Engineering Reports

WILEY

Engineering Reports Open Access

RESEARCH ARTICLE OPEN ACCESS

Genetic Algorithm Based Multipath Optimization for Multimobile Robot Navigations

K. Somasundaram[1] | Juha Plosila[2]

[1]Department of Mathematics, Amrita School of Physical Sciences, Amrita Vishwa Vidyapeetham, Coimbatore, India | [2]Department of Computing, University of Turku, Autonomous Systems Laboratory, Turku, Finland

Correspondence: K. Somasundaram (s_sundaram@cb.amrita.edu)

Received: 8 October 2025 | **Revised:** 21 January 2026 | **Accepted:** 3 February 2026

Keywords: genetic algorithm | mobile robots | multipath optimization | networks

ABSTRACT

Multimobile Robot Flow Network Problem (MM-RNP) is to find optimum navigation paths in a network without robot collisions. Very few works have been done for the multirobots path allocation problem. Most of the work concentrated on the single robot path allocation problem. This MMRNP is one of the hardest combinatorial optimization problems. MMRNP solutions can be derived using exhaustive enumeration and branch-and-bound linear programming methods. However, the computation required by these procedures is enormous even for a small size problem. In this paper, we present a heuristic approach using Genetic Algorithm (GA) to achieve the near-optimal solution. We propose a new population initialization for our GA with different operators. The proposed GA optimizes path allocation for mobile robots navigating in a static network environment. We follow the edge-level perspective to solve the MMRNP. Our experimental results show that the proposed method gives a better navigation strategy than the traditional methods.

1 | Introduction

In recent years, the navigation of autonomous mobile robots got attention by academicians and industrialists due to the deployment of mobile robots in various places, such as airports, hotels and agricultural fields. Multimobile robots refer to a system where multiple robots work together and coordinate to achieve shared goals. These robots communicate with each other to divide tasks efficiently and improve overall performance. Multimobile robots are used in different areas such as military, traffic control, supply chain industries, and so on. In future, many service sectors will deploy mobile robots to replace the human service due to cost efficiency and other aspects. At the same time, there are several challenges in the deployment of multimobile robots. Figure 1 shows multirobots path navigation. This consists of path planning, obstacle avoidance and coordinated movements. For example, the design of floor planning, job allocations and routing are the major challenges in the real-time implementation. One of the major issues in the multimobile robot's deployment is the path allocations to different robots. These types of multimobile robot navigation systems are embedded in distributed control systems. Sensor-based fusion techniques are used in these multimobile robots. High-sensitivity cameras and monitoring systems are employed in these robots. These robots will share the information such as position, task allocation, and available paths with each other. Also, they use interrobot communication protocols for coordination and data sharing. Globalized mapping for path planning is also required. Efficient algorithms are essential to ensure collision-free mobile robot navigation. Recently, different learning-based methods such as reinforcement learning were studied for these navigation tasks. Various control mechanisms are required to control mobile robots. For example, dynamic path planning and path allocation are the most important aspects in the multirobot navigation system.

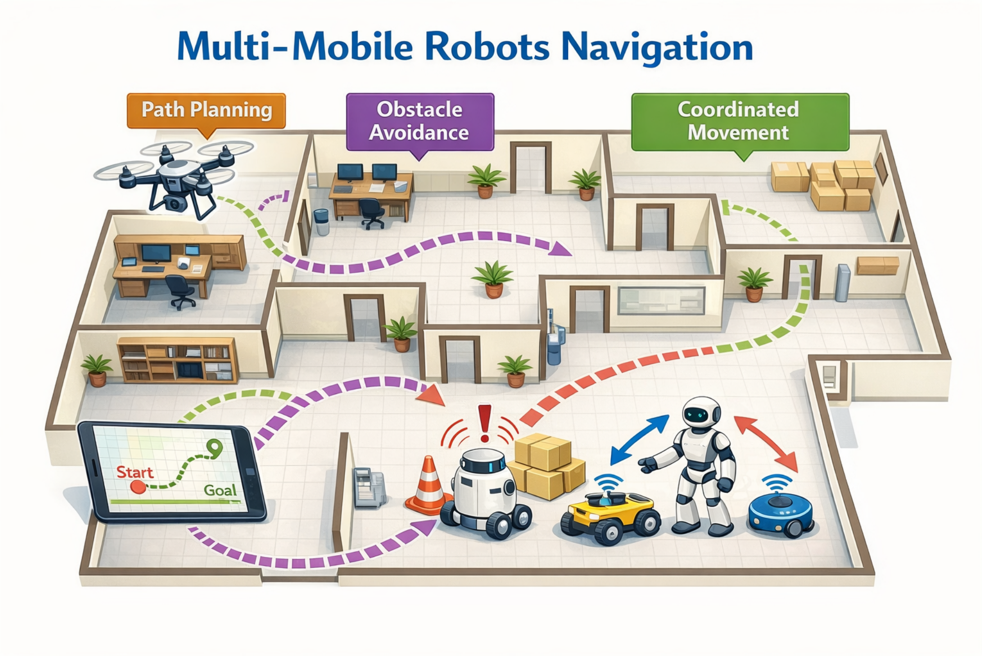


FIGURE 1 | Multirobots navigation.

