satellite station is two minutes, pick up time of a drone in a satellite station is two minutes, the upper limit of the drone variable is 10, and no customers are assigned initially to be satisfied by the first echelon vehicle.

The solution is: the first echelon vehicle visits sat4, sat3, C6, C5, C4, C3, and sat2 one after another in a row before coming back to the depot; one drone is launched form sat4 to satisfy C10, ine drone is launched form sat3 to satisfy C9, and four drones are launched from sat2 to satisfy C1, C2, C7, and C8. The solution of this experiment is shown in Figure 5.12.

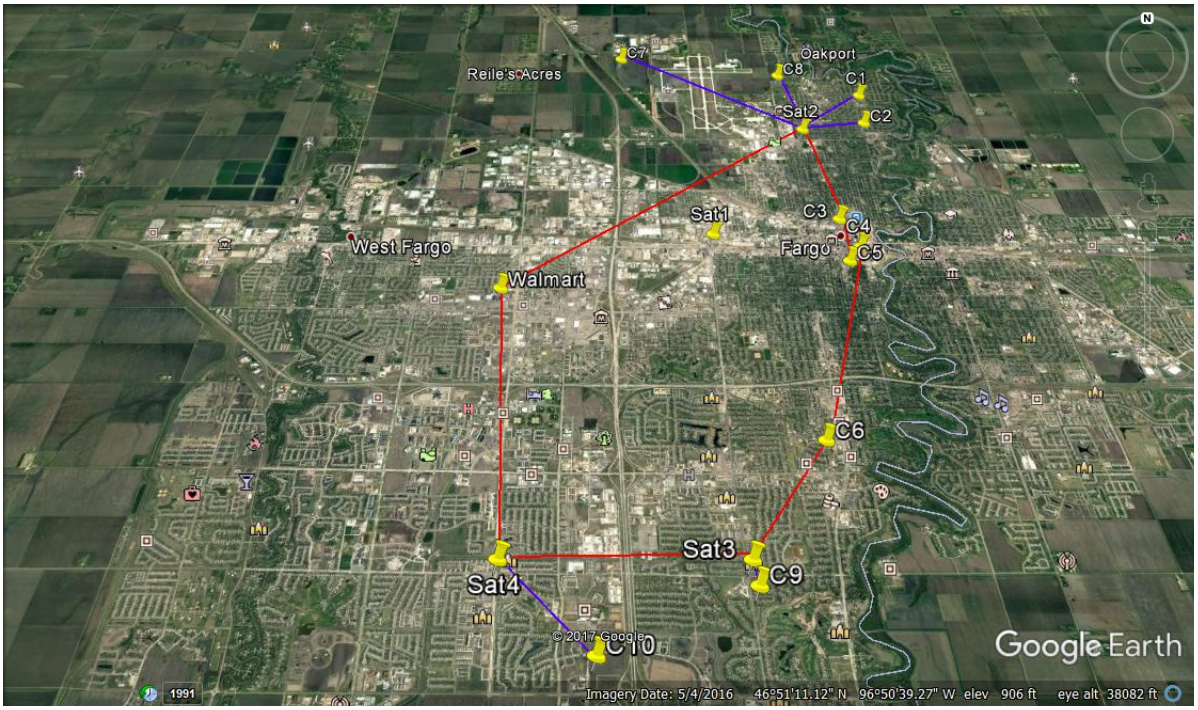

Figure 5.12. Solution of the second model for bordering customers when E=2, and G=1

5.2.2.3. Scenario 3 for Bordering Customer Locations

A combination of first echelon vehicle and second echelon drone routing solution to satisfy customer demand is obtained from running this model. The parameters of this model are: unit operating cost of the vehicle (E)=3 and unit operating cost of each drone (G) =1,the first echelon vehicle can visit utmost 14 locations, a drone can satisfy only one customer in its each tour, service time for each customer is two minutes, set up time for each satellite station is two minutes, pick up time of a drone in a satellite station is two minutes, the upper limit of the drone variable is 10, and no customers are assigned initially to be satisfied by the first echelon vehicle.

The solution is: the first echelon vehicle equipped with drones visits sat3 and deploys two drones to satisfy C9 and C10, when drones are back then the vehicle moves to C6, C5, C4, and C3 one after another in a row to satisfy their demand, then the vehicle moves to sat2 and deploys four drones to satisfy C7, C8, C1, and C2 and waits, when drones are back the vehicle moves to the depot. The solution of this experiment is shown in Figure 5.13.

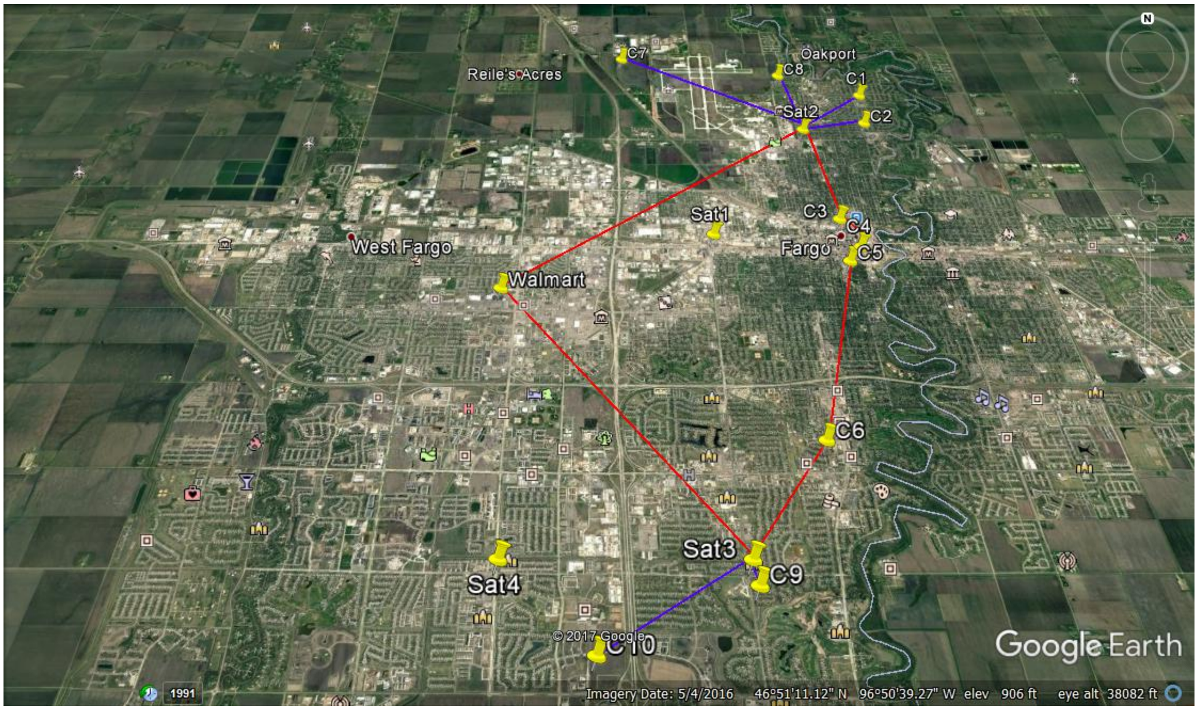

Figure 5.13. Solution of the second model for bordering customers when E=3, and G=1

5.2.3. Multiple Vehicle Drone and Truck Delivery with Bordering Customer Locations

In this experiments three scenarios are considered as mentioned above and three first echelon vehicles are used.

5.2.3.1. Scenario 1 for Bordering Customer Locations

In this experiment a combination of first echelon vehicles and second echelon drone routing solution is achieved to satisfy customers. The parameters of this model are: unit operating cost of the vehicle (E)=1 and unit operating cost of each drone (G) =1, a first echelon vehicle can visit utmost 14 locations, a drone can satisfy only one customer in its each tour, service time for each customer is two minutes, set up time for each satellite station is two minutes, pick up time of a drone in a satellite station is two minutes, the upper limit of the drone variable is 10, and no customers are assigned initially to be satisfied by the first echelon vehicle.

