

RESEARCH

Open Access



Multiobjective route finding in a multimode transportation network by NSGA-II

Hamed Faroqi^{1*}

*Correspondence:
h.faroqi@uok.ac.ir

¹ School of Civil Engineering,
University of Kurdistan, Sanandaj,
Iran

Abstract

Route finding is an everyday challenge for urban residents. While many route planner applications exist, they cannot find suitable routes based on user preferences. According to user preferences, routing in a multimode urban transportation network can be considered a multiobjective optimization problem. Different objectives and modes for transportation, along with many routes as decision elements, give rise to the complexity of the problem. This study uses an elitism multiobjective evolutionary algorithm and the Pareto front concept to solve the problem. The data of a simulated multimode network consisting of 150 vertexes and 2600 edges are used to test and evaluate the proposed method. Four transport modes are considered: the metro, bus, taxi, and walking. Also, three minimization objective functions are considered: expense, discomfort, and time. The results show the competence of the algorithm in solving such a complex problem in a short run time. The optimal setting for the algorithm parameters is found by considering the algorithm run time, diversity of solutions, and convergence trend by running sensitivity analyses. A repeatability test is applied using the optimal setting of the algorithm, which shows a high level of repeatability. While NSGA-II (Non-dominated Sorting Genetic Algorithm II) may be a well-established algorithm in the literature, its application in multiobjective route finding in multimode transport networks is unique and novel. The outcomes of the proposed method are compared with existing methods in the literature, proving the better performance of the NSGA-II algorithm.

Keywords: Heuristics, Genetic algorithms, Multiple objective programming, Decision support systems

Introduction

Route planning is one of the daily decisions of every citizen. The parameters and priorities each person considers when selecting their path can vary based on their characteristics: age, occupation, financial status, education level, etc. Individuals' priorities and objectives would differ, but these objectives may be limited to minimizing the route cost, discomfort, time, and distance. Available travel modes in urban areas have different characteristics. The three significant modes in the public transport system of metropolitans usually are the metro, taxi, and bus. Usually, people assume the metro as a fast mode, the taxi as a comfortable mode, and the bus as a low-cost mode. Multiobjective routing in the multimode network means finding a route between two points using

different modes while optimizing desired objectives simultaneously. The availability of other modes and different individual concerns make planning a complex and multiobjective problem.

The routing problem is one of the classic problems in network analysis and GIS domains, which has been the subject of much research [1–4]. Single objective methods have already been used to solve problems like Travel Salesman Problem (TSP) and single-mode and multi-mode routing. Multiobjective practices can incorporate different objectives and preferences of the users and are consequently more acceptable. Multiobjective routing in multimode networks means searching among a large space of routes and finding a group that simultaneously optimizes different objectives. Deterministic methods cannot be used for such a problem because of the low speed of these methods and the complexity of the problem-solving space. On the contrary, meta-heuristic strategies can overcome such complexities. These methods usually offer a group of optimal solutions, called Pareto Front, which are not superior to (completely dominated by) each other.

This study develops an Elitism MultiObjective Evolutionary (EMOE) algorithm to solve a multiobjective routing problem in a multimode public transport network. The routing objectives are to minimize trip expense, time, and discomfort. The modes of the metro, bus, and taxi are considered, and the walking mode is used for exchanging between the modes. The proposed algorithm is implemented and evaluated using the simulated data of a large network (150 vertexes and 2600 edges). While NSGA-II (Non-dominated Sorting Genetic Algorithm II) may be a well-established algorithm in the literature, its application in multiobjective route finding in multimode transport networks is unique and novel. In the next section, the existing methods and research are reviewed. In [Methods/experimental](#) section, the algorithm's principal elements and structure are introduced. [Results and discussion](#) section describes the implementation of the proposed method for solving the problem. Finally, the main achievements of the study, along with conclusions and recommendations, are provided in [Conclusions](#) section.

Review of related research

In general, multiobjective decision-making algorithms are divided into three categories: without weighting, weighting before solving, and weighting after solving. In without weighting methods, no preference among factors is declared by the decision-maker. Instead, an ideal state is defined, and the criterion for selecting the options is the closeness to this perfect state [5]. In weighting before methods, the objectives are first weighted by the user and then combined to form a single objective [6]. In contrast, a collection of optimal solutions is obtained first in weighting after solving methods. Then, using the preferences declared by the user, the final solution can be selected [7]. Therefore, the third category can be more user-friendly by providing desirable and competitive choices for each end user.

Most research on shortest-path finding in multimode networks is based on a single objective and limited transportation networks [8, 9]. Batista et al. offered a heuristic "Path Composition Approach" method that finds the suitable path in a two-mode network based on user preferences. The basis of this method is dividing the whole way into some sub-paths that are recombined to make the final paths. In fact, in this method,

the idea that “each part of the optimal path is itself an optimal path” is used [10]. Modesty et al. presented a framework for the problem of multiobjective routes in multimode transport networks. In this method, all possible routes are searched, and eventually, the routes that optimize each respective objective are selected and offered to the user [11]. Golnarkar et al. presented a deterministic method based on fuzzy weighting for the weight of multimode network edges implemented on a small and single objective simulated network [12]. Most deterministic algorithms for solving multiobjective problems combine objectives into one objective (weighting before solving) and produce a single optimal route in each run. Therefore, having an optimal solution group (suggestion of better routes) is not feasible in a single objective algorithm run.

Evolutionary algorithms are implemented and evaluated for solving single objective optimization problems from different viewpoints. In comparison, a few researches are available on applying evolutionary algorithms for solving multiobjective routing problems regarding traffic planning, route design, transport, etc. For the first time, in 1997, a genetic algorithm was used with constant-length chromosomes to solve the routing problem [13]. Jene and Leen studied the usage of a hybrid multiobjective genetic algorithm to minimize expense and maximize traffic flow in a two-mode network. This study demonstrates better results for the evolutionary algorithm than the deterministic methods [14]. Mounely et al. studied and reported on the adequacy and efficiency of their developed evolutionary algorithm for solving multiobjective shortest path problems (MSPP) in a GIS environment. The results of their implementation on simulated data and actual road network data were better than the Dijkstra algorithm in terms of implementation time and convergence towards the Pareto solution front [15]. Abbaspour and Samadzadegan used a genetic algorithm to solve the shortest path problem in a two-mode transport network [16]. In addition to this research, some applied systems are developed for single objective multi-mode route planning [17, 18].

