IoT Enabled Efficient Waste Collection System

Md. Hasan Masroor Khan
Student Id: 012161003

A Thesis

in

The Department

of

Computer Science and Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Science in Computer Science and Engineering

United International University

Dhaka, Bangladesh

October 2020

Approval Certificate

This thesis titled **"IoT Enabled Efficient Waste Collection System"** submitted by **Md. Hasan Masroor Khan**, Student ID: **012161003**, has been accepted as Satisfactory in fulfillment of the requirement for the degree of Master of Science in Computer Science and Engineering on ______________________.

Board of Examiners

1.
______________________________ Supervisor
Dr. A.K.M. Muzahidul Islam
Professor
Department of Computer Science and Engineering
United International University, Dhaka, Bangladesh

2.
______________________________ Co-Supervisor
Dr. Mohammad Nurul Huda
Director, Master of Science in Computer Science and Engineering
Professor, Department of Computer Science and Engineering
United International University, Dhaka, Bangladesh

3.
______________________________ Head Examiner
Dr. Md. Motaharul Islam
Professor
Department of Computer Science and Engineering
United International University, Dhaka, Bangladesh

4.
______________________________ Examiner-I
Ms. Rubaiya Rahtin Khan
Assistant Professor
Department of Computer Science and Engineering
United International University, Dhaka, Bangladesh

5.
______________________________ Examiner-II
Mr. Suman Ahmmed
Assistant Professor
Department of Computer Science and Engineering
United International University, Dhaka, Bangladesh

6.
______________________________ Ex-Officio
Dr. Salekul Islam
Professor and Head
Department of Computer Science and Engineering
United International University, Dhaka, Bangladesh

Declaration

This is to certify that the work entitled **"IoT Enabled Efficient Waste Collection System"** is the outcome of the research carried out by me under the supervision of Professor Dr. A.K.M. Muzahidul Islam and Professor Dr. Mohammad Nurul Huda.

Md. Hasan Masroor Khan
012161003
Master of Science in Computer Science and Engineering
United International University
Dhaka, Bangladesh

In my capacity as supervisor of the candidate's thesis, I certify that the above statements are true to the best of my knowledge.

Dr. A.K.M. Muzahidul Islam
Supervisor
Professor, Department of Computer Science and Engineering
United International University
Dhaka, Bangladesh

Dr. Mohammad Nurul Huda
Co-Supervisor
Director, Master of Science in Computer Science and Engineering
Professor, Department of Computer Science and Engineering
United International University
Dhaka, Bangladesh

Abstract

In order to have a secure, comfortable and healthy life, smart cities are regarded as the future of human habitation. Within the concept of Internet of Things (IoT), the development of sensors and actuators empowers a city to serve its citizens in an intelligent, efficient and automated way. Waste management is one of those services that smart cities should provide. Rapid population growth as well as high urbanization rate and high consumption of resources contribute to the generation of large quantities of waste in a city, which is almost an uncontrollable problem faced by the municipal authorities of Asia's developing countries. Inadequate resources, inadequate financial strength and inadequate manpower take it beyond their ability to adequately handle the waste. Among all the waste management processes, waste collection is a significant process. The major part of the total budget selected for the waste management is used for waste collection only. Besides, long time travelling of the waste collecting trucks in the roads may increase fuel cost, human labor cost, CO_2 emissions etc. An effective solution to reduce these costs is an IoT enabled efficient waste collection system. By implementing IoT, the current fill level of waste bins can be detected. Therefore, trucks only visit those bins which are full or almost full thus reducing the length of the waste collection route as well as waste collection cost. In this study, a smart bin is developed to display the process of detecting the current fill level of waste bins. Moreover, to optimize the waste collection routes, different route optimization algorithms such as Dijkstra’s algorithm, Bellman-Ford algorithm and Genetic algorithm are experimented on the real life waste container locations of Mirpur, Dhaka. The performances of these algorithms are compared and the result shows that, Genetic algorithm performed better than the other route optimization algorithms and generated more optimized waste collection routes in the case of the real life waste container locations of Mirpur, Dhaka.

Acknowledgement

First and foremost, I would like to express my pure obedience and gratitude to Allah and salutation to the beloved Prophet Hazrat Muhammad (S).

I would like to express my sincere gratitude to Professor Dr. A.K.M. Muzahidul Islam, my Supervisor, and to Professor Dr. Mohammad Nurul Huda, my Co-Supervisor, for their inspiration, patience and precious guidance. Their persistent encouragement and motivation helped me to remain motivated and to do the thesis, even in challenging situations.

In my MSCSE course, I would also like to show gratitude to all of my subject teachers. I was inspired to do this thesis by their insightful teaching and my knowledge is also enriched.

My heartfelt thanks and love go to my family and in particular to my late father, Md. Seraj Uddin Khan, and Rahima Khatun, my mother. I want to show my sincere respect and cordial love to my late aunt Sabiha Begum, who, like a mother, always stood beside me and tried to keep me on the right track.

I would like to show my sincere thanks to my friends Md. Mokhlesur Rahman Maruf and Md. Shah Jalal Jamil for their valuable support in my thesis. I appreciate their support from the core of my heart. I would like to thank my friends and classmates especially Tajim, Sunny, Rahul, Asif and Faysal for their support and inspiration. We spent tremendous time together that inspired me to study effectively and obtain better results in MSCSE program.

I would like to thank my friends and colleagues of the ICT department of Dhaka Residential Model College, especially to Syed Mahbub Hasan Amiri and Mainul Islam who constantly encouraged me and provided me mental support. I would like to thank my friends, relatives, colleagues and one of my students of Dhaka Residential Model College, Shudipto Mondal who supported me with their sincere cooperation and good wishes.

Finally, I would like to express my thanks to every individual who has inspired me and wished me good success in MSCSE program.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

The future of human civilization is considered to be smart cities, where the Internet of Things (IoT) is the technology that is considered to be the main component of smart cities. In order to increase the potential for new innovative services, IoT-enabled smart devices are expected to be integrated into the backbone of the smart city community [1]. The development of sensors and actuators allows for the transformation of a city into a smart city. In order to create a sensor network between all smart devices in a smart city, the potential Internet with IPv6 (i.e. 6LoWPN) will be strongly effective [2]. Smart cities must provide citizens with various automated and intelligent cloud-based services. Waste management is one of the most significant IoT powered services offered by a smart city [3].

Waste management is currently one of the world's most critical problems for any country. The rate of urbanization has increased so much over the past few decades that experts predicted that a large amount of the world's population (i.e. 70 percent) will move to urban environments by 2050 [2]. The key causes of producing large quantities of solid waste are urbanization, economic development, rapid population growth and high resource consumption [5] [6]. Among all the solid waste management processes, waste collection is a significant process.

Waste is not properly collected in Asia's developing countries due to improper technology for collection, inadequate trained manpower, inadequate financial strength etc [7]. As a result, waste is dumped in inappropriate ways, such as on roadsides and low-lying lands that causes air pollution, contamination of water, contamination of land, contamination of underground water, etc [7] [8]. In addition, bad odor begins to emerge and nuisance of insects and mosquitoes continues to spread due to roadside dumping, which can result in severe health issues for city residents [9] [10].

Bangladesh is the third most populous country in the region of South-East Asia [11]. Dhaka, the capital city of Bangladesh, is one of the world's most densely populated cities where per square kilometer 1, 29, 501 residents live [12]. Dhaka North City Corporation (DNCC) and

Dhaka South City Corporation (DSCC) jointly generate waste of 1.6 million tons per year [13], where the daily waste generation is approximately 4634.52 tons [12].

As a developing country, Bangladesh is also experiencing solid waste management challenges owing to a lack of financial capital, a lack of infrastructure, a lack of proper planning and a lack of skilled workers [8] [14]. A substantial volume of waste (40-60%) is thus not adequately collected and disposed of at final disposal sites [15]. There are 181 open trucks, 40 compactor trucks and 194 container carriers in city corporations [16]. Besides, DSCC only has 16 compactor trucks and 135 open trucks that are not sufficient to properly collect this enormous amount of waste [17].

In Dhaka city, the collection of waste is currently performed in a static way in both Dhaka city and Asia's developing countries [2]. Since modern technology such as IoT are not used, it is not possible to distinguish from the side of the municipal authority which bins are empty and which are almost full that demand urgent collection. In addition, there is a high likelihood of encountering critical circumstances (e.g. dense traffic jam, road under construction) several times in order to collect these bins. It thus increases the length of travel, fuel costs, human labor and a lot of environmental pollution due to emissions of traffic gas on the road for a long period of time. An effective solution to decrease those costs would be to use IoT technology to determine the full bins and to optimize the waste collection routes.

