

UDC: 004.021+004.94

A hybrid multi-objective carpool route optimization technique using genetic algorithm and A* algorithm

R. S. Beed¹, S. Sarkar^{2,a}, A. Roy³, S. D. Biswas¹, S. Biswas¹

¹ Department of Computer Sc., St. Xavier's College (Autonomous),
30 Mother Teresa Sarani Kolkata 700016 West Bengal, India

² Department of Computer Sc. & Engineering, Assam University,
Silchar, Assam 788011, India

³ Department of Computer Sc., Assam University,
Silchar, Assam 788011, India

E-mail: ^a sunitasarkar@rediffmail.com

Received 26.06.2020, after completion — 02.12.2020.

Accepted for publication 02.12.2020.

Carpooling has gained considerable importance as an effective solution for reducing pollution, mitigation of traffic and congestion on the roads, reduced demand for parking facilities, lesser energy and fuel consumption and most importantly, reduction in carbon emission, thus improving the quality of life in cities. This work presents a hybrid GA-A* algorithm to obtain optimal routes for the carpooling problem in the domain of multi-objective optimization having multiple conflicting objectives. Though the Genetic Algorithm provides optimal solutions, the A* algorithm because of its efficiency in providing the shortest route between any two points based on heuristics, enhances the optimal routes obtained using the Genetic algorithm. The refined routes obtained using the GA-A* algorithm, are further subjected to dominance test to obtain non-dominating solutions based on Pareto-Optimality. The routes obtained maximize the profit of the service provider by minimizing the travel and detour distance as well as pick-up/drop costs while maximizing the utilization of the car. The proposed algorithm has been implemented over the Salt Lake area of Kolkata. Route distance and detour distance for the optimal routes obtained using the proposed algorithm are consistently lesser for the same number of passengers when compared to the corresponding results obtained from an existing algorithm. Various statistical analysis like boxplots have also confirmed that the proposed algorithm regularly performed better than the existing algorithm using only Genetic Algorithm.

Keywords: carpooling, A* algorithm, genetic algorithms, pathfinding, Pareto optimality

Citation: *Computer Research and Modeling*, 2021, vol. 13, no. 1, pp. 67–85 (Russian).

1. Introduction

In the twenty-first century, due to the constant development of society and industry, the need for mobility has increased rapidly and so has the use of cars, especially in the developing and under-developed countries [Mulders, 2012, 2013]. This has resulted in traffic congestion, environmental pollution as well as caused various cardiovascular and respiratory problems. Carpooling is an effective solution to overcome the abovementioned problems [Manzini, Pareschi, 2012; Martino et al., 2011]. Research also suggests that carpooling results in less stress than commuting alone. Carpooling is the most rapidly evolving solution for the shift from vehicle ownership to shared vehicle usage mobility. The future of mobility consists of technology-enabled, door-to-door, multi-modal travel encompassing pre-trip, in-trip, and post-trip services to improve journey experience to the Mobility User [Briggs, 2015]. Carpooling helps users to share a ride to destinations in the same area, by either casual carpooling or by real-time carpooling. According to the most used terminology, carpooling is the agreement of sharing the use of a particular car by many passengers, usually commuting along the same route/ journey at mutually compatible times [Luè, Colorni, 2009]. As carpooling decreases the number of vehicles used by travelers, it results in various benefits like mitigation of traffic or congestion on the roads, reduced demand for parking facilities, lesser energy or fuel consumption, and most importantly, reduction in carbon emission, thus improving the quality of life in cities.

Genetic Algorithms (GA) represent a class of optimized, adaptive, and iterative algorithms that function upon existing data sets and design the developing concepts based on genetic information as observed in nature. GA operates on a set of solutions using operations like selection, crossover and mutation and stops only when concurrency of the required criterion takes place. These algorithms, even though irregular or randomized, mostly use heuristic data to utilize promising regions within the search space [Deb, 2011]. A population in a GA is a set of various coinciding search points or solutions. A new population is produced for each iterative step, called a generation. A solution, often labeled as a chromosome, $y = [y_1, \dots, y_n]$, is basically a set of variables in a search space of n dimensions or variables [Wan, Birch, 2013]. These n variables are like n -genes. This paper utilizes the essence of GAs to provide multiple solutions that are non-dominated equally important when considering a broad array of conflicting objectives. Any of these non-dominated solutions can be used as the final route without having the driver or the passengers suffer a loss in their interests [Konak et al., 2006].

One of the most extensively used path-finding algorithms is the A* algorithm, which is a heuristic or an informed search algorithm. A* uses the fundamental concepts of the Greedy Best First Search technique like the Dijkstra Algorithm and provides the shortest accessible path between the source and the destination. A* algorithm is used majorly in the fields of game development, robotics, traversal of graphs and maps, etc. The most important features of this algorithm are its high efficiency and its convenience. It uses a valuation function, denoted by $f(n)$, as a guiding capacity to find the required path, both effectively and precisely. This valuation function that gives an estimated cost of the path from the starting node to the target node, via the intermediate node is expressed as

$$f(n) = g(n) + h(n), \quad (1)$$

where $g(n)$ and $h(n)$ are the actual cost from the starting node to the current node(n) and the assessed/heuristic cost of the shortest path from this current node(n) to the target node, respectively. The heuristic cost function, $h(n)$, for each intermediate node is calculated by taking the Euclidean distance between the current node(n) and the target node. $h^*(n)$ denotes the actual cost of the best possible path from the present node(n) to the objective node. Now, if $h^*(n) \geq h(n)$ for all intermediate nodes, then it is accepted as a reachable route-finding process. It has vast uses in the fields of GIS systems along with game routing systems and maps.

There are numerous pieces of existing literature that have proposed algorithms for the carpooling problem. Some authors used Genetic Algorithm, weighted sum methodology, pathfinding algorithms like A* algorithm, Dijkstra algorithm, etc. Unfortunately, there are not many optimized car-pooling algorithms that support the users along with the service providers to choose the most optimal routes, keeping in mind the various real-life constraints that affect this decision-making process. This work

proposes a hybrid algorithm that implements carpool route optimization using a Genetic Algorithm and refines the route using the A* algorithm. This work aims at providing a choice of optimal routes, to facilitate the passengers and the service provider/driver by maximizing car utilization, minimizing total distance travelled, as well as keeping in consideration the individual passenger's cost. Instead of using the A* algorithm in its traditional form, where the parameter used to select the most optimal route is the distance, here the A* algorithm is used to optimize the set of routes obtained from the Genetic Algorithm. Considering the aforementioned conflicting objectives, viz. maximizing car utilization, minimizing total distance travelled, minimizing individual passenger's cost, the authors provide a list of non-dominated routes which are all considered equally good for both the passengers and the service provider.

The primary objective for this work is to provide a set of non-dominated routes to solve the traditional carpooling problem. It has been motivated by the fact that carpooling is very essential in the present scenario and is one of the most effective means in dealing with the detrimental effects of pollution. Carpooling works towards providing a greener environment by encouraging riders to share rides. Not only does this technique greatly reduce the fuel consumption per person and benefits the environment, but it also provides a cost-effective mode of travel for the riders by allowing them to travel together and share the cost leading to financial savings. Another objective of this work is to deal with the problem of deviating from the main route to pick-up/drop passengers and returning to that point before continuing the journey. This leads to excess and redundant travel. This motivated the authors to integrate the A* algorithm with GA to produce optimized routes with greater efficiency by fetching shorter new routes from the pick-up/drop point to the destination.

2. Literature survey

Varied research works have been published in the arena of carpooling to find optimal routes as well as to allocate riders while matching their requirements. The concept of carpooling consisted of picking up passengers in sequence and dropping them later. It later developed into a "park and ride" concept having a common pick up point for all the riders. Recently, due to the growth and spread of the internet technology, dynamic carpooling, i.e. picking up and dropping off passengers as and when requests arise while travelling, has reached its peak. This modern version of car-pooling witnessed its first practical use when John Zimmer, from Cornell University along with Logan Green, from University of California, created "Zimride", a dynamic match-making service to connect drivers and passengers using GPS on android phones [Zhang et al., 2014].