The main limitation of many of these researches in the existing literature is their inefficiency when dealing with the data of real networks. The multimode nature of the network and the need for multiobjective routing dramatically increase the problem's computational complexity. In dealing with such complex issues, meta-heuristic methods are usually more promising than deterministic ones because they can provide a set of optimal solutions instead of one single solution. Therefore, the proposed method in this study can overcome the limitations of traditional and deterministic route-finding algorithms.

Methods/experimental

Main concepts and definitions of the EMOE algorithm

The natural evolution process inspires evolutionary algorithms in engineering. They are used for solving optimization problems with large search space. The evolutionary algorithm starts with a random initial population of solutions and continues with generating new and better answers from the previous ones using some operators. In the elitism evolutionary algorithm, the superior solutions from the population of each generation are preserved and transferred to the next generation as the elite population.

In multiobjective problems, users usually prefer a group of solutions, among which they can select according to their preferences. Using algorithms like EMOEA (EMOE Algorithm), the final result is a group of optimum solutions such that none is entirely superior to (dominated by) any other, considering all objectives. In Fig. 1, the main stages of the EMOE algorithm are presented.

In EMOEA, like other evolutionary algorithms, each generation consists of a solutions group, and each solution is a chromosome (can be imagined as a vector). Each chromosome consists of independent units called genes, which are the components or parts of a solution. The algorithm procedure consists of initialization, main loop (operators), and determining non-dominated solutions. Initialization is the assignment of initial values to the parameters of the algorithm, which are the number of chromosomes in each generation, the algorithm's break conditions, and the rates for crossover and mutation operations. Each generation contains some solutions evolved from the previous generation by the operators [19].

When a new generation is created, the break conditions should be tested. A typical break condition is to reach a pre-defined maximum number of iterations for the main loop. The second condition for stopping the main algorithm loop can specify the minimum difference between the best solutions competencies in two consecutive generations [7]. Reaching this minimum difference can indicate that the algorithm cannot improve the solutions further.

Finally, the concepts of domination, non-dominated solutions, and the Pareto front are used to determine the best solutions. According to the definition, solution X_1

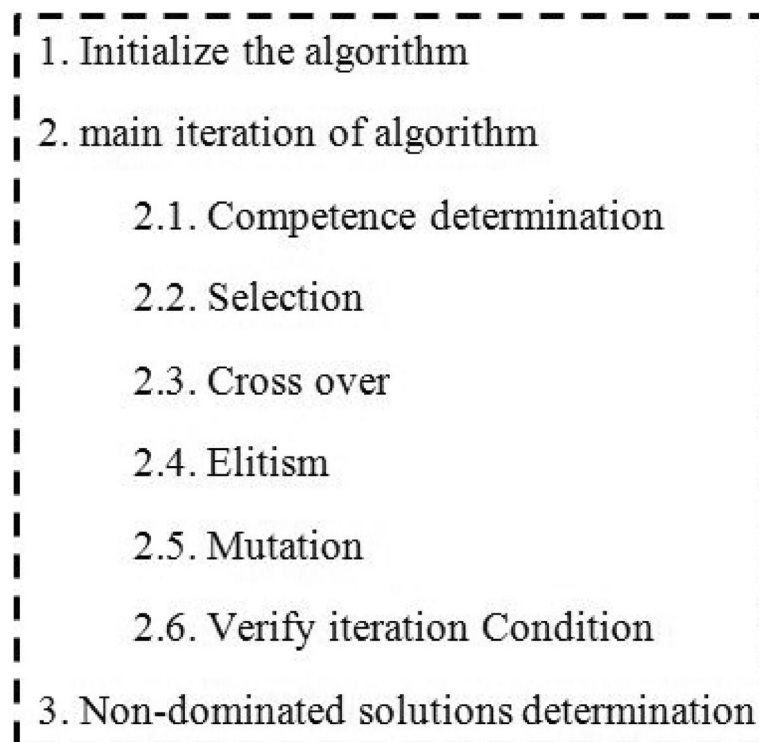


Fig. 1 Pseudo code of elitism multiobjective evolutionary algorithm

dominates solution X_2 if and only if X_1 is not worse than X_2 in all objectives and X_1 is better than X_2 at least in one objective function [6]. In Fig. 2, the area of solutions dominated by a solution X_1 is shown as a rectangle in the two-dimensional space of minimization objective functions F_1 and F_2 . A solution is a member of the first level of the Pareto front if and only if any other solution does not dominate it. In the Pareto front, the solutions are placed in some solution fronts according to their objective values. The solutions categorized in each front are not dominated by each other. However, the solutions of each front are dominated by higher front solutions. They also dominate the solutions of lower fronts.

As already mentioned, at each loop of the second stage of the algorithm, a new generation is created by applying the operators on the previous generation, and at the end of the loop, break conditions are tested. The procedures related to these operators are described in the following.

Each solution has a specified value for each objective function. Competence-determination operator compares all solutions' objective values and determines the Pareto Fronts' first front using the earlier definition. The rest of the solutions are then compared to select the second front, and it continues until determining all fronts. After finishing this process, each solution's front level is determined [6].

The selection operator selects the parent solutions for the production of new solutions, based on pre-specified values of crossover rate. The parents are selected probabilistically with probabilities proportional to the Pareto level of the solutions. The crossover operator combines the two parent solutions, which creates two offspring solutions as members of the new generation [19].

After crossover, the number of resulting solutions exceeds the specified number of generation solutions. To reduce their number, an elitism operator is used. The elitism operator ranks the available solutions according to two factors of competence (Pareto front number) and swarm distance for selecting the required solutions based on this ranking. The swarm distance of each solution means the density of other solutions situated in its vicinity within the objectives space [19].