The solution is: one first echelon vehicle visits sat1, C4, C5, C3, C2,C1, C8, and C7 in a row before coming back to the depot; another first echelon vehicle visits sat4, C10, and sat3 in a row and at sat3 launches a drone to satisfy C9, when the drone is back the vehicle visits C6 before

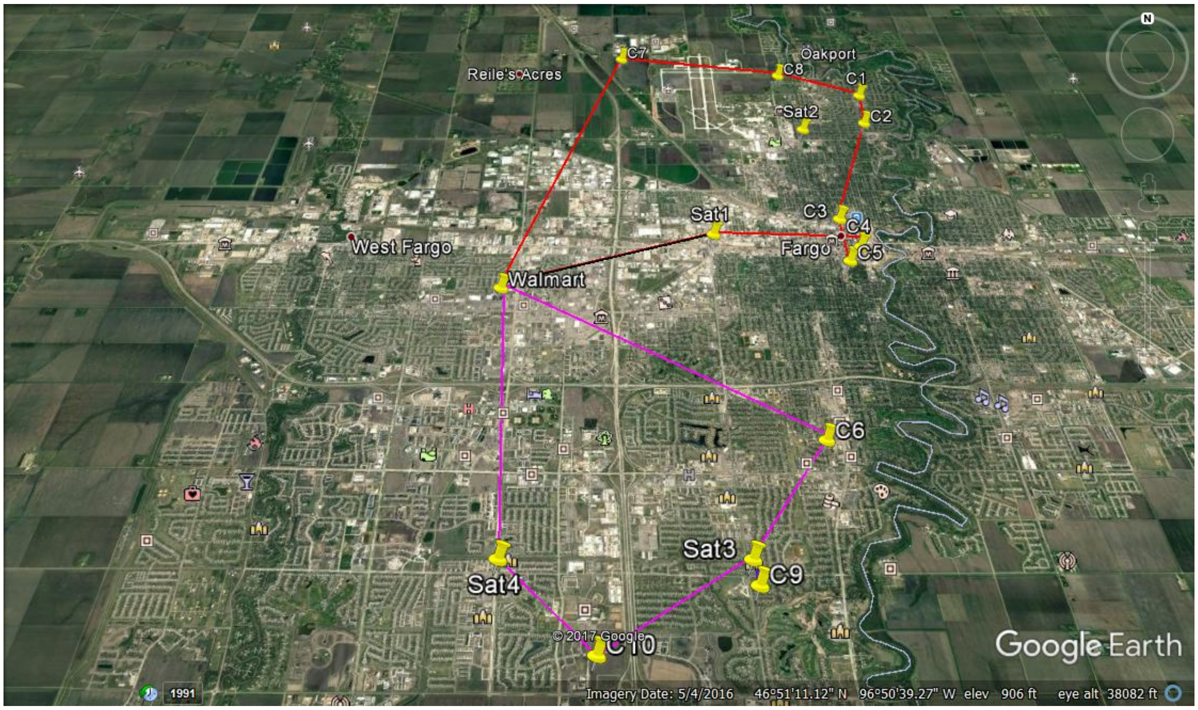

Figure 5.14. Solution of the third model for bordering customers when E=1, and G=1

going back to the depot. The other first echelon vehicle visits sat1 and comes back to the depot for no reason. The routing solution of this model is shown in Figure 5.14.

5.2.3.2. Scenario 2 for Bordering Customer Locations

A combination of first echelon vehicles and second echelon drone routing solution is obtained from running this model for bordering customers. The parameters of this model are: unit operating cost of each first echelon vehicle (E)=2 and unit operating cost of each drone (G) =1,the first echelon vehicle can visit utmost 14 locations, a drone can satisfy only one customer in its each tour, service time for each customer is two minutes, set up time for each satellite station is two minutes, pick up time of a drone in a satellite station is two minutes, the upper limit of the drone variable is 10, and no customers are assigned initially to be satisfied by the first echelon vehicle.

The solution is: one first echelon vehicle first visits C4, then C5, C3, and sat2 one after another in a row and at sat2 deploys four drones to satisfy C1, C2, C7, and C8, then when drones are back the vehicle moves to the depot; another first echelon vehicle visits sat4 and deploys two drones to satisfy C9 and C10, and when drones are back it goes back to the depot; the other first echelon vehicle visits sat1 and deploys a drone to satisfy C6 and receiving the drone back it comes back to the depot location. The solution of this experiment is shown in figure 5.15.

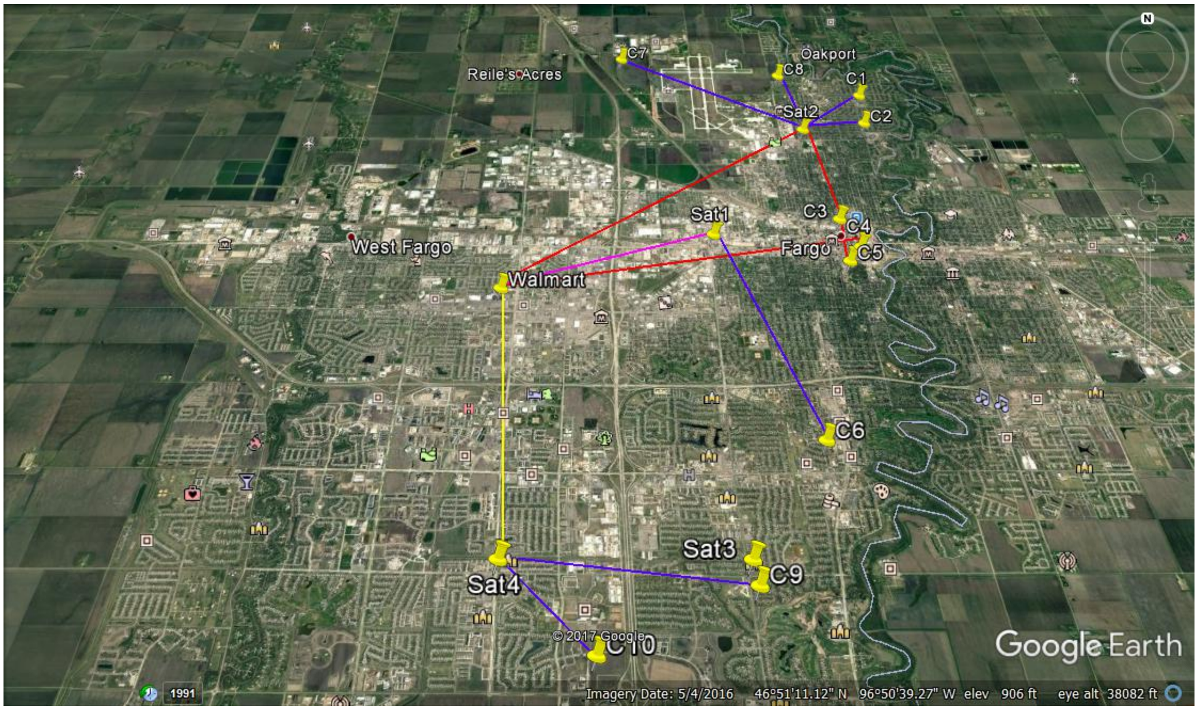

Figure 5.15. Solution of the third model for bordering customers when E=2, and G=1

5.2.3.3. Scenario 3 for Bordering Customer Locations

TM scenario 2 for bordering customers a combination of first echelon vehicles and second echelon drone routing solution with slight difference is obtained from running this model for bordering customers. The parameters of this model are: unit operating cost of each first echelon vehicle (E)=3 and unit operating cost of each drone (G) =1,the first echelon vehicle can visit utmost 14 locations, a drone can satisfy only one customer in its each tour, service time for each customer is two minutes, set up time for each satellite station is two minutes, pick up time of a drone in a satellite station is two minutes, the upper limit of the drone variable is 10, and no customers are assigned initially to be satisfied by the first echelon vehicle.

The solution is: one first echelon vehicle first visits C5, then C4, C3, and sat2 one after another in a row and at sat2 deploys four drones to satisfy C1, C2, C7, and C8, then when drones are back the vehicle moves to the depot; another first echelon vehicle visits sat4 and deploys two drones to satisfy C9 and C10, and when drones are back it goes back to the depot; the other first echelon vehicle visits sat1 and deploys a drone to satisfy C6 and receiving the drone back it comes back to the depot location. The solution of this experiment is shown in Figure 5.16.

