Smart Dustbin

Internship Report

Submitted in partial fulfillment of the requirements

For the degree of

Bachelor of Technology in Computer Science & Engineering

By

Dip Patel 19BCE166

Parv Patel 19BCE190

Guided By

Dr. Smita Agrawal

Prof. Parita Oza

[DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING]

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Ahmedabad 382481

This is to certify that the Internship entitled “Smart Dustbin” submitted by Dip Patel(19BCE166) and Parv Patel(19BCE190), towards the partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering of Nirma University is the record of work carried out by him/her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination.

Dr. Smita Agrawal	Prof. Parita Oza	Dr. Madhuri Bhavsar,
Assistant Professor	Assistant Professor	Professor and HOD,
Computer Science and Engineering Dept.,	Computer Science and Engineering Dept.,	Computer Science and Engineering Dept.,
Institute of Technology,	Institute of Technology,	Institute of Technology,
Nirma University,	Nirma University,	Nirma University,
Ahmedabad	Ahmedabad	Ahmedabad

Acknowledgement

This dissertation would not have been possible without the guidance and the help of Guiding faculties, who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

We would like to express our gratitude to the Institute of Technology under Nirma University for including an internship program as a credit course, which has provided an opportunity to work on such an Evolutionary Research project.

Our sincere gratitude to **Dr. Smita Agrawal** and **Prof. Parita Oza** for devoting time and explaining how work is being done in research and assigning us with various tasks during these 8 weeks of internship period.

We would also like to extend special thanks to the entire lab staff for their cooperation and help for internet access.

CONTENTS

Abstract

Looking at the contemporary scenario of the waste management system, we see various issues. Route finding for garbage collection is part of a waste management system that can be more cost-efficient with the help of technology. This research paper aims to find the optimized route for trucks to collect garbage from smart dustbins, reducing costs and making a more efficient and profitable system. Route finding is an NP-hard problem that can be solved by many algorithms, some are heuristics, meta heuristics, VRP, and genetic algorithms. In this paper, k mean clustering and genetic algorithm is used to find route optimization for the Ahmedabad city dataset In comparison to previous studies**[1]** it gives better results by 95 meters. It also considers the comparison between Genetic algorithm without clustering and Genetic algorithm with clustering, where 129 km are required to visit all the nodes of a city. It is concluded that real GA approach is robust, represents an efficient search method and is easily applied to dynamic and complex systems of the well-known TSP in the field of solid waste routing system in the large cities but with the help of k means clustering results as well as computation time can be improved.

Introduction

Looking at the contemporary scenario of the current waste management system, we see various issues and flaws in the system which are cost inefficient, route optimization, and it fails to utilize resources efficiently, manage collected garbage. Integrating current technologies into existing systems helps to achieve optimization in every phase. Utilizing technology may improve the process of deducing the best-fit route required to optimize waste collection procedures.In this research paper, try to find a solution to one of the issues of route optimization for trucks to reduce cost and save time.This study aims to optimize the waste collection process by minimizing the total traveled distances (cost), Ahmedabad city has six different zones: center, north, east, south, west and new west. The route is found with a combination of both cluster and genetic algorithm for each zone. First, filled garbage bins are clustered according to the number of trucks available and capacity of the truck by applying the k-means cluster algorithm for specific zones. The genetic algorithm is further applied in the resultant clusters to find the best optimized route for the truck (about genetic algorithm). Genetic algorithm utilizes available genetic operators to derive a solution to complex problems. A growing population results in increased waste, which in turn increases the requirement for additional trash collectors. Nevertheless, their job becomes more difficult since a rising population creates a spike in waste collection points. However, establishing route optimization techniques can help in the management of waste collection[3].

2. Literature review

In [1] Author applies various heuristic, enhanced heuristic, genetic algorithms to solve the route optimization for waste management systems where Municipality has a truck for different zones. According to this study genetic algorithms with consideration of TSP and VRP give an efficient Solution. Where he gets the optimal results in case of genetic algorithm 6585 meters to tral the 16 nodes of Ibrid city .In[2]Author uses VRP with GA to find the route. The genetic algorithm was chosen because of its reliability when compared with the rest of the metaheuristic algorithms that could be used to optimize a VRP for any dynamic route setup. In[3] Author applies a Heuristic algorithm to find the optimized location of Depot so that we can reduce the Distance traveled during waste collection.In[4] As Vehicle routing algorithms are NP hard problems to find optimal solutions, it does not work for large scale datasets. After dividing the big problem into several sub-problems, VRP gives a better optimized route, and it can be done with the help of a balanced k-means algorithm.