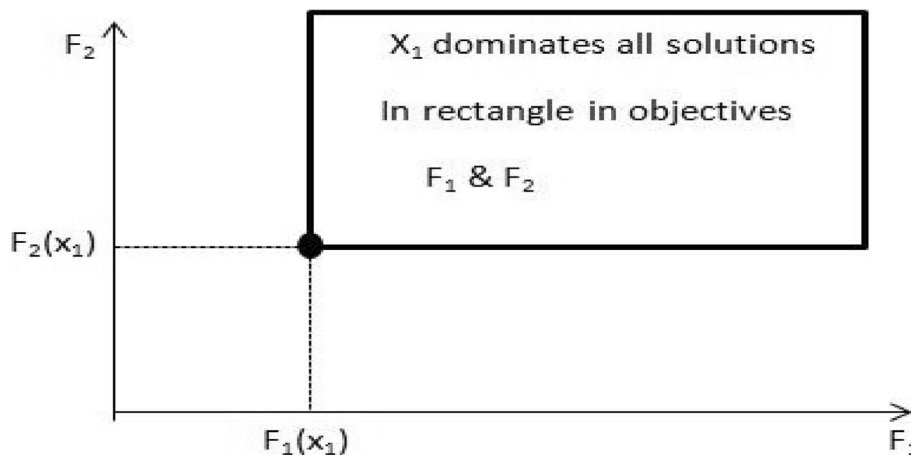


Fig. 2 Dominated solution group for two objective functions

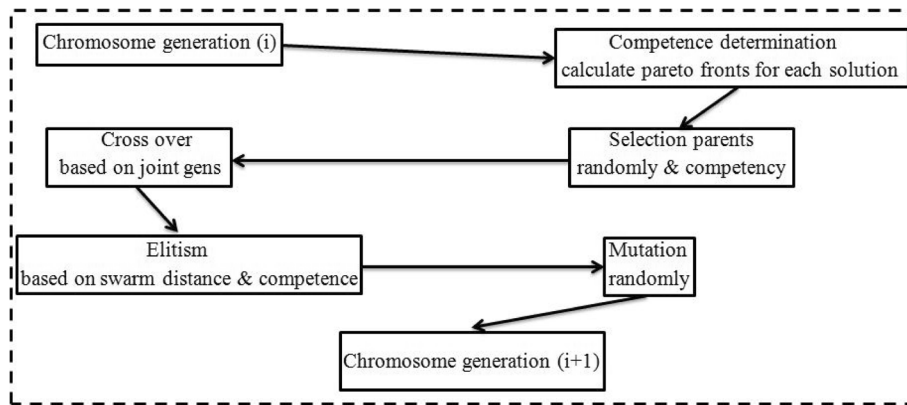


Fig. 3 Sequence of operators in the main loop of the EMOE algorithm

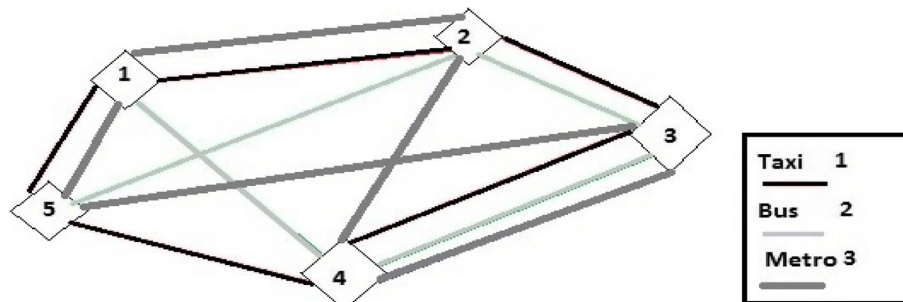


Fig. 4 An illustration of a multimode network

The mutation operator selects the solutions randomly based on a predefined mutation rate and then chooses a gene of that solution randomly. Another gene replaces the selected gene. Fig. 3 shows the sequence of operators in the algorithm’s main loop.

Using EMOE algorithm to solve the multimode route planning problem

This study aims to use the EMOE algorithm to find optimal routes that satisfy three objectives simultaneously in a multimode transport network. The route consists of a sequence of vertexes (nodes) that starts with a specified origin vertex and ends with a destination vertex without crossing any vertex more than once. A multimode transport network comprises different modes such as cars, buses, and walking. Similarly, a multimode route can be defined as a route comprising some parts covered by various modes. The network can be presented as a graph in which more than one edge might exist between two vertexes related to different modes. For any existing path between two vertexes with a specified mode, an edge is assumed with a specified code for that mode and three weight values for the three objectives. The walking mode is assumed in each vertex to exchange between the modes. Figure 4 illustrates a small multimode network with five vertexes and 14 edges for three metro, taxi, and bus modes.

The evolutionary algorithm runs in four main steps. Firstly, it is required to encode the problem into the algorithm’s space which defines the problem by chromosomes (made of genes) and objective functions. The second step is initializing the algorithm, which comprises two steps of valuing the evolutionary parameters and generating a random

generation as generation zero. Thirdly, the main loop of the algorithm starts to create better generations, and it repeats until the breaking conditions are met. Finally, the outcomes of the algorithm must be decoded from the chromosomes to the routes.

The main loop starts with a generation of chromosomes and aims at improving it through its operators. The first operator of the main loop selects parents for crossover and mutation operators. The second operator combines parents to generate new chromosomes. The third operator mutates parents to generate new chromosomes. Both crossover and mutation aim at expanding the search space and adding more diversity to the pool of chromosomes. The fourth step selects and removes some of the chromosomes to decrease the population to the initial size. Finally, the breaking conditions are verified to decide on repeating the loop or stopping it.

Each chromosome in the evolutionary algorithm is a solution comprised of genes. It defines a route as a sequence of vertexes between two specified origin and destination vertexes. Considering these, the number of each vertex can be regarded as a gene, and a route, which is the sequence of vertexes, can be regarded as a chromosome. The length of each route or chromosome is different and limited by the total number of network vertexes. In a multimode route, to define the route thoroughly, we need to declare both the vertex numbers and the mode of the edge between any consecutive vertexes.