Poor planning in the path allocation leads to collision of the robots. In the case of service robots in a big restaurant, it is essential to find an optimum path from source places to the destination places (customer places).

Path allocation problem is to determine an optimal path without collision. For single source and single destination problems, they are well studied and are in polynomial time. But the multisource and multidestination problems are much more difficult than single source-destination problems. Determining optimal navigation in an multipath constrained environment is much harder than the simple case. In general, it is an NP-hard problem [1].

In this paper, we address the path allocation problem for autonomous mobile robot navigation. We use the graph model for our problem. Our proposed algorithm will work for any type of mobile robot navigation system like agricultural field, industrial service, and restaurant service. Network flow path problem serves as the basis for several network load balancing algorithms and therefore has been extensively investigated by many researchers. The aim of the maximum network flow algorithm, presented below, is to maximize the path allocation in a network with capacity and balancing constraints. A mobile robot path planning using genetic algorithm in a static environment was shown by Choueiry et al. [2]. Ismail et al. [3] proposed a genetic algorithm for path planning for a single mobile robot in a static environment. In this paper, we propose a genetic algorithm for multimobile robots' navigation in a static environment.

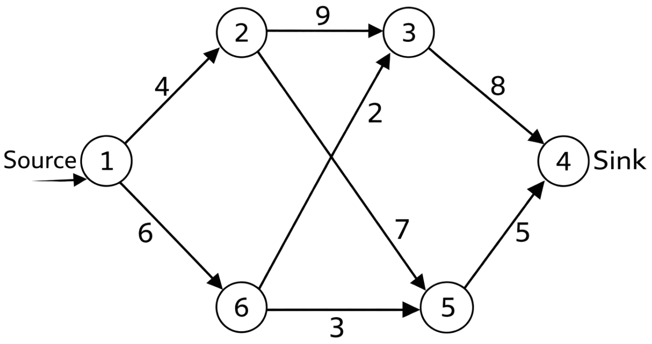


FIGURE 2 | A typical transport network with six nodes and a single robot. The numbers on the edges represent the corresponding edge's capacity.

Figure 2 shows a navigation network with single source and single destination. A graph $G=(V, E, W)$ consists of a set V of nodes/vertices, a set E of links/edges and a set of weights W. Throughout this paper, we assume that the graph network is simple and directed. Any network can be visualized as a directed graph. The weights are assigned to either nodes or links. In a given transport network G, a flow is an assignment of a nonnegative number f_{ij} to every directed edge (i, j) such that the following conditions are satisfied:

1. For every directed edge (i, j) in G

$$f_{ij} \leq C_{ij}. \tag{1}$$

2. There is a specified node s in G, called the source, for which

$$\sum_i f_{si} - \sum_i f_{is} = w, \tag{2}$$

3. There is another specified node t in G, called the destination, for which

$$\sum_i f_{ti} - \sum_i f_{it} = -w. \tag{3}$$

4. All other nodes are called intermediate nodes. For each intermediate node j

$$\sum f_{ij} - \sum f_{ji} = 0. \tag{4}$$

1.1 | Research Question

Is there any optimum way to transport the commodities from different source nodes to different destination nodes using the multiple mobile robots?

This is a paper; we address this problem through graph model.

1.2 | Major Contributions

The following are our major contributions in this paper.

- We proposed a graph mathematical model for the multipath and multirobots navigation problem.
- Our mathematical model is a multiobjective constrained optimization problem. For solving this constrained optimization problem, we proposed new fitness and energy functions and applied them to a genetic algorithm.
- Our results show the proposed genetic algorithm gives us a good navigation strategy.
- The proposed graph-based GA model gives better convergence than the other models.
- We compare our results with some of the existing works.

