

# Genetic Algorithm to Optimize Solid Waste Collection

**Kabil Bhargava, Rajiv Gupta, Anupam Singhal, Ajay Shrinivas**

Department of Civil Engineering, Birla Institute of Technology & Science  
Pilani, Rajasthan, India  
rajiv@pilani.bits-pilani.ac.in

**Abstract** – Urban solid waste management has become a critical issue due to the growth of population as well as an upsurge in living standards with inefficient management leading to negative socio-economic and environmental impacts. Solid waste management includes segregation at site, storage and waste characterization, collection, transfer and transportation, processing and recovery and eventually disposal of waste. Among all the phases, it has been seen that 60-80% of the total capital is spent on the collection phase and hence, optimizing this phase is crucial as we move toward an era of Smart cities and Internet of Things. In the paper, Genetic Algorithm (GA) is employed to determine cost-effective routes for solid waste collection in order to reduce distance travelled by waste collection trucks (in turn reducing fuel consumption and CO<sub>2</sub> emissions), taking into account the collection capacity of trucks, waste generated in bins, traffic conditions and time constraints. The results indicate that GA can be used effectively for allocating least distance path while incorporating all factors effectively.

**Keywords:** Waste Management, Genetic Algorithm, Urban Waste, Waste Collection.

## 1. Introduction

Solid waste is mainly generated by human activities. Solid waste management (SWM) is associated with the control of generation, source separation, storage, collection, transfer and transportation, processing and disposal of waste according to the principles of public health, economic and other environment factors (von Poser and Awad, 2006). The overall objective of SWM is to minimize the adverse environmental effects caused by indiscriminate disposal of solid wastes. Here, we specifically look at the one of the most important phase, i.e. collection.

The rapid growth in urbanization and that of population have made the situation alarming with respect to SWM (Koushki et al., 1997). Solid waste collection (SWC) has become an important issue in the larger cities of India. At a time when the country look to move toward Smart Cities, optimized plan for garbage collection is an important area of study and is addressed here.

The optimized routing for municipal SWC is important as it is estimated that, of the total amount of money spent for the collection, transportation, and disposal of solid waste, approximately 60-80% is spent on the collection phase (Karadimas et al., 2007; Municipality Of Athens, 2003). Hence, reducing cost in the collection phase could bring down the SWM cost drastically and would lead to lower CO<sub>2</sub> emissions, traffic congestion and travel time. This would allow municipalities to provide better services. Inadequate and improper disposal of waste leads to visual pollution, bad odor and serious hazard to people's health (Melo et al., 2007).

The algorithm developed optimizes the collection route in a way that the minimum path is followed cumulatively by the trucks (containers), balancing the garbage quantity as per capacity. This takes into consideration that no truck moves to an empty trash bin and the truck is sent to a route in which the sum of the wastes in the bins do not exceed the capacity of the truck (it will first go to the nearest recycling station and then proceed to the next point), thus optimizing the number of trucks used in a given area.

It is assumed that the city's solid waste quantities to be picked up on the particular day are known before the trucks are sent out. The collection could be further optimized if infrastructures of sensors capture the weight of garbage bin in real time. Furthermore, the bin areas can be allocated based on the population density of a region, and the tentative time of deposition of waste in the bin. Since the actual wastes in each of the bins would be more than the wastes collected before trucks are sent out, real time monitoring helps improve the efficiency of waste collection and ensures that no truck goes to a bin whose wastes exceed the truck capacity.

The algorithm aims to minimize the total cumulative distance travelled by the trucks while balancing the workload among trucks, with subject to the capacity of collection truck and the collection requirement for each street. Time and traffic are also considered as factors in the algorithm and helps convert the actual distance into effective distance.

## **2. Previous Work & Genetic Algorithm**

Genetic Algorithm has been used to define efficient routes in order to reduce the fuel consumption of garbage-trucks taking into account parameters related to traffic and trash bin positions. They considered that the city has an infrastructure with sensors spread, such as sensors that capture the weight of the garbage bin and send them to the servers where the optimization algorithm is running. In addition, the system also needs to receive up-to-date traffic data within the city and this is important in defining the routes, as the algorithm considers the traffic in the streets. Each Chromosome has  $N$  bits;  $N$  is the number of dumpsters to be collected. Poser and Awad (2006) represent the chromosome as a permutation of bin nodes and determine the least distance. The chromosomes with worst, average and best fitness in the population are determined, and the gradual convergence of the GA to the best fit solution with an increase in number of iterations is observed.