Therefore, the odd genes in each chromosome may be assigned to the index of vertexes and even genes to the modes used between two consecutive vertexes. In Fig. 5, two different routes or chromosomes are specified for travel between vertexes 1 and 3 of Fig. 4. Numbers of modes are presented in a bold quadrangle and located in even genes.

The competence of each solution is calculated according to its front. To determine the competence of each chromosome, its front level should be determined. First, the values of all three objective functions related to discomfort, time, and expense are calculated. According to the concept of the Pareto front, the solution front related to each route is determined [19]. Figure 6 presents the method for defining the domination between two solutions, including three objective functions. In this figure, $X_1 (E_1, C_1, T_1)$ denotes route X_1 , and $E_1, C_1,$ and T_1 represent values of the three objective functions for route X_1 .

After determining the competence of each solution, it is time to select existing solution chromosomes for crossover and mutation operations. According to the “roulette wheel” method of selection operator, the area of a circle is divided among the solutions, proportional to their competence value. In single objective problems, the width of the section allocated to each solution is proportional to its objective function. However, in multiobjective problems, the width of this section is proportional to the solution competence, which is the level of its front. In the selection stage, each one of the routes is selected to be a parent probabilistically, with probabilities proportional to its competence [7, 19].

The crossover operator creates two new child chromosomes by combining parts from both parent chromosomes. Each chromosome is supposed to be a complete route between the origin and destination vertexes. Therefore, for connecting some parts of the



Fig. 5 Two chromosomes with different lengths between vertexes 1 and 3

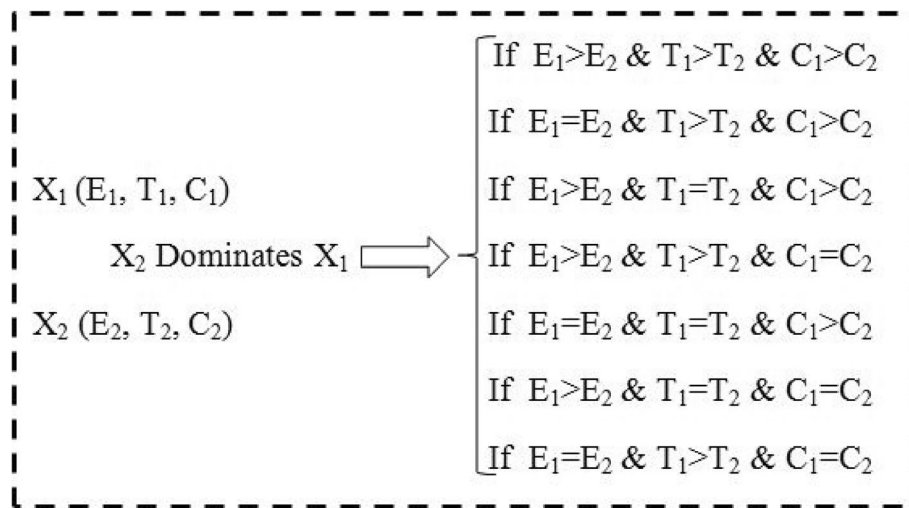


Fig. 6 Determination of X_2 solution domination over X_1 with three objective functions

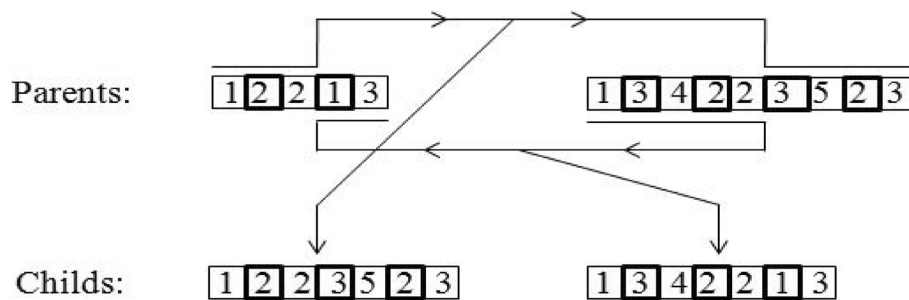


Fig. 7 Crossover operator

two parent routes, their common vertexes or genes along the two routes are needed to be found and used. At first, the first similar vertex (excluding origin and destination) between the two routes is found. Any of the two routes is split at this vertex. Then, the parts between the two routes are switched to create two new routes between the origin and destination. At the crossover stage, the number of newly created solutions equals the product of the crossover rate and the number of generation population. Figure 7 shows an example of a crossover operator for the two routes of Fig. 5.

In elitism, two factors of competence and swarm distance are considered. According to Outcomes section, the competence of each route is proportional to the level of its front. A route’s swarm distance equals the density of solutions around it within a three-dimensional space of objective functions. The solutions are ranked according to their competence. For organizing two chromosomes with equal competence, a chromosome with a lower swarm distance is prioritized; this means that more attention will be paid to the less-explored area of the search space. After ranking all the routes using these two factors, the routes are selected in order from the top of the list up to the number of the initial population count.

The mutation operator randomly selects one route and one of its vertexes and then replaces the selected vertex with a randomly found new vertex. The replacement is

subject to the availability of edges between the new vertex and vertexes before and after the selected vertex in the route. After the completion of the mutation operation, a new generation of routes is created. The algorithm's main loop will be stopped or continued by examining the break conditions explained earlier.

Results and discussion

The proposed methodology is implemented on simulated public transport network data from Tehran, Iran. In the following, the main stages of implementation, along with the results, are explained, and different algorithm implementations are evaluated.