In this study, to optimize the routes, different shortest path algorithms have been applied besides the IoT based system and the performance has been compared to determine the best performing algorithm. IoT technology is used to detect the fill level of every bin and to classify the full bins that have greater priority on collection over the empty bins.

Figure 1 presents the system workflow of an IoT-enabled waste management system. The waste is dumped in smart bins in the beginning. Then, the smart bins' fill levels and the corresponding bin IDs are transmitted to the cloud server. A web application is used to receive the fill level data of the bins. The data is collected by the web application and displayed in the browsers. Cloud server generates the optimized paths. The optimized paths could also be displayed in the web application. Waste is collected from the IoT based smart bins and disposed of to the final waste dumping location. The collection trucks follow the generated optimized path during the waste collection process.

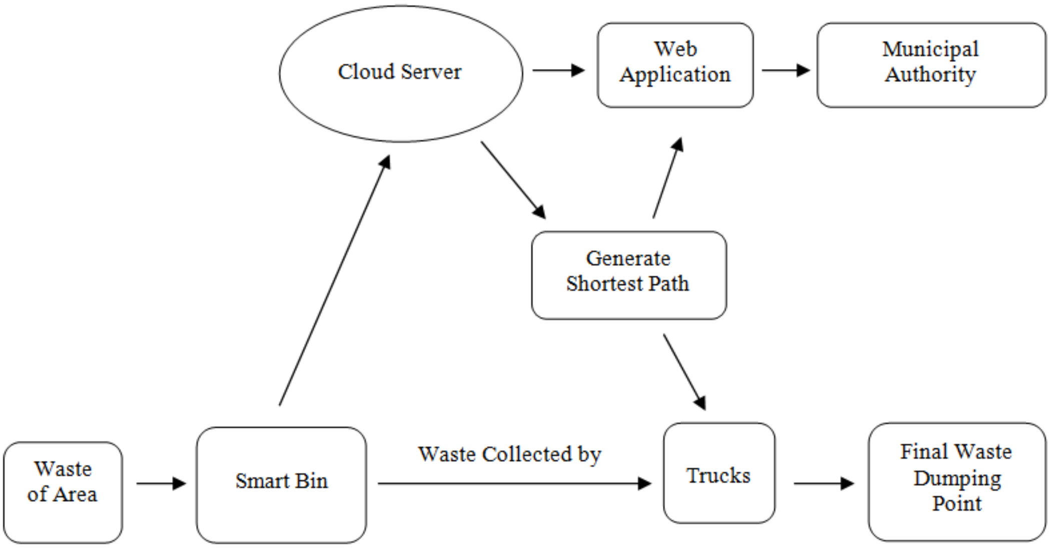

Figure 1: Work flow of an IoT enabled waste management system

In this study, a smart bin is developed to demonstrate the procedure of current fill level detection of a bin. Moreover, Dijkstra's algorithm, Bellman-Ford algorithm and Genetic algorithm are applied to calculate the shortest path from source to destination. Here, the source and the destination are the two fixed real life locations in Dhaka which are selected according to [7] and [13]. The performances of these algorithms are compared and Genetic algorithm is found better than the other algorithms. The contributions of this thesis are given below:

1. Dijkstra's algorithm, Bellman-Ford algorithm and Genetic algorithm are applied for achieving effective and efficient IoT enabled waste collection by optimizing the waste collection route.
2. The performances of these algorithms are also compared to identify the better route optimization. The performances are compared in the context of the real life waste container locations of Mirpur, Dhaka. Genetic algorithm is found as the better route optimization algorithm in the context of the real life waste container locations of Mirpur, Dhaka.
3. A smart bin is developed to demonstrate the process of fill level detection of a bin.

The rest of the report is structured as follows. Chapter 2 reviews the related works regarding waste management, waste management using IoT technology and route optimization for waste collection. Chapter 3 presents the methodology of using IoT in waste management and the route optimization algorithms to generate optimized paths to efficiently collect the bins which require immediate collection. In chapter 4, the optimization performances of the algorithms are compared and the results for this study are shown. This report is concluded in chapter 5 by stating the future directions.

Chapter 2

Background and Literature Review

Different current literatures have studied the waste management system. A robust waste management technique was suggested in [1]. Two scenarios for waste collection were considered here, where scenario 1 only used low-capacity waste collection trucks and scenario 2 used both low-capacity and high-capacity waste collection trucks. Scenario 2 became the more cost-efficient method as a result of the performance assessment.

In [2], details on the k number of total waste bins could be found using the Top-k query technique. Information about the bins was stored in a database with each bin’s GPS latitude, longitude and remaining space. The remaining capacity was changed in real time as waste entered a bin, using the IoT infrastructure. The Top-k query returned the number of k full bins according to the waste level of the bins in a descending order which should be visited.

In [3], two layered models were proposed by the authors to efficiently collect the waste from the bins. At the beginning of the trip, the Upper Layer Semi-static model provided the shortest path to reach all the waste collection bins. The Lower Layer Dynamic model immediately generated a new alternate route when a truck faced an emergency situation (e.g. traffic congestion, road under construction), to resolve the situation and to continue the waste collection process.

The ArcGIS tool was used in [7] to evaluate the pattern of bin relocation to optimize waste disposal in the bins. Here, on the area of Mirpur, the waste generation and management process e.g. waste generation amount, different types of wastes generated, number of waste containers, the waste dumping practices etc. are highlighted. A new pattern of waste bins relocation is proposed for optimized waste dumping, so that more waste can be covered and dumped properly.

In [9], a smart bin is developed where an ultrasonic sensor is used to calculate the fill level. Using the SIM900A GSM module, when the fill level exceeds the threshold level, an SMS is sent to the municipal authorities' telephone.

In [19], a smart bin is developed where the bin data measured by different sensors are transmitted to a web server using a GSM module. Truck Driver gets notification as message using GSM module. Force sensor is used to measure the weight of the waste inside the bins. An Android app is used by citizens to find the nearest bin to dispose of household waste.

In [20], a smart bin is developed where the bin data is transmitted to the open IoT platform ‘ThingSpeak’ using ESP8266 Wi-Fi module. Humidity, temperature, VOC (Volatile Organic Components) gas measurement are done to check the waste quality. Whenever the waste level of the bin exceeds the threshold level, an email notification is sent to the corresponding truck driver. A navigation system is proposed where Google Map API is used to show the route to collect all the required bins. Here, the bins to be collected are selected manually.

In [21], a smart bin is developed where the bin data is transmitted to a web server using SIM900. The Gas sensor and the Load sensor are used to detect gas emission inside the bin and to measure the weight of waste inside the bin respectively. A Heuristic algorithm is proposed to calculate the shortest path.

In [22], The GIS based software ArcGIS was used for the municipal waste collection in Kanpur, India. The shortest path was generated considering the real life waste collection locations of Kanpur. The zone with eighteen wards with thirty-three locations for the collection of municipal waste was used as the study area. The network analyst tool of the ArcGIS software used Dijkstra's algorithm to calculate the shortest path. After applying Dijkstra's algorithm, on average a 27.78% decrease in the total haul distance was observed.

In [23], route optimization is done using network analysis in ArcGIS in the city of Malang, Indonesia. Here, the optimized route is compared with the ordinary route. The result shows that, after optimization, the travel distance of the collection route are decreased almost 40%. Thus, the travel duration and the total fuel consumption are also optimized.

In [24], using the load sensor and the ultrasonic sensor, the measurement of the waste weight and the waste level of the smart bin are detected. GPRS device is used to transfer those data to the web server. Thereafter, the shortest optimized path was obtained using the Ant Colony Optimization (ACO) algorithm to reach those bins. In this study, some complex scenarios such as traffic jam, road blockage and insufficient truck capacity were also considered, thus making this process more effective.

In [25], the Dijkstra’s algorithm is used to determine the shortest optimized path where the bin positions were prioritized for collection according to the waste level of the bins. Prior to opening the lid of the waste container, RFID tags are used to check the waste collector. Using the ESP8266 Wi-Fi module, information including the bin status, exact cleaning time and the bin ID was transmitted to the ThingSpeak server.