An automated system was developed [Knappen et al., 2013] to coordinate ridesharing trips. Clients registered their profile and periodical information about repeating trips, and the service provider prompted the enlisted users to combine their outings through ridesharing. The service provider assessed the satisfaction quotient of co-travelers based on prior information. Another model was developed [Schreieck et al., 2016] which focused on matching ridesharing offers with ride requests and also storing and retrieving routes using inverted index data structures. Google API was used for geocoding the source and destination address. This system employed the matching mechanism by emphasizing various shortest path algorithms, such as Dijkstra's Algorithm and A* Search. It was observed that the proposed technique performed well enough for real-time applications while being simpler than existing optimization-based techniques. Another model [He et al., 2014] concentrated on the profitability of the ride. Different GPS directions were mined to get the frequently utilized routes using route parting and gathering, grid mapping techniques, etc. An improved carpool system was developed [Karande, Bogiri, 2015] that allowed users to avail the services of ridesharing via a smartphone. This model defined an advanced Genetic Algorithm based carpool route and matching algorithm that provided a solution by securing ideal match arrangements.

An innovative model was proposed using GA [Masum et al., 2011] to solve the carpooling problem that used a fitness function to select desirable parents to reproduce and create the next generation. The process of preparation ensured the removal of duplicate genes within the child because

of crossover and mutation. Missing genetic information was re-inserted using a heuristic method. A GA based carpooling service was proposed in 2016 [Rathod et al., 2016] that generated optimal routes of travel within a short period of time. The proposed algorithm generated intermediate paths that were used to find the solution to the empty seats available in the car. Later another model was developed [Boukhater et al., 2014] that was map-based and provided shared rides for all customers, considering their personal inclinations. The proposed algorithm performed better than the traditional algorithm. A heuristic algorithm for Maximum Carpool Matching was proposed [Hartman et al., 2014] and it demonstrated the Maximum Carpool Matching problem was NP-hard even for the situation where the weight function is binary. They introduced a natural integer linear program and demonstrated that if the arrangement of drivers is known, an optimal assignment of travelers to drivers can be found in polynomial time utilizing a reduction to Network Flow.

A* Search is a procedure majorly utilized in the field of Artificial Intelligence. A bi-directional search technique model was proposed [Sharma, Pal, 2015] as an improvement over the traditional A* algorithm for finding the shortest path. As, A* algorithm is in general one of the most optimal path-searching algorithms that use heuristics, optimizing it even further by applying the bi-directional search, resulted in a system that provided the shortest possible path, in very less search time. The A* algorithm, both in its unidirectional and its bi-directional forms, provided results much better than those of the Dijkstra algorithm in its traditional and bi-directional forms, respectively. The authors concluded that the A* algorithm outdid the Dijkstra algorithm in all informed search situations, with and without obstacles. A hybrid algorithm was proposed [Arnates et al., 2017] for Unmanned Aerial Vehicles (UAVs) that applied a heuristic approach over Genetic Algorithm to provide facilities of pathfinding and re-routing. The algorithm involved greedy heuristic to find possible paths and then used the GA to provide the most optimal solutions within a comparatively low amount of time. To prove the efficiency of the proposed algorithm, experimental simulations were conducted, results of which showed that this combination of the given heuristic approach with GAs was a good strategy for routing UAVs.

Researchers [Zeng, Church, 2009] performed various tests on road maps of two regions of California to compare the effectiveness of the shortest path algorithms of the Gallo-Pallottino (GP) class with the A* algorithm and its three variations. The authors successfully proved that the A* algorithm and its variations performed much faster and better compared to what the GP-class algorithms do on real-life road maps or networks. It also showed that on-road networks, A* algorithm's performance exceeds even the most optimal execution of the Dijkstra Algorithm, that too by a very large margin. This work proved that the A* algorithm's optimality increases with the increase in the size of the road networks. The experiments in this research work also showed that one of the three variations of A*, A star with approximate buckets (ASBA) outperformed all the other algorithms that were considered.

A variation of the native A* algorithm, called the A* Hamilton algorithm [Arnates et al., 2017] was proposed to navigate to many destinations in any order. The algorithm provided the shortest path from a source location to many destinations without any order. Another model was proposed [Meng, Zhang, 2019] where a salient feature of the A* algorithm was to move towards the direction of the destination by utilizing directional elements, with the goal that the intermediate route procedure will move towards the shortest path as soon as possible. Then, the direction factor was utilized to guarantee that the priority of the path finding of the A* algorithm was to move towards the direction of the target. The proposed technique improved the efficiency of the algorithm as the outcome of the A* optimization algorithm was around 20–50% better than the traditional A* algorithm. Also, the best case was achieved at around 89%. A dynamic multi-objective ride-sharing model [Herbawi, Weber, 2012] was proposed that assigned passengers to the car drivers, characterized user requests, and coordinated the passengers' pickup and drop off timings, optimally. They proposed a hybrid algorithm that acted at two levels and divided each day into a group of time periods, to deal with the ride-matching problem using time windows. A hybrid path-finding model using Genetic Algorithm was proposed very recently in 2018 [Yui et al., 2018] that provided a multi-weighted heuristic (MWH) function, which was then used in the A* algorithm to find the most optimal routes. GA provided

multiple heuristic functions that acted as agents, which in turn competed to produce children chromosomes or agents. On optimizing all these agents, the final MWH function was returned.

The authors of this paper, inspired by the aforementioned research, thus proposes a hybrid model that uses a genetic algorithm for route optimization and aims to use A* algorithm to refine the proposed optimal solution for the carpooling problem. The hybrid route search technique controls the search towards the destination node by using lower limits on the distance to the target. The proficiency of this approach relies on the lower values. A* search utilizes path costs along with heuristic values. Here, along with the lower bounds on the distance to the target, the authors optimize the traditional A* algorithm by incorporating other parameters, like the density of ride requests generated in a route, the total detour taken for picking up and dropping off passengers and the length of the route, to provide a result that is optimal for both the passengers and the driver.

3. Proposed car-pooling model

It has been observed that hierarchical decision structures [Ibrahim, 2010] suitably represent multi-objective problems better in real life business scenario. It is also commonly seen that the higher-level parameters do affect but does not totally control the lower level objectives. If lower level objectives are given priority of choice making at the local level and proper synchronization and coordination be maintained with the upper level objectives, it is expected to generate improved results. [Takama, Loucks, 1981]. A multi-level multi-objective optimization algorithm was earlier proposed [Beed et al., 2020] to solve the carpooling problem. The model used Genetic Algorithm to obtain optimized routes maximizing the carpool service provider's profit. A detailed analysis of the car routing problem brings into light the presence of multiple conflicting objectives. The essence of multi-objective optimization is that it strives to obtain a set of solutions, by applying various mechanisms, so that no objective is neglected on the behest of others. This work aims to improve the Hierarchical Multi-objective Route Optimization for Solving Carpooling Problem [Beed et al., 2020] by applying the proposed hybrid GA-A* algorithm for route optimization.

While solving a carpooling problem, the basic intuition is frequently directed towards minimizing the distance traveled by the passengers. Considering other factors such as occupancy, detour, and passenger pickup and drop cost into consideration can significantly provide better selection of a route. The proposed model aims to solve the car-pooling problem by dividing these multiple objectives into a hierarchical model, to optimize the solution. Since the higher-level objectives do not completely control the parameters of the lower level, the lower level of this hierarchy consists of those conflicting objectives that are confined to the individual passengers only, namely detour distance for pickup and drop of a passenger and passenger density of the surrounding area. The higher-level objectives concentrate on route optimization and consists of minimizing the distance traveled by the car, maximizing the utilization of the vehicle, and reducing the pick-up and drop-off cost for a particular passenger. Hierarchical decision structures help in realizing real-life situations better. The conflicting objectives are hierarchically arranged as in Figure 1.