Data collection

The data required to implement the algorithm includes the map of the transportation network and the values of the objective functions assigned to the respective network. The case study is located in the central district of Tehran, Iran, as a rectangle with coordinates of $\{(35.68404, 51.34352), (35.68571, 51.45682), (35.75595, 51.45613), (35.75651, 51.34009)\}$: This area covers the central Bazar of Tehran and includes many educational buildings and organizations. The final simulated network consists of 150 vertexes (intersections of roads) and covers approximately an area of 80 square kilometers: Each vertex is a station where edges of different modes may enter or exit there/ from. If more than one mode (edges) converge to a certain vertex (station), the interchange between the modes is assumed to take place through a walking mode (edge); the expense of walking mode is zero, and a constant value of 3 min is considered for the time value of a walking edge. The network contains 120 edges for the taxi mode, 45 for the bus mode, 25 for the metro mode, and 70 for walking between different modes. The value of the expense objective function for the taxi mode is proportional to the traveled distance: The traveled distance is in km units, and the expense for each trip leg of the taxi mode is 75 (money) units per km. For the metro and bus modes, this value is proportional to the number of stations in that mode and edge; for the metro, the expense is 35 units per station; for the bus, the expense is 20 units per station. The value of the time objective is calculated based on the distance between vertexes and the average speed assumed for that mode: According to the traffic congestions in the study area, the average speed for the taxi is considered as 15 km/hour, for bus 10 km/hour, and metro 20 km/hour. The objective function of route discomfort assumes a predetermined value for each type of mode, which is 1 for taxi, 2 for metro, 3 for bus, and 4 for walking; Taxi is the most comfortable mode.

Outcomes

Date sets are gathered (in Shapefile format) and preprocessed (converted to CSV format) by QGIS. The Rstudio environment is the programming environment to implement the proposed method: Libraries from "rmo" package are mainly used. Three 2-dimensional diagrams show the Pareto front in the space of three assumed objective functions for each implementation. To implement the algorithm on the simulated network, it is required to determine the break conditions and set the initial values of the crossover rate, mutation rate, and solutions population number. Initialization values of those parameters are selected after running sensitivity analyses. The first break condition is to reach

the maximum iteration number of the main loop. Also, the algorithm will stop if the difference between objective function values of two consecutive generations becomes less than 3. The crossover and mutation rates in each generation are set to 0.1, and for other parameters, different values are considered in several algorithm implementations.

To analyze the algorithm results, two vertexes with the longest possible distance on the network are assumed as the origin and destination, and the algorithm is implemented with different parameters of solutions population and iteration numbers. Table 1 summarizes the results for nine algorithm implementations with different parameter values for the two vertexes. In general, upon increasing the iterations count and the number of the initial solutions population, algorithm run time is increased, more routes are found to be at the first front, and the total routes found are increased. Run number 6 found nine non-dominated solutions in the first front. Run number 8 found 51 routes, the maximum number of found solutions, of which five routes were situated at the first front. Runs 5 and 9 are stopped based on the second break condition, i.e., the difference between objective function values of two consecutive generation routes.

Based on the concept of non-dominated solutions and to illustrate the relationship between routes obtained from different implementations, the diagrams related to objective function values of routes in the first front are presented in Fig. 8; for easier comparison of the charts, equal ranges are assumed for all of the horizontal axes (objectives) in all graphs: The expense function values are drawn within the range (216,315), time function values within the range (51, 69), and discomfort function values within the range (28, 46). The non-dominance feature of the routes is defined according to Fig. 6 rules and is observable in the values of objective functions in Fig. 8. In Fig. 8, each little square represents a solution in the first front of a run. Numbers above the little squares represent the ID of each solution to compare across different objective functions.

Run 6 is the only implementation that includes solutions in the whole range of all three objectives and has searched and covered a larger area in the objective function space (larger search area). Also, runs 5, 8, and 9 have explored a relatively wide range of spaces. In comparison, run 5, having only four routes at the first front, has a better distribution in searching the objective functions space. The size of the searched space in objective functions space and the variety of the non-dominated solutions provides the user with a better condition for selecting amongst solutions and initial parameters.

Table 1 Results of implementations with different solutions population and iteration numbers

No of run	Population no	Maximum iteration number	No found routes	No of routes in 1 st front	Run time (seconds)	Iteration stop
1	30	50	21	4	5.5	50
2	30	500	18	4	5.8	500
3	30	1000	16	6	6.0	1000
4	60	50	42	3	9.6	50
5	60	500	28	4	9.9	152
6	60	1000	38	9	11.6	1000
7	80	50	48	4	12.8	50
8	80	500	51	5	13.3	500
9	80	1000	50	6	11.9	883