3. Waste collection process flow:

General flow of the system starts with the insertion of waste into the IOT based smart dustbins and then according to the priority of the dustbins it starts checking whether it has reached the required level for the collection process . And dustbins filled up to required level will send their details like location, temperature, humidity, weight etc. to the server. Other Dustbin will be put on hold until new waste enters into them . And then the route optimization algorithm will generate the optimized path for each Truck and that path will be sent to the truck's GPS system and the truck will follow the path, collect the Dustbins and Dump the waste at the Dump yard. After this, the Flow starts again from the initial stage.

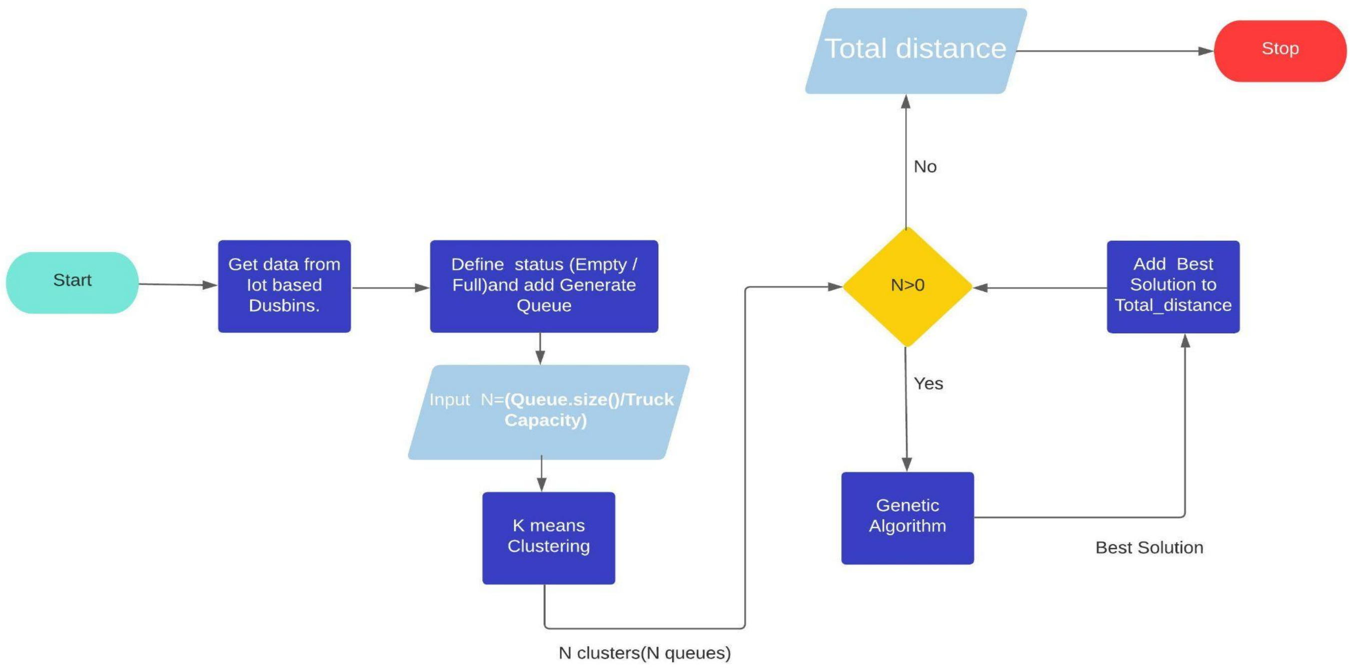


Fig. 3.0

4. Implementations:

4.1. Dataset Generation

Here In this paper , Ahmedabad (India) city is being considered for a smart waste collection system. Collection of accurate data like city maps , sectors , trucks availability , service staff availability , population, and waste collection by each sector is very hard as it needs to deal with lots of government Offices and service branches. So In this paper the Data provided by 'MINISTRY OF

HOUSING AND URBAN AFFAIRS' by government of India for 2018-2019 has been considered. According to this, Ahmedabad is divided into 47 different sectors and 5 basic zones (NORTH, SOUTH, EAST, WEST, CENTRAL). And Household waste collection for each sector has been described for the year 2018-2019. Coordinates for each sector have been added to the data using 'MAPQUEST' geocoding API. These Coordinates are further used to find the distance between two city points with the help of Mathematical terminologies between two Geo coordinates.