Ahn and Ramakrishna (2002) use variable-length chromosomes (strings) and their genes (parameters) for encoding the problem. The crossover operation exchanges partial chromosomes (partial routes) at positionally independent crossing sites and the mutation operation maintains the genetic diversity of the population. The proposed algorithm can cure all the infeasible chromosomes with a simple repair function.

Karadimas et al. (2007) employ GIS to account for all the required parameters for solid waste collection like static and dynamic data, such as the positions of waste bins, the road network and its related traffic, as well as the population density in the area under study. In addition, waste collection schedules, truck capacities and their characteristics are also taken into consideration. But, the problem is still considered as a Single Vehicle Origin Routing trip.

Hsieh and You (2014) adopt the Immune based approach for the SWC problem, where the workload balance among vehicles, collection requirements per street as well as frequency of collection is considered.

The current work accounts for the number of vehicles, the different permutations of vehicle trips, the actual capacity of waste that is collected from the bin, truck capacity and the time and traffic related factors. Also, recycling stations are considered at various places to renew truck capacity when it is unable to load the next bin waste. Although the municipal waste collection has already been optimized through programming and mathematical methods, the results were not satisfactory due to the large amount of variables that resulted in a great computational time. Mathematical programming approaches have had rather limited success with this problem. According to Thieraut and Klekamp (1975), for a 20-node problem, integer linear programming requires 8000 variables and 440 constraints while dynamic programming is limited to 13-node problems. Heuristic algorithms for solving the travelling salesman problem (TSP) are usually very fast and not limited to a definite number of nodes. Evolutionary algorithms like GA can be used to solve such problems as constraints mentioned are not large in number.

Evolutionary algorithms (EA) are a promising alternative to problems where adaptively, complex computations, innovation and large scale parallelism are required. GA is a heuristic search algorithm belonging to the class of evolutionary algorithms. Search method is based on natural selection and transformation. The algorithm mimics Darwin's theory of evolution by means of natural selection, or survival of the fittest.

GAs use probabilistic transition rules during iterations, while the traditional methods that use predefined or fixed transition rules. This makes them more robust, random and applicable to a large range of problems. Furthermore, GA only uses the information about the objective function. They do not require knowledge about the problem domain or any other auxiliary information, allowing a number of problems to be solved without the need to inculcate restrictive assumptions.

To solve SWC problems of realistic size, a heuristic approach is a good option. The genetic algorithm is a highly parallel, random and self-adaptive search algorithm based on the mechanism of the natural selection and genetics. The algorithm has proven to be effective and has been widely used to solve TSP (Potvin, 1999; Sallabi and Haddad, 2009) and Multiple travelling salesman problem (Carter and Ragsdale, 2006; Zhou and Li, 2010; Li et al., 2013). Major

advantage of this algorithm is when it converges to a solution it is able to find another set of possible solutions that the destination can be reached.

### **3. Methodology**

Genetic Algorithm is a heuristic search algorithm belonging to the class of EA. EAs is a promising alternative for problems where adaptivity to problems, complex computations, innovations and large scale parallelism is required. It is a Search method based on Natural Selection and Transformation.

GA begins with encoding all the possible solutions into finite length array of bits. These strings are called chromosomes and represent the potential solution to problem in the form of genes. These chromosomes form the initial population for the problem and are tested for quality/fitness. Fitness is a measure of how good that probable solution is. After calculating fitness, selection is done in such a way that chromosomes with greater fitness value have greater probability of getting selected. Then crossover and mutation operations are performed on these selected chromosomes and a new population is formed by replacing the bad solutions with the good ones. This process is repeated until a satisfactory fitness value is reached or in some cases other conditions bound us to terminate the process. The control parameters of GA are population size, crossover probability (normally consider around 0.5), stopping criteria and mutation probability (should be very low around 0.05).