2 | Literature Review

Efficient path allocation algorithms for autonomous mobile robots reduce the service time, costs, and capital investment. Finding an optimum path in the navigation of the mobile robots is challenging and remains open. Several methodologies have been proposed and reported in the literature for the path planning of mobile robots. There are two major classifications in these algorithms such as global and local path allocations. Ge and Cui [4] proposed a repulsive potential function for mobile robot path planning. The path planning problem belonging to NP-hard classes [1]. There are several soft computing methods for this problem. Ismail et al. [3] proposed a genetic algorithm for a path planning for mobile robot in a static environment. Lamini et al. [5] also proposed a genetic algorithm-based approach for autonomous mobile robot path planning. Tingping Feng et al. [6] proposed an adaptive genetic algorithm to identify the hazard levels during the navigation on the road. Variations in the genetic algorithms were proposed for this navigation problems [7–9]. Ant colony algorithms [3, 5, 10] were proposed for the path planning problems. A particle swarm optimization-based approach. The Table 1 shows the comparison between different techniques. A decentralized fault-tolerant weights based algorithm for coordination of swarm robots for a disaster scenario was discussed in [16]. Several reviews are available in the literature [13, 15, 17]. Ganesan et al. [18] proposed a hybrid sampling based path planning algorithm for autonomous mobile robot navigation.

3 | Multipath Network

3.1 | Problem Formulation

In many practical situations it becomes necessary to deal with several distinct commodities flowing simultaneously through a given transport network. Multimobile robot navigation is one of the ways to transport the commodities in an optimum way. In multimobile robot navigation, each robot has its own sources and its own destinations. Our aim is to optimize multimobile robot navigation such that the robots navigate from their source places to destinations at an optimal cost and without collision. Such a problem is called a Multirobot and Mutipath problem. All the approaches mentioned in the literature review section have used a single path-level perspective to solve the problem. In this paper, we propose an edge-level perspective to solve the problem.

TABLE 1 | Important highlights of the literature review.

References	Path selection	Method	Results
Ismail AL-Taharwa et al. [11]	Single path planning optimization	Genetic algorithm	The fitness of the path is 0.38
Razif Rashid et al. [10]	Mobile robot path planning	Ant colony optimization	Performance evaluation is improved
Mohd Nadhir et al. [12]	Mobile robot path planning in static environments	Genetic algorithm	The average maximum fitness of the best solution for the improved
Tingping Feng et al. [13]		Hybrid Adaptive Genetic Algorithm	The optimal path distance is reduced by 2.74% at the minimum.
Raúl-Alberto and Ernesto [8]	Optimization of layout of the workstations	Genetic algorithm	A reduction of 0.98 s in transfer time was achieved.
Samsuria et al. [9]	Scheduling problem involving mobile robot transportations	Adaptive fuzzy-genetic algorithm The average time taken for the best solutions was 4 *s* per run.	
Junfei Li et al. [14]	Single path optimization	Knowledge-based genetic algorithm	40 obstacles, take an average f 18.96 s.
Samia Choueiry et a. [15]	Single path planning optimization	Genetic algorithm	The fitness of the path is 0.42

In multirobot path navigation networks, all source nodes share the edge bandwidth, and therefore, as in the single-robot navigation case, the sum of all robot flows through an edge must not exceed the bandwidth of the edge. Figure 3 shows a multirobot navigation network with two source nodes and two destination nodes. For each robot, the movement is preserved at every intermediate node. The problem can be easily represented by an adjacency matrix. An adjacency matrix A is a two-dimensional array of integer values where the number of rows and the number of columns are equal to the number of nodes in the network. Thus $A(m, n)$ indicates the number of robots navigating from the node m to the node n. It is assumed that robots do not move within a node itself and hence the diagonal elements are always zero. Every node has the following attributes (Figure 4):

In-Flow: No. of robots coming into the node.

Out-Flow: No. of robots going out of the node.

Excess-Flow: The difference between the In-flow and the Out-flow.

Total-Flow: Average of In-flow and Out-flow.

Balanced: Boolean variable set to TRUE if In-flow = Out- flow else FALSE.

Extra-Flow: No. of robots coming into the network or going out of the network.

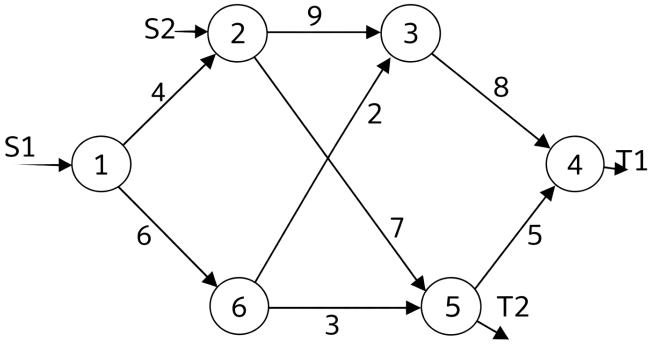


FIGURE 3 | A multirobot navigation network, where S_1 and S_2 are sources and T_1 and T_2 represent the destinations.

Energy: Fitness of a node.