4.2. Considerations

For Better utilization of the resources, we first need to consider some criteria according to the availability of the resources at Municipal corporations. Some of them which are considered in this paper are No of Trucks available, Average No of Trash can get filled every day, Maximum possible time for 1 complete routing of truck, Capacity of Truck (How much waste it can contain).

- If a dustbin is filled above R% then it is deemed as full, otherwise it is considered as empty.
- In one phase we are going to route all the nodes in one turn only.
- 'W' is the capacity of the Truck, which means the truck can take waste from maximum 'W' dustbins.
- Number of requests came is Q.
- Number of Trucks allocated to complete the one whole phase = ceiling(Q/W)
- Truck's Average speed is V km/hr.
- Sufficient number of trucks are available.

4.3. Clustering

In this study, There is a Q node to travel and the capacity of each truck is W dustbins, so each truck can take the waste from at most W nodes. Q nodes should be divided in such a way that each truck gets allocated at most W nodes to visit by minimizing the number of trucks required. That is achieved using K Means clustering algorithm here.

K-means clustering algorithm computes the centroids and iterates until it finds the optimal centroid. The number of clusters identified from data by algorithm is represented by 'K' in K-means.
This algorithm minimizes the sum of distances between the data point and their corresponding clusters, which is the base requirement of grouping of nodes here, As the distance traveled by each truck can be minimized if the distance between the nodes of a cluster is minimized.
Clustering helps in minimizing iteration and computation power to find optimized solutions than by using only genetic algorithms. It is feasible to first cluster the nodes and run GA on each cluster instead of using only GA.

4.4. Genetic Algorithm

Genetic algorithms belong to Evolutionary Algorithms, which are generally used for optimization problems to minimize or maximize some parameters. It is a technique which reduces the search space for getting optimal results using Natural selection procedures and genetics. Initially the known possible set of solutions is given to the algorithm These solutions then undergo recombination and mutation (like in natural genetics), producing new children, and the process is repeated over various generations. Each individual (or candidate solution) is assigned a fitness value (based on its objective function value) and the fitter individuals are given a higher chance to mate and yield more "fitter" individuals. This is in line with the Darwinian Theory of "Survival of the Fittest". **Genetic algorithms simulate the process of natural selection,** which means those species who can adapt to changes in their environment are able to survive and reproduce and go to the next generation. In simple words, they simulate "survival of the fittest" among individuals of consecutive generations for solving a problem. **Each generation consists of a population of individuals,** and each individual represents a point in search space and possible solution.

Input parameters :

- N_population (size of population)
- mutation _rate
- Truck's capacity
- No of epochs
- Truck speed
- Diesel price

Output parameters:

- Iteration number to get best solution
- Min distance
- Order to travel such that it will take minimum distance

Five phases are considered in a genetic algorithm.

1. Initial population
2. Fitness function
3. Selection
4. Crossover
5. Mutation

4.4.1. Initial Population

The process begins with a set of individuals, which is called a **Population**. Each individual is a solution to the problem you want to solve.

An individual is characterized by a set of parameters (variables) known as **Genes**. Genes are joined into a list to form a **Chromosome** (solution). Here we represent a solution as a list of nodes which represent the order to visit the nodes. Let's say city_list is a list of N nodes that represent that solution as visiting the nodes in city_list[0]->city_list[1]->city_list[2]...........->city_list[n-1] order. Multiple initial solutions can be achieved using random shuffling of the inserted request queue.

Population_set[]=set of randomly suffered solutions

4.4.2. Fitness Calculation

The **fitness function** determines how fit an individual is (the ability of an individual to compete with other individuals). It gives a **fitness score** to each individual. The probability that an individual will be selected for reproduction is based on its fitness score. Total distance covered from starting node to

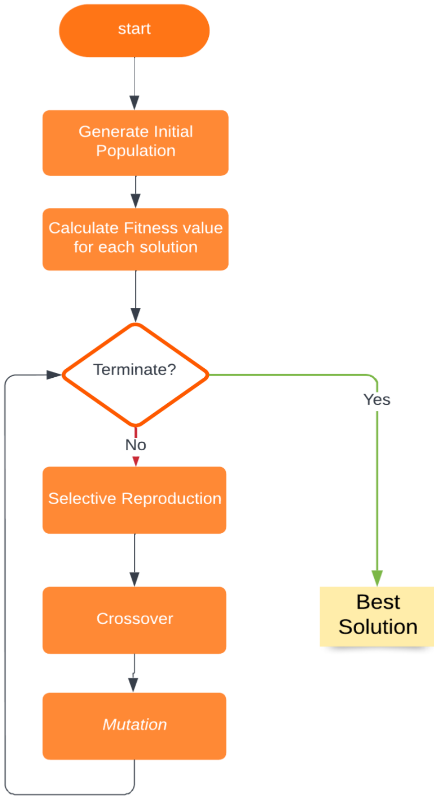


Fig.4.4.2

A dump yard has been considered as the fitness value of the particular solution(Chromosome) . Total distance from S to E is calculated as