The main focus in [26] is on the route optimization and truck requirement optimization. An optimal route planning model and algorithm based on Dijkstra’s algorithm are proposed. An equation is proposed to calculate the cost of the links or roads considering three parameters, which are, distance between the nodes, the status (fill level) of the bin, and congestion of the roads. Here, the highest priority has been given to the status (fill level) of the bin, then the roads distances and the least priority for the roads congestion. Distance measurements of the roads have been proposed using a GIS based software from the map. Embedded sensors of the smart bins could be utilized to detect the waste level of bins. Congestion of the roads could be measured by speedometer and the counter sensor by using a laser or Ultrasonic sensor. An algorithm of using both big trucks and small trucks has been proposed. The proposed model is compared with the scenario where only big trucks are used as well as the scenario where only small trucks are used. According to the result, the travel distance has been reduced around 26.12% for the big truck situation and 37.08% for the small truck situation. Moreover, the usage cost per day has been reduced around 59.81% and 41.14% for the big truck usage and small truck usage situations respectively.

In [27], the Dijkstra’s algorithm has been used to determine the shortest path for the waste collection among the real waste bin locations in the 75. Yıl Neighborhood of Yunus Emre District, Manisa Province, Turkey. 66 real life locations are considered for the shortest path calculation. The distances between the locations are taken from the Google Maps service. When no shortest path was calculated, the obtained travel distance of the total road was 3348 km. After applying Dijkstra's algorithm to calculate the shortest path, the obtained travel distance of the total road was increased to 2016 km. As a result, approximately 40% of time and fuel costs were saved. Moreover, some additional data such as traffic congestion, real time road condition and capacity of the waste collection vehicles were not considered in this study.

In [28], Genetic Algorithm (GA) is employed to determine cost effective routes for solid waste collection in order to reduce the distance travelled by waste collection trucks which will result in reduction in the fuel consumption and the CO_2 emission. The capacities of the trucks to collect waste, the traffic condition and the road condition are also considered. The algorithm is run multiple times while varying different parameters in a suitable manner. It is observed that the performance is the best when the population size is 100, the number of trucks is 3, the crossover probability is 0.2 and the mutation probability is 0.04. After the program runs for a number of times, it is seen that the average distance decreases from 1550 in first Iteration to 1280 at the end of 50th Iteration. The result indicates that, by applying a genetic algorithm, the path distance is optimized and a more efficient route has been generated. Therefore, it can be used in other similar routing problems too.

In [29], Genetic algorithm is used to optimize the waste collection procedure. Firstly, the distances between nodes are obtained by applying XML parsing over a SVG file with graph map. This procedure is followed by the application of the Floyd-Warshall algorithm to recalculate the distances. Afterwards, the Genetic algorithm has been used which results in generation of more efficient waste collection routes for the garbage collecting trucks. In this paper, the possible cost savings found is 15% by applying Genetic algorithm. To generate the result, different parameters are varied in different ways. The best result has been found for the value of the parameters where throwing of old population is 40%, maximum crossover is 20%, and minimum mutation is 80%. Also, if the system requires a solution for low-power devices, then the total no. of population was considered as 50 to obtain the best result. If the system doesn't have any power constraint, then the highest no. of population has been proposed to get the most optimized result.

In [30], the shortest path calculation performances have been compared between Dijkstra's algorithm, Symmetrical Dijkstra's algorithm, A* algorithm, Bellman-Ford algorithm, Floyd-Warshall algorithm and Genetic algorithm. In this study, Bellman-Ford algorithm performed better than the other algorithms in most of the cases. Results also show that the solution provided by Genetic algorithm has varied with the number of generations. As the number of generations increases, the solution becomes more optimized as well as the running time also increases since the higher number of generations needs more number of operations to perform. Therefore, proper adjustment in the number of generations is suggested so that the optimum ratio of running time to solution can be achieved and Genetic algorithm can be used efficiently in the shortest path calculation problems. So, Bellman-Ford algorithm has been considered as the most efficient shortest path algorithm compared to the other algorithms.

Table 1: Summary table of the related works

Author (s)	**Focus**	**Approach/Outcome**
Islam, S.M.D. et al. 2016	Waste bin location	Used GIS application to propose the new locations of waste bins to cover maximum waste dumping
S. Singh et al., 2019	Route optimization	Used ArcGIS and Dijkstra's algorithm to optimize the waste collection route which achieved 27.78% reduced distance.
A.H Putra et al., 2020	Route optimization	Used ArcGIS to optimize the waste collection route which achieved almost 40% reduced distance besides fuel consumption and travel time reduction
A. Mishra et al., 2019	Smart bin and route optimization	Developed smart bin and used Dijkstra's algorithm to optimize the waste collection route
M. A. Hossain et al., 2020	Route optimization and truck usage optimization	Used Dijkstra's algorithm and proposed a system consisting both big trucks and small trucks to optimize the waste collection route as well as to optimize the truck usage
A. Osman, 2020	Route optimization	Used Dijkstra's Algorithm to optimize the waste collection route

Table 1: Summary table of the related works (Continued)

Author (s)	Focus	Approach/Outcome
K. Bhargava et al., 2019	Route optimization	Used Genetic Algorithm to optimize the waste collection route
R. Fujdiak et al.	Route optimization	Used Genetic Algorithm to optimize the waste collection route
S. Chan et al., 2016	Comparison between different algorithms	Performed a comparison between the Dijkstra’s Algorithm, the Symmetrical Dijkstra’s Algorithm, the A* Algorithm, the Bellman-Ford Algorithm, the Floyd-Warshall Algorithm and the Genetic Algorithm which results in the Bellman-Ford Algorithm as the most efficient shortest path algorithm for that scenario.

The features that make this work exceptional than the others are mentioned below:

1. Dijkstra's algorithm, Bellman-Ford algorithm and Genetic algorithm are applied for IoT enabled efficient waste collection considering the real life locations of Mirpur, Dhaka. None of the reviewed papers applied any of these algorithms considering these real life locations.
2. The performances of these algorithms are compared considering the real life waste container locations of Mirpur, Dhaka. None of the reviewed papers compared these algorithms considering these locations.
3. Genetic algorithm is found as the better route optimization algorithm in the case of these real life waste container locations.

Thus, this work is exceptional than the other works.

Chapter 3

Methodology

In this study, the source, the waste bin locations and the destination can be represented as vertices and the roads connecting those nodes can be represented as edges. Since, the source and the destination always remain the same; therefore this problem can be represented as a single source shortest path problem. In this chapter, a smart bin which is used to represent the application of IoT in the waste management system is discussed. Moreover, the methodology and pseudocode of the algorithms i.e. Dijkstra’s algorithm, Bellman-Ford algorithm and Genetic algorithm are discussed.

3.1. Smart Bin

The smart bin is a waste container connected to modern technology (e.g. microcontrollers, sensors etc.) that automates the entire bin system. The waste collectors had to collect all the bins, whether those were empty or full, prior to using these smart devices. Using smart technologies, the real-time waste fill level of the bins can be detected. According to that information, only the full bins that require urgent collection should be emptied.

3.1.1 Components of smart bin

The components of smart bin are:

- Arduino Uno R3 (ARD-00028)
- Ultrasonic sensors (HC-SR04)
- SIM900A GSM module (GGG-00033)
- Servo motor SG90
- Jumper wires
- Breadboard (MIS-00002)

3.1.1.1 Arduino Uno R3 (ARD-00028)

Arduino Uno R3 (ARD-00028) is a microcontroller based development board manufactured by 'Arduino.cc' that allows smart operation of systems. It consists of an ATmega328P microcontroller microchip that is used in various autonomous systems. Arduino Uno does not require that the chip be programmed before installation; developers can use the USB-to - Serial converter to easily upload their own code. It is commonly used in smart technologies such as Internet of Things. This development board is presented in figure 3.

3.1.1.2 Ultrasonic sensor (HC-SR04)

Ultrasonic sensor (HC-SR04) is utilized to calculate the distance between it and the nearest thing. VCC, GND, TRIG and ECHO are the 4 pins of this sensor. A high frequency sound signal is sent via the transmitter or TRIG pin. The signal is mirrored when an object comes in front of this signal, and eventually received by the receiver or ECHO pin. The Ultrasonic sensor returns the duration taken by the sound signal in microseconds in this travel. This device is presented in figure 4.

3.1.1.3 SIM900A GSM module (GGG-00033)

The SIM900A GSM Module (GGG-00033) is produced by SIMCOM. It utilizes an attached SIM card and AT commands for making calls, receiving calls, sending SMS etc. GPRS internet of the SIM card is used to establish internet connection. An image of this module is presented in figure 5.