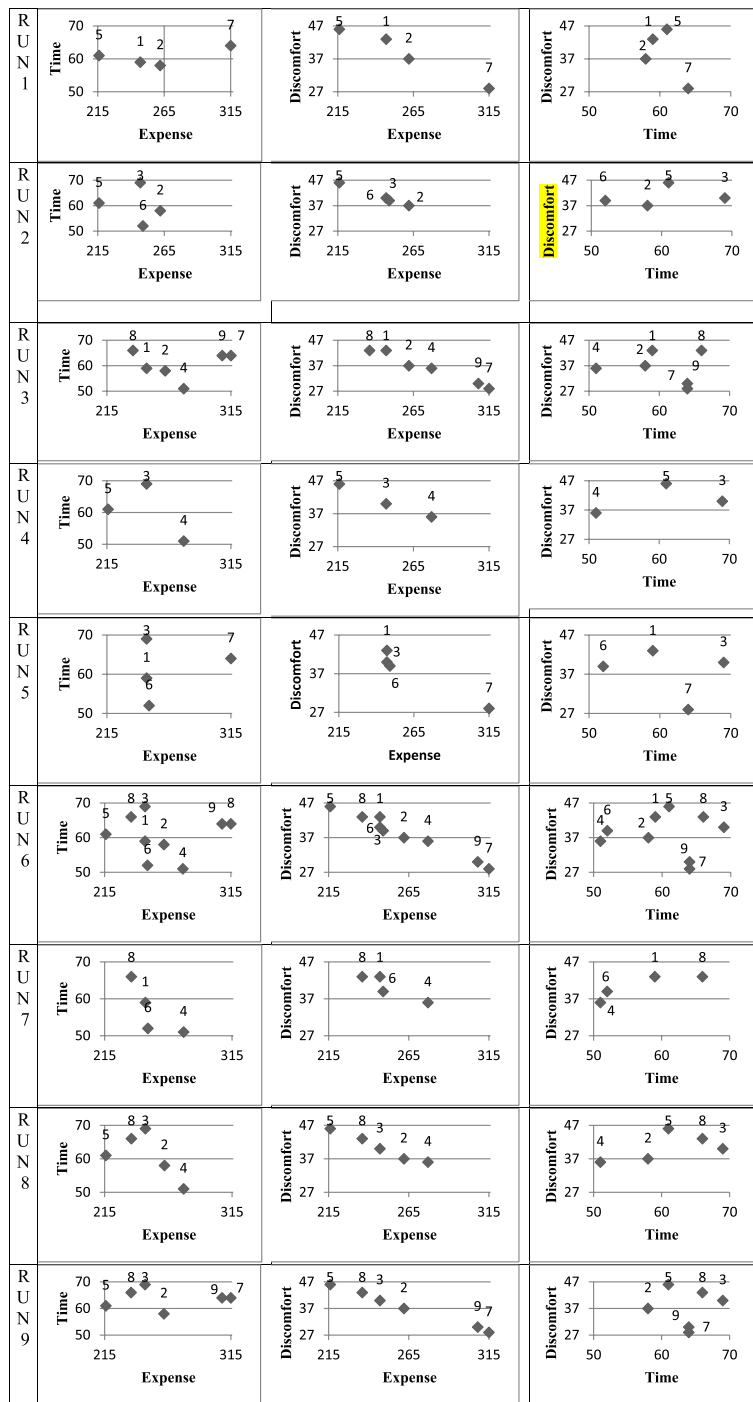


Fig. 8 Pareto fronts of all runs presented in two-dimensional space of objective functions (elongated squares are two or more overlapped regular squares)

Table 2 specifies the minimum values found for any objective function for all implementations. The minimum values for all objective functions are bolded. Also, the route number (found solution/chromosome) related to objective values is presented in parentheses. Run 6 has found the minimum values in all objective functions. Runs

Table 2 Minimum values found for the objective functions in runs

Run NO	Min expense _(path no)	Min time _(path no)	Min discomfort _(path no)
1	216 ₍₅₎	58 ₍₂₎	28 ₍₇₎
2	216 ₍₅₎	52 ₍₆₎	37 ₍₂₎
3	236 ₍₈₎	51 ₍₄₎	28 ₍₇₎
4	216 ₍₅₎	51 ₍₄₎	36 ₍₄₎
5	247 _(1,3)	52 ₍₆₎	28 ₍₇₎
6	216 ₍₅₎	51 ₍₄₎	28 ₍₇₎
7	236 ₍₈₎	51 ₍₄₎	36 ₍₄₎
8	216 ₍₅₎	51 ₍₄₎	36 ₍₄₎
9	216 ₍₅₎	58 ₍₂₎	28 ₍₇₎

1, 3, 4, 8, and 9 found minimum values for two objective functions. Runs 2, 5, and 7 have found a minimum value for just one of the objective functions. Therefore, run 6 can be assumed as the best implementation due to its extent of search in solutions and objective functions space, diversity, and number of routes found in the first front, and minimum values found for all objective functions.

The convergence test is used to study the convergence trend of the results obtained from the algorithm at each run. The convergence test analyzes the variations of objective function values over time (increase of iterations). Figure 9 shows the convergence diagrams regarding the three objectives for all runs. Each chart considered the variation of the minimum values related to the respective objective function. To compare the convergence trend of different algorithm runs, gradient diagrams and the range of accurate values can be considered. The high gradient of the graph (steeper graph) shows fast convergence of the algorithm in finding the solutions with minimum objective function values. The convergence diagram of the runs is mostly descending or stepping towards the objective functions' lower values. The gradient of runs 1 and 7 are almost constant in all three objective parts. A wider range of objective values in each diagram demonstrates better covering of the search space by that run. The best convergence trend in the chart related to the time function is related to runs 4, 8, and 6. In the expense objective function diagram, runs 1, 6, and 9 show the maximum gradient with maximum variation range. In the convergence diagram of the discomfort objective function, runs 9, 3, and 6 show the best convergence trend. Therefore, run number 6 is assumed as a good run in the convergence diagram of all three objective functions.

According to the previous sensitivity analyses, run 6 can be chosen as an optimal run. It has found more routes in the first front than others. Also, run 6 in Table 2 found the minimum values of each objective function compared to others. For the repeatability test, run 6 is implemented ten times with the same initial parameters and origin and destination points. The result of the repeatability test is shown in Fig. 10. The count of routes found at each run is presented on the horizontal axis, and the counts of the runs, with that number of routes in the first front, are specified on the vertical axis. The figure shows that four iterations found nine routes, and four others found eight courses in the first front. The other two runs found 7 and 6 routes. This indicates 80% repeatability for the algorithm, meaning that in 80% of the situations, the algorithm finds similar results.