Total_distance = d[S][city_list[0]]+d[E][city_list[N]]+**Σ**d[city_list[i]][city_list[i+1]]

where i∈(0,N-1)

Where d = distance matrix

city_list[]=A Solution(chromosome)

N= number of nodes in request queue (number of nodes to be visited)

d[S][city_list[0] = distance of the starting node of the solution from S(initial location of truck).

d[E][city_list[N]] = distance of the ending node of the solution from E(dump yard).

The solution having the least distance will be considered as a more fitter solution and the chance of surviving it is more.

4.4.3. Progenitors selection

The idea of the selection phase is to select the fittest individuals and let them pass their genes to the next generation.Two pairs of individuals (parents) are selected based on their fitness scores. Individuals with high fitness have more chances to be selected for reproduction. Based on the previously computed fitness, the progenitors are selected. The higher fitness an individual solution has, the higher the probabilities it has to mate and produce offspring (descendants). We simply need to select a set of progenitors based on their mating probability. The selection process is executed as many times as necessary until we obtain enough progenitors to produce N kids to replace the original set of N solutions. It should be noticed that each progenitor can mate more than once and with different partners. To select the parents, there are several strategies (sorted from most to least common):

- **Roulette Wheel Selection**

  The roulette wheel is a selection method based on fitness probabilities. So, to do that you need to add up all the fitness metrics for each of the solutions and give each solution a probability of being chosen so that probability is individual_fitness/total_fitness.

total_fit=**Σ**fitness[i] ∀ i∈(0,N-1)

probability[i]=fitness[i]/total_fit

progenitor_list_a=random selection using probability of each chromosome (parent list-1)

progenitor_list_b=random selection using probability of each chromosome (parent list-2)

Progenitor_list[0] =progenitor_list_a

progenitor_list[1]=progenitor_list_a

4.4.4. Mating or reproduction

Two different solutions are expected to mate. For simplicity here, it is assumed that reproduction is done between two solutions, although there seems to be evidence that more progenitors increase the performance of the algorithm. The mating consists of merging the two solutions into one, keeping bits of each of the parents. The mating can be done by exchanging fixed sections of the solution, but also selecting random bits of each parent. In either case, we need to make sure that the solution is still consistent. For example, if no repeats are allowed, we need to check that no repeats exist in the offspring, otherwise, that needs to be corrected, so the solution satisfies the problem constraints.

i∈(0,N-1)

Offspring = mate(progenitor_list[0][i],progenitor_list[0][i])

Mate method crossovers the node in between two list

new_popiulation.add(offspring)

4.4.5. Mutating

The mutation step is important to prevent that our algorithm gets stuck at the local optima. The local optimum is a solution that seems the best if we look at nearby solutions, but it's not the best possible solution. The best possible solution or global optima may be behind a dip and therefore we need randomness to jump across hills. The mutation step generates this necessary randomness. Depending on the selected randomness, the mutation step changes different parts of the solution arbitrarily. If it's too little we get stuck at the local optima, if it's too much it will break the best solutions preventing them from consolidating and keep improving the general population fitness. There should be noted that some implementations do have inclusion criteria for the offspring. Some applications have a filter that prevents terrible offspring from being added to the set of solutions. Other solutions include mechanisms to prevent that the generated offspring are too similar for the sake of variation. The higher the variation the higher the chances of achieving a better solution, otherwise it can converge to all the solutions being the same.

Number of nodes to be swapped in offspring p= mutation_rate * number of nodes

$\forall \;\; i \in (0,p-1)$

T1,T2= two different randomly selected node from offspring

swap(T1,T2)

mutated_pop.add(New_offspring)

mutated_pop is a newly generated population set after mutation operation on each offspring of given population set

4.4.6. Stopping criteria

If the stopping criteria are met, then we stop the execution. Otherwise, we go to step two and repeat the whole process again. Since the best solution is unknown, a set of rules can be defined in order to stop the computation.