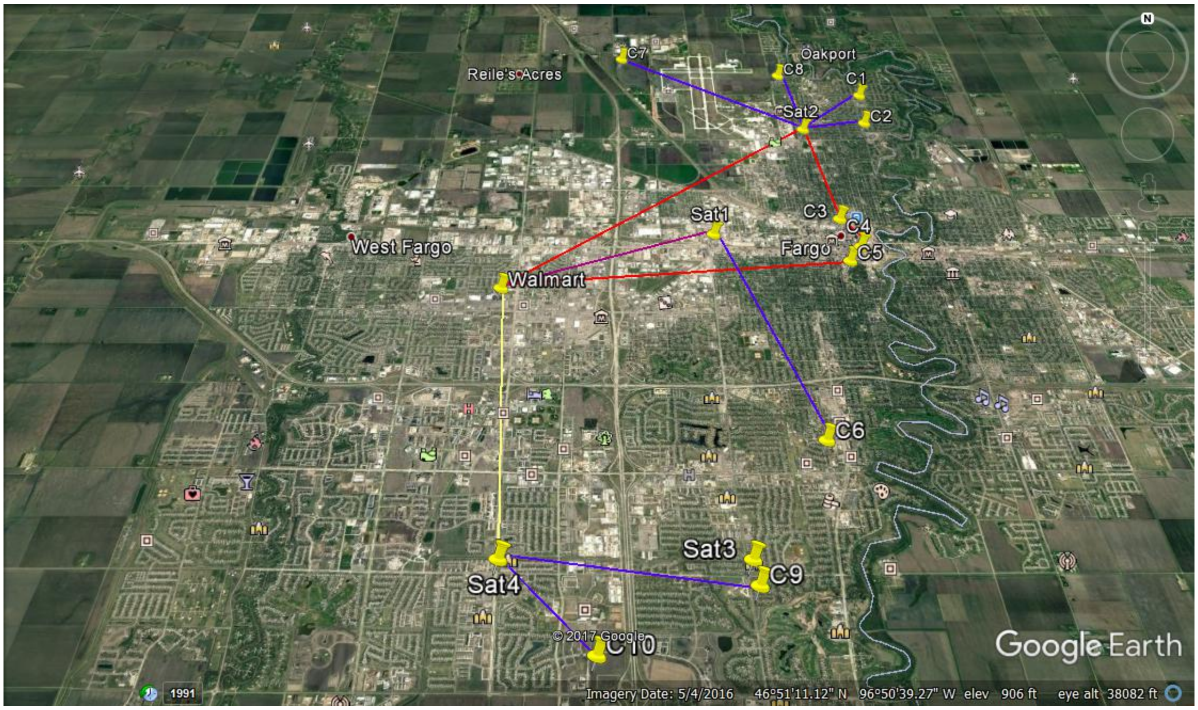

Figure 5.16. Solution of the third model for bordering customers when E=3, and G=1

5.3. Comparison of Results

5.3.1. Primary Experiments

The scenarios of our experiments are also tested with single vehicle routing model and multiple vehicle routing model. The results of our experiments are shown in Table 5.5 and other results are shown in Table 5.6.

From the results of Tables 5.5 and 5.6 we summarize our observations as follows:

I. Single Vehicle Drone only Delivery:

1. Duration of operation is better (minimum is better) in case of random customers but is worse in case of bordering customers when compared with traditional single vehicle routing model.
2. Total routing cost is worse in case of both type of customers when compared with traditional single vehicle routing model.

Table 5.5. Experimental results.

Model	Customer Distribution	Scenario	Routing Cost	Duration of Operation
Single Vehicle Drone-Only Delivery	Random	-	81,153	52,327
	Bordering	-	73,100	53,280
Single Vehicle Drone and Truck Delivery	Random	1	58,349	58,353
	Random	2	101,890	91,816
	Random	3	119,039	50,183
	Bordering	1	48,256	48,266
	Bordering	2	90,460	80,544
	Bordering	3	121,335	110,660
Multiple Vehicle Drone and Truck Delivery	Random	1	80,113	58,447
	Random	2	135,556	50,340
	Random	3	183,305	50,191
	Bordering	1	71,301	35,569
	Bordering	2	128,918	54,712
	Bordering	3	173,738	77,209

Table 5.6. Single and multiple vehicle routing results.

Scenario Name	Single Vehicle Routing		Multiple(3) Vehicle Routing	
	Routing Cost	Duration of Operation	Routing Cost	Duration of Operation
Scenario 1 for Random Customers	53,260	53,280	82,831	46,012
Scenario 2 for Random Customers	106,520	106,540	165,662	92,010
Scenario 3 for Random Customers	159,780	159,800	248,493	137,994
Scenario 1 for Bordering Customers	48,560	48,580	74,422	32,916
Scenario 2 for Bordering Customers	97,120	97,140	148,844	65,822
Scenario 3 for Bordering Customers	145,680	145,700	223,266	98,728

II. Single Vehicle Drone and Truck Delivery:

1. Except scenario 1 of random customers in all other cases this model outperforms the traditional single vehicle routing model when duration of operation is considered.
2. Except scenario 1 of random customers in all other cases this model outperforms the traditional single vehicle routing model when total routing cost is considered.

III. Multiple Vehicle Drone and Truck Delivery:

1. In all scenarios and all type of customers this model outperforms multiple vehicle routing model when total routing cost is considered.
2. Except scenario 1 this model outperforms the multiple vehicle routing model when duration of operation is considered.

5.3.2. Extensive Experiments

Six set of experiments are conducted with the same set up of depot and satellites locations as before. Only varying thing in this experiments is customer locations. Customer locations generated using random customer generator and each set of experiments has different customer locations. Experimental results are given in table 5.7 and 5.8.

From the results of Tables 5.7 and 5.8 we summarize our observations as follows:

I. Single Vehicle Drone only Delivery:

1. In case of scenario 1 total routing cost is worse for all experiments when compared with traditional single vehicle routing model.
2. In case of scenario 2 total routing cost is worse for two experiments but is better for four experiments when compared with traditional single vehicle routing model.
3. In case of scenario 3 total routing cost better in all experiments when compared with traditional single vehicle routing model.

II. Single Vehicle Drone and Truck Delivery:

1. In case of scenario 1 total routing cost is same for two experiments, is better for three experiments, and is worse for one experiment when compared with traditional single vehicle routing model.

Table 5.7. Experimental results 1.

Model	Experiment No.	Scenario	Routing Cost	Job No
Single Vehicle Drone-Only Delivery	1	1	95,536	5487887
	1	2	1,12,352	5487891
	1	3	1,23,063	5487894
	2	1	84,279	5487943
	2	2	1,06,306	5487946
	2	3	1,23,771	5487950
	3	1	78,814	5487986
	3	2	1,03,712	5487988
	3	3	1,14,423	5487990
Single Vehicle Drone and Truck Delivery	1	1	51,174	5487900
	1	2	1,03,782	5487905
	1	3	1,27,270	5487906
	2	1	53,105	5487952
	2	2	88,710	5487955
	2	3	1,16,076	5487956
	3	1	52,818	5488182
	3	2	87,614	5488187
	3	3	1,12,823	5488190
Multiple Vehicle Drone and Truck Delivery	1	1	70483	5487907
	1	2	1,24,578	5487911
	1	3	1,61,629	5487914
	2	1	65,236	5487959
	2	2	1,06,762	5487962
	2	3	1,49,028	5487965
	3	1	58,226	5488193
	3	2	98,028	5488196
	3	3	1,30,322	5488200
Single Vehicle Routing	1	1	51,174	5487924
	1	2	1,02,348	5487926
	1	3	1,53,522	5487929
	2	1	53,656	5487969
	2	2	1,07,312	5487972
	2	3	1,60,968	5487974
	3	1	53,092	5488209
	3	2	1,06,184	5488212
	3	3	1,59,276	5488215
Multiple Vehicle Routing	1	1	66,981	5487933
	1	2	1,33,962	5487935
	1	3	2,00,943	5487937
	2	1	62,099	5487979
	2	2	1,24,198	5487980
	2	3	1,86,297	5487983
	3	1	57,226	5488220
	3	2	1,14,452	5488223
	3	3	1,71,678	5488228

Table 5.8. Experimental results 2.