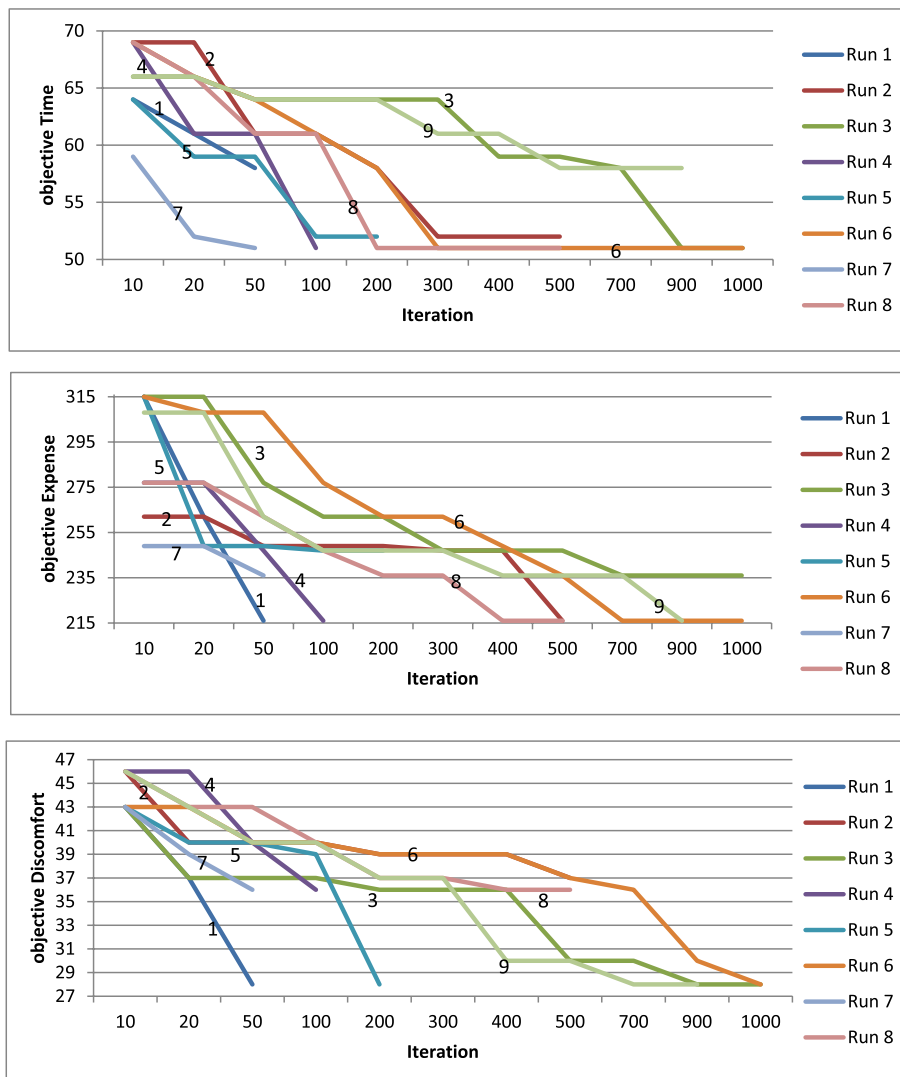


Fig. 9 Convergence trend of all runs regarding each objective function

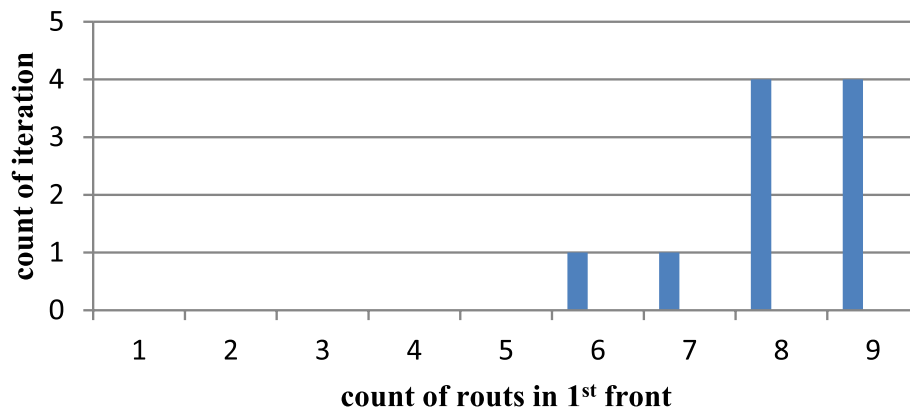


Fig. 10 The results of the repeatability test

For evaluation, the proposed method is compared with the methods developed in [20]. And [20] focused on two variations of ant colony optimization (ACO) algorithms, multi-pheromone and multi-colony, for solving a similar route finding problem. Following [20], two metrics of error ration (ER) for convergency [21, 22] and spacing metric (SM) [23, 24] for diversity purposes are used to compare the performance of the algorithms. Table 3 summarizes the evaluation outcomes. NSGA-II has better performance in both convergency and diversity metrics than two variations of multiobjective ACO algorithms.

Conclusions

Multiobjective routing in multimode networks is a complex problem. This might be due to the presence of different modes, various and even incompatible objectives, numerous possible combinations of edges in creating the routes, etc. Using deterministic algorithms for solving such complex problems usually results in difficulties such as non-convergence, long running time, and failure in finding optimal solutions. In this study, an elitism multiobjective evolutionary algorithm is used to find the optimum routes, which overcome the shortcomings of deterministic algorithms. It has found optimal routes within proper and acceptable running time while simultaneously considering three objectives: minimizing time, discomfort, and expense. The elitism technique used in the offered algorithm leads to the fast convergence of the algorithm, even when using different initial parameter values. The outcomes of the proposed method are compared with existing methods in the literature, proving the better performance of the NSGA-II algorithm. While NSGA-II may be a well-established algorithm in the literature, its application in multiobjective route finding in multimode transport networks is unique and novel.

The proposed method runs a large multimode network. The outcomes present the suitability for implementing the proposed heuristic as a core algorithm of the trip planner applications. Finding several optimal solutions as algorithm output would allow end-users to select routes according to their preferences of objectives. The algorithm uses the Pareto front diagrams for presenting the optimal front solutions, resulting in a better understanding of the solutions status by the user. These diagrams present the values achieved for all three objective functions in all optimal solutions. Using these diagrams, the user can compare the solutions in terms of meeting different objectives and select the most appropriate route according to his preferences. The use of evolutionary algorithms is a promising approach to solving routing problems. Yet, further research is required on adapting this technique in different scenarios of route planning.

Table 3 Evaluation outcomes

Algorithm	ER	SM
NSGA-II	0.516	0.060
Multi-Pheromone ACO	0.599	0.066
Multi-Colony ACO	0.569	0.064

Future research could focus on three directions. Firstly, larger datasets could be used to evaluate the performance of the proposed methodology. Secondly, the proposed method could be further developed to use prior knowledge as input for a better understanding of users' preferences. Finally, other heuristic algorithms (e.g., particle swarm optimization (PSO)) could be studied to solve the routing problem and integrate with the NSGA-II algorithm.