- There is a solution that satisfies the minimum criteria. That means that there is a solution that has a fitness equal or better than we expected to be satisfied.
- We already performed too many iterations. The algorithm has looped enough times, so we assume that nothing better can be found.
- We spent the budget. The computation/time/money has been used and there are no more resources left to continue iterating.
- The solution has plateaued. The solution does not seem to improve and maybe it is not worsening either. The algorithm seems unable to find something better.
- We think that the solution is good enough. After checking the results manually, we can decide that we're satisfied with the results and decide to stop the experiment.

- A combination of all. We can merge all of them and find the best stopping rule for our taste.

best_solution=min(best_solution,min(fitness_list))

4.4.7. Cost Evaluation:

In real life scenarios several costs like truck maintenance , Human resource required , Computation cost , Server cost , Application management cost, cost of petrol need to be considered . For simplicity here, Total distance is considered as the main cost factor.

Cost[i]=(distance traveled by I th truck / Average kilometers it travels per liter diesel)*diesel price.

$$\text{Total cost}=\Sigma \text{cost}[i] \quad \forall \; i\in(0,K)$$

where K= number of clusters

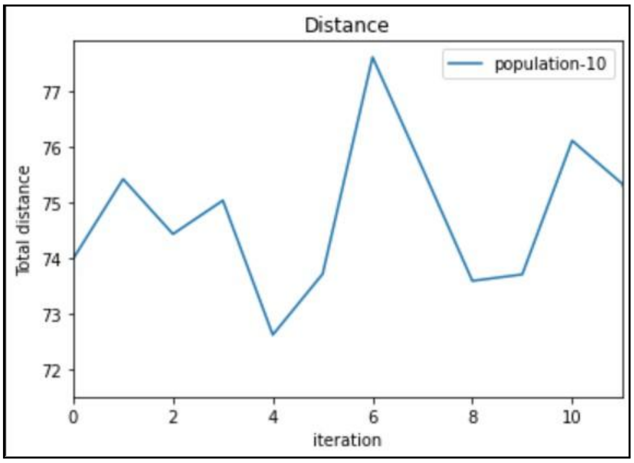
5. Experimental Analysis

As mentioned earlier, the research study is implemented on Ahmedabad city dataset, As part of the research study various experimental analyses were carried out to find the best way to achieve an optimized solution. How change in parameter will impact on end results were analysis like change in no population, mutation rate, capacity, number of iteration during the process.

Consider Scenario-1:

The generated request queue of 15 size, mutation rate of 0.3 and capacity of the truck takes 5 , 1000 epochs. These parameters are considered constant and analyze the change in population to total distance.

The result shows that as population size increases, the line of the distance tends to approach a plateau and after some specific number of population sizes only optimized solutions will be achieved. The analysis is further carried out for the same mutation rate, epoch and choose the best values for the algorithm