#### **3.1. Objective**

The considered SWC routing problem aims to minimize the total distance travelled by collection vehicles and also balance the workload among vehicles with subject to the capacity of collection vehicle and the collection requirement for each bin. Traffic and time variables are multiplied by factors corresponding to heavy, moderate and light traffic. These traffic factors are also dependent on time which have been divided as working and non-working hours and night time.

#### **3.2 Problem Constraints**

Each vehicle returns to starting point at the end of each collection trip. Each vehicle (truck) travels to a unique set of bins are in between. Each bin is visited by exactly one vehicle. If the road network graph is not perfect, the lack of any direct road between two bins is replaced with an imaginary road (edge) that has a very large distance and consequently would have to travel large distance for the collection trip, making it less fit and having less chance of getting selected for further iterations.

The vehicle goes to next bin only if it has sufficient capacity left equal or more than that of the bin collecting capacity. The quantity of waste collected is updated and stored once it goes to the bin and collects the amount of waste. This way we will be able to incorporate the changes in bin quantity during the routing process.

The vehicle cannot interrupt the collection of waste on a street, that is, the vehicle will not go to the next bin if the carrying capacity is not enough. For such a case, the vehicle has to first visit the recycling station and empties the waste before collecting the waste on the street even though it does not reach the carrying capacity limit.

#### **3.3. Chromosome Representation**

Chromosome representation is a crucial step when applying GA since it directly determines the performance of the algorithm. The chromosome is used to represent probable solution. In this case the chromosome was defined to represent the dustbin locations and the nodes, where each truck has to travel in the collection region, in a single chromosome itself. Each chromosome has  $M+N$  bits;  $M$  is the number of bins and  $N$  is the number of vehicles available on a particular day. To respect the restrictions that you cannot have repeated bin positions in the chromosome- the first  $N$  bits will have a permutation (indicating order) of numbers from 1 to  $M$  with none being repeated and the next  $N$  bits will have permutations from 1 to  $M$  where the summation of these bits will be equal to  $M$ - so that all the trucks combined will visit the  $M$  dustbin positions. Hence, the first  $M$  bits will represent the path to be followed for collection and the next  $N$  bits would break the first  $N$  bits into  $N$  parts- each part representing a path for a particular truck (Sub-tour).

In Figure 1, we have 16 bins and 4 vehicles, so we need to have a chromosome of 20 bits where first 16 will have bin permutation and the next 4 will indicate vehicle going to respective bins. Here, vehicle 1 (with no. 6) goes to 6 bins namely 8, 9, 11, 12, 4 and 1 (in that order), vehicle 2 (with no. 2) goes to bins 3 and 10 (in that order), vehicle 3 (with no. 5) goes to bins 5, 16, 2, 6 and 14 (in that order) and vehicle 4 (with no. 3) goes to bins 15, 13 and 7(in that order).

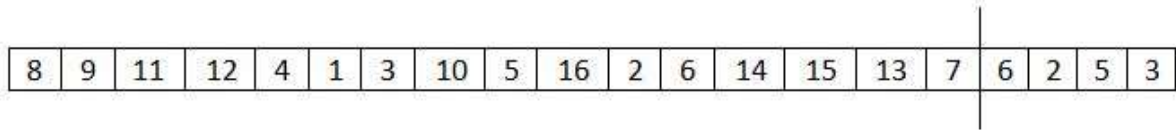


Fig. 1: Chromosome representation, where first M (= 4) bits is the permutation of bin pickups and following N( = 16) bits represent the sub-tours for vehicle.