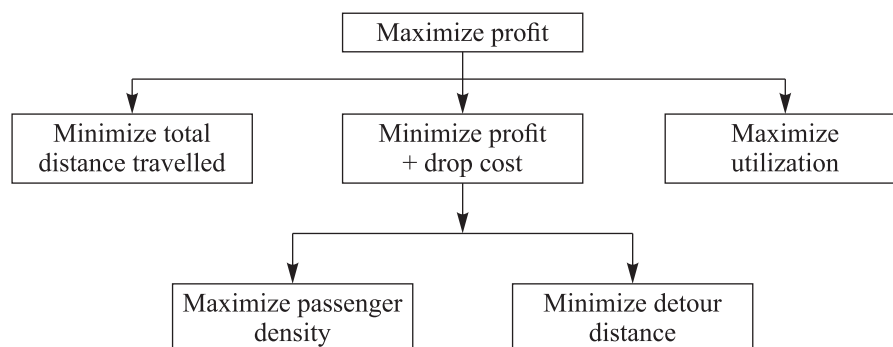


Fig. 1. Hierarchical optimization structure

Mathematically, the **upper level** objectives are defined as follows:

Conflicting objectives

f_1 : minimization of travel distance

Here, the travel distance is the net distance between the starting and drop location of the 1st passenger. A presumption has been made that the trip ends at the drop location of the 1st passenger.

Considering n^{th} feasible route,

$$\text{Dist}_n = \sum \text{dist}_n(x, y), \quad (2)$$

where $\text{dist}_n(x, y)$ are the distances between junctions on the route between pick up and drop point of first passenger and Dist_n is the summation of such distances. It is beneficial to have minimum distance as distance is directly proportional to fuel consumption and hence, cost.

f_2 : maximization of vehicle utilization

The vehicle should always be packed to capacity to generate maximum revenue.

f_3 : minimization of passenger's pickup-drop cost

The cost to pick a particular passenger and drop him/her should be minimized to maximize revenue. However, this is not a single objective but a combination of few conflicting objectives which influence this cost factor. The lower level objectives are defined later.

Linking the three conflicting objectives, the main objective is formulated as follows:

Main objective

$$\text{maximize } f \equiv (f_1, f_2, f_3). \quad (3)$$

Constraints

$$\text{dist}_n(x, y) \geq 0 \quad \forall n,$$

$$0 \leq x \leq 4, \text{ as the maximum carrying capacity of the vehicle.}$$

At the **subordinate level** the conflicting objectives are as follows.

f_4 : minimization of detour distance

Detour distance is the sum of the pickup and drop off distance. As explained earlier, shorter the distance travelled, less is the fuel consumption, lower is the cost. For the i^{th} passenger, let x_i and y_i be the pickup and drop off distance respectively,

$$\text{Detour_dist}_i = x_i + y_i. \quad (4)$$

f_5 : maximize passenger density

There is a greater likelihood of finding more passengers in densely populated areas leading to greater occupancy of the vehicle. Popular areas are expected to be favored over scarcely populated areas. n_i is considered to be the passenger density inside radius r with respect to passenger i .

Main objective. Minimize individual's cost

$$f_3 \equiv (f_4, f_5). \quad (5)$$

Constraints

$$x_i, y_i \geq 0 \quad \forall i,$$

$$n_i \geq 0 \quad \forall i,$$

$$t_{\max} > t > 0.$$

The waiting time for a passenger should not exceed the maximum t_{\max} defined in the system.

In figure 2, person A wants to seek a carpool ride from location 'A' to location 'a'. There are three probable routes between these two points marked orange, green and blue. On closer observation, it is seen that the route colored green has the minimum distance but there are just three passengers in this route. The second route colored blue has the highest count of passengers but is definitely longer than the previous route. The third route colored orange does not have the highest count of passengers on the route, but most of the passengers are placed very close to the original route thereby minimizing the detour distance. All the routes are equally favorable as they are superior with respect to a certain objective.

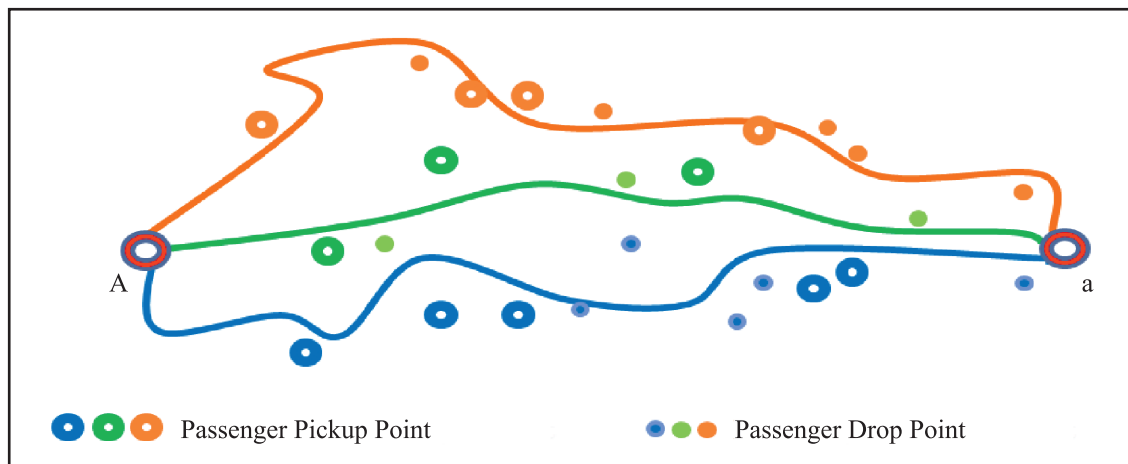


Fig. 2. Higher level parameters

In order to formulate the above as a multi-objective problem, the objectives need to conflict with one another where improving one leads to the deterioration of the other/s. As minimizing the travel distance is directly proportional to the cost incurred, it directly leads to poor utilization of the vehicle capacity. Hence it can be concluded that if the service provider maximizes the vehicle's capacity utilization, it is highly likely that the vehicle needs to travel more. Greater the travel, greater are the expenses and minimization of profit, thus the conflict. Secondly, minimizing the pickup and drop cost may lead to lesser travel which in turn may result in under-utilization of the vehicle thus justifying the multi-objective nature of the problem. While selecting an individual passenger, one may consider the detour distance ($x_B + y_B$) for Passenger B to be much less than the detour distance for Passenger C ($x_C + y_C$) in Figure 3. However, it may be observed that two other passengers are in the proximity of C in contrast to none in the proximity of B. Picking up C would ensure better utilization of the car capacity while picking up B would ensure lower detour distance. Hence both the solutions are equally enterprising justifying the multi-objective nature, once again.

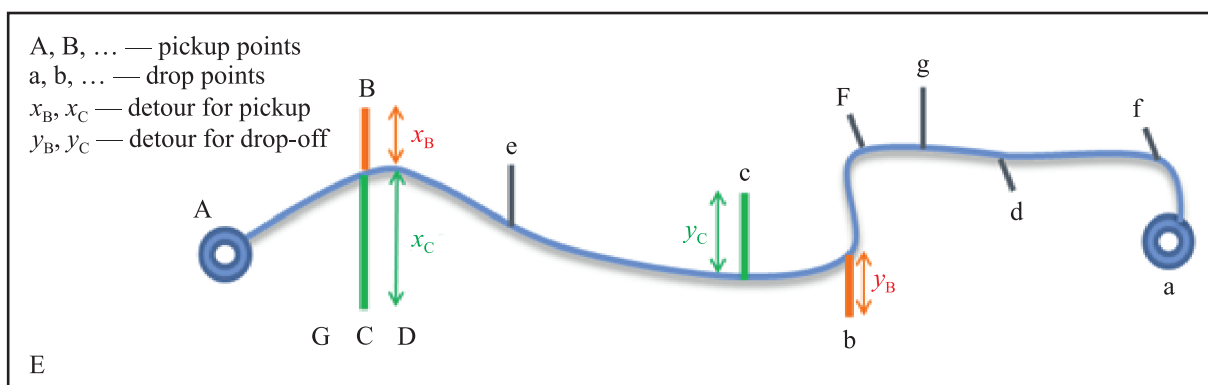


Fig. 3. Lower level parameters

The proposed hybrid algorithm aims to improve the existing algorithm by combining GA with A* algorithm. Since the A* algorithm is efficient in providing the shortest route between any two points, the authors have utilized this characteristic to generate a route incorporating all the pick-up and drop-off points of passengers for every corresponding elite route. This leads to maintaining the same level of occupancy while benefiting greatly in other aspects of total distance and detour.