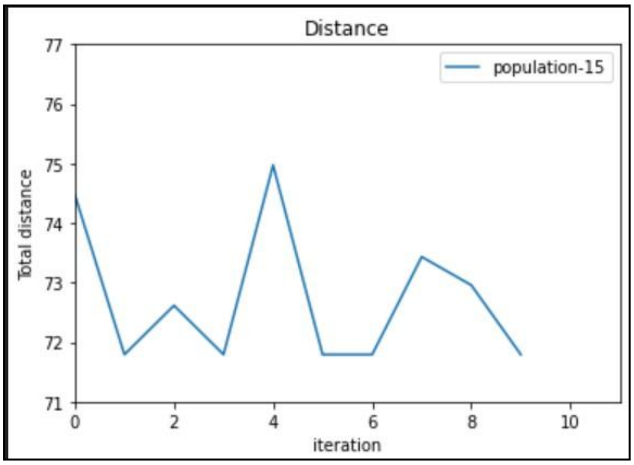


Fig. 5.1.1

Fig. 5.1.2

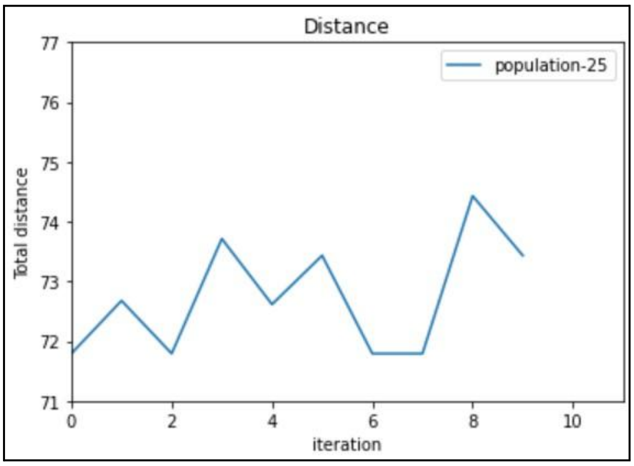


Fig. 5.1.3

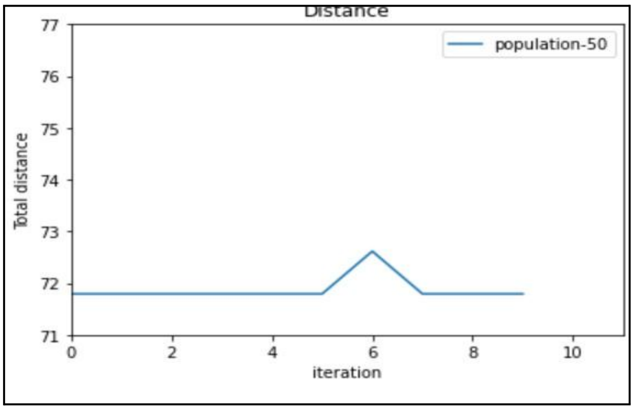


Fig. 5.1.4

Consider-Scenario-2:

Second scenario is considered as the request queue is filled with all requests from the city. With the capacity of 9 dustbins, mutation rate of 0.3 and population size of 25, these 47 towns are grouped into 6 clusters with help of K-Means algorithm and genetic algorithm runs on each cluster and finds the route as below shown in the generated map.