3.1.1.4 Servo motor SG90

The servo motor SG90 is a little machine that has an output shaft which can be placed at any angle between 0 and 180 degrees through programming. The devices attached to the shaft can also be managed to move by relocating the shaft. In various industries, servo motors are used such as toys, robots, automation, vehicles etc. This machine is shown in figure 6.

3.1.1.5 Jumper wires

To establish connection between various devices, jumper wires are utilized. In several situations, circuit boards work as a foundation to build connections through jumper wires. Male-to-Female (C&C-00071), Female-to-Female (C&C-00062) and Male-to-Male (C&C-00061) are three styles of jumper wires. These wires are presented in Figure 7.

3.1.1.6 Breadboard (MIS-00002)

A breadboard is the plastic board which is used for electronic prototyping and circuit design experimentation. Since, the breadboard is not soldered it can easily be reused. Connection between different electronic devices can be established through the interconnected little holes of it. An image of this board is presented in figure 8.

3.1.1.7 Setup of the structure

In this study, two ultrasonic sensors are utilized. The first ultrasonic sensor is attached on the front side of the bin as the entry point. The other ultrasonic sensor, the servo motor, the Arduino Uno, and the SIM900A module are located on the top of smart bin. The servo motor opens and closes the smart bin's lid. Two separate 5 Volts – 2 Amperes external power supply adapters power the Arduino Uno and the SIM900A module. The other components are linked to Arduino Uno via jumper wires and breadboard. Figure 9 and Figure 10 present the prototype of a smart bin.

3.1.2 Work flow of smart bin

In the beginning, the first ultrasonic sensor takes measurements every second and calculates the distance between it and the nearest object. The ultrasonic sensor returns the travel duration taken by the sound signal in microseconds. Afterwards, considering the travel duration, the distance to the nearest object is calculated by implementing the equation given below:

Distance (in inches) = (time required for sound signal to travel / 2) / 74;

The Arduino board sends signal to the servo motor to open the smart bin's lid by relocating the shaft from 0 to 180 degrees whenever any object is placed within 10 inches of the first ultrasonic sensor. Thereafter, Arduino waits for 5 seconds and then sends a signal again to the servo motor to relocate the shaft back to 0 degree and thus close the smart bin's lid. Afterwards, the second ultrasonic sensor measures the waste level distance from the top of the smart bin to calculate the percentage of waste fill level by considering the equation given below:

*Percentage of waste fill level = ((height of bin – waste level distance) * 100) / height of bin;*

The waste fill level percentage and the corresponding bin ID are transmitted by SIM900A module to the developed web application. AT commands are used to transmit this information using the attached SIM card's GPRS internet connection. After transmitting the data, the first ultrasonic sensor continues to take measurements and the processes goes on. The working procedure of smart bin is shown in figure 2.

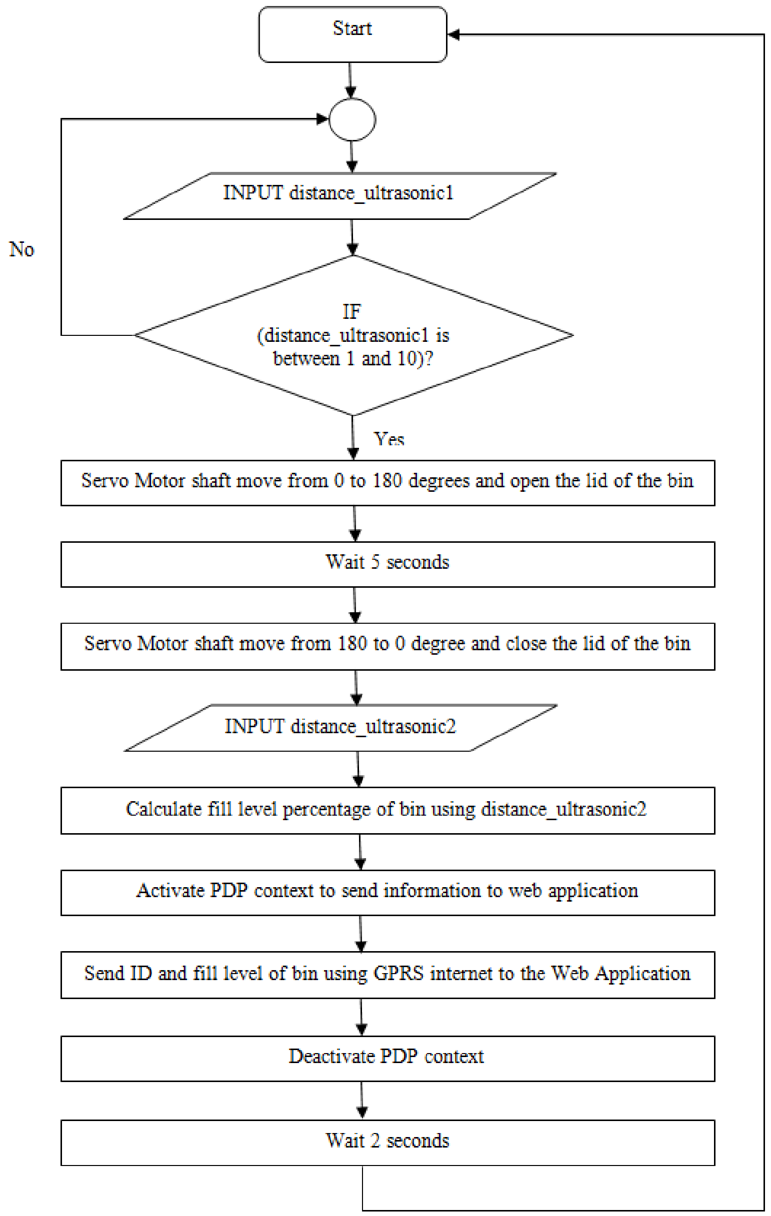

Figure 2: Smart bin's working procedure

Figure 3: Arduino Uno R3 (ARD-00028)

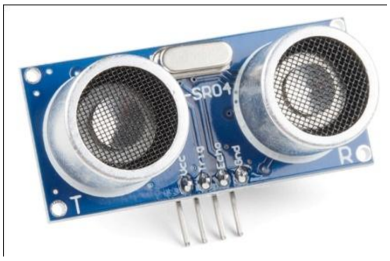

Figure 4: Ultrasonic sensor (HC-SR04)

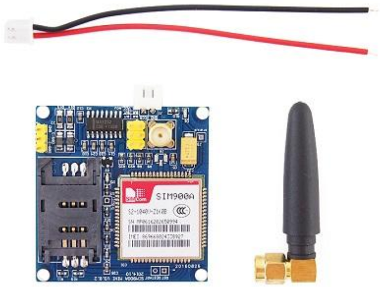

Figure 5: SIM900A GSM Module (GGG-00033)

Figure 6: Servo motor SG90

Figure 7: Jumper wires

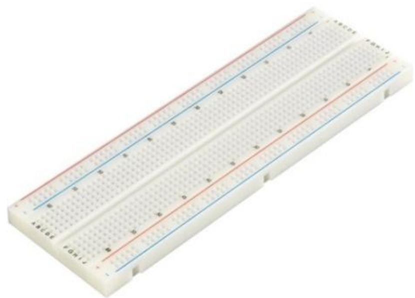

Figure 8: Breadboard (MIS-00002)