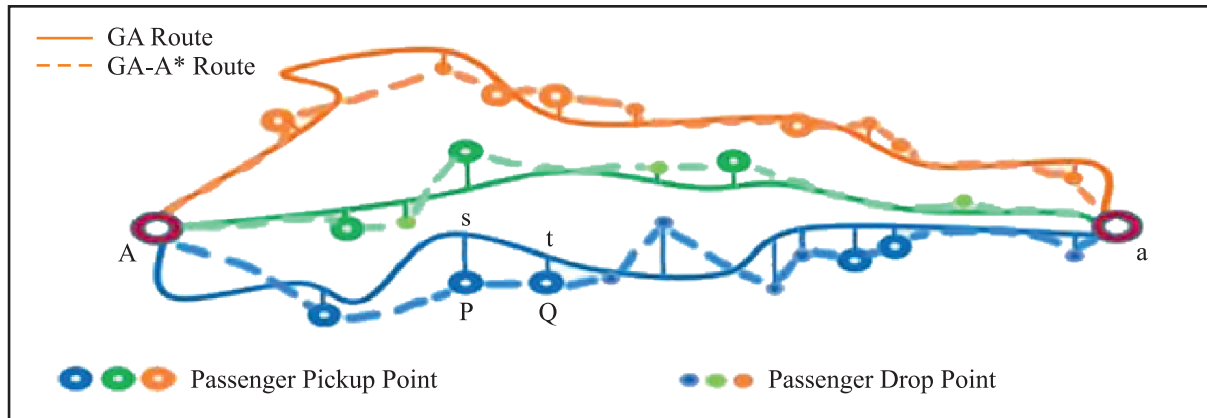


Fig. 4. Route Map of GA and GA-A* algorithms represented by solid and dotted lines respectively

Once the optimal routes are generated by one generation of GA, the routes are now processed using the A* algorithm to obtain better solutions. As can be observed from Figure 4, the routes generated by the GA (solid lines) require the cars to take additional detours from the route to pick-up/drop passengers and return back to the original route (solid line). This leads to an unnecessary increase in travel distance. Consider the car has picked up Passenger P. The next pickup is Passenger Q. Using the traditional algorithm, the car would travel to Point 's', travel along the main route (solid line) till Point 't' and then take a detour to pick passenger Q. The proposed hybrid GA-A* algorithm improves this drawback by obtaining the shortest route from P to Q using the A* algorithm. The process repeats this shortest route search algorithm from every pickup/drop point to its next pickup/drop point on the route using this A* algorithm. These routes are further subjected to dominance tests to obtain Pareto Optimal solutions.

It might seem counter-intuitive that the shortest path between the starting and ending points is not considered explicitly during route selection by the GA. However, a careful study of the selection process across the various generations of the GA reveals that in most cases, the crossover and mutation procedures implicitly add the shortest path to the gene pool but it gets eventually eliminated by the algorithm on account of a poor fitness score. Although the shortest path might offer the best distance, it tremendously neglects the other objectives of better occupancy and car utilization. This makes it unwanted from the perspective of a multi-objective model.

The hybrid algorithm is as follows:

ALGORITHM(GA-A*):

- Step 1: Read passenger request log
- Step 2: If request log is empty then go to Step 20
- Step 3: If the passenger pickup point is within a radius of t kms
then store pickup and drop location of passenger i into X and Y
go to step 4
- else go to step 1
- Step 4: Set X and Y as source and destination of route; $gen = 1$
- Step 5: Generate randomly a pool of m chromosomes being routes between X and Y
- Step 6: Select randomly q chromosomes for first generation of Genetic Algorithm
- Step 7: For each generation of GA perform Rank Selection, Crossover and Mutation

- Step 8: Perform Dominance test to obtain Pareto Optimal n chromosomes / optimal routes.
- Step 9: For each route obtained in Step 8, do
- Step 10: For each point P (pickup/drop point) on the route, do
- Step 11: Use A* algorithm to obtain shortest path from P to the next pickup/drop point Q .
- Step 12: Append path X to P with this new shortest path from P to Q .
- Step 13: Implement dominance tests on the newly generated route to maintain Pareto Optimality.
- Step 14: End of Step 10 Loop
- Step 15: End of Step 9 Loop
- Step 16: Combine n Pareto optimal chromosomes with another set of k random chromosomes from the pool for a total of q chromosomes.
- Step 17: $gen = gen + 1$
- Step 18: Go to Step 7 till $gen < max_gen$
- Step 19: Print n Pareto optimal solutions
- Step 20: Exit

The A* algorithm functions in the following manner:

- Step 1: Insert starting node into Open List (OL)
- Step 2: Retrieve the first node of OL as the current node (CN)
- Step 3: If CN is the destination node, then exit
- Step 4: Explore the neighboring nodes of CN
- Step 5: Set CN as their parent and calculate valuation functions
- Step 6: If neighbors of CN are not present in Closed List (CL)
- Step 7: Insert neighbors of CN in OL in increasing order of valuation functions.
- Step 8: Remove CN from OL and add to CL.
- Step 9: Go to Step 2.

where X is the starting point of first passenger (marked A in Figures 2–4),

Y is the destination of the first passenger (marked a in Figures 2–4),

P is the current node being processed (marked P in Figure 4),

Q is the next node to be serviced which may be a pickup point or a drop point (marked Q in Figure 4).

Step 1 of the algorithm reads the passenger request log. The log comprises of the name of the passenger and his geographical location of pickup and drop points. Once the passenger's pickup location is within radius t kms, he is selected and his pickup and drop locations are assigned as the route's start and end points, X and Y respectively. Step 5 generates random routes between these two points X and Y and creates the chromosome pool. Step 6 marks the beginning of the Genetic Algorithm. The typical operations of GA, i.e. selection (here rank selection has been used), crossover (ordered crossover has been used) and mutation (creep mutation) has been performed. Step 8 focuses on use of Pareto Optimality for solving multi-objective optimization problems and the utilization of the dominance test to generate Pareto Optimal results. As explained earlier, after the GA routes are obtained an improvisation is done using the A* algorithm to obtain shorter routes. The main objective of this improvisation is to obtain the shortest path using A* algorithm from the current node to the next node to be serviced (pickup/drop). The obtained results are subjected to dominance test to obtain Pareto Optimal results. This is performed using steps 9 to 15. The process is then repeated for a specified number of generations to refine the results obtained thus far.

Coming to the A* algorithm, the **open list** contains the nodes that are not yet accessed and may be visited. The **closed list** contains the nodes that have already been visited. At the beginning, both the open list and closed list are empty. Step 1 of the algorithm adds the starting node to the open list. Step 2 retrieves the topmost node of the open list, i.e., the node with the lowest value of the valuation function and sets it as the current node (CN). Step 3 checks whether the current node is the destination node. If so, the path is complete, and the algorithm is terminated. In steps 4 and 5, the valuation functions of the neighboring nodes of the current node are calculated. In steps 6 and 7, the neighbors which are not present in the closed list are updated in the open list with their positions in the open list

determined by the valuation function. A smaller value is placed nearer to the top because the smaller value denotes a tendency for the path through that node to be shorter. Step 8 adds the current node to the closed list and the algorithm is executed again from step 2.

For each generation of GA, the routine represented in Figure 5 is executed to select the passengers served along each route. This is done as part of the Rank Selection procedure. The fitness score is defined as the ratio of the length of the route and the number of passengers served. Each chromosome is assigned a fitness score. The following sub-functions / data structures are used to generate the passenger list (i) **passenger request**: stores the pickup and drop-off points of passenger, (ii) **serviceable request**: A passenger request R whose pickup point is in the neighborhood of a node in the route and the drop-off point of the request is in the direction the car's journey. The total detour of such a request from the initial route should not exceed the predetermined threshold value, (iii) **request-list**: A list to store all the passenger requests that is served by the car while travelling from X to Y (iv) **service-list**: A list to store the pickup and drop-off points in the order that they are serviced (v) **base route**: It is the route generated by the GA, (vi) **occupancy**: It is the number of passengers currently present in the car. The maximum permissible occupancy is four.

For each node in the base route, the request-list is searched for nearby drop-off points. If a drop-off exists, the drop-off point is added to the service-list and the occupancy is reduced by one. If the occupancy is less than the maximum permissible limit, the global request pool is then searched for nearby serviceable requests. If found, the request is added to the request-list, the occupancy incremented by one, and its pickup point is added to the service-list. Standard algorithms have been used for ordered crossover and creep mutation.