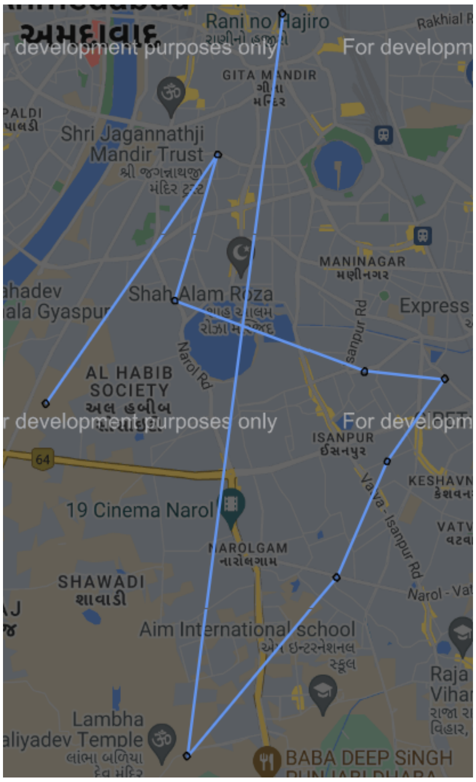


Fig. 5.2.1

Cluster1

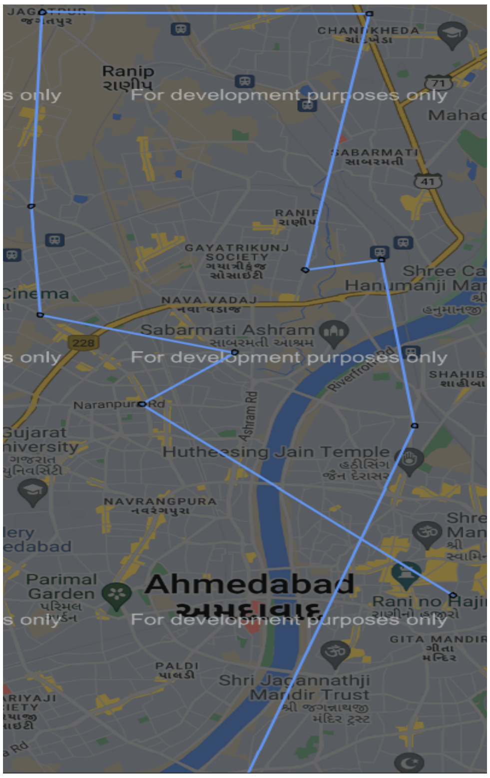


Fig. 5.2.2

Cluster 2

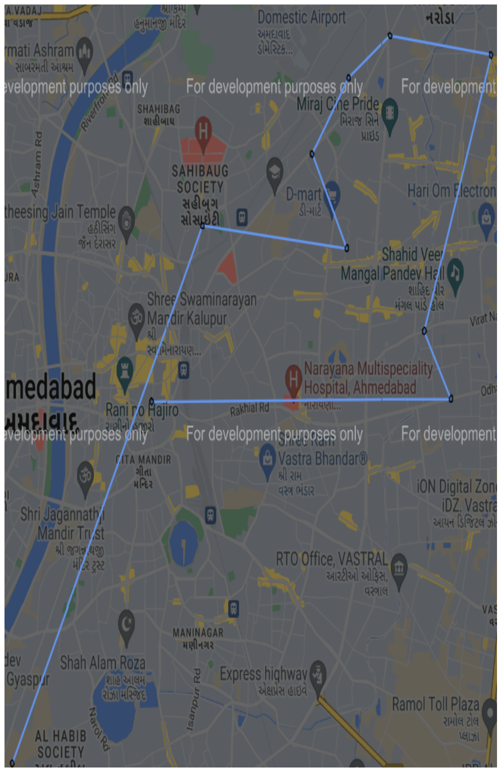
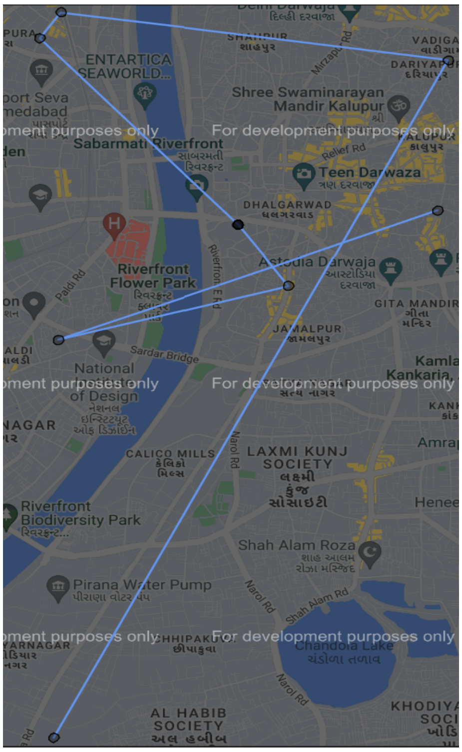
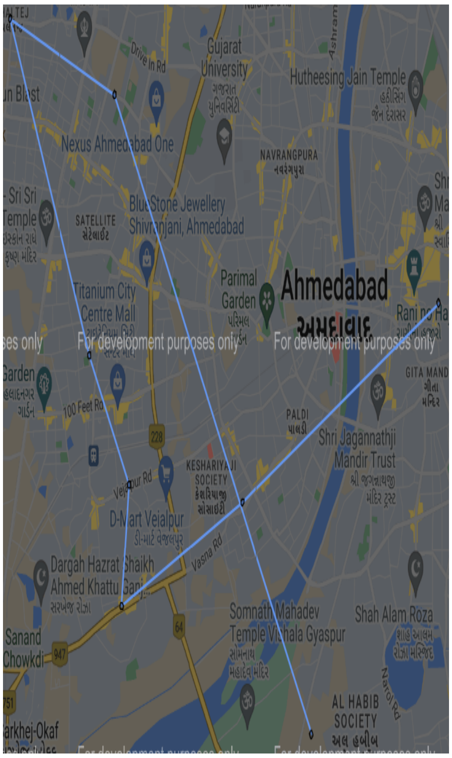
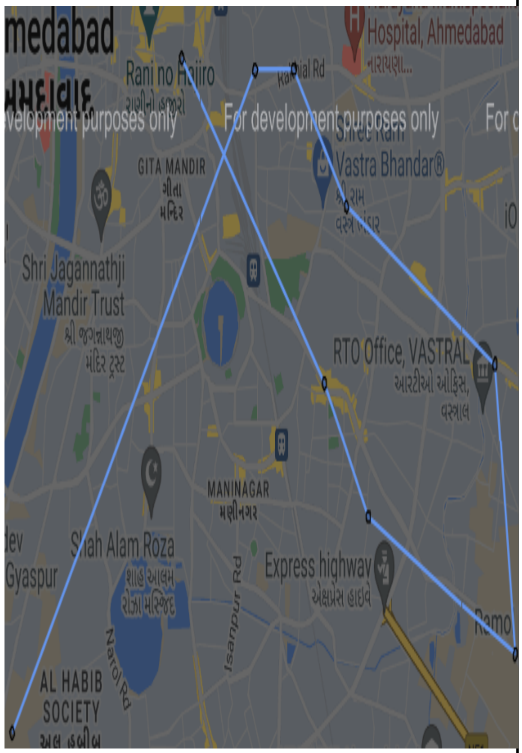