### 3.4. Initialization

After the representation of chromosomes in a population, create a number of feasible solutions (chromosomes) which are randomly in nature. In general, there are two issues to be considered for population initialization of GAs: the initial population size and the procedure to initialize the population. Size of initial population plays an important role in the convergence rate of GA. Secondly, there are two ways to generate the initial population: heuristic initialization and random initialization. Although the mean fitness of the heuristic initialization is already high so that it may help the GAs to find solutions faster, it may just explore a small part of the solution space and never find global optimal solutions because of the lack of diversity in the population. Therefore, random initialization is effected in this paper so that the initial population is generated with the encoding method already explained.

$$x = \begin{bmatrix} 6 & 14 & 1 & 4 & 5 & 13 & 3 & 16 & 7 & 15 & 8 & 12 & 9 & 11 & 2 & 10 \\ 10 & 2 & 4 & 7 & 15 & 9 & 11 & 12 & 16 & 1 & 5 & 8 & 14 & 6 & 3 & 13 \\ 3 & 11 & 10 & 7 & 16 & 6 & 9 & 8 & 4 & 1 & 2 & 14 & 12 & 13 & 15 & 5 \\ 10 & 5 & 11 & 12 & 4 & 7 & 1 & 9 & 15 & 3 & 16 & 13 & 14 & 6 & 2 & 8 \\ 3 & 5 & 11 & 7 & 4 & 8 & 2 & 16 & 12 & 10 & 9 & 13 & 14 & 1 & 15 & 6 \\ 5 & 11 & 3 & 1 & 8 & 10 & 14 & 6 & 12 & 9 & 7 & 15 & 16 & 2 & 13 & 4 \end{bmatrix}$$

Fig. 2: Sample population of Size 6 showing the permutation of 16 bin positions.

### 3.5. Fitness

Fitness is the important criteria in determining the chance that an individual chromosome can be selected to the next generation and to keep the solution converging in the right direction. Fitness directly related to the quality of the chromosome (potential solution). For a GA, many selection strategies based on the proportion of fitness require a non-negative fitness and the larger fitness the better individual.

Since, algorithm is to find the minimum distance for the collection problem, so fitness is inversely proportional to the distance travelled and hence, we take the fitness function as  $1/(\text{Sum of distance of all sub-tours})$ .

Here, the sum of distance will be between various bin positions, provided the vehicle capacity is available. When enough capacity is not available, the distance will be from one bin location to the nearest recycling station and then to the next bin, that is-

If  $d(i,j)$ : distance between node  $i$  and node  $j$

and  $d(i,t)$ : distance from node  $i$  to nearest recycle station

and  $d(t,j)$ : distance from node recycle station to node  $j$

then

$d(i,j) = d(i,j)$  if vehicle has enough capacity to collect waste at node  $j$ .

$d(i,j) = d(i,t) + d(t,j)$  if the truck does not have enough capacity to collect waste at node  $j$ .

Then, we also include time, traffic and road condition factors so as to convert the actual distance to effective distance, which give more realistic representation of how much time is actually required to cover the given distance.

Example, let us say that the traffic factor at 8:00am in the morning is 1.4 between nodes 5 and 8 and the actual distance between nodes 5 and 8 is 4km. Also, let the road condition factor is 1.2 between the nodes due to bad condition. Hence, the effective distance between nodes 5 and 8 becomes  $1.4 \times 1.2 \times 4 = 6.72\text{km}$ .

These traffic, time and road factors could be obtained based on time-history data for the road network or dynamic data obtained through Geographical Information System or other Internet of Things related network.

### 3.6. Selection

The selection operator is used to improve the average fitness (quality) of the population by providing the fitter chromosomes a better chance to get selected into the next generation, that is, the selection function is used to explore those regions of the solution space which are more promising to the solution.