Model	Experiment No.	Scenario	Routing Cost	Job No
Single Vehicle Drone-Only Delivery	4	1	88,600	5488235
	4	2	1,04,402	5488237
	4	3	1,15,113	5488240
	5	1	92,356	5488288
	5	2	1,17.726	5488290
	5	3	1,35,744	5488294
	6	1	70,402	5488350
	6	2	91,764	5488352
	6	3	1,03,469	5488355
Single Vehicle Drone and Truck Delivery	4	1	43,803	5488244
	4	2	87,606	5488247
	4	3	1,11,962	5488251
	5	1	55,334	5488298
	5	2	1,08,772	5488301
	5	3	1,35,744	5488303
	6,	1	44,318	5488358
	6	2	76,876	5488362
	6	3	87,897	5488365
Multiple Vehicle Drone and Truck Delivery	4	1	58,126	5488253
	4	2	1,18,632	5488258
	4	3	1,45,545	5488260
	5	1	74,559	5488307
	5	2	1,28,886	5488314
	5	3	1,62,276	5488318
	6	1	51,562	5488369
	6	2	90,088	5488374
	6	3	1,11,170	5488378
Single Vehicle Routing	4	1	43,803	5488262
	4	2	87,606	5488267
	4	3	1,31,409	5488271
	5	1	54,301	5488321
	5	2	1,08,602	5488326
	5	3	1,62,903	5488329
	6	1	50,594	5488386
	6	2	1,01,188	5488394
	6	3	1,51,782	5488399
Multiple Vehicle Routing	4	1	59,825	5488276
	4	2	1,19,650	5488279
	4	3	1,79,475	5488284
	5	1	73,170	5488335
	5	2	1,46,340	5488339
	5	3	2,19,510	5488344
	6	1	57207	5488405
	6	2	1,14,414	5488408
	6	3	1,71,621	5488414

2. In case of scenario 2 total routing cost is same for one experiments, is better for three experiments, and is worse for two experiments when compared with traditional single vehicle routing model.
3. In case of scenario 3 total routing cost is better for all experiments when compared with traditional single vehicle routing model.

III. Multiple Vehicle Drone and Truck Delivery:

1. In case of scenario 1 total routing cost is better for two experiments, and is worse for four experiments when compared with multiple vehicles routing model.
2. In case of scenario 2 total routing cost is better for all experiments when compared with multiple vehicles routing model.
3. In case of scenario 3 total routing cost is better for all experiments when compared with multiple vehicles routing model.

5.3.3. Distance Matrices for Random Customers and Bordering Customers

Distance matrices obtained from the road distance matrix generator and aerial distance matrix generator are given in table 5.9 and table 5.11 for random customers, and in table 5.10 and table 5.12 for bordering customers respectively.

Table 5.9. Road distance matrix for random customers

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	5278	10140	9011	5412	14275	6856	10427	8758	8619	6521	12151	8583	9309	15790
2	5433	0	5141	7419	10549	12772	1333	5546	5533	6285	1997	13831	5359	4392	14287
3	10022	5271	0	15105	15130	5100	3748	2850	3032	7876	4001	18413	2857	5377	4989
4	9005	6852	14381	0	3258	18516	7809	11534	12999	4392	8472	3174	12824	11010	20031
5	6200	13650	18281	3997	0	22417	14586	18699	16899	9852	15489	6355	16724	14934	23932
6	14185	13375	5100	19050	19075	0	8178	7963	8044	18658	10812	22358	6902	12887	2995
7	6576	3231	3748	7809	11663	10868	0	4153	4140	5837	714	11272	3966	4315	8375
8	10443	5839	2850	11603	15462	7950	4173	0	5882	7376	4443	13721	5707	3766	6290
9	10438	8608	4830	15522	15547	9843	5938	7680	0	15130	5882	18830	2941	9374	11359
10	8393	6342	7740	4394	7651	18407	6340	7423	12890	0	6925	6512	12715	5918	19922
11	6516	2104	4012	8115	11603	11220	846	4417	4084	6419	0	14885	3910	4579	8319
12	11581	12837	17468	3136	6394	21604	13773	17886	16086	6351	14676	0	15911	12862	23119
13	8465	6635	2857	13549	13574	6902	3966	5707	1142	13157	3910	16857	0	7401	8418
14	9228	4359	5204	10999	14230	10305	4139	3774	7377	5953	4409	12987	7202	0	9923
15	15700	9590	4989	20565	20590	2995	8067	6290	7350	20173	8337	23873	8418	9893	0

Table 5.10. Road distance matrix for bordering customers

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	5278	10140	9011	5412	12615	11649	8207	7462	7638	9069	11367	11802	9388	9482
2	5433	0	5141	7419	10549	7600	6634	2760	3039	3351	6735	9864	7767	7796	11162
3	10022	5271	0	15105	15130	2475	1509	3655	4363	4942	8326	5233	2333	15482	15744
4	9005	6852	14381	0	3258	16856	15890	8719	8204	7632	3425	15608	16043	377	3651
5	6200	13650	18281	3997	0	20756	19790	12982	12467	11895	7431	19509	19943	4375	3004
6	12497	7893	2475	17580	17605	0	1076	4951	5462	6100	10790	8260	2447	17957	18219
7	11404	6927	1383	16488	16513	1076	0	3985	4496	5134	9824	6615	3479	16865	17127
8	8199	2760	3014	8682	12290	4951	3985	0	1383	1412	5816	7399	5348	9050	13358
9	7476	2607	4020	8288	11895	5473	4507	1609	0	578	5421	8404	6354	8656	13206
10	7651	2783	4377	7671	11279	6322	5356	1967	578	0	4804	8762	6711	8039	12236
11	8917	6441	8264	3426	6726	10723	9757	5853	5338	4766	0	16023	16458	3803	7120
12	11276	10466	5233	16141	16166	8260	6742	7395	8370	8682	16199	0	5915	16518	16780
13	11684	6934	2333	16768	16793	2447	3487	5318	6293	6605	9989	5915	0	17145	17407
14	11273	7229	17160	377	3635	19635	18669	9087	8572	8000	3802	18388	18822	0	3546
15	9942	11198	15829	3651	2666	18304	17338	13281	13382	12194	7437	17057	17491	3624	0

Table 5.11. Aerial distance matrix for random customers

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	3785	7117	6218	5120	8330	4685	7893	6282	6218	4757	8642	5714	7228	10065
2	3785	0	3723	6478	7234	5995	939	4125	3692	4389	1220	9022	3598	3524	6949
3	7117	3723	0	9841	10947	3218	2851	2046	1827	6976	2511	12262	2594	3310	3262
4	6218	6478	9841	0	3233	12466	7111	9194	10166	3451	7590	2578	10043	7586	13076
5	5120	7234	10947	3233	0	12973	8101	10877	10725	6038	8455	4507	10357	9521	14183
6	8330	5995	3218	12466	12973	0	5395	5142	2303	9949	4891	14990	2635	6527	2686
7	4685	939	2851	7111	8101	5395	0	3208	3132	4639	528	9608	3236	2813	6105
8	7893	4125	2046	9194	10877	5142	3208	0	3819	5956	3202	11419	4511	1732	4292
9	6282	3692	1827	10166	10725	2303	3132	3819	0	7749	2615	12698	826	4773	3834
10	6218	4389	6976	3451	6038	9949	4639	5956	7749	0	5166	5470	7865	4264	10077
11	4757	1220	2511	7590	8455	4891	528	3202	2615	5166	0	10101	2710	3091	5729
12	8642	9022	12262	2578	4507	14990	9608	11419	12698	5470	10101	0	12603	9734	15457
13	5714	3598	2594	10043	10357	2635	3236	4511	826	7865	2710	12603	0	5281	4557
14	7228	3524	3310	7586	9521	6527	2813	1732	4773	4264	3091	9734	5281	0	6002
15	10065	6949	3262	13076	14183	2686	6105	4292	3834	10077	5729	15457	4557	6002	0