Each Chromosome again has the attributes of Fitness (fitness of the chromosome) and Num_Balanced (number of balanced nodes in the chromosome). Thus, the problem is belonging to a multicriteria decision-making problem.

4 | Proposed Genetic Algorithm

4.1 | Steps Involved in a Primitive Genetic Algorithm

The common evolutionary operators can act on selection, crossover, and mutation. Similarly, the genome must uniquely describe with nodes, links, and weights. If a network with n nodes, then the adjacency matrix A is of size $n \times n$. We encode the adjacency matrix A as a vector of length n^2. The values in the *ith* row in A are mapped to the *ith* array in the vector. For example, Figure 5 shows a simple directed network with four nodes and four links. The corresponding adjacency matrix A and encoding array v are given below;

$$A = \begin{vmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix} \text{and } v = [0110; 0001; 0001; 0000].$$

The genetic algorithm approach includes the following criteria:

1. Initialization
2. Evaluation of fitness and energy

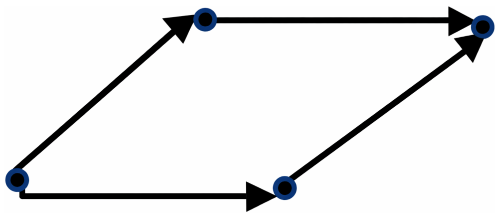


FIGURE 5 | Network.

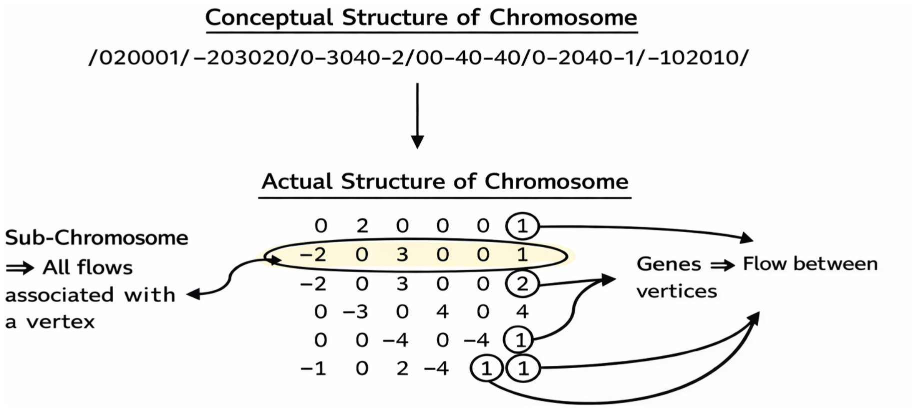


FIGURE 4 | Conceptual and actual structure of a chromosome.

Graph Network
↓
Define encoding scheme (matrix / edge list / NEAT genes)
↓
Convert graph → genome (list of genes)
↓
Apply GA operators:
- Mutation (add edge, change weight, add node)
- Crossover (combine two genomes)
- Selection (fitness-based)
↓
Decode genome → graph again
↓
Evaluate fitness
↓
Repeat evolution

FIGURE 6 | Transition pipeline.

3. Selection/reproduction
4. Cross over and mutation
5. If termination criterion is met stop else repeat steps 2 to 4

Figure 6 the conceptual transition pipeline. In particular, if a network G is of the form $G = (V, E, W)$ then the

$$\text{Genotype} = \left\{ \left(v_i, v_j, w_{ij}, \text{enabled}, \text{innovation}\right) \mid \left(v_i, v_j\right) \in E \right\}.$$

The design and implementation of each module is problem dependent. The exact design that we have followed is explained in detail in the following section.

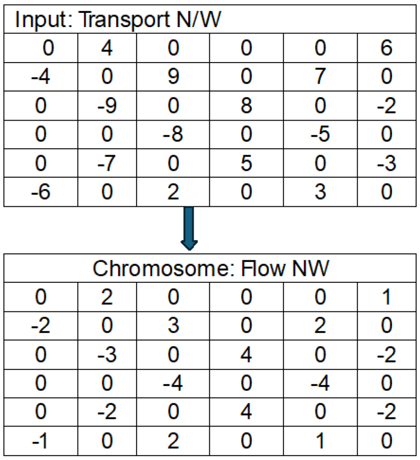
4.2 | Proposed Methodology

The primary task in implementing GA is to represent a typical solution to the problem in the form of a string and to device appropriate data structure to evaluate the string called chromosome.

1. Chromosome Representation and Initialization: The design of chromosomes is completely problem dependent. In our problem, chromosomes are used to model individual paths or nodes or edges of the network. In this paper, we use the adjacency matrix structure to model the chromosome. The initialization stage can be considered the vital module of any genetic algorithm-based problem solving approach. Chromosomes are generated randomly. The structure of a typical chromosome is as shown in Figure 4. Each row indicates a subchromosome of length equal to the number of nodes. A chromosome is generated from the adjacency matrix as shown in Figure 7.