Abbreviations

EMOE	Elitism MultiObjective Evolutionary algorithm (the main algorithm developed by this study)
NSGA-II	Non-dominated Sorting Genetic Algorithm II
ACO	Ant colony optimization
ER	Error ration
SM	Spacing metric
PSO	Particle swarm optimization

Glossary

Chromosome	A set of parameters that define a proposed solution to the problem that the EMOE algorithm is trying to solve. It is formed by a combination of genes.
Crossover Operator	It combines the gene information of at least two parents (chromosomes) to generate new offspring (chromosomes).
Gene	The property of each element that makes up the chromosomes.
Mutation Operator	It maintains the diversity of the population by tweaking the chromosomes.
Objective Function	A mathematical formulation to quantify how good chromosomes are.
Population	A set of chromosomes.
Repeatability Test	A test for examining the reliability of the EMOE algorithm.
Swarm Distance	A measure for crowdedness and density of chromosomes in the space of objective functions in the EMOE algorithm.

Research involving human participants and/or animals

There is no involvement with human participants and/or animals.

Author's contributions

HF: conceptualization, data collection, methodology, programming, writing.

Funding

There is no funding.

Availability of data and materials

The datasets will be shared upon request.

Declarations

Competing interests

There are no conflicts of interest.

Received: 26 December 2023 Accepted: 19 March 2024

Published online: 25 March 2024

References

1. Barrett CL, Bisset K, Jacob R, Konjevod G, Marathe MV (2002) Classical and contemporary shortest path problems in road networks: implementation and experimental analysis of the TRANSIMS router. In: *Lecture Notes in Computer Science*, vol 2461. Springer, Heidelberg, pp 126–138
2. Aifadopoulou G, Ziliaskopoulos A, Chrisohoou E (2007) Multiobjective optimum path algorithm for passenger pretrip planning in multimode transportation networks. *Transp Res Rec* 2032:26–34
3. Bérubé J, Potvin J, Vaucher J (2006) Time-dependent shortest paths through fixed sequence of nodes: application to a travel planning problem. *Comput Oper Res* 33(6):1838–1856
4. Azari M, Paydar A, Feizizadeh B, Hasanlou VG (2023) A GIS-based approach for accident hotspots mapping in mountain roads using seasonal and geometric indicators. *Appl Geomat* 15(1):127–139
5. Kaveh M, Kaveh M, Mesgari MS, Paland RS (2020) Multiple criteria decision-making for hospital location-allocation based on improved genetic algorithm. *Appl Geomat* 12:291–306
6. Coello Coello CA, Lamount GB, Veldhuizen DA (2007) *Evolutionary algorithms for solving multiobjective problems*, 2nd edn. Springer, New York
7. Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms- a comparative case study. In: *Parallel problem solving from nature*. Springer-Verlag, Berlin

8. Spiess H, Florian M (1989) Optimal strategies: a new assignment model for transit networks. *Transp Res B* 23(2):83–102
9. Nguyen S, Pallotino S, Malucelli F (2001) A modeling framework for the passenger assignment on a transport network with timetables. *Transp Sci* 35(3):238–249
10. Battista, M.G., Lucertini, M., & Simeone, B. (1995). Path composition and multiple choices in a bimode transportation network, In Proceedings of the 7th World Conference on Transport Research Society. Sydney, Australia.
11. Modesti P, Sciomachen A (1998) A utility measure for finding multiobjective shortest paths in urban multimode transportation networks. *Eur J Oper Res* 111(3):495–508
12. Golnarkar A, Aleshekih AA, Malek MR (2010) Solving best path problem on multimode transportation networks with fuzzy costs. *Iran J Fuzzy Syst* 7(3):1–13
13. Gen, M., Cheng, R., & Wang, D. (1997). Genetic algorithms for solving shortest path problems, Proceedings of 1997 IEEE International Conference on Evolutionary Computing, pp. 401–406.
14. Gen, M., & Lin, L. (2004). Multiobjective genetic algorithm for solving network design problem, presented at the 20th Fuzzy Systems Symposium, Kitakyushu, Japan.
15. Mooney P, Winstanley A (2006) An evolutionary algorithm for multicriteria path optimization problems. *Int J Geogr Inf* 20:401–423
16. Abbaspour A, Samadzadegan F (2011) An evolutionary solution for multimode shortest path problem in metropolises. *ComSIS* 7(4):789–804
17. JPL: Journey Planner for London (2009). <http://journeyplanner.tfl.gov.uk>.
18. Faroqi, H., Mesbah, M., & Kim, J. (2018, January). Inferring socioeconomic attributes of public transit passengers using classifiers. In Proceedings of the 40th Australian transport research forum (ATRF).
19. Kalyanmoy D, Pratap A (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *TEVC* 6(2):182–197
20. Faroqi H, saadi Mesgari, M. (2016) Performance comparison between the multi-colony and multi-pheromone ACO algorithms for solving the multi-objective routing problem in a public transportation network. *J Navig* 69(1):197–210
21. Van Veldhuizen DA (1999) Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. *J Evol Comput* 8(2):125–147
22. Fasina E, Sawyerr BA, Abdullahi YU, Oke SA (2023) A comparison of two hybrid optimization techniques: the Taguchi-BBD-firefly and the Taguchi-regression-firefly methods on the IS 2062-E250 steel plates boring problem. *J Eng Appl Sci* 70(1):47
23. Schott, J. (1995). Fault tolerant design using single and multicriteria genetic algorithms optimization. Department of Aeronautics and Astronautics (No. AFIT/CI/CIA-95–039). Air Force Inst of Tech Wright-Patterson AFB OH
24. Abdelati MH, Abd-El-Tawwab AM, Ellimony EEM, Rabie M (2023) Solving a multi-objective solid transportation problem: a comparative study of alternative methods for decision-making. *J Eng Appl Sci* 70(1):82

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.