Table 5.12. Aerial distance matrix for bordering customers

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	3785	7117	6218	5120	8880	8216	6019	6080	5785	5722	7419	8333	6556	6616
2	3785	0	3723	6478	7234	5447	4641	2257	2479	2319	4693	5953	5395	6854	8039
3	7117	3723	0	9841	10947	1766	1234	2761	3523	3865	7729	4310	1967	10201	11643
4	6218	6478	9841	0	3233	11325	10362	7243	6629	6196	2306	12334	11720	379	2374
5	5120	7234	10947	3233	0	12633	11757	8785	8379	7942	4870	12365	12599	3302	1821
6	8880	5447	1766	11325	12633	0	988	4087	4759	5167	9126	4929	1711	11673	13239
7	8216	4641	1234	10362	11757	988	0	3119	3776	4189	8149	5224	2266	10707	12308
8	6019	2257	2761	7243	8785	4087	3119	0	777	1108	5043	6502	4727	7588	9219
9	6080	2479	3523	6629	8379	4759	3776	777	0	443	4382	7250	5490	6963	8688
10	5785	2319	3865	6196	7942	5167	4189	1108	443	0	3961	7438	5830	6533	8245
11	5722	4693	7729	2306	4870	9126	8149	5043	4382	3961	0	10647	9657	2610	4578
12	7419	5953	4310	12334	12365	4929	5224	6502	7250	7438	10647	0	3252	12713	13568
13	8333	5395	1967	11720	12599	1711	2266	4727	5490	5830	9657	3252	0	12086	13425
14	6556	6854	10201	379	3302	11673	10707	7588	6963	6533	2610	12713	12086	0	2231
15	6616	8039	11643	2374	1821	13239	12308	9219	8688	8245	4578	13568	13425	2231	0

6. CONCLUSION AND FUTURE WORK

In this thesis, we are discussing a new paradigm for organizing and operating logistics systems using unmanned aerial vehicles. By orienting these systems to take advantage of the advancements in unmanned autonomous vehicles, aerial and ground, we can expect significant savings and an increase in the efficiency and reliability of the overall system. If we especially consider the infrastructure limitations when under duress, then the opportunities that arise for better and faster procurement when using two-echelon routing systems become clear.

Our preliminary results do indeed reveal that there are significant cost and time savings to be had by properly setting up a logistics system to take advantage of independent actors and/or autonomous vehicles as second echelon units. More specifically, from the results of our experiments we can observe that with the increase of the unit operating cost of first echelon vehicles, the proposed logistics setup works better. Of course, our results point to the same direction if the unit operating costs of first echelon vehicles stay the same, when the same unit costs for operating drones are decreased. Seeing as the cost of operations for manned and slow trucks is significantly higher under scenarios of war zones, emergency operations, and humanitarian applications, our proposed paradigm here could lead to significant time and cost savings.

Last, although the use of unmanned vehicles in supply chain logistics is inevitable, it has yet to become widespread. The reasons behind that are technical and legislative limitations. Some of the most important ones as identified in this work are the following. Better spoof resistance would prevent logistical and economical nightmares of cybersecurity during logistics operations. Improved battery life would also prove beneficial as it will provide unmanned vehicles with better autonomy. Similarly, improving navigation capabilities (and/or navigation systems) would also prove highly useful for drone adoption in logistics operations.

Overall, unmanned vehicles provide an excellent technology for improving our replenishment operations, especially when the logistics systems are under duress, autonomously, seamlessly, and with less costs. Our work here is the first step towards establishing public trust towards the economic viability of such projects.

REFERENCES

[1] Niels Agatz, Paul Bouman, and Marie Schmidt. Optimization approaches for the traveling salesman problem with drone. 2016.

[2] Daniela Ambrosino and Maria Grazia Scutella. Distribution network design: New problems and related models. *European journal of operational research*, 165(3):610–624, 2005.

[3] Lisa L Arnold and Paul A Zandbergen. Positional accuracy of the wide area augmentation system in consumer-grade gps units. *Computers & Geosciences*, 37(7):883–892, 2011.

[4] Geoff Clarke and John W Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4):568–581, 1964.

[5] Teodor Gabriel Crainic, Guido Perboli, Simona Mancini, and Roberto Tadei. Two-echelon vehicle routing problem: a satellite location analysis. *Procedia-Social and Behavioral Sciences*, 2(3):5944–5955, 2010.

[6] Teodor Gabriel Crainic, Antonio Sforza, and Claudio Sterle. *Location-routing models for two-echelon freight distribution system design*. CIRRELT, 2011.

[7] George Dantzig, Ray Fulkerson, and Selmer Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, 2(4):393–410, 1954.

[8] George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.

[9] Burak Eksioglu, Arif Volkan Vural, and Arnold Reisman. The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57(4):1472–1483, 2009.

[10] Bruce L Golden, Subramanian Raghavan, and Edward A Wasil. *The vehicle routing problem: latest advances and new challenges*, volume 43. Springer Science & Business Media, 2008.

[11] Todd Humphreys. Statement on the vulnerability of civil unmanned aerial vehicles and other systems to civil gps spoofing. *University of Texas at Austin (July 18, 2012)*, 2012.

[12] Gilbert Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European journal of operational research*, 59(3):345–358, 1992.

[13] Jan Karel Lenstra and AHG Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227, 1981.

[14] Ning Lu, Nan Cheng, Ning Zhang, Xuemin Shen, and Jon W Mark. Connected vehicles: Solutions and challenges. *IEEE internet of things journal*, 1(4):289–299, 2014.

[15] Chase C Murray and Amanda G Chu. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54:86–109, 2015.

[16] Simon Parkinson, Paul Ward, Kyle Wilson, and Jonathan Miller. Cyber threats facing autonomous and connected vehicles: Future challenges. *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[17] Paolo Toth and Daniele Vigo. *Vehicle routing: problems, methods, and applications*. SIAM, 2014.

[18] Marjolein Veenstra, Kees Jan Roodbergen, Leandro C Coelho, and Stuart X Zhu. A simultaneous facility location and vehicle routing problem arising in health care logistics in the netherlands. 2016.

[19] Xingyin Wang, Stefan Poikonen, and Bruce Golden. The vehicle routing problem with drones: several worst-case results. *Optimization Letters*, pages 1–19, 2016.

[20] Yan Xia, Rajan Batta, and Rakesh Nagi. Controlling a fleet of unmanned aerial vechicles to collect uncertain information in a threat environment. *Operations Research*, 2017.

APPENDIX A. JAVA CODES FOR RDM GENERATOR

```
package DbProject;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.HttpURLConnection;
import java.net.URL;
import java.sql.*;
import java.util.Scanner;
import org.json.*;
import com.mysql.jdbc.PreparedStatement;

public class ProjectCode {
public static void main(String[] args) throws FileNotFoundException {
// TODO Auto-generated method stub
String File1= "DistanceMatrix3.txt";
String File2= "TravelTimeMatrix3.txt";
PrintWriter outDistance= new PrintWriter(File1);
PrintWriter outTime= new PrintWriter(File2);

Scanner input= new Scanner(new File("walmartExperiment.txt"));
double[] lattitude = new double[61];
double[] longitute = new double[61];
```

```
for(int k=0; k¡61; k++){
lattitude[k]= input.nextDouble();
longitute[k]= input.nextDouble();
}
input.close();

for(int i=0; i¡61; i++){
for(int j=0; j¡61; j++){

double originLat = lattitude[i];
double originLong = longitute[i];
double distLat = lattitude[j];
double distLong = longitute[j];

int [] arr = calcDistance(originLat, originLong, distLat, distLong);

System.out.println(”Distance: ” + arr[0]);
System.out.println(”Duration: ” + arr[1]);
outDistance.print(arr[0] + ” ”);
outTime.print(arr[1] + ” ”);

int a=i+1;
int b=j+1;
int c=arr[0];
int d=arr[1];
int[] data= dataToDatabase(a, b, c, d);
}
outDistance.print(””);
outTime.print(””);
}
```

```
outDistance.close();
outTime.close();
if(true){
return; }
}

public static int[] dataToDatabase(int a, int b, int c, int d){
String url = "jdbc:mysql://localhost:3306/demo";
String username = "";
String password = "";
FileInputStream input1= null;
FileInputStream input2= null;

try {
// 1. create a connection to the database
Connection myConn = DriverManager.getConnection(url, username, password);
// 2. make a statement
PreparedStatement myStmt = (PreparedStatement) myConn.prepareStatement("insert
into wal3(Origin, Destination, Distance, TTime)" +" values(?, ?, ?, ?)");

myStmt.setInt(1, a);
myStmt.setInt(2, b);
myStmt.setInt(3, c);
myStmt.setInt(4, d);

// 3. Execute SQL query
myStmt.executeUpdate();
System.out.println("Update Complete");}
catch (Exception exc) {
exc.printStackTrace();
```

```
}
return null;
}

public static int[] calcDistance(double originLat, double orignLong, double distLat, double dist-
Long) {
String url = ”https://maps.googleapis.com/maps/api/directions/json?origin=url = String.format
(url, originLat, orignLong, distLat, distLong);

try {
URL obj = new URL(url);
HttpURLConnection con = (HttpURLConnection) obj.openConnection();
// optional default is GET
con.setRequestMethod(”GET”);
//add request header
con.setRequestProperty(”User-Agent”, ”Mozilla/5.0”);
int responseCode = con.getResponseCode();
// System.out.println(”Sending ’GET’ request to URL : ” + url);
// System.out.println(”Response Code : ” + responseCode);
BufferedReader in = new BufferedReader( new InputStreamReader(con.getInputStream()));
String inputLine;
StringBuffer response = new StringBuffer();
while ((inputLine = in.readLine()) != null) { response.append(inputLine);
}
in.close();
System.out.println(response.toString());
JSONObject json = new JSONObject(response.toString());
JSONObject routes = json.getJSONArray(”routes”).getJSONObject(0);
JSONObject legs = routes.getJSONArray(”legs”).getJSONObject(0);
int distance = legs.getJSONObject(”distance”).getInt(”value”);
```

```
int duration = legs.getJSONObject("duration").getInt("value");
return new int[]distance, duration;
}
catch (Exception e) {
System.out.println(e.toString());
return null;
}
}
}
```