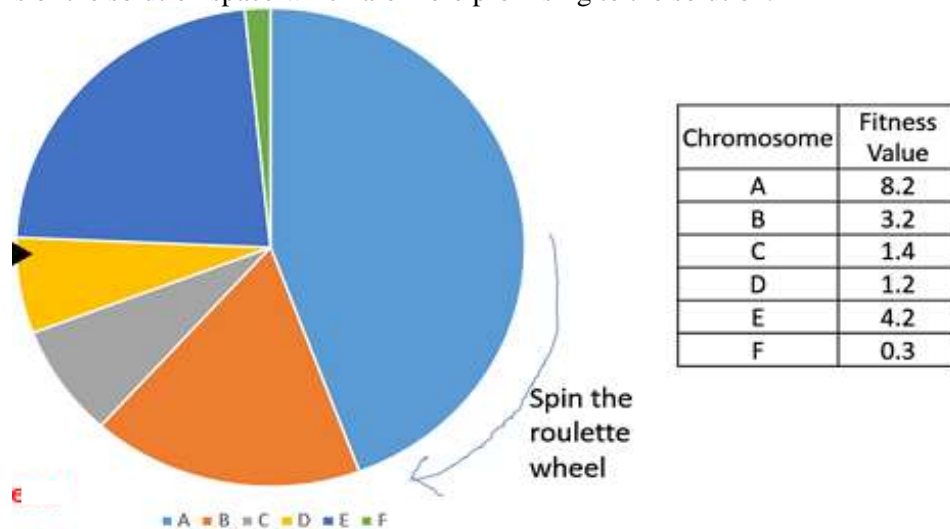


Fig. 3: Roulette Wheel Selection Method.

Here, we have used Roulette wheel selection (stochastic method). In this selection, each individual in the population is assigned a roulette wheel slot sized in proportion to its fitness. Hence, good solutions have a larger slot size than less fit solutions. The roulette wheel is spun (by taking random numbers) to obtain a reproduction candidate.

### 3.7. Crossover

As GA is an algorithm that will return the best solution within the analyzed ones, contained in the population during the various interactions, it is necessary to recombine the individuals using crossing and mutation techniques. In crossover, two individuals are randomly selected and are recombined with a probability  $p_c$  (here, it is taken as 0.5). Here, position-based crossover is used. In this crossover, a subset of positions is selected in the first parent. Then, the cities found at these positions are copied to the offspring (at the same positions). The other positions are filled with the remaining cities, in the same order as in the second parent.

In the figure, position based crossover is shown where positions 2,3 and 6 are selected from parent 1 and copied to offspring 1 at the same positions. Then starting from position 1 in parent 2, the elements are copied to offspring 1 in the same order leaving aside elements at position 2, 3 and 6. Similarly for offspring 2, elements at position 2, 3 and 6 in parent 2 are copied to offspring 2 at the same positions and then remaining positions are copied from parent 1 in the same order as in parent leaving aside elements at position 2, 3 and 6.

Parent 1:	1	<u>2</u>	<u>3</u>	4	5	<u>6</u>	7	8
Parent 2:	8	<u>7</u>	<u>6</u>	5	4	<u>3</u>	2	1

---

Offspring 1:	8	2	3	7	5	6	3	1
Offspring 2:	1	7	6	2	4	3	5	8

Fig. 4: Position based crossover.

### 3.8. Mutation

Mutation is vital for improving exploitation (local search ability) and exploration (maintaining variability of the population). It is done so that the solution does not converge at the local minima by varying chromosomes so that more solution space is search and the Global optimum is reached. It also prevents the premature termination in GA. The mutation is doing only small changes with exchanging two nodes.

Here, in the mutation phase, two different functions are used. The user can select the mutation operator of his preference through the user interface. The first method simply reverses the order of visit of bins between two randomly selected positions. The second randomly selects two loading spots and swaps their positions. Through experimentation, it was discovered that that the SECOND mutation mechanism yielded better results than the FIRST one.

Chromosome:	4	<u>6</u>	1	7	3	8	<u>5</u>	2
-------------	---	----------	---	---	---	---	----------	---

---

Mutation Method 1:	4	<u>5</u>	1	7	3	8	<u>6</u>	2
Mutation Method 2:	4	6	<u>8</u>	<u>3</u>	7	<u>1</u>	<u>5</u>	2

Fig. 5: The two mutation methods.

### 3.9. Termination

Two conditions are used for terminating the genetic algorithm:

- 1) The population becomes stable, which is read as percentage change in fitness function is less than 0.1%.
- 2) Based on the number of iterations.

The code has not been included in the paper owing to length and formatting constraints. The results have instead been graphically displayed in the next section.

## 4. Results & Discussion

The Control Parameters in the proposed GA are 1) Population Size, 2) Number of Vehicles, 3) Crossover Method, 4) Crossover Probability, 5) Mutation Method, 6) Mutation Policy, 7) Fitness Function, and 8) Number of Iterations.