4.3 | Evaluation

The robot movement (also called flow) through each edge is initialized by randomly generating a number in the range [0, capcity]. The flow is only initialized for those indices at which adjacency matrix has a positive value. Since the adjacency matrix is symmetric about the diagonal with respect to magnitude and antisymmetric with respect to direction, we initialize the inflow of edges by negating the outflow values of the edges. In addition to in-Flow and out-Flow, sources and destination nodes are also having an extra-Flow. This is flow comes into or goes out of the sources or destination, respectively. It should be noted that a source node and a destination node belonging to the same commodity are assigned the same extra flow. The extra flow is now added to inflow of the source or outflow of the destination. Other nodes have extra-flow = 0. A node is said to be balance if the net flow cancels out each other to zero. That is,



Input: Transport N/W					
0	4	0	0	0	6
-4	0	9	0	7	0
0	-9	0	8	0	-2
0	0	-8	0	-5	0
0	-7	0	5	0	-3
-6	0	2	0	3	0

Chromosome: Flow NW					
0	2	0	0	0	1
-2	0	3	0	2	0
0	-3	0	4	0	-2
0	0	-4	0	-4	0
0	-2	0	4	0	-2
-1	0	2	0	1	0

FIGURE 7 | Adjacency matrix representation of the network and the chromosome.

$$\text{Balanced} = \begin{cases} 1 & \text{if(inflow} = \text{outflow)} \\ 0 & \text{otherwise.} \end{cases}$$

In this paper, the fitness is evaluated using a weighted aggregate objective function given below:

$$f = \sum_{i=1}^{N} w_i f_i, \tag{5}$$

where f is the fitness of the chromosome, f_i is the fitness evaluated on the basis of *ith* criterion, and w_i is the weight of the fitness f_i.

The fitness of the chromosome is evaluated based on the following criteria:

i. The number of balanced nodes in the network.
ii. The total flow in the network.

The first criterion is used to assert absence of congestion in the network and the second criterion helps to maximize the robot movements (flow). Let b be the number of balance nodes, n be the total number of nodes, f be the total flow of the network and c be the total capacity of the network. Thus, we get the following expression for evaluating the fitness of each chromosome

$$\text{Fitness} = \frac{w_1 * \left(\frac{b}{n}\right) + w_2 * \left(\frac{f}{c}\right)}{w_1 + w_2}, \tag{6}$$

where w_1 and w_2 are the weights assigned to each criterion. A term called energy is associated with each node in a network. The energy is again a function of flow and balance, similar to fitness. In evaluating energy, the expression is slightly modified as follows:

$$\text{Energy} = \frac{\text{Excessflow}}{\text{Totalflow}} + \frac{\text{Totalflow}}{\text{Totalcapacity}},$$

where Excess flow = | Inflow − Outflow |. In addition, total flow and total capacity are those associated only with that node.

4.4 | Selection

We follow a tournament based selection [4] approach based on the fitness value of the chromosomes. The tournament is conducted by the conventional method of randomly creating two fixtures of length equal to the population size. A fixture here means an array of integers in the range [1, maxPop], arranged in a random order, where maxPop is the number of chromosomes in a given population. The adjacent integers in the fixture arrays indicate the indices of the two chromosomes which will be compared. Thus, each fixture will lead to $\frac{\text{maxPop}}{2}$ comparisons and eventually the same number of selections. The procedure is repeated once more, using the second fixture, to obtain maxPop number of new chromosomes. The selection is based on the fitness values of the chromosomes.

1. *Selection of Elite Set of Chromosomes:* Since a strategy of variable mutation rate is followed, there is a possibility of losing near optimal solutions. Hence to avoid such circumstances, an elitist approach is employed. In the elitist strategy, we save the best few chromosomes of each generation and breed them in the following generation. The breeding is done by replacing the worst few chromosomes resulting after cross-over by these elite chromosomes. Thus, the average fitness as well as near-optimal solution is preserved from being lost or extinct.

4.5 | Crossover and Mutation

Essence of GA is inheriting from parents the learning process and mutation to impart exploration in search.

1. *Crossover:* Crossover is done by replacing the nodes based on the energy of nodes. When two chromosomes are compared, the nodes with higher energy are marked dominant, and others are marked otherwise. The flow is then altered as follows:

Consider two chromosomes A and B considered for cross-over. Let the node j to be crossed-over be the ith node.