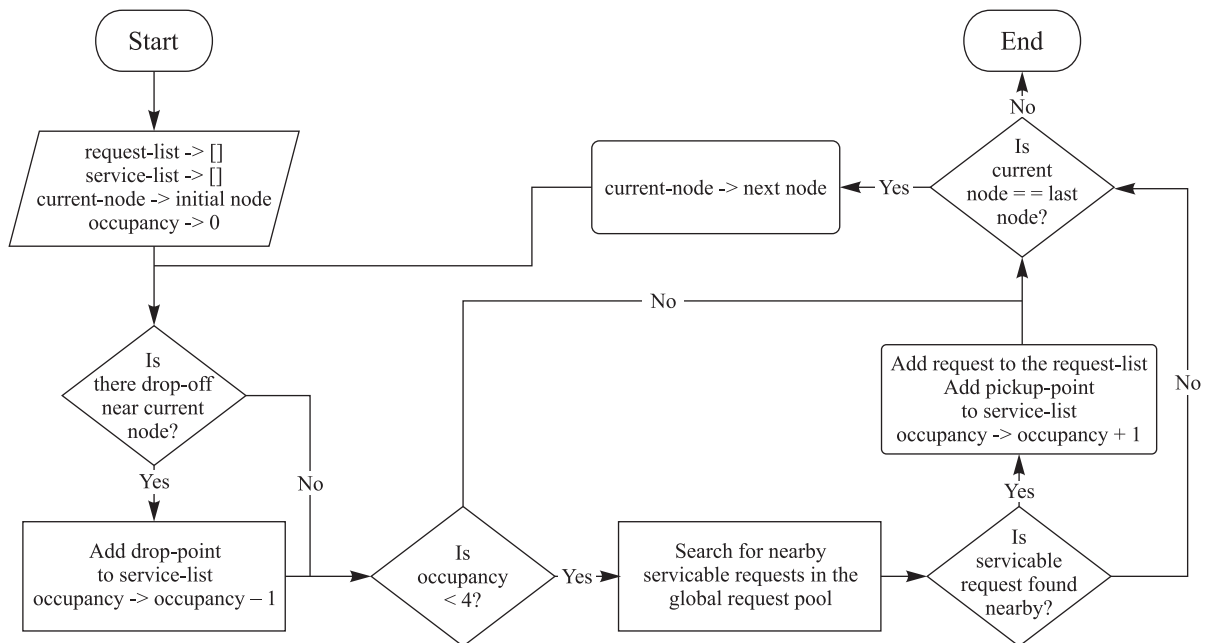


Fig. 5. Diagrammatic representation of the passenger selection module

4. Experimentation and results

The map of Salt Lake area of Kolkata has been used as a prototype to implement this algorithm. The map was divided into 116 junction points or nodes. The Google Map API was used to obtain the actual distance between the nodes and a corresponding distance matrix was created. A request matrix was dynamically generated and used by the algorithm to generate routes. The initial request was randomly generated.

On executing the program code, the following results are obtained.

Table 1. The data displayed in the above table represents the result of a particular execution

GA Route	GA-A* Route	Occu- pancy	Total Dist. GA	Total Dist. GA-A*	Detour GA	Detour GA-A*	Improve- ment in distance
73 84 92 93 105 106 107 115	73 84 92 93 105 106 112 106 103 106 112 115	5	7280	4320	4300	1400	40.66%
73 84 92 93 94 93 105 106 112 115	73 84 92 93 105 102 94 95 94 93 105 106 112 106 103 106 112 115	6	9120	6220	5600	3300	31.80%
73 60 73 84 92 93 105 106 107 115	73 84 92 93 105 106 112 106 103 106 112 115	5	7880	4320	4300	1400	45.18%
73 84 92 108 109 112 115	73 84 92 93 105 106 112 106 105 106 112 115	4	8020	4420	4400	1500	44.89%
73 84 92 108 109 112 109 112 109 112 115	73 84 92 93 105 106 112 106 105 106 112 115	5	11420	4420	6000	1500	61.30%
73 84 92 93 105 102 94 102 103 104 107 106 105 109 112 115	73 84 92 93 105 102 103 102 94 95 94 102 105 102 103 97 103 106 107 106 112 106 103 106 105 109 112 115	10	15760	9680	10060	6760	38.58%
73 74 61 74 73 84 92 108 109 112 115	73 74 61 74 86 87 95 94 93 105 106 112 115	4	9120	3880	3600	960	57.46%
73 74 73 84 92 93 94 85 94 93 105 109 112 106 107 115	73 84 92 93 105 102 94 95 87 95 94 93 105 109 112 106 105 102 103 106 112 115	7	14230	7540	7900	4620	47.01%
73 84 92 84 92 108 92 108 109 112 115	73 84 92 93 105 109 105 93 92 93 105 106 112 106 105 106 112 115	5	12620	6720	5800	3800	46.75%
73 84 92 84 73 74 86 88 86 88 89 90 91 97 103 106 107 106 112 115	73 84 92 93 92 84 73 74 75 76 77 78 79 80 98 104 103 106 105 102 103 106 112 106 105 106 105 109 112 115	9	17640	11570	10560	8650	34.41%

A single execution of the proposed algorithm generates several non-dominated solutions which are all equally good. Table 1 lists all such feasible non-dominated routes between starting location 73 and end location 115. The other columns of the table provide a comparative study with respect to the detours and total distance of the routes provided by the GA and the GA-A* algorithms. As seen in Figure 4, the solid lines represent the distances obtained using the existing algorithm [Beed et al., 2020] and the route is given in Column 1, the route distance in Column 4 and the detour distance in column 6, measured in metres. Column 2 represents the route obtained using the dotted line in the same figure, columns 5 represents the route distance using the proposed hybrid GA-A* algorithm. Column 7 gives the detour distance which has been calculated as the difference between the total route distance displayed in column 5 and the shortest distance between the between the start and end points of the request. Although the routes under the Column 1 appears to contain a fewer number of nodes and hence, by intuition, should have a lesser total distance, it is to be noted that these routes just represent the basic travel path of the car and does not take into account the detours for picking up and dropping off passengers, which need to be added to the main route distance while assessing total distance and cost. Since the routes generated by the GA-A* algorithm includes the pickup and drop-off points of the passengers, there is no additional detour. The following chart (Figure 6) displays the percentage improvement in the total distance of the routes over a single execution for ten different routes after being optimized using the GA-A* algorithm (Column 8 of Table 1).

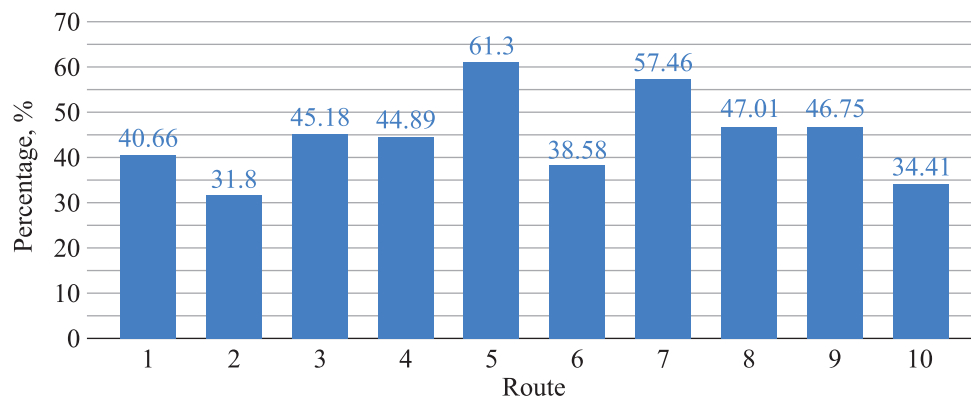


Fig. 6. Percentage improvement in Distance over a single execution

Statistical Analysis for comparison of Results obtained by Hybrid GA-A* algorithm and existing GA

The box plots in Figure 7 clearly show consistently lower values for total distance using the proposed algorithm as compared to the existing GA in ten different routes generated using a single run. Also, the variability in the values of total distances by GA-A* as compared to that of GA is less.