Figure 9: Smart bin’s front side

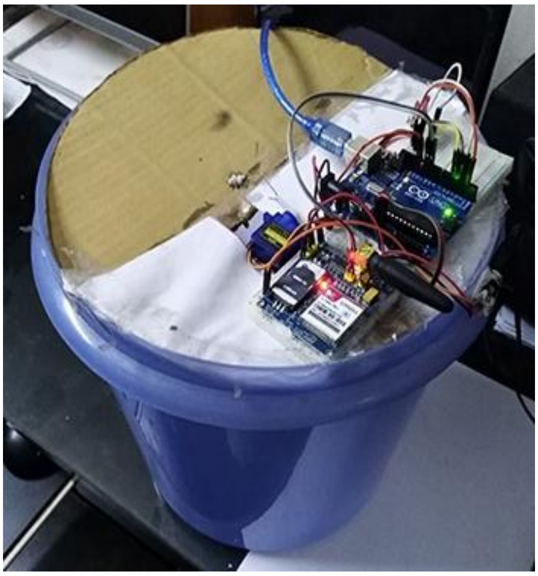

Figure 10: Smart bin’s back side

Table 2: AT commands used for data transmission

AT Command	Purpose
AT+CMEE=2	Show Error Message
AT+CPIN?	SIM is inserted or not
AT+CGREG?	Device is registered or not
AT+COPS?	SIM Network is correct or not
AT+CSQ	Check the signal quality
AT+CGATT=1	Attach GPRS
AT+CGDCONT=1,\"IP\",\"gpinternet\"	Define PDP context
AT+CGDCONT?	List all the PDP contexts that are defined
AT+CGACT=1, 1	Activate the PDP context where ID is 1
AT+SAPBR=3, 1,\"Contype\",\"GPRS\"	Type of internet connection
AT+SAPBR=3,1,\"APN\",\"gpinternet\"	Access point name
AT+SAPBR=2,1	Check whether bearer 1 is open
AT+SAPBR=1,1	Enable bearer 1
AT+HTTPINIT	Initiate the HTTP service
AT+HTTPPARA=\"CID\", 1	Set the HTTP session
AT+HTTPPARA=\"URL\",\"http://www.hasansprojects.com/updateFillData.php?bin_ID=3&fill_level=95 \"	Set the HTTP URL (here the URL used for this study is given)
AT+HTTPACTION=1	Set up the HTTP POST action
AT+HTTPTERM	Terminate the HTTP service
AT+CGACT=0, 1	Deactivate the PDP context where ID is 1

3.1.3 Data transmission to web application

A web application is partially developed to detect the current fill level of the smart bins. The web technologies used to develop the web application are HTML 5, PHP 5.6.24, MySQL (MariaDB 10.1.16), CSS 3 and JavaScript. The fill level data of waste bins along with the bin ID are transmitted by SIM900A GSM module using the GPRS internet service of the attached SIM card. Before starting the transmission, the PDP (Packet Data Protocol) context is activated for data transmission. The SIM900A GSM module used HTTP protocol to send an HTTP POST request to a URL of the web application to update the bin ID and the bin fill level. The web application receives the data, stores those data in the database and updates the fill level according to the bin ID. The requests are sent using AT commands by SIM900A GSM Module. The AT commands which are used for data transmission are shown in Table 2.

3.2 Dijkstra's Algorithm

3.2.1 Motivation of using Dijkstra's algorithm

Dijkstra's algorithm is a widely used algorithm in computer science to solve the single source shortest path problem for a given graph where the edges have nonnegative weights [26]. Some literatures have implemented Dijkstra's algorithms in their works. In [22], [25], [26] and [27], Dijkstra's algorithm is implemented to optimize and calculate the shortest path. In [30], the performance of Dijkstra's algorithm has been compared with other route optimization algorithms. Since this study is similar to the single source shortest path problem, therefore Dijkstra's algorithm is implemented here.

3.2.2 Methodology of Dijkstra's algorithm

Let us consider a basic example of Dijkstra's algorithm. Figure 11 represents a simple graph containing 5 nodes and the costs between those nodes.

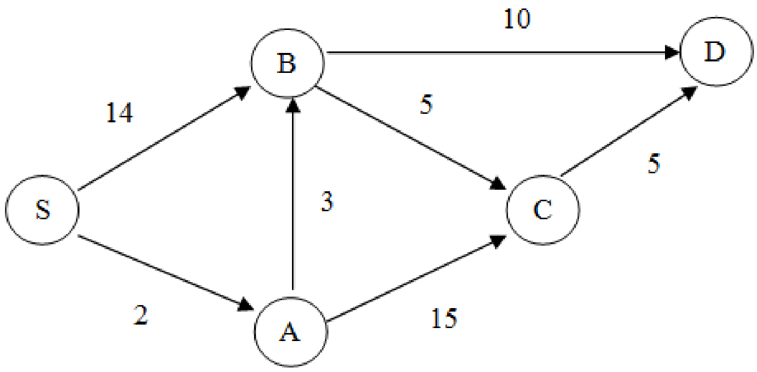

Figure 11: Graph representing costs between nodes

Here, the source or the starting point is S and the destination or end point is D. Using Dijkstra's algorithm, the total cost from source node to all the other nodes can be calculated. The cost can be the distance between the nodes, the travel time between the nodes etc. In this study, the travel time between the nodes is considered as the cost between the nodes. An adjacency matrix or cost matrix can be used to represent the graph as follows:

	S	A	B	C	D
S	0	2	14	∞	∞
A	∞	0	3	15	∞
B	∞	∞	0	5	10
C	∞	∞	∞	0	5
D	∞	∞	∞	∞	0

Figure 12: Cost matrix of the example of Dijkstra's algorithm

In this algorithm, let us consider *d* as a set of the costs or travel times from source or S to all other nodes. Since, no time needs to travel from S to S; therefore the cost from S to S is kept as 0. Since, the cost from source or S to the other nodes is unknown in the initial stage, therefore, the costs from source to the other nodes are kept as ∞ (infinity). Therefore the current costs of the *d* are:

	S	A	B	C	D
d	0	∞	∞	∞	∞

Figure 13: Initial costs of the example of Dijkstra's algorithm

Let us consider visited [ ] as a set of visited nodes. In the initial stage, visited [ ] is an empty set.

3.2.2.1 Cost calculation by edge relaxation

A greedy approach is used in Dijkstra's algorithm to select the unvisited nodes with the minimum cost. Edge relaxation process is performed on all the outgoing edges to calculate cost. Let us consider two nodes u and v, where d[u] represents the cost from source to u, d[v] represents the cost from source to v and cost (u, v) represents the cost from u to v. So, to calculate the cost from source to v, edge relaxation process is used which can be stated as:

If (d[v] > d[u] + cost (u, v)) **then**
d[v] = d[u] + cost (u, v)

3.2.2.2 Iteration 1

First of all, the visited nodes and their distances will not be included in the calculation. Here, in the costs list or the set of costs, the minimum cost is 0 which is the cost of S. So, S is inserted into the visited [ ] set. Therefore the visited nodes are:

visited = [S]

Here, the adjacent nodes of S are node A and node B.

Cost from S to A:
d[A] > d[S] + cost (S, A) = ∞ > 0 + 2 = ∞ > 2 = true
So, d[A] = 2

Cost from S to B:
d[B] > d[S] + cost (S, B) = ∞ > 0 + 14 = ∞ > 14 = true
So, d[B] = 14

After this iteration the new costs are:

	S	A	B	C	D
d	0	2	14	∞	∞

Figure 14: Costs after the first iteration of the example of Dijkstra's algorithm

3.2.2.3 Iteration 2

The adjacent unvisited nodes of S are A and B and their costs are 2 and 14 respectively. Since, A has the minimum cost; therefore, A is selected and inserted in the visited nodes set. Therefore the visited nodes are:

visited = [S, A]

Here, the adjacent unvisited nodes of A are B and C.

Cost from A to B:

d[B] > d[A] + cost (A, B) = 14 > 2 + 3 = 14 > 5 = true

So, d[B] = 5

Cost from A to C:

d[C] > d[A] + cost (A, C) = ∞ > 2 + 15 = ∞ > 17 = true

So, d[C] = 17

After this iteration the new costs are:

	S	A	B	C	D
d	0	2	5	17	∞

Figure 15: Costs after the second iteration of the example of Dijkstra's algorithm

3.2.2.4 Iteration 3

The adjacent unvisited nodes of A are B and C and their costs are 5 and 17 respectively. Since, B has minimum cost; therefore, B is selected and inserted in the visited nodes set. Therefore the visited nodes are:

visited = [S, A, B]

Here, the adjacent unvisited nodes of B are C and D.

Cost from B to C:

d[C] > d[B] + cost (B, C) = 17 > 5 + 5 = 17 > 10 = true
So, d[C] = 10

Cost from B to D:

d[D] > d[B] + cost (B, D) = ∞ > 5 + 10 = ∞ > 15 = true

So, d[D] = 15

After this iteration the new costs are:

	S	A	B	C	D
d	0	2	5	10	15

Figure 16: Costs after the third iteration of the example of Dijkstra's algorithm

3.2.2.5 Iteration 4

The adjacent unvisited nodes of B are C and D and their costs are 10 and 15 respectively. Since, C has minimum cost; therefore, C is selected and inserted in the visited nodes set. Therefore the visited nodes are:

visited = [S, A, B, C]

Here, the adjacent unvisited node of C is only D.