APPENDIX B. JAVA CODES FOR ADM GENERATOR

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.util.Scanner;

public class GrtCrlDist {
public static void main(String[] args) throws FileNotFoundException {
// TODO Auto-generated method stub

String File1= "GCDistanceMatrix.txt";
PrintWriter outDistance= new PrintWriter(File1);

final double Average_Earth_Radius = 6371010;
Scanner input= new Scanner(new File("walmartExperiment.txt"));
double[] lattitude = new double[61];
double[] longitute = new double[61];
double d;
int e;

for(int k=0; k¡61; k++){
lattitude[k]= input.nextDouble();
longitute[k]= input.nextDouble();
}
input.close();

for(int i=0; i¡61; i++){
for(int j=0; j¡61; j++){
```

```
double originLat = lattitude[i];
double originLong = longitute[i];
double distLat = lattitude[j];
double distLong = longitute[j];
d= Average_Earth_Radius*

Math.acos(Math.sin(Math.toRadians(originLat))*Math.sin(Math.toRadians(distLat))+
Math.cos(Math.toRadians(originLat))*Math.cos(Math.toRadians(distLat))*
Math.cos(Math.toRadians(originLong)-Math.toRadians(distLong)));

        e = (int) Math.round(d);
System.out.print(e);
outDistance.print(e+ " ");

}
outDistance.print("");
}
outDistance.close();
}
}
```

APPENDIX C. JAVA CODES FOR RL GENERATOR

```
package randomNumber;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.util.Random;
import java.util.Scanner;

public class mainClass {
private static Random Random;
public static void main(String[] args) throws FileNotFoundException {
// TODO Auto-generated method stub

String File1= ”ExpCusLatLong.txt”;
PrintWriter outTime= new PrintWriter(File1);

Random lat= new Random();
double lmin=46.789837;
double lmax=46.934513;
double lrange=lmax-lmin;

Random lon= new Random();
double lomin=-96.882492;
double lomax=-96.768813;
double lorange=lomax-lomin;

for(int counter=1; counter¡=10; counter++){
```

```
outTime.println(lat.nextDouble()*lrange+lmin);
//outTime.print(' ');
outTime.println(lon.nextDouble()*lorange+lomin);
outTime.print(");
}
outTime.close();


}
}
```

APPENDIX D. GAMS CODES FOR THE FIRST MODEL

```
set K2 set of second echelon drones /1,2,3,4,5,6,7,8,9,10/;
set P /1,2,3,4,5,6,7,8,9,10,11,12,13,14,15/;
alias (P, O);
alias (P, S);
alias (P, C);
parameter cap1/4/;
parameter cap2/1/;

parameter b(C)
/
1 0
2 0
3 0
4 0
5 0
6 0
7 0
8 0
9 0
10 0
11 0
12 0
13 0
14 0
15 0
/;
```

parameter lambda(C) service time for customer c

/

1 0

2 0

3 0

4 0

5 0

6 2

7 2

8 2

9 2

10 2

11 2

12 2

13 2

14 2

15 2

/;

parameter d(C) demand for customer c

/

1 0

2 0

3 0

4 0

5 0

6 1

7 1

8 1

9 1

10 1

11 1

12 1

13 1

14 1

15 1

/;

parameter mu(S) set up time for satellite station s

/

1 0

2 2

3 2

4 2

5 2

6 0

7 0

8 0

9 0

10 0

11 0

12 0

13 0

14 0

15 0

/;

table Tk1(S,P) travel time for the first echelon vehicle

(this table is same as table 5.9 or table 5.10)

;

table Tk2(K2,S,P) travel time for the second echelon drones

(this table is created by the table 5.11 or table 5.12)

;

parameter $V_(S)$

/

1 0

2 2

3 2

4 2

5 2

6 0

7 0

8 0

9 0

10 0

11 0

12 0

13 0

14 0

15 0

/;

binary variable y(S) if station s is set up;

binary variable v(S,C) if customer c is assigned to a drone from station s;

binary variable f(P,S) if vehicle visits node s from node p;

binary variable x(K2,S,P,C) if drone K2 deployed from station s visits node c from node p;

positive variable tau(S) time the first echelon vehicle spends at station s;

positive variable t(K2,S) time drone k2 spends during a round trip at station s;

```
positive variable DurationOfOperation;
positive variable NumberOfDronesRequired;
integer variable L(P);
integer variable M(P);
free variable obj;
equations objEquation1, equation2(C), equation3(C,S), equation4(S), equation5(P), equation6(P,S), equation7, equation8(C), equation9(C), equation10(C,P), equation11(P,S,K2), equation12(C,S,K2), equation13(S,K2), equation14(S,K2), equation15, equation16(S);

objEquation1.. obj =e= sum(P,sum(C,Tk1(P,C)*f(P,C)))+sum(K2,sum(S$(ord(S) gt 1 and ord(S) lt 6),sum(P$(ord(P) gt 1),sum(C$(ord(C) gt 1),Tk2(K2,P,C)*x(K2,S,P,C)))));

equation2(C)$(ord(C) gt 5).. sum(S$(ord(S) gt 1 and ord(S) lt 6),v(S,C))=e=1;

equation3(C,S)$(ord(C) gt 5 and ord(S) gt 1 and ord(S) lt 6).. v(S,C)=l= y(S);

equation4(S)$(ord(S) gt 1 and ord(S) lt 6).. sum(P$(ord(P) ne ord(S)),f(P,S))=e= y(S);

equation5(P).. sum(S$(ord(S) ne ord(P)),f(S,P))-sum(S$(ord(S) ne ord(P)),f(P,S))=e=0;

equation6(P,S)$(ord(P) gt 1 and ord(S) gt 1 and ord(S) ne ord (P)).. M(P)-M(S)+cap1*f(P,S) = l = (cap1-1);

equation7.. sum(P$(ord(P) lt 2),sum(S$(ord(S) gt 1),f(S,P)))=e= 1;

equation8(C)$(ord(C) gt 5).. sum(K2,sum(S$(ord(S) gt 1 and ord(S) lt 6),sum(P$(ord(P) gt 1 and ord(P) lt 6 and ord(P) eq ord(S)),x(K2,S,C,P))))+sum(K2,sum(S$(ord(S) gt 1 and ord(S) lt 6),sum(P$(ord(P) gt 5 and ord(P) ne ord(C)),x(K2,S,C,P))))=e=1;