The box plots in Figure 8 clearly show consistently lower values for detour distances using the proposed algorithm as compared to the detour distances obtained using the existing GA for ten different routes generated in a single execution between the same start and end points. It is also noted that on an average, the total distance and the detour in case of the proposed GA-A* algorithm are clearly less than the existing GA algorithm. The above observations are confirmed using statistical tests given below:

Table 2. Two-sample t test for Total_Dist GA vs Total_Dist GA-A*

	N	Mean	StDev	SE Mean
Total_Dist GA	10	11309	3629	1148
Total_Dist GA-A*	10	6309	2618	828

Difference = mean (Total_Dist GA) – mean (Total_Dist GA-A*)

t-Test of difference = 0 (vs >): t-Value = 3.53, P-Value = 0.001

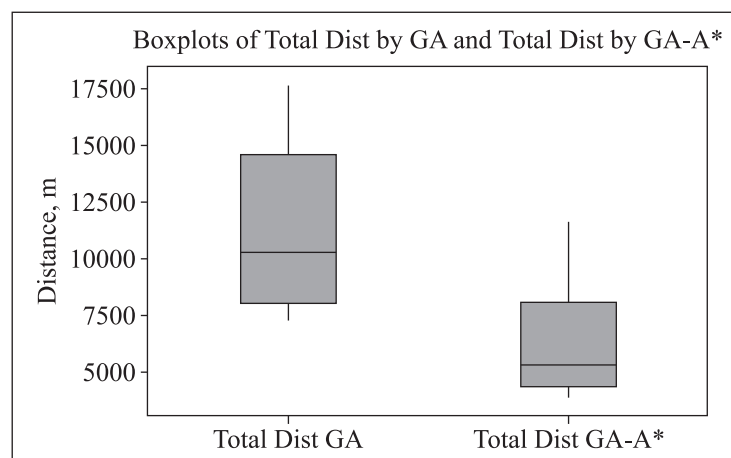


Fig. 7. Showing boxplots of total distance using GA and total distance using GA-A*. The distances represented along the y-axis are in meters

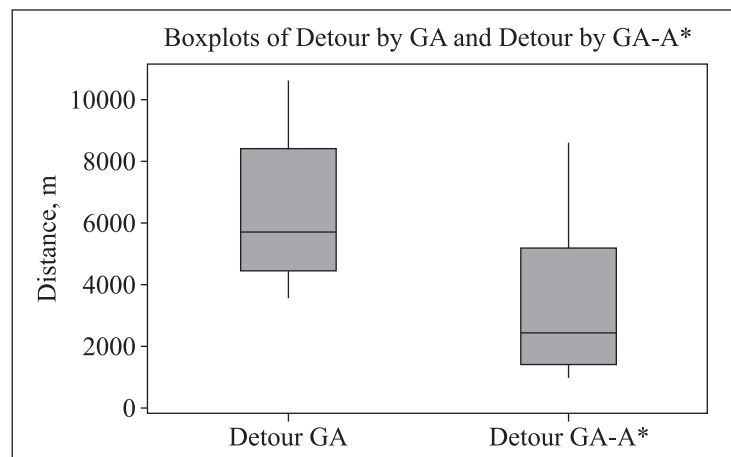


Fig. 8. Showing boxplots of Detour distance using GA and Detour distance using GA-A*. The distances represented along the y-axis are in meters

p-value clearly indicates rejection of the null hypothesis that there is no difference in the mean total distances against the greater than type alternative hypothesis at 5% level of significance. Hence we conclude that the average total distance by GA is significantly greater than the average total distance under the proposed algorithm under 5% level of significance.

Table 3. Two-sample t test for Detour GA vs Detour GA-A*

	N	Mean	StDev	SE Mean
Detour GA	10	6252	2462	779
Detour GA-A*	10	3389	2618	828

Difference = mean (Detour GA) – mean (Detour GA-A*)

t-Test of difference = 0 (vs >): t-Value = 2.52, P-Value = 0.011

p-value clearly indicates rejection of the null hypothesis that there is no difference in the mean total detour distances against the greater than type alternative hypothesis at 5% level of significance. Hence, we conclude that the average total detour distance by GA is significantly greater than the average total detour distance under the proposed algorithm under 5% level of significance.

A simulation run of twenty different executions has revealed that the average GA-A* distance triumphs over the average total distance provided by GA in all executions, thus providing evidence that the routes have been optimized while maintaining an identical level of occupancy. The following table records the observed average total travel distance of the routes selected by the GA and the routes improvised by the GA-A* Hybrid algorithm for each of the 20 executions.

The following chart (Figure 9) uses the data from Table 4 to compare the average Total GA Distance and GA-A* Distance for 20 different executions of the algorithm.

The following chart (Figure 10) uses the data from Figure 9 to demonstrate the percentage improvement in the average distance per execution obtained on using the A* algorithm to optimize the routes generated by the GA.

The following three charts (Figures 11, a–c) compare the distance of the route generated by the GA and the corresponding route generated by GA-A* for three different iterations. It can be observed that every route in each iteration is shortened after optimization using the A* algorithm.

Table 4. Data for Average Total GA Distance and Average GA-A* distance for 20 executions

Execution No	Average Total GA Distance	Average GA-A* Distance	Improvement in Hybrid Algorithm	Execution No	Average Total GA Distance	Average GA-A* Distance	Improvement in Hybrid Algorithm
1	16169	12775	21%	11	19851	12938	35%
2	2257	1905	16%	12	14809	9954	33%
3	6058	3330	45%	13	24684	16856	32%
4	25667	13837	46%	14	13189	8068	39%
5	17724	13708	23%	15	13104	9928	24%
6	15808	11823	25%	16	29620	16988	43%
7	2882	2122	26%	17	3139	2614	17%
8	3084	1203	61%	18	9925	6674	33%
9	5302	3950	25%	19	20810	11830	43%
10	13081	6178	53%	20	13106	9454	28%

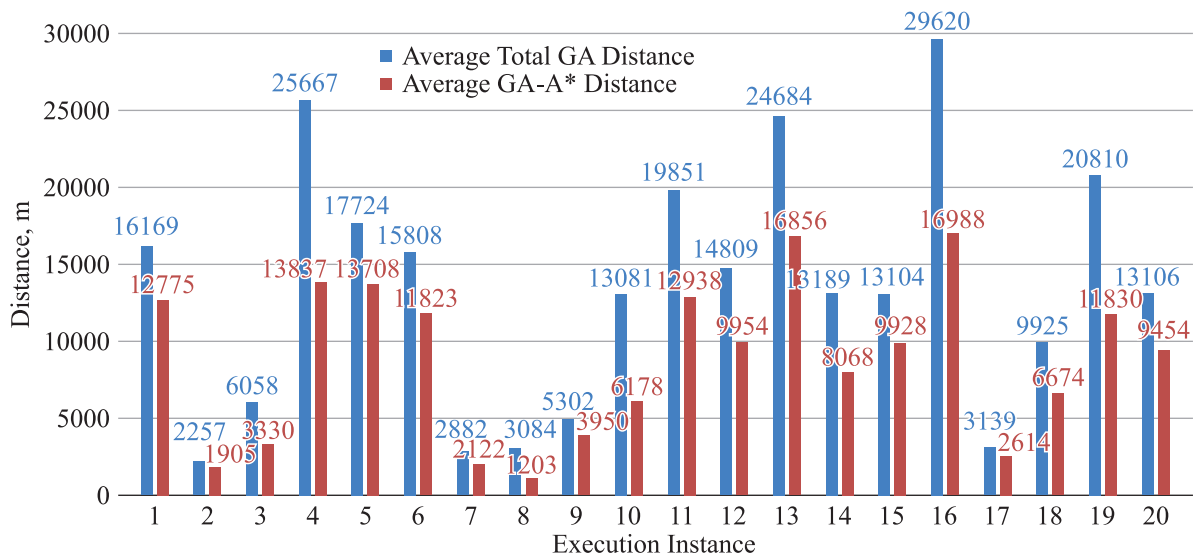


Fig. 9. Comparison of the average distance per execution of GA and GA-A* for 20 executions