Cost from C to D:

d[D] > d[C] + cost (C, D) = 15 > 10 + 5 = 15 > 15 = false

So, the cost of d[D] is unchanged.

After this iteration the new costs are:

	S	A	B	C	D
d	0	2	5	10	15

Figure 17: Costs after the fourth iteration of the example of Dijkstra's algorithm

3.2.2.6 Iteration 5

The adjacent unvisited node of C is D and the cost is 15. Since, D has the minimum cost; therefore, D is selected and inserted in the visited nodes set. Therefore the visited nodes are:

visited = [S, A, B, C, D]

Since, there is no adjacent unvisited node of D; therefore cost calculation from source will not occur here. The iterations will stop here.

After this iteration the new costs are:

	S	A	B	C	D
d	0	2	5	10	15

Figure 18: Final Costs of the example of Dijkstra's algorithm

The shortest path can be obtained by following the visited nodes or by rearranging the costs of *d* from lower to higher. So, total cost of the shortest path from S to D is 15 and the path is:

S – A – B – C – D

3.2.3 Pseudocode of Dijkstra's algorithm

```
Function Dijkstra (G, source):
    Set queue [ ] as the empty set of vertices
    Set distance [ ] as the empty set of distances of the vertices
    Set visited [ ] as the empty set of vertices which are already visited
    cost [ ] = G

    Insert (source, queue)

    for each vertex v in G:
        distance[v] = infinity
        Insert (v, queue)
    end for each

    distance [source] = 0

    while queue is not empty:
        u = Get the node with minimum distance from (queue)
        Remove (u, queue)
        Insert (u, visited)

        for each vertex from u to v in G:
            if distance [v] > distance [u] + cost [u] [v]
                distance [v] = distance [u] + cost [u] [v]
            end if
        end for each
    end while
    return visited
end function
```

3.3 Bellman-Ford Algorithm

3.3.1 Motivation of using Bellman-Ford algorithm

Bellman-Ford algorithm was developed by Richard E. Bellman and Lester R. Ford, Jr., to solve the shortest path problem of a graph [30]. This algorithm works iteratively where its number of iterations was based on the number of edges path from starting node to destination node [30]. The performance of Bellman-Ford algorithm has been compared with the other route optimization algorithms in [30]. Since, this study is similar to the single source shortest path problem, therefore Bellman-Ford algorithm is implemented in this study.

3.3.2 Methodology of Bellman-Ford algorithm

Let us consider the same example considered for Dijkstra’s algorithm. In the example of Dijkstra’s algorithm, the shortest costs from source to all other nodes or points are calculated. In the case of Bellman-Ford algorithm, the calculations of the shortest paths from the source to all other nodes are also done in the same edge relaxation process like Dijkstra’s algorithm. But the difference is, Dijkstra's algorithm uses a greedy approach to select the closest unvisited node and performs the edge relaxation process on all the outgoing edges. In Bellman-Ford algorithm, without using a greedy approach, simply all the edges are relaxed. Here, the whole process will run for N-1 times, where N represents the total number of nodes in a graph.

In this algorithm, a variable d is also considered as a set of the costs or travel times from source to all other nodes. But, the variable visited [ ], which has been used to contain the visited nodes will not be considered here.

Like the example of Dijkstra’s algorithm, after N-1 times, the new costs will become:

	S	A	B	C	D
d	0	2	5	10	15

Figure 19: Final Costs of the example of Bellman-Ford algorithm

So, by rearranging the costs from lower to higher the shortest path can be obtained as:

S – A – B – C – D

3.3.3 Pseudocode of Bellman-Ford algorithm

```
Function Bellman-Ford (G, source):
    Set distance [ ] as empty set of distances of the vertices
    cost [ ] = G
    Set N as total number of vertices

    for each vertex v in G:
        distance[v] = infinity
    end for each

    distance [source] = 0

    for count from 1 to N-1:
        for each vertex from u to v in G:
            if distance [v] > distance [u] + cost [u] [v]
                distance [v] = distance [u] + cost [u] [v]
            end if
        end for each
    end for

    return distance
end function
```

3.4 Genetic Algorithm

3.4.1 Motivation of using Genetic algorithm

Genetic algorithm is a heuristic search algorithm belonging to the class of Evolutionary algorithms, where complex computations, innovations and large scale parallelism are required [28]. The Genetic algorithm is considered as a strong computational and optimization tool, which could be used for more efficient municipal waste management [29]. The performance of Genetic algorithm has been compared with the other route optimization algorithms in [30]. Since, one of the main goals of this study is to optimize the waste collecting route; therefore, Genetic algorithm is implemented here.

3.4.2 Methodology of Genetic algorithm

Let us consider a new example where S, A, B, C and D are 5 locations, S represents the source, D represents the destination and A, B and C represent the bin locations. In this study, the routes represent the chromosomes. Since, this algorithm is permutation based and the source and the destination locations are fixed, first of all the permutation is done among the bin locations A, B and C. Let the size of the population is 4. Therefore, after the permutation, 4 random paths have been selected as the initial population. The costs of each path represent the fitness value of the respected paths. Let's consider some random values as the distances of the paths. So, the initial populations, which are selected as for the next iteration, are as follows:

S A B C D Cost: 27
S B A C D Cost: 38 (maximum)
S C B A D Cost: 32
S C A B D Cost: 22 (minimum)

Here, the current minimum path cost is 22 and the route is "S C A B D". The maximum path cost is 38 and the route is "S B A C D". Since, the main objective of this algorithm is to optimize the route cost, so the less the cost the better the fitness value the path has. Therefore, a selection is performed based on minimum cost to select the paths with the better fitness values where the route with the maximum cost is removed from the populations. So, the remaining routes are:

S A B C D Cost: 27
S C B A D Cost: 32
S C A B D Cost: 22

Afterwards, the swap mutation is performed on any of the remaining routes and a new route is generated. Swap mutation is done by swapping any two elements of the path except the source and the destination since these locations are fixed. The newly generated route is checked whether this route matches with the existing routes. Another checking is also performed whether the newly generated route matches with the routes which were removed. If any of the checking returns true, then swap mutation is done again. If all the permutations are checked and a new route cannot be generated, then the route with the lowest cost will be considered as the shortest route.

After performing the swap mutation, the new generation of population including a newly generated route is as follows:

S A B C D Cost: 27
S C B A D Cost: 32 (maximum)
S C A B D Cost: 22 [Swap C and A]
S A C B D Cost: 15 [After swapping C and A] (minimum)

Here, the current minimum path cost is 15 and the route is “S A C B D”. The maximum path cost is 32 and the route is “S C B A D”. This process will continue until the minimum cost of the current generation is equal to the minimum cost of the previous generation. Since, the current minimum cost is 15, which does not match with the previous minimum cost 22 thus, the process should continue.

After performing selection based on minimum cost and removing the route with maximum cost from the populations, the remaining routes are:

S A B C D Cost: 27
S C A B D Cost: 22
S A C B D Cost: 15

After performing the swap mutation, the new generation of populations is as follows:

S A B C D Cost: 27 (maximum)
S C A B D Cost: 22
S A C B D Cost: 15 [Swap A and B] (minimum)
S B C A D Cost: 18 [After swapping A and B]

Here, the current minimum cost is equal to the previous minimum cost. Therefore, all the iterations should stop here and no more iteration will be continued. In both the cases, the route is “S A C B D” and the cost of this route is 15.

So, in this example the cost of the optimized route is 15 and the optimized shortest route is:

S – A – C – B – D

3.4.3 Pseudocode of Genetic algorithm

```
Function Genetic Algorithm (Graph, source, destination):
        PreviousMinimumCost = 0
        Set costs_list [ ] as the empty set of costs of different routes
        N = Size of population
        Select N routes randomly as initial population keeping the source and the destination
fixed

        For each route in population
                cost = Calculate cost of the route
                Insert (cost, costs_list)
        End for each

        CurrentMinimumCost = Get minimum cost from (costs_list)

        While PreviousMinimumCost ≠ CurrentMinimumCost
                PreviousMinimumCost = CurrentMinimumCost
                Remove the route with maximum cost from population
                Do swap mutation on any of the remaining routes of the population
                CurrentMinimumCost = Get minimum cost from (costs_list)
        End while

        shortestPath = Path with the CurrentMinimumCost from (population)
        return shortestPath
end function
```

Chapter 4

Results

4.1 Study Location

According to [7], Mirpur, Zone – 2 (Mirpur – Pallabi) of DNCC (Dhaka North City Corporation) is selected as the study location, where the area of the location is 10.40 km2. In 2016, the amount of solid waste generation per day was 353.34 tons in that location [7].