1. If in $A(i, j)$ both ith and jth nodes are dominant then replace $B(i, j)$ by $A(i, j)$.
2. If ith vertex is dominant and jth vertex is not dominant (implies the other way in chromosome B), then replace $A(i, j)$ and $B(i, j)$ by their average.
3. If neither ith vertex nor jth vertex is dominant then replace $A(i, j)$ by $B(i, j)$.

2. *Mutation:* The concept of adaptive mutation is employed in our work which enables the heuristics to adapt according to the current optimum solution. If a solution is a near optimum solution, then the mutation rate is very low compared to the one when the solution is not optimum. This is by assuming a constant mutation rate and a variable mutation rate. The final mutation rate is a function of these two—in our case, the sum of the two terms. The variable mutation rate is taken as the ratio between the excess flow and the total flow. When there is no excess flow, the variable rate has the lowest possible value, that is, zero. Thus, the probability of losing the chromosome by mutation is reduced. On the other hand, a high excess flow implies high probability of mutation, thus transforming the unfit chromosomes. Mutation is incremented by either increasing or decreasing the flow associated with the vertex by factor of one unit.

5 | Experimental Results and Discussions

The GA parameters were chosen on a trial and error procedure and assigned the values shown in Table 2.

Two runs were executed, first without elitist strategy and then using the elitist approach. The nondominated sets obtained by varying the fitness weights w_1 and w_2, namely weight for the first objective, namely flow and weight for the second objective, namely number of balanced nodes, respectively. Initially

TABLE 2 | Constant parameters initialized.

#define	**POPSIZE**	**500**
#define	GENNUM	100
#define	MUTATPROB	0.1
#define	XOVERPROB	0.7
#define	NUM_COMMODITY	2
#define	IS_ELITE_REO	1
#define	ELITE_SIZE	10

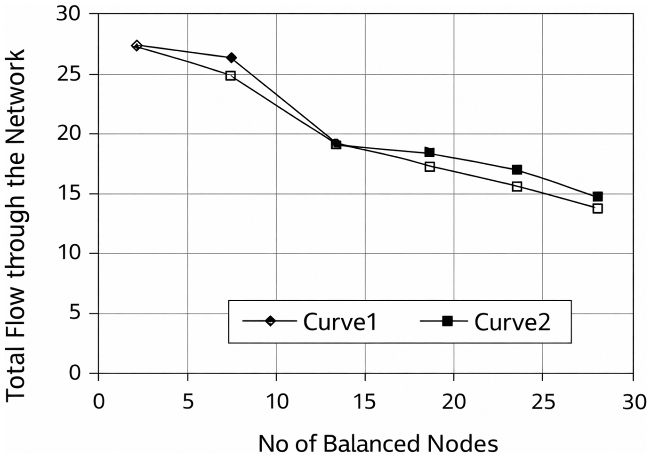


FIGURE 8 | Nondominated front for the problem in Figure 3 obtained using elitist strategy (Curve 1) and without using elitist strategy (Curve 2).

w_1 is taken as zero and incremented by $0.05(\delta)$ after every run. The value for w_2 is taken as $(1 - w_1)$ by default. Thus, $((^{1}) + 1)$ sets of nondominated solutions are obtained after all the runs. Finally, a net nondominated front is obtained from the population formed by combining all the nondominated solution sets. The above computational experiment was carried out twice. It is shown in Figure 8. First, elite solutions were not preserved, and later elite strategy was used. Curve1 is the net nondominated front obtained without preserving elite solutions and the Curve2 shows the net nondominated front obtained by preserving the elite chromosomes. Table 3 shows the fitness values for different path. It also shows the elapsed time.

TABLE 3 | Comparison of best path and their fitness values.

Steps	Best path chromosome	Generations	Elapsed time (sec.)	Fitness
5	1 1 2 2 2	72	198.8	2.2
6	1 1 2 2 2 3	111	523.1	1.333
7	2 2 8 2 2 3 3	140	553.6	0.429
8	3 3 8 8 2 2 2 33	167	766.4	1.375

TABLE 4 | Bench mark results.

Agents	Success (%)	Makespan	Sum cost	Runtime
10	100.0	32.4	480.2	0.12
50	94.0	71.8	3210.5	0.88
100	72.0	121.3	8055.7	2.91

TABLE 5 | Bench mark parameters.

Parameter	Values
Agents	10, 20, 50, 100
Map size	32 × 32, 64 × 64, 256 × 256
Obstacles	10%–30%
Runs	10–30 per instance

In this paper, we use the following standard setup for each map:

Agent counts: 10, 20, 50, 100, 150, 200 For each agent count: run 25–50 scenarios.

Report: Success rate, Avg makespan, Avg sum of costs and Avg runtime.