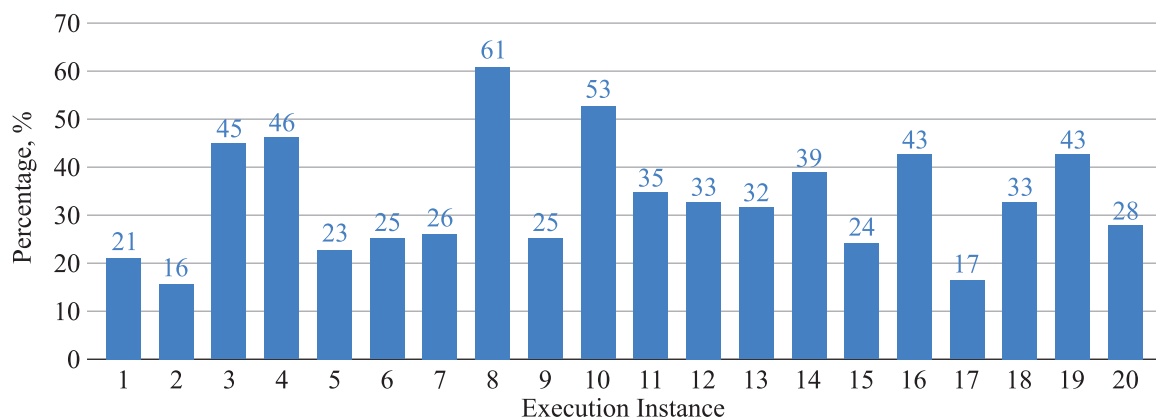


Fig. 10. Percentage improvement in Distance over 20 executions

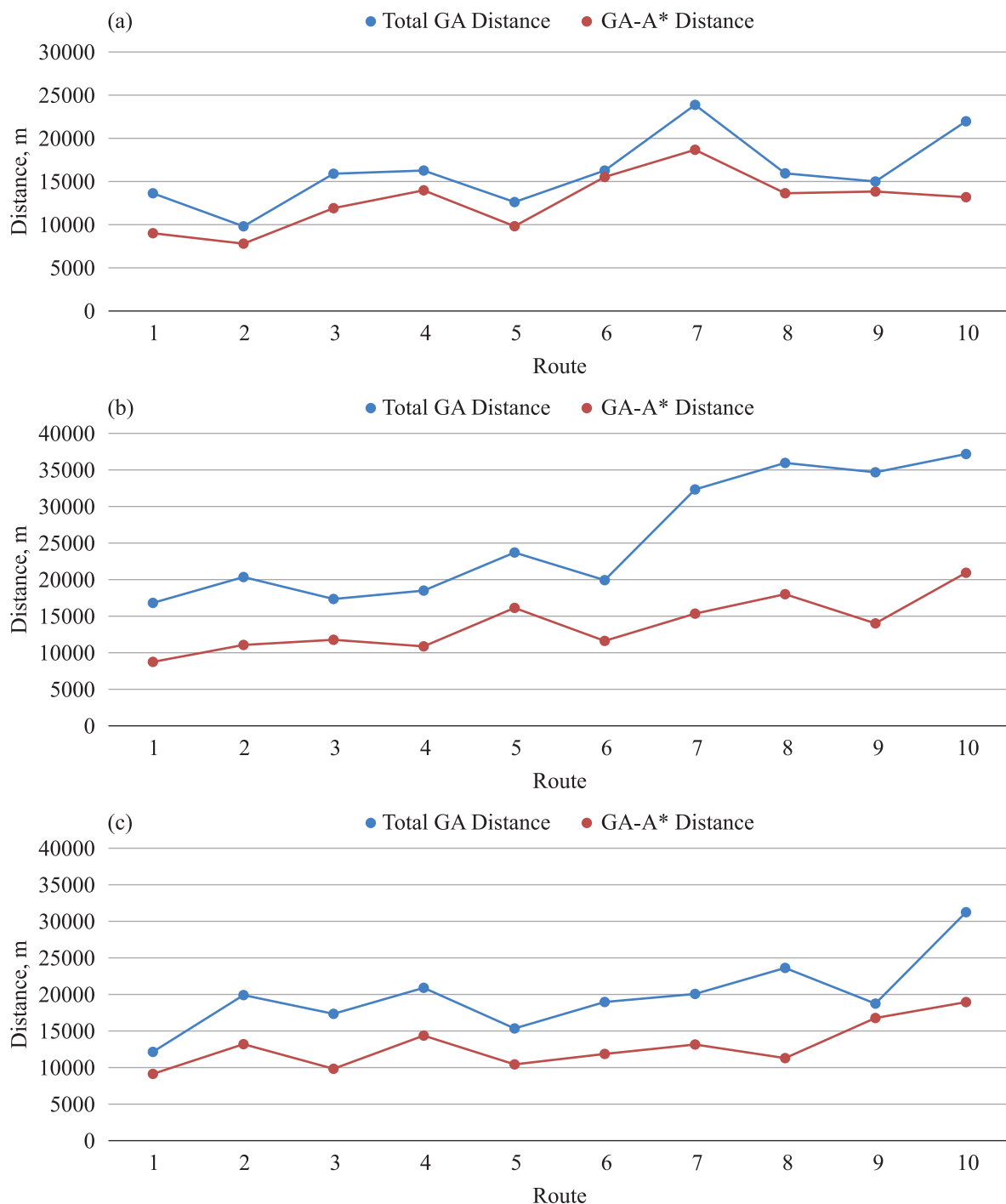


Fig. 11. Distances of the routes generated by GA and GA-A* in different instances of execution

Statistical Analysis for comparison of Results obtained by Hybrid algorithm and existing algorithm

The box plots (Figure 12) are clear indicating lesser average distances by the proposed algorithm as compared to the existing GA algorithm. The proposed algorithm, having lesser spread in the values, seems to be more consistent than the existing one. Statistical test to confirm the above observations are given below:

Table 5. Two-sample t test for average Total_Dist in GA vs average Total_Dist in GA-A* in 20 independent runs

	N	Mean	StDev	SE Mean
Total_Dist GA	20	13513	8110	1813
Total_Dist GA-A*	20	8807	5074	1135

Difference = mean (Average Total_Dist_GA) – mean(Average Total_Dist_GA-A*)

t-Test of difference = 0 (vs >): t-Value = 2.20 , P-Value = 0.018

p-value clearly indicates rejection of the null hypothesis that there is no difference in the mean total distances against the greater than type alternative hypothesis at 5% level of significance in 20 independent runs of the algorithms. Hence it can be concluded that the average total distance by GA is significantly greater than the average total distance under the proposed algorithm under 5% level of significance.

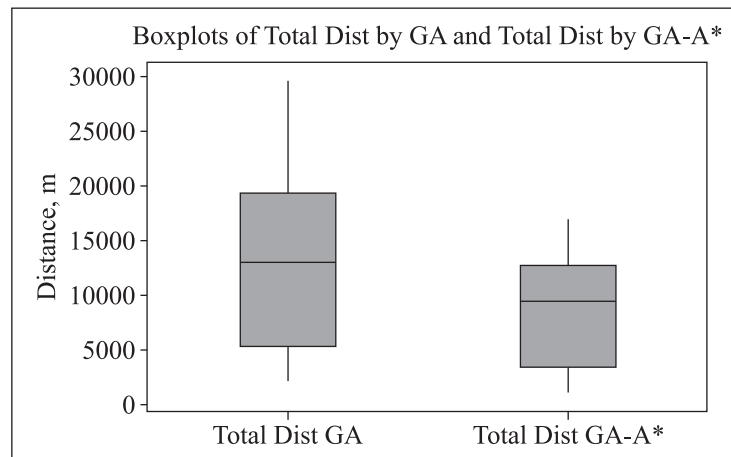


Fig. 12. Boxplots of average total distances generated using GA and that using hybrid GA-A* for 20 executions. The distances represented along the y-axis are in metres

5. Conclusion

The carpooling or ride-sharing algorithms proposed in the past have focused mainly on the shortest route to generate maximum profit though the real life scenario has clearly indicated there are multiple factors which influence this passenger and route selection. The hybrid algorithm proposed in this paper adds a new concept of considering multiple conflicting objectives to solve the carpooling problem in a hierarchical manner. This hierarchical structure has a better resemblance of the reality. Pareto-optimal technique has been preferred over the popular weighted sum technique for solving multiple conflicting objectives as the latter suffers from certain drawbacks. Genetic Algorithm has been used to generate optimal results and routes have been refined over generations. To these routes, A* algorithm has been applied to further reduce the route distances. The simulated experiments conducted by this model to provide car-pooling routes in a real road map proved that the proposed Hybrid algorithm, using both GA and A* algorithms, provides more optimal routes than in the case where only GA is used. Also, on an average, the total distance and the detour in case of the proposed algorithm are clearly less than the existing GA algorithm. So, this work has successfully contributed in producing a hybrid algorithm that can be used for real life carpooling situations to provide time efficient results, for both the users and the service provider.