4.1.1 Starting Point

The authors in [13] mentioned the Zonal Office (Zone - 4, Mirpur) of DNCC in their study as the location of the waste collection trucks. According to their field survey, in the Zonal Office (Zone - 4, Mirpur) of DNCC, different types of garbage trucks were found to collect waste from containers and transport the waste to the disposal site. Therefore, in this study, the Zonal Office (Zone - 4, Mirpur) of DNCC is considered as the source or the starting point to start the waste collection.

4.1.2 Bin Locations

According to [7], the study area contains 41 different sized waste containers which are kept in 17 locations. In this study, among those 17 locations, 7 random locations are considered as the waste bin or container locations to calculate the shortest path and to compare the performance of the algorithms.

4.1.3 Destination Point

According to [7], the trucks collect waste from the study area and dispose of it to the Amin Bazar landfill site at different trips. The location of the Amin Bazar waste dumping site is 23º47'48''N and 90º17'50''E, which is within the low-lying floodplain of the Karanachhali River in Savar, Dhaka [7]. The area is a 50 acres semi-aerobic landfill site which facilitates rapid decomposition of waste [7]. Dhaka-Aricha Highway is used to dispose of the collected waste on this site. Therefore, in this study, the Amin Bazar waste dumping site is considered as the destination point to dispose of all the collected waste. The geographic coordinates of all these locations are collected from Google Maps which are given in Table 3.

Table 3: Geographic coordinates of the route locations

Location	Latitude	Longitude
Source	23.808216	90.368621
Bin 1	23.812160	90.367057
Bin 2	23.810344	90.363458
Bin 3	23.813523	90.361660
Bin 4	23.820511	90.363806
Bin 5	23.820502	90.366317
Bin 6	23.821061	90.370104
Bin 7	23.804795	90.349225
Destination	23.797936	90.300171

4.1.4 Cost Data Collection

In this study, the travel time between the locations is considered as the cost or weight between the locations. The travel time required between all the locations which include the starting location, the bin locations and the destination location are collected using the Distance Matrix API of Google Maps [31]. The travel durations are returned in seconds. The "traffic_model" parameter of the Distance Matrix API is used to calculate the travel time [31]. The value "best_guess" is used as the value of the "traffic_model" parameter of the Distance Matrix API, which indicates that the returned duration in traffic is the best estimate of travel time based on both historical traffic conditions and live traffic [31]. Cost matrices have been generated using those time costs. Using the cost matrices the total costs of the routes which start from the starting point and end to the destination point, are calculated from the values of the cost matrices. The values in the cost matrices are represented in seconds.

4.2 Comparison between Algorithms

In this study, the number of travel locations is varied and the shortest path was calculated for various numbers of locations which include the starting point and the destination point. The current fill levels of the bins can be identified using IoT. To optimize the total cost of the route, the trucks should only visit the full or almost full containers which require collection rather than the empty or half full containers. Since, 7 waste container locations are selected randomly from the 17 locations of the study area; therefore, the total number of the full waste containers is varied between 3, 4, 5, 6 and 7 in this study. So, by including the starting point and the destination point, the total numbers of travel locations become 5, 6, 7, 8 and 9 respectively. The selection process of the full bins and the travel points is presented in Table 9. Here, the shortest paths are calculated and the performances are compared considering the total numbers of travel locations. In the case of Genetic algorithm, the size of the population is considered as 5 in this study.

In this study, the waste container locations are chosen randomly. In the case of 5 travel locations, the container locations 1, 4 and 6 are chosen randomly along with the source and the destination. The cost matrix is presented in Table 4.

Table 4: Cost matrix for 5 locations

0	270	590	481	1563
243	0	320	449	1700
532	452	0	256	1751
485	422	283	0	2021
1633	1634	1743	1999	0

In the case of 6 travel locations, the container locations 1, 2, 4 and 6 are chosen randomly along with the source and the destination. The cost matrix is presented in Table 5.

Table 5: Cost matrix for 6 locations

0	270	326	590	481	1563
243	0	226	320	449	1700
218	136	0	337	585	1524
532	452	347	0	256	1751
485	422	468	283	0	2021
1633	1634	1547	1743	1999	0

In the case of 7 travel locations, the container locations 1, 2, 3, 4 and 6 are considered randomly along with the source and the destination. The cost matrix is presented in Table 6.

Table 6: Cost matrix for 7 locations

0	270	326	415	590	481	1563
243	0	226	253	320	449	1700
218	136	0	121	337	585	1524
389	307	178	0	216	471	1583
532	452	347	304	0	256	1751
485	422	468	507	283	0	2021
1633	1634	1547	1527	1743	1999	0

In the case of 8 travel locations, the container locations 1, 2, 3, 4, 5 and 6 are considered randomly along with the source and the destination. The cost matrix is presented in Table 7.

In the case of 9 travel locations, the container locations 1, 2, 3, 4, 5, 6 and 7 are considered along with the source and the destination. The cost matrix is presented in Table 8.

Table 7: Cost matrix for 8 locations

0	270	326	415	590	615	481	1563
243	0	226	253	320	344	449	1700
218	136	0	121	337	480	585	1524
389	307	178	0	216	398	471	1583
532	452	347	304	0	182	256	1751
421	358	404	443	316	0	74	1957
485	422	468	507	283	274	0	2021
1633	1634	1547	1527	1743	1926	1999	0

Table 8: Cost matrix for 9 locations

0	270	326	415	590	615	481	600	1563
243	0	226	253	320	344	449	504	1700
218	136	0	121	337	480	585	328	1524
389	307	178	0	216	398	471	387	1583
532	452	347	304	0	182	256	555	1751
421	358	404	443	316	0	74	733	1957
485	422	468	507	283	274	0	797	2021
574	492	354	334	550	732	805	0	1243
1633	1634	1547	1527	1743	1926	1999	1165	0

Table 9: Waste containers and total travel locations

Waste Container Locations	**Number of Waste Container Locations**	**Total Travel Locations**
1, 4, 6	3	5
1, 2, 4, 6	4	6
1, 2, 3, 4, 6	5	7
1, 2, 3, 4, 5, 6	6	8
1, 2, 3, 4, 5, 6, 7	7	9

From Table 10, it can be observed that, Genetic algorithm generates a more optimized route compared to Dijkstra’s algorithm and Bellman-Ford algorithm. When the number of travel locations is 5, then all the algorithms generate the shortest path with the same optimized cost. As the number of travel locations increased, Genetic algorithm performed better than the other shortest path algorithms.

Table 10: Comparison of time cost efficiency between algorithms

Total Travel Locations	**Dijkstra's Algorithm**	**Bellman-Ford Algorithm**	**Genetic Algorithm**
5	45 min 53 sec	45 min 53 sec	45 min 53 sec
6	51 min 55 sec	51 min 55 sec	47 min 18 sec
7	52 min 2 sec	52 min 2 sec	51 min 50 sec
8	58 min 30 sec	58 min 30 sec	51 min 50 sec
9	76 min 55 sec	76 min 55 sec	52 min 9 sec

In [30], performances of various shortest path algorithms including Dijkstra's algorithm, Bellman-Ford algorithm and Genetic algorithm are compared. As a result, Bellman-Ford algorithm performed better than the other algorithms in most of the cases.

In this study, the performances of Dijkstra's algorithm, Bellman-Ford algorithm and Genetic algorithm are compared for various number of travel locations. Here, as the number of travel locations increased, Genetic algorithm performed better than the other shortest path algorithms.

Chapter 5

Conclusion and Future Works

5.1 Conclusion

An IoT-enabled effective waste collection process has been suggested in this study. A smart bin is developed to show the current fill level detection process of waste bins. Therefore, it will be easier to detect the full or almost full bins which have high requirement of collection. Afterwards, Dijkstra's algorithm, Bellman-Ford algorithm and Genetic algorithm are implemented on the real life waste container locations of Mirpur, Dhaka and their performances are compared. Google's *Distance Matrix API* is used to obtain the travel duration or journey times between all locations. Finally, according to real life waste container locations, Genetic algorithm is found as the better performing algorithm than the other algorithms.

5.2 Limitations

Before determining the optimized route, the capability of a vehicle to collect the waste from the required bins is not considered. Hence, it is unclear if a vehicle has sufficient capacity for collecting the remaining waste. The vehicle's availability for collection is not taken into consideration in this study. The vehicle's fitness is also not taken into account.