equation9(C)$(ord(C) gt 5).. sum(K2,sum(S$(ord(S) gt 1 and ord(S) lt 6),sum(P$(ord(P) gt 1
```

and ord(P) lt 6 and ord(P) ne ord(S)),x(K2,S,C,P))))+sum(K2,sum(S$(ord(S) gt 1 and ord(S) lt 6),sum(P$(ord(P) gt 1 and ord(P) lt 6 and ord(P) ne ord(S)),x(K2,S,P,C))))=e=0;

equation10(C,P)$(ord(P) gt 5 and ord(C) gt 5 and ord(C) ne ord(P)).. L(P)-L(C)+ cap2* sum(K2, sum(S$(ord(S) gt 1 and ord(S) lt 6),x(K2,S,P,C)))=l=cap2-1;

equation11(P,S,K2)$(ord(S) gt 1 and ord(S) lt 6 and ord(P) gt 1).. sum(C$(ord(C) gt 1),x(K2,S,P,C))-sum(C$(ord(C) gt 1),x(K2,S,C,P))=e=0;

equation12(C,S,K2)$(ord(C) gt 5 and ord(S) gt 1 and ord(S) lt 6).. sum(P$(ord(P) gt 1 and ord(P) lt 6),sum(O$(ord(O) gt 1),x(K2,P,C,O)))+sum(P$(ord(P) gt 1 and ord(P) lt 6),sum(O$(ord(O) gt 1),x(K2,P,S,O)))-v(S,C)=l= 1;

equation13(S,K2)$(ord(S) gt 1 and ord(S) lt 6).. sum(P$(ord(P) gt 1), sum(C$(ord(C) gt 1), x(K2,S,P,C)* Tk2(K2,P,C)))+ sum(P$(ord(P) gt 1), sum(C$(ord(C) gt 5),x(K2,S,P,C)* lambda(C))) =e= t(K2,S);

equation14(S,K2)$(ord(S) gt 1 and ord(S) lt 6).. tau(S)=g= t(K2,S)+ (mu(S)+$V_{(}S)$)*y(S);

equation15.. DurationOfOperation =e= sum(P,sum(C,Tk1(P,C)*f(P,C)))+ sum(S$(ord(S) gt 1 and ord(S) lt 6),tau(S));

equation16(S)$(ord(S) gt 1 and ord(S) lt 6).. NumberOfDronesRequired =g= sum(C$(ord(C) gt 5),v(S,C));

model FirstModel /all/;
solve FirstModel using mip minimizing obj;

APPENDIX E. GAMS CODES FOR THE SECOND MODEL

```
set K2 set of second echelon drones /1,2,3,4,5,6,7,8,9,10/;
set P /1,2,3,4,5,6,7,8,9,10,11,12,13,14,15/;
alias (P, O);
alias (P, S);
alias (P, C);
parameter cap1/14/;
parameter cap2/1/;
parameter E unit operating cost of the vehicle /2/;
parameter G unit operating cost of drones /1/;

parameter b(C)
/
1 0
2 0
3 0
4 0
5 0
6 0
7 0
8 0
9 0
10 0
11 0
12 0
13 0
14 0
```

15 0

/;

parameter lambda(C) service time for customer c

/

1 0

2 0

3 0

4 0

5 0

6 2

7 2

8 2

9 2

10 2

11 2

12 2

13 2

14 2

15 2

/;

parameter d(C) demand for customer c

/

1 0

2 0

3 0

4 0

5 0

6 1

7 1

8 1

9 1

10 1

11 1

12 1

13 1

14 1

15 1

/;

parameter mu(S) set up time for satellite station s

/

1 0

2 2

3 2

4 2

5 2

6 0

7 0

8 0

9 0

10 0

11 0

12 0

13 0

14 0

15 0

/;

table Tk1(S,P) travel time for the first echelon vehicle

(this table is created by the table 5.9 or table 5.10)

;

table Tk2(K2,S,P) travel time for the second echelon drone

(this table is created by the table 5.11 or table 5.12)

;

parameter $V_(S)$

/

1 0

2 2

3 2

4 2

5 2

6 0

7 0

8 0

9 0

10 0

11 0

12 0

13 0

14 0

15 0

/;

binary variable y(S) if station s is set up;

binary variable u(C) if the vehicle visits or satisfies customer c;

```
binary variable v(S,C) if customer c is assigned to a drone from station s;

binary variable f(P,S) if the vehicle visits node s from node p;

binary variable x(K2,S,P,C) if drone k2 deployed from station s visits node c from node p;

positive variable tau(S) time the first echelon vehicle spends at station s;

positive variable t(K2,S) time a drone k2 spends during a round trip at station s;

positive variable DurationOfOperation;

positive variable NumberOfDronesRequired;

integer variable L(P);

integer variable M(P);

free variable obj;


equations objEquation1, equation2(C), equation3(C),equation4(C,S), equation5(S), equation6(P),
equation7, equation8(C), equation9(P,S), equation10(C), equation11(C), equation12(P,S,K2), equa-
tion13(C,P), equation14(C,S,K2), equation15(S,K2), equation16(S,K2), equation17, equation18(S);


objEquation1.. obj =e= E*sum(P,sum(C,Tk1(P,C)*f(P,C)))+ G*sum(K2,sum(S$(ord(S) gt 1 and
ord(S) lt 6), sum(P$(ord(P) gt 1), sum(C$(ord(C) gt 1), Tk2(K2,P,C)*x(K2,S,P,C)))));


equation2(C)$(ord(C) gt 5).. u(C)+sum(S$(ord(S) gt 1 and ord(S) lt 6),v(S,C))=e=1;


equation3(C)$(ord(C) gt 5).. u(C)=g=b(C);


equation4(C,S)$(ord(C) gt 5 and ord(S) gt 1 and ord(S) lt 6).. v(S,C)=l= y(S);


equation5(S)$(ord(S) gt 1 and ord(S) lt 6).. sum(P$(ord(P) ne ord(S)),f(P,S))=e= y(S);


equation6(P).. sum(S$(ord(S) ne ord(P)),f(S,P))- sum(S$(ord(S) ne ord(P)),f(P,S)) =e= 0;


equation7.. sum(P$(ord(P) lt 2), sum(S$(ord(S) gt 1),f(S,P))) =e= 1;
```

```
equation8(C)$(ord(C) gt 5).. sum(P$(ord(P) ne ord(C)),f(P,C))=e= u(C);

equation9(P,S)$(ord(P) gt 1 and ord(S) gt 1 and ord(S) ne ord (P)).. M(P)-M(S)+ cap1* f(P,S)
=l= (cap1-1);

equation10(C)$(ord(C) gt 5).. sum(K2,sum(S$(ord(S) gt 1 and ord(S) lt 6), sum(P$(ord(P) gt
1 and ord(P) lt 6 and ord(P) eq ord(S)), x(K2,S,C,P))))+ sum(K2,sum(S$(ord(S) gt 1 and ord(S)
lt 6), sum(P$(ord(P) gt 5 and ord(P) ne ord(C)), x(K2,S,C,P)))) =e= sum(S$(ord(S) gt 1 and
ord(S) lt 6), v(S,C));

equation11(C)$(ord(C) gt 5).. sum(K2,sum(S$(ord(S) gt 1 and ord(S) lt 6),sum(P$(ord(P) gt
1 and ord(P) lt 6 and ord(P) ne ord(S)), x(K2,S,C,P))))+ sum(K2,sum(S$(ord(S) gt 1 and ord(S)
lt 6), sum(P$(ord(P) gt 1 and ord(P) lt 6 and ord(P) ne ord(S)), x(K2,S,P,C)))) =e= 0;

equation12(P,S,K2)$(ord(S) gt 1 and ord(S) lt 6 and ord(P) gt 1).. sum(C$(ord(C) gt 1), x(K2,S,P,C))-
sum(C$(ord(C) gt 1), x(K2,S,C,P))=e=0;

equation13(C,P)$(ord(P) gt 5 and ord(C) gt 5 and ord(C) ne ord(P)).. L(P)-L(C)+ cap2*sum(K2,
sum(S$(ord(S) gt 1 and ord(S) lt 6), x(K2,S,P,C))) =l= cap2-1;

equation14(C,S,K2)$(ord(C) gt 5 and ord(S) gt 1 and ord(S) lt 6).. sum(P$(ord(P) gt 1 and ord(P)
lt 6), sum(O$(ord(O) gt 1), x(K2,P,C,O)))+ sum(P$(ord(P) gt 1 and ord(P) lt 6), sum(O$(ord(O)
gt 1), x(K2,P,S,O)))-v(S,C) =l= 1;

equation15(S,K2)$(ord(S) gt 1 and ord(S) lt 6).. sum(P$(ord(P) gt 1), sum(C$(ord(C) gt 1),
x(K2,S,P,C)* Tk2(K2,P,C)))+ sum(P$(ord(P) gt 1),sum(C$(ord(C) gt 5), x(K2,S,P,C)* lambda(C)))
=e= t(K2,S);

equation16(S,K2)$(ord(S) gt 1 and ord(S) lt 6).. tau(S)=g= t(K2,S)+ (mu(S)+ V_(S))* y(S);
```