We use Multiagent Path-Finding (MAPF) bench marks [19, 20] for multiagent path-finding. A MAPF benchmark consists of a grid map (obstacles + free cells) and a set of agents. Each agent has Start position and Goal position. Performance metrics are Makespan, Sum of costs, Number of collisions, and Computation time. Table 4 shows these values for our benchmark. Table 5 shows the bench mark parameter values. A comparison plot (Makespan vs. Number of Agents) for GA, CBS, and Prioritized Planning is shown in Figure 9. The convergence behavior of the proposed Genetic Algorithm for MAPF is shown in Figure 10. It is observed that there is an improvement in best makespan over generations. The convergence graph shows how the solution quality improves across generations of the Genetic Algorithm. Our problem is minimization problem and therefore the curve usually decreases over generations. It also shows that a large drop in makespan at the beginning. Population diversity is high; crossover and mutation are exploring effectively. The algorithm quickly finds better paths. The algorithm exhibits rapid convergence during the initial generations due to effective exploration of the search space.

6 | Conclusions

In this paper, we proposed a GA based multirobots navigation network optimization. This study is an attempt to generate a

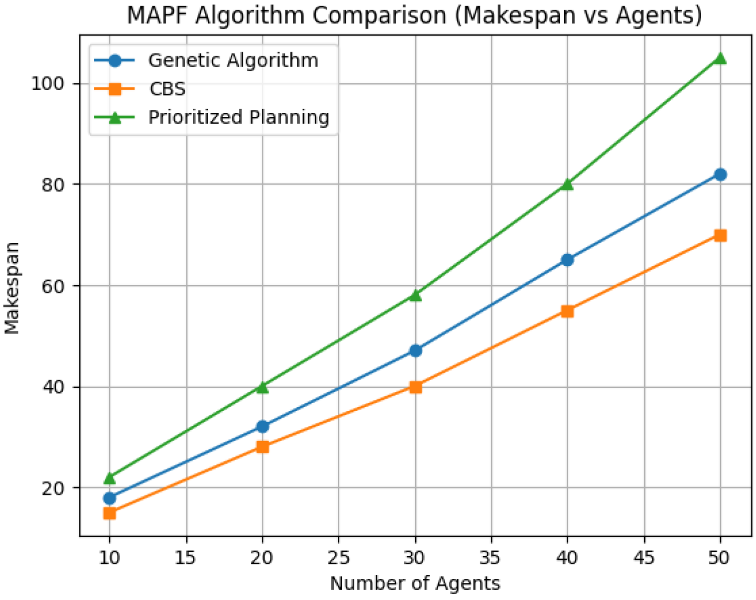


FIGURE 9 | Comparison plot.

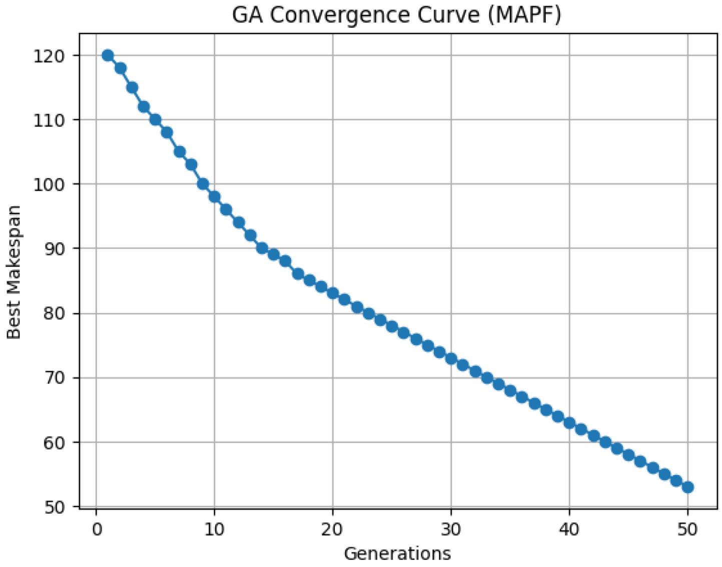


FIGURE 10 | GA convergence.

nondominated solution set for the problem considering more than one objective considered simultaneously. Result indicates that the robots flow through the network is better when elitist strategy is used. We have considered the static network.

As a feature work, one can develop the same for dynamic networks. Bio-inspired based GA algorithm can be developed with this energy and fitness functions. This may improve the accuracy of the navigation. Also, recent AI methods can be applied to this problem.

Author Contributions

K. Somasundaram: investigation, methodology, formal analysis, visualization, writing – review and editing. **Juha Plosila:** conceptualization, investigation, funding acquisition, writing – original draft, visualization, methodology, project administration, and funding.