In order to best execute the GA and search the entire solution space, the program was run multiple times while varying all the proposed parameters suitably. The results are presented below

1. In the first case, the Population size was varied, that is, the number of chromosomes competing for the best solution was varied keeping other parameters like number of bins, number of vehicles etc. same. The objective is to see the impact of population size on the performance of Genetic Algorithm and find out the best population size. It was found that Population size 100 performed the best among sizes 50,100,150 and 200 when program run for a number of times (the average performance of these population value 100 performed the best).

2. In the second case, the number of vehicles are altered keeping the bin locations and other parameters same. This is important since we do not want to send in more vehicles than desired for collection phase as it will lead to increased traffic as well as capital costs. We took the number of bin locations as 16 and it was found that best no. of trucks is 3 when program run for a number of times (the average performance of having three trucks performed the best).

3. In the third and fourth cases, the behavior of algorithm was checked for different values of Probability for Crossover as well as Mutation. There are important parameters to maintain a balance between exploration and exploitation and should be well adjusted to have a good performance for the Genetic Algorithm. It was found that Crossover Probability of 0.2 and Mutation probability of 0.04 performed best when program run for a number of (the average performance of these two values performed the best).

4. Finally, we make use of the optimum parameters obtained from above cases and verified the convergence of Genetic Algorithm. This is important because Convergence is an important indicator for the performance of GA. If it is convergent, it means that the algorithm is evolving in the right direction and is stable.

The optimum values of each of the parameters, Population size 100, Number of trucks 3, Crossover Probability 0.2 and Mutation probability 0.04 were obtained from the above variation of parameters when the program was run multiple times. It is observed that the Average Fitness of the Population Increases with each passing Iteration indicating that the Genetic Algorithm is converging as a whole. It is seen that the Average distance decreases from 1550 in first Iteration to 1280 at the end of 50th Iteration, when program run for a number of times.

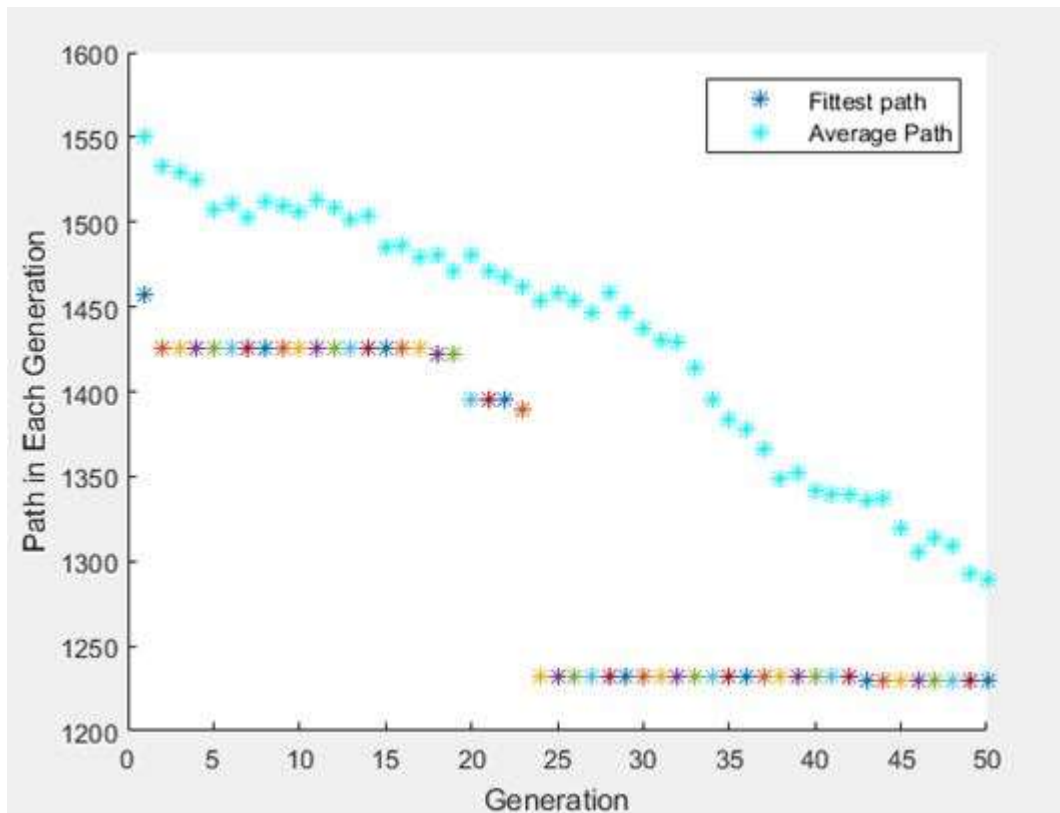


Fig. 5: Results showing an improvement in the Average population distance.