5.3 Future Works

- Before determining the optimized route and appointing a driver, the capability to collect the waste from the required bins, the availability and the fitness of the vehicles will be considered.

- The real life waste bin locations of the entire Dhaka North City Corporation (DNCC) will be considered to calculate the shortest path

- An image recognition system will be developed using machine learning to detect different waste types for appropriate waste segregation.

- HTTP (Hypertext Transfer Protocol) has been used to transmit the bin data from the bin to the web application server to update the fill level of the bin. HTTP is designed for internet based devices and applications where low power and any other constraints are not crucial. IoT devices which often run on batteries have different constraints such as power limitations, limited memory, limited processing capabilities etc. Constrained Application Protocol (CoAP) is a special type of protocol which is designed to be used for the resource constrained devices and applications in the field of IoT (Internet-of-Things), WSN (Wireless Sensor Network) and M2M (Machine-to-Machine) communication system. CoAP is simple and draws less power compared to HTTP. The packet sizes in CoAP are much smaller than HTTP. Moreover, CoAP is designed to interoperate with HTTP, so that CoAP messages can be translated to HTTP which enables simplified integration of CoAP with the web technology. Therefore, CoAP will be considered in the future for data transmission.

- The web application is partially developed but it is in a demonstration phase. More functionality will be added to make it robust for monitoring the waste management process.

In this study, an IoT-enabled waste level detection process of a bin is presented. Moreover, a better route optimization algorithm is also observed for efficient waste collection in this study. IoT-enabled various smart cloud-based services are provided in the current world of technology. One of these services is presented in this report.

References

[1] T. Anagnostopoulos, A. Zaslavsky, and A. Medvedev, “Robust waste collection exploiting cost efficiency of IoT potentiality in Smart Cities,” *2015 Int. Conf. Recent Adv. Internet Things, RIoT 2015*, no. April, 2015.

[2] T. Anagnostopoulos, A. Zaslavsy, A. Medvedev, and S. Khoruzhnicov, “Top - k Query Based Dynamic Scheduling for IoT-enabled Smart City Waste Collection,” *Proc. - IEEE Int. Conf. Mob. Data Manag.*, vol. 2, no. June, pp. 50–55, 2015.

[3] A. Lukashin, L. Laboshin, V. Zaborovsky, and V. Mulukha, “Internet of Things, Smart Spaces, and Next Generation Networks and Systems,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8638, no. August 2014, pp. 535–543, 2014.

[4] I. Hong, S. Park, B. Lee, J. Lee, D. Jeong, and S. Park, “IoT-Based Smart Garbage System for Efficient Food Waste Management,” vol. 2014, 2014.

[5] F. A. S. Islam, “Solid Waste Management System in Dhaka City of Bangladesh,” *J. Mod. Sci. Technol.*, vol. 4, no. 1, pp. 192–209, 2016.

[6] A. Abedin and M. Jahiruddin, “Asian Journal of Medical and Biological Research Waste generation and management in Bangladesh : An overview,” *Asian J. Med. Biol. Res.*, vol. 1, no. 1, pp. 114–120, 2015.

[7] E. Sciences, “Municipal Solid Waste Management using GIS Application in Mirpur Area of Dhaka City, Bangladesh,” *Pollution*, vol. 2, no. 2, pp. 141–151, 2016.

[8] M. Zahur, “Solid Waste Management of Dhaka City : Public Private Community Partnership,” vol. IV, no. 2, pp. 93–97, 2007.

[9] M. K. A, N. Rao, and P. S. B, “Smart Dustbin-An Efficient Garbage Monitoring System,” *Rev. Int. J. Eng. Sci. Comput.*, vol. 6, no. 6, pp. 7113–7116, 2016.

[10] K. Bahauddin and M. Uddin, “Prospect of Solid Waste Situation and An Approach of Environmental Management Measure (EMM) Model for Sustainable Solid Waste Management: Case Study of Dhaka City,” *J. Environ. Sci. Nat. Resour.*, vol. 5, no. 1, pp. 99–111, 2012.

[11] M. Parvin and A. Begum, “Organic Solid Waste Management and the Urban Poor in Dhaka City,” *Int. J. Waste Resour.*, vol. 08, no. 01, pp. 1–9, 2018.

[12] A. Saifullah and M. Tasbirul Islam, “Municipal solid waste (MSW) management in Dhaka City,” *Am. J. Eng. Res.*, no. 5, pp. 88–100, 2016.

[13] M. R. Kabir, “Municipal Solid Waste Management System: A Study on Dhaka North and South City Corporations,” *J. Bangladesh Inst. Planners*, vol. 8, no. December, pp. 35–48, 2015.

[14] S. R. Das, M. Lokman Hossain, S. Talukder, and M. K. Hossain, “01 Solid Waste Management in Commercial City of Bangladesh,” *J. Environ. Treat. Tech.*, vol. 3, no. 1, pp. 60–70, 2015.

[15] P. C. Roy and S. Roy, “Status of Solid Waste Disposal and Management Practices in Bangladesh - a Study on Dhaka City,” vol. 3, no. Dcc, pp. 123–128, 2013.

[16] S. A. Mondal and M. Rahman, “Comparative Analysis of Traditional Open Truck and Hauled Container Systems of Waste Collection in Dhaka City JICA Expert Team , 2 Bangladesh University of Engineering and Technology Conceptualization of Primary and Secondary Collection : Conceptualizatio,” vol. 77, no. February, pp. 978–984, 2013.

[17] M. R. Khan, F. Sultana, and F. Akter, “Solid Waste Management Scenario of Some Selected Places in Dhaka City,” vol. 1, no. 8, pp. 1–6, 2016.

[18] A. Medvedev, P. Fedchenkov, A. Zaslavsky, T. Anagnostopoulos, and S. Khoruzhnikov, *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, vol. 9247, no. August. 2015.

[19] S. V. Kumar, T. S. Kumaran, A. K. Kumar, and M. Mathapati, “Smart Garbage Monitoring and Clearance System using Internet of Things,” no. August, pp. 184–189, 2017.

[20] K. F. Haque, R. Zabin, K. Yelamarthi, P. Yanambaka, and A. Abdelgawad, “An IoT Based Efficient Waste Collection System with Smart Bins,” pp. 1–5, 2020.

[21] M. Johnson and C. E. Shyni, “SMART GARBAGE BIN WITH EFFICIENT ROUTING AND MANAGEMENT SYSTEM,” vol. 14, no. 5, pp. 80–89, 2019.

[22] S. Singh and S. N. Behera, *Advances in Waste Management*. Springer Singapore, 2019.

[23] E. Science, “Waste Transportation Route Optimization in Malang using Network Analysis Waste Transportation Route Optimization in Malang using Network Analysis,” 2020.

[24] S. Sharmin and S. T. Al-Amin, “A cloud-based dynamic waste management system for smart cities,” *Proc. 7th Annu. Symp. Comput. Dev. ACM DEV-7 2016*, no. November, 2016.

[25] A. Mishra, N. Ghosh, and P. Jena, “Internet of Things based Waste Management System for Smart Cities : A real time route optimization for waste collection vehicles,” no. 4, pp. 496–503, 2019.

[26] M. A. Hossain, M. Z. M. Z. Harith, M. Yamani, and I. Idris, “Route Optimization by using Dijkstra ’ s Algorithm for the Waste Management System,” pp. 110–114, 2020.

[27] A. Osman, “The Shortest Road Algorithm Approach In Determining The Route Of Solid Waste Collection Vehicles : The Case Of Manisa 75 . Y ı l Neighborhood Katı Atık Toplama Araçlarının Güzergâh Tespitinde En Kısa Yol Algoritması Yaklaşımı :,” no. March, 2020.

[28] K. Bhargava, R. Gupta, A. Singhal, and A. Shrinivas, “Genetic Algorithm to Optimize Solid Waste Collection,” pp. 1–8, 2019.

[29] R. Fujdiak, P. Masek, P. Mlynek, J. Misurec, and E. Olshannikova, “Using Genetic Algorithm for Advanced Municipal Waste Collection in Smart City.”

[30] S. Chan, Y. Meng, N. Adnan, S. S. Sukri, and W. Mohd, “An Experiment on the Performance of Shortest Path Algorithm,” no. August, pp. 29–30, 2016.

[31] Google. Google Maps Distance Matrix API. Developer’s Guide. Available from: https://developers.google.com/maps/documentation/distance-matrix/overview