Fig. 5.2.3

Cluster 3

[33, 34, 43, 35, 10, 45, 44] Distance = 17.352 kms Epoch : 144 Expected Travel Time : 52 minutes	[15, 3, 4, 2, 0, 1, 6, 5, 8] Distance:31.937 Epoch : 946 Expected Travel Time :96 minutes	[14, 21, 11, 12, 13, 23, 24, 38] Distance : 26.57 Epoch : 979 Expected Travel Time : 80 minutes
 Fig. 5.2.4 Cluster 4 [20, 9, 17, 16, 40, 25, 22, 28, 29] Distance:12.852 Epoch : 522 Expected Travel time : 39 minutes	 Fig. 5.2.5 Cluster 5 [18, 7, 19, 31, 32, 30] Distance : 23.352 Epoch : 0 Expected Travel time :70 minutes	 Fig. 5.2.6 Cluster 6 [36, 26, 37, 39, 46, 41, 42] Distance : 17.865 Epoch : 10 Expected Travel time : 54 minutes

Total distance traveled by all trucks to visit all the city points is 129.9347 kilometers which costs less or equal to 30-35 liters of Diesel price. Truck 2 is taking the highest time (96 minutes) to complete the trip. From these results it stated that Ahmedabad Municipal corporation can have the ability to restart the new collection phase after 120 minutes, by this it is possible to cover the whole city more than 5 times in a day which will be advantageous during festival seasons.Hence Overall analysis stated that AMC can cover all the city points in less than 2 hours using only 9 trucks by spending 35 liters of Diesel.

6. Comparative analysis:

For the validation purpose , The Genetic algorithm implemented here is applied on the dataset used by previous study [1] . Where the author gives the route optimization algorithm for the smart waste management system of Irbid City, Syria. In [1] 16 city points have been taken as dustbin nodes and A single truck will travel all 16 nodes considering 1 as start and 16 as ending point. Best solution for total distance taken in [1] is 6585 meters after 2060 simulation, Which is improved by applying genetic algorithms defined in this paper by 95 meters by giving the total distance required to visit all

16 nodes as 6490 meters. The Smallest Distance between any two nodes has been achieved using the Floyd Warshall algorithm. Improved Order of visiting nodes is

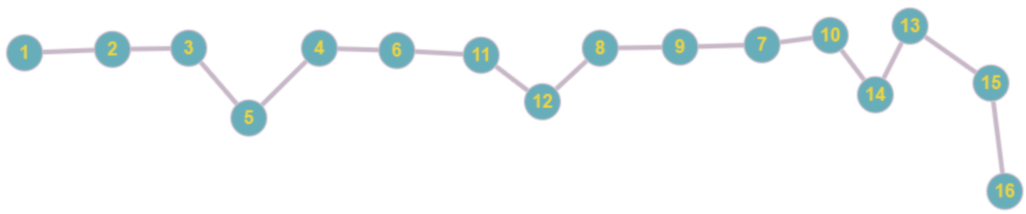


Fig. 6

7. Conclusion:

In this paper, we have presented our approach to optimize the route for the waste collection process to reduce expenses on the waste collection process.We suggested Iot based experimental approach by which AMC have server and dustbins are well set with wireless sensors such that AMC will priorly get the information about the status of the dustbins and by using that information using k means clustering and Genetic algorithm can reduce the waste collection trip distance significantly which lead to increase in substantial savings significantly.Accordingly, it is concluded that GA approach is robust, represents an efficient search method and is easily applied to dynamic and complex system of the well-known TSP in the field of solid waste routing system in the large cities.

References

1. Ingo von Poser and Ingenieurtechnik, Merck KGaA "Optimal Routing for Solid Waste Collection in Cities by Using Real Genetic Algorithm", 2020.
2. Narayanaswamy V. Ramu I and William J. Kennedy 2 "Heuristic algorithm to locate solid waste disposal site", 1994.
3. Zainab Adnan Jwad and Saad Talib Hasson "An Optimization Approach for Waste Collection Routes Based on GIS in Hillah-Iraq", 2018
4. Abdullah Alwabli and Ivica Kostanic "Dynamic Route Optimization For Waste Collection Using Genetic Algorithm", 2020
5. Nikolaos Baras, Dimitris Ziouzios, Minas Dasygenis, Constantinos Tsanaktsidis University of Western Macedonia "Waste collection vehicle navigation in modern cities", on June 02, 2022 .
6. Amirhossein Malakahmad and Noor Diana Khalil Civil Engineering Department, Universiti Teknologi PETRONAS "Solid Waste Collection System In Ipoh City", 2011.
7. Willapa Wongsinlatam Faculty of Applied Science and Engineering, Khon Kaen University, Nong Khai Campus "Finding the Shortest Route of Waste Collection by Metaheuristic Algorithms using SCILAB", 2020