```
equation17.. DurationOfOperation =e= sum(P,sum(C,Tk1(P,C)* f(P,C)))+ sum(S$(ord(S) gt 1
and ord(S) lt 6), tau(S));

equation18(S)$(ord(S) gt 1 and ord(S) lt 6).. NumberOfDronesRequired =g= sum(C$ (ord(C)
gt 5), v(S,C));

model SecondModel /all/;
solve SecondModel using mip minimizing obj;
```

APPENDIX F. GAMS CODES FOR THE THIRD MODEL

```
set K1 set of first echelon vehicles /1,2,3/;
set K2 set of second echelon drones/1,2,3,4,5,6,7,8,9,10/;
set P /1,2,3,4,5,6,7,8,9,10,11,12,13,14,15/;
alias (P, O);
alias (P, S);
alias (P, C);
parameter cap1/14/;
parameter cap2/1/;
parameter E/2/;
parameter G/1/;

parameter b(C)
/
1 0
2 0
3 0
4 0
5 0
6 0
7 0
8 0
9 0
10 0
11 0
12 0
13 0
14 0
```

15 0

/;

parameter lambda(C) service time for customer c

/

1 0

2 0

3 0

4 0

5 0

6 2

7 2

8 2

9 2

10 2

11 2

12 2

13 2

14 2

15 2

/;

parameter d(C) demand for customer c

/

1 0

2 0

3 0

4 0

5 0

6 1

```
7 1
8 1
9 1
10 1
11 1
12 1
13 1
14 1
15 1
/;

parameter mu(S) set up time for satellite station s
/
1 0
2 2
3 2
4 2
5 2
6 0
7 0
8 0
9 0
10 0
11 0
12 0
13 0
14 0
15 0
/;
```

table Tk1(K1,S,P) travel time for the first echelon vehicles

(this table is created using table 5.9 or table 5.10)

;

table Tk2(K2,S,P) travel time for the second echelon vehicle

(this table is created using table 5.11 or table 5.12)

;

parameter $V_(S)$

/

1 0

2 2

3 2

4 2

5 2

6 0

7 0

8 0

9 0

10 0

11 0

12 0

13 0

14 0

15 0

/;

binary variable y(K1,S) if station s is set up;

binary variable u(K1,C) if vehicle k1 satisfies customer c;

binary variable v(S,C) if customer c is assigned to a drone from station s;

```
binary variable f(K1,P,S) if vehicle k1 visits node s from node p;

binary variable x(K2,S,P,C) if vehicle k2 deployed from station s visits node c from node p;

positive variable tau(S) time a first echelon vehicle spends at station s;

positive variable t(K2,S) time vehicle k2 spends during a round trip at station s;

integer variable L(P);

integer variable M(P);

free variable obj;


equations objEquation1, equation2(C), equation3(C), equation4(C,S), equation5(P,K1),
equation6(P,S,K2), equation7(S,K1), equation8(C,S,K2), equation9(C,P), equation10(C),
equation11(C), equation12(P,S), equation13(S,K2), equation14(S,K2), equation15(K1),
equation16(K1,C);


objEquation1.. obj =e= E*sum(K1,sum(P,sum(C,Tk1(K1,P,C)* f(K1,P,C))))+
G*sum(K2,sum(S$(ord(S) gt 1 and ord(S) lt 6), sum(P$(ord(P) gt 1), sum(C$(ord(C) gt 1),
Tk2(K2,P,C)* x(K2,S,P,C)))));


equation2(C)$(ord(C) gt 5).. sum(K1,u(K1,C))+ sum(S$(ord(S) gt 1 and ord(S) lt 6),v(S,C))
=e=1;


equation3(C)$(ord(C) gt 5).. sum(K1,u(K1,C)) =g= b(C);


equation4(C,S)$(ord(C) gt 5 and ord(S) gt 1 and ord(S) lt 6).. v(S,C) =l= sum(K1, y(K1,S));


equation5(P,K1).. sum(S$(ord(S) ne ord(P)), f(K1,S,P))-sum(S$(ord(S) ne ord(P)), f(K1,P,S))
=e=0;


equation6(P,S,K2)$(ord(S) gt 1 and ord(S) lt 6 and ord(P) gt 1).. sum(C$(ord(C) gt 1),
x(K2,S,P,C))-sum(C$(ord(C) gt 1), x(K2,S,C,P))=e=0;
```

equation7(S,K1)$(ord(S) gt 1 and ord(S) lt 6).. sum(P$(ord(P) ne ord(S)), f(K1,P,S)) =e= y(K1,S);

equation8(C,S,K2)$(ord(C) gt 5 and ord(S) gt 1 and ord(S) lt 6).. sum(P$(ord(P) gt 1 and ord(P) lt 6),sum(O$(ord(O) gt 1), x(K2,P,C,O)))+ sum(P$(ord(P) gt 1 and ord(P) lt 6), sum(O$(ord(O) gt 1), x(K2,P,S,O)))- v(S,C)=l= 1;

equation9(C,P)$(ord(P) gt 5 and ord(C) gt 5 and ord(C) ne ord(P)).. L(P)-L(C)+ cap2*sum(K2, sum(S$(ord(S) gt 1 and ord(S) lt 6), x(K2,S,P,C))) =l= cap2-1;

equation10(C)$(ord(C) gt 5).. sum(K2,sum(S$(ord(S) gt 1 and ord(S) lt 6), sum(P$(ord(P) gt 1 and ord(P) lt 6 and ord(P) eq ord(S)), x(K2,S,C,P))))+ sum(K2,sum(S$(ord(S) gt 1 and ord(S) lt 6), sum(P$(ord(P) gt 5 and ord(P) ne ord(C)), x(K2,S,C,P))))=e= sum(S$(ord(S) gt 1 and ord(S) lt 6), v(S,C));

equation11(C)$(ord(C) gt 6).. sum(K2,sum(S$(ord(S) gt 1 and ord(S) lt 6),sum(P$(ord(P) gt 1 and ord(P) lt 6 and ord(P) ne ord(S)), x(K2,S,C,P))))+ sum(K2,sum(S$(ord(S) gt 1 and ord(S) lt 6), sum(P$(ord(P) gt 1 and ord(P) lt 6 and ord(P) ne ord(S)), x(K2,S,P,C))))=e=0;

equation12(P,S)$(ord(P) gt 1 and ord(S) gt 1 and ord(S) ne ord (P)).. M(P)-M(S)+ cap1*sum(K1, f(K1,P,S)) =l= (cap1-1);

equation13(S,K2)$(ord(S) gt 1 and ord(S) lt 6).. sum(P$(ord(P) gt 1),sum(C$(ord(C) gt 1), x(K2,S,P,C)* Tk2(K2,P,C)))+ sum(P$(ord(P) gt 1),sum(C$(ord(C) gt 5), x(K2,S,P,C)*lambda(C))) =e= t(K2,S);

equation14(S,K2)$(ord(S) gt 1 and ord(S) lt 6).. tau(S) =g= t(K2,S)+ (mu(S)+$V_(S)$)* sum(K1,y(K1,S));

```
equation15(K1).. sum(P$(ord(P) lt 2),sum(S$(ord(S) gt 1), f(K1,S,P)))=e= 1;

equation16(K1,C)$(ord(C) gt 5).. sum(P$(ord(P) ne ord(C)), f(K1,P,C))=e= u(K1,C);


model ThirdModel /all/;
solve ThirdModel using mip minimizing obj;
```