From the results it is clear that the routes generated by the existing genetic algorithm are generally longer and the cars have to retract back to the point of diversion to return to the original route after completing the pick-up/drop. However, in real life, it might not be feasible to retract back to the original route. This shortcoming is greatly improved by using the A* algorithm to reroute and optimize the original route. As is evident from the results that have been provided, the optimization provides vast improvements on the previously observed results. This mainly stems from the fact, that A* is an algorithm that emphasizes finding the shortest possible route between two points while limiting the computations to a minimum. The new route that is generated includes the pick-up and drop-off points of the passengers while maintaining the same level of occupancy, thus providing shorter paths. This, in turn, decreases the overall travel of the car, thus ensuring shorter trip times for passengers as well as lesser fuel cost on the part of the driver. This also deals with the negative aspect of having to return to the point of deviation, in cases where it is completely unnecessary. This presents a more practical approach to determining the route thus making the algorithm implementable in the real world.

According to the above experiments and observations, the paths generated by the Genetic Algorithm are not generally the shortest as these paths initially do not include the pick-up and drop-off distance of the passengers. As a result, the cars must take various detours by deviating from the main route to pick-up or drop-off a passenger and then again return back to the point of deviation. GA generates final paths which are comparatively longer than the paths generated by the optimized GA-A* algorithm. Secondly GA requires considerably longer time to generate the route which is considerably reduced by using A* algorithm. The model proposed by this work can be further improved by taking into consideration various other conflicting objectives like (i) Traffic lights: Larger number of crossings and/or traffic lights along the route may cause congestion and thus loss of time. (ii) Road networks: Aspects like blockage of various roads, or restricted movements of vehicles in particular directions, (iii) Road surface quality: This may also be a factor affecting the choice of routes and can sometimes be very important to reduce the total travel time and (iv) Congestion: There might be an enormous amount of traffic congestion along the shortest route whereas the longer routes may have lesser traffic.

References

- Arantes J., Arantes M., Toledo C., Trindade O., Williams B.* Heuristic and Genetic Algorithm Approaches for UAV Path Planning under Critical Situation // International Journal on Artificial Intelligence Tools. — 2017. — Vol. 26, No. 01. — P. 1760008.
- Beed R. S., Sarkar S., Roy A., Bhattacharya D.* Hierarchical Multi-objective Route Optimization for Solving Carpooling Problem / In *Mandal J., Mukhopadhyay S.* (eds) // Proceedings of the Global AI Congress 2019, Advances in Intelligent Systems and Computing, Springer. — Vol. 1112.
- Boukhater C., Dakroub O., Lahoud F., Awad M., Artail H.* An intelligent and fair GA carpooling scheduler as a social solution for greener transportation // Proceedings of 17th IEEE Mediterranean Electro-technical Conference, Beirut, Lebanon. — P. 182–186.
- Briggs M.* Future of Mobility: Introducing the New Business Models // Proceedings of Frost & Sullivan's Intelligent Mobility Workshop, London, United Kingdom. — July 2015.
- Deb K.* Multi-objective Optimization using Evolutionary Algorithms. — New York, United States: John Wiley & Sons Inc., 2001.
- Halaoui H.* Intelligent Hamilton Path: Using Artificial Intelligent A* Algorithm and Hamilton Path to Navigate Multiple Destinations // Lecture Notes in Networks and Systems. — P. 344–357. — DOI: 10.1007/978-3-319-56994-9_24
- Hartman I., Keren D., Abu dubai A., Cohen E., Knapen L., Yasar A., Janssens D.* Theory and Practice in Large Carpooling Problems // Procedia Computer Science. — 2014. — Vol. 32. — P. 339–347.

- He W., Hwang K., Li D. Carpool Routing for Urban Ridesharing by Mining GPS Trajectories // IEEE Transactions on Intelligent Transportation Systems. — 2014. — Vol. 15, No. 5. — P. 2286–2296.
- Herbawi W., Weber M. A genetic and insertion heuristic algorithm for solving the dynamic ride matching problem with time windows // Proceedings of the 14th International Conference on Genetic and Evolutionary computation conference, New York, USA. — 2012. — P. 385–392.
- Ibrahim A. Solving multi-level multi-objective linear programming problems through a fuzzy goal programming approach // Applied Mathematical Modelling. — 2010. — Vol. 34, Iss. 9. — P. 2377–2387.
- Karande N., Bogiri N. Solution to Carpool Problems using Genetic Algorithms // International Journal of Engineering and Technique. — 2015. — Vol. 1, No. 3. — P. 113–122.
- Knapen L., Hartman I., Keren D., Yasar A., Cho S., Bellemans T., Janssens D., Wets G. Scalability issues in optimal assignment for carpooling // Journal of Computer and System Sciences. — 2014. — Vol. 81. — P. 568–584.
- Knapen L., Keren D., Yasar A., Cho S., Bellemans T., Janssens D., Wets G. Estimating scalability issues while finding an optimal assignment for carpooling // Proceedings of 4th International Conference on Ambient Systems, Networks and Technologies, Elsevier Procedia Computer Science 19, Halifax, Canada, June, 2013.
- Konak A., Coit D., Smith A. Multi-objective optimization using genetic algorithms. A tutorial // Reliability Engineering and System Safety. — 2006. — Vol. 91. — P. 992–1007.
- Luè A., Colorni A. A software tool for commute carpooling: a case study on university students in Milan // International Journal of Services Sciences. — 2009. — Vol. 2, No. 3–4.
- Manzini R., Pareschi A. A decision-support system for the car pooling problem // Journal of Transportation Technologies. — 2012. — Vol. 2.
- Martino S., Galiero R., Giorio C., Ferrucci F., Sarro F. A matching-algorithm based on the cloud and positioning systems to improve carpooling // DMS, Knowledge Systems Institute. — 2011. — P. 90–95.
- Masum A., Shahjalal Md., Faruque F., Md. Sarker I. Solving the Vehicle Routing Problem using Genetic Algorithm // International Journal of Advanced Computer Science and Applications. — 2011. — Vol. 2, No. 7. — P. 126–131.
- Meng Q., Zhang J. Optimization and application of artificial intelligence routing algorithm // Cluster Computing. — 2019. — Vol. 22. — P. 8747–8755.
- Mulders C. Carpooling, a vehicle routing approach // Université catholique de Louvain, Thesis submitted for the Master in Computer Science and Engineering, option Artificial Intelligence. — 2012. — P. 1–52.
- Rathod S., Pavankumar M., Ravindra A., Sandip S., Kiran J. Carpooling using Genetic Algorithm // International Journal of Engineering and Management. — 2016. — Vol. 6, Iss. 3. — P. 127–129.
- Schreieck M., Safetli H., Siddiqui Ali S., Pflügler C., Wiesche M., Krcmar H. A Matching Algorithm for Dynamic Ridesharing // Transportation Research Procedia. — 2016. — Vol. 19. — P. 272–285.
- Sharma S., Pal B. Shortest Path Searching for Road Network using A* Algorithm // International Journal of Computer Science and Mobile Computing. — 2015. — Vol. 4, Iss. 7. — P. 513–522.
- Takama N., Loucks D. Multi-level optimization for multi-objective problems // Applied Mathematical Modelling. — 1981. — Vol. 5. — P. 173–178.
- Wan W., Birch J. B. An Improved Hybrid Genetic Algorithm with a New Local Search Procedure // Journal of Applied Mathematics. — 2013. — Vol. 2013. — Article ID 10359.

- Yiu Y., Du J., Mahapatra R.* Evolutionary Heuristic A* Search: Heuristic Function Optimization via Genetic Algorithm // IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), Laguna Hills, CA. — 2018. — P. 25–32.
- Zeng W., Church R.* Finding shortest paths on real road networks: the case for A* // International Journal of Geographical Information Science. — 2009. — Vol. 23, Iss. 4. — P. 531–543.
- Zhang D., He T., Liu Y., Lin S., Stankovic J.* A Carpooling Recommendation System for Taxicab Services // IEEE Transactions on Emerging Topics in Computing. — 2014. — Vol. 2, Iss. 3. — P. 254–266.