#### 4. Conclusion

In this paper, we have proposed an efficient routing and scheduling problem for Solid Waste Collection using Genetic Algorithm, considering the following factors-

- 1) Truck Capacity.
- 2) Recycling Stations to refill capacity.
- 3) Multiple vehicles(sub-routes).
- 4) Traffic and Time consideration.

The results indicate that the Genetic Algorithm applied produces remarkably low distance solutions in consecutive iterations and effectively incorporates the constraints. Hence, it can be used for other similar routing problems as well with ease.

The future course for the work can be including dynamic data with the help of sensors for bin waste and dynamic data for traffic. This can be used in developed countries but for developing countries like India, due to lack of advance infrastructure time-history data could be best used.

## References

- [1] A. B. Melo, A. M. Oliveira, D. S. d. Souza and M. J. d. Cunha, "Optimization of Garbage Collection Using Genetic Algorithm," *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, Orlando, FL, 2017, pp. 672-677.
- [2] I. von Poser and A. Awad, "Optimal Routing for Solid Waste Collection in Cities by using Real Genetic Algorithm," vol. 1, pp. 221 – 226, 2006.
- [3] R. Thieraut, and R. Klekamp, *Decision Making Through Operations Research*, John Wiley and Sons Inc., NY, USA, 1975.
- [4] A. P. Koushki, J. L. Hulsey, and E. K. Bashaw, "House Hold Solid Waste Traits and Disposal Site Selection", *J. Urban Plan. & Devel.*, ASCE, vol. 123, no. 1, pp.1-9. 1997.
- [5] C. W. Ahn and R. S. Ramakrishna, "A genetic algorithm for shortest path routing problem and the sizing of populations," in *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 6, pp. 566-579, Dec. 2002.
- [6] Karadimas, Nikolaos & Papatzelou, Katerina & Loumos, Vassili, "Genetic Algorithms for Municipal Solid Waste Collection and Routing Optimization," *International Federation for Information Processing Digital Library, Artificial Intelligence and Innovations 2007: from Theory to Applications*, vol. 247, pp. 223-232, 2007.
- [7] Municipality Of Athens, *Estimation, Evaluation and Planning Of Actions for Municipal Solid Waste Services During Olympic Games 2004*, Athens, Greece, 2003.
- [8] J. Potvin, "Genetic algorithm for the traveling salesman problem," *Annals of Operations Research*, vol. 63, no. 4, pp. 330-370, 1999.
- [9] M. Sallabi and E. Haddad, "An Improved Genetic Algorithm to Solve the Traveling Salesman Problem," *World Academy of Science, Engineering and Technology*, vol. 52, no. 3, pp. 471-474, 2009.
- [10] A. E. Carter and C. T. Ragsdale, "A new approach to solving the multiple traveling salesperson problem using genetic algorithms," *European Journal of Operational Research*, vol. 175, no. 1, pp. 246–257, 2006.
- [11] Wei Zhou, Yuanzong Li, "An Improved Genetic Algorithm for Multiple Traveling Salesman Problem," in *2nd International Asia Conference on Informatics in Control, Automation and Robotics*, Wuhan Hubei, China, pp. 493-495, 2010.
- [12] J. Li, Q. Sun, M. Zhou, and X. Dai, "A New Multiple Traveling Salesman Problem and Its Genetic Algorithm-Based Solution," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, 2013.
- [13] Y.-C. Hsieh and P.-S. You, "An Artificial Intelligence Approach for the Solid Waste Collection Problem," *Applied Mathematics & Information Sciences*, vol. 8, no. 1L, pp. 283–291, Jan. 2014.