Acknowledgments

The first author thanks the University of Turku, Finland for their support during his visit in January to February, 2025. The authors would like to thank the reviewers for their valuable suggestions. Open access publishing facilitated by Turun yliopisto, as part of the Wiley - FinELib agreement.

Funding

This work was supported by the University of Turku, Finland.

Ethics Statement

The authors have nothing to report.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

References

1. J. Canny, *The Complexity of Robot Motion Planning* (MIT press, 1988).

2. S. Choueiry, M. Owayjan, H. Diab, and R. Achkar, "Mobile Robot Path Planning Using Genetic Algorithm in a Static Environment," In 2019 Fourth International Conference on Advances in Computational Tools for Engineering Applications (ACTEA), 1–6. IEEE, 2019.

3. A. T. Ismail, A. Sheta, and M. Al-Weshah, "A Mobile Robot Path Planning Using Genetic Algorithm in Static Environment," *Journal of Computer Science* 4, no. 4 (2008): 341–344.

4. S. S. Ge and Y. J. Cui, "New Potential Functions for Mobile Robot Path Planning," *IEEE Transactions on Robotics and Automation* 16, no. 5 (2000): 615–620.

5. C. Lamini, S. Benhlima, and A. Elbekri, "Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning," *Procedia Computer Science* 127 (2018): 180–189.

6. T. Feng, J. Li, H. Jiang, et al., "The Optimal Global Path Planning of Mobile Robot Based on Improved Hybrid Adaptive Genetic Algorithm in Different Tasks and Complex Road Environments," *IEEE Access* 12 (2024): 18400–18415.

7. T. Mohanraj, S. Arunkumar, M. Raghunath, and M. Anand, "Mobile Robot Path Planning Using Ant Colony Optimization," *International Journal of Research in Engineering and Technology* 3, no. 11 (2014): 1–6.

8. E. Samsuria, M. S. A. Mahmud, N. A. Wahab, M. Z. Romdlony, M. S. Z. Abidin, and S. Buyamin, "Adaptive Fuzzy-Genetic Algorithm operators for Solving Mobile Robot Scheduling Problem in Job-Shop FMS Environment," *Robotics and Autonomous Systems* 176 (2024): 104683.

9. R.-A. Sánchez-Sosa and E. Chavero-Navarrete, "Robotic Cell Layout Optimization Using a Genetic Algorithm," *Applied Sciences* 14, no. 19 (2024): 8605.

10. J. Li, Y. Hu, and S. X. Yang, "A Novel Knowledge-Based Genetic Algorithm for Robot Path Planning in Complex Environments," *IEEE Transactions on Evolutionary Computation* 29, no. 2 (2025): 375–389.

11. M. A. P. Garcia, O. Montiel, O. Castillo, R. Sepulveda, and P. Melin, "Path Planning for Autonomous Mobile Robot NaviGation With Ant Colony Optimization and Fuzzy Cost Function Evaluation," *Applied Soft Computing* 9, no. 3 (2009): 1102–1110.

12. E. Masehian and D. Sedighizadeh, "A Multi-Objective PSO-Based Algorithm for Robot Path Planning," In 2010 IEEE International Conference on Industrial Technology, 465–470. IEEE, 2010.

13. B. K. Patle, A. Pandey, D. R. K. Parhi, and A. J. D. T. Jagadeesh, "A Review: On Path Planning Strategies for Navigation of Mobile Robot," *Defence Technology* 15, no. 4 (2019): 582–606.

14. M. N. Ab Wahab, A. Nazir, A. Khalil, et al., "Improved Genetic Algorithm for Mobile Robot Path Planning in Static Environments," *Expert Systems with Applications* 249 (2024): 123762.

15. H. Zhang, W. Lin, and A. Chen, "Path Planning for the Mobile Robot: A Review," *Symmetry* 10, no. 10 (2018): 450.

16. R. Aniketh, E. B. Manohar, G. R. S. P. R. Yazwa, M. Nithya, and M. R. Rashmi, "A Decentralized Fault-Tolerant Weights Based Algorithm for Coordination of Swarm Robots for a Disaster Scenario," In 2016 IEEE Annual India Conference (INDICON), 1–5, 2016.

17. L. Liu, X. Wang, X. Yang, H. Liu, J. Li, and P. Wang, "Path Planning Techniques for Mobile Robots: Review and Prospect," *Expert Systems With Applications* 227 (2023): 120254.

18. S. Ganesan, B. Ramalingam, and R. E. Mohan, "A Hybrid Sampling-Based RRT* Path Planning Algorithm for Autonomous Mobile Robot Navigation," *Expert Systems With Applications* 258 (2024): 125206.

19. R. Stern, N. Sturtevant, A. Felner, et al., "Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks," In Proceedings of the International Symposium on Combinatorial Search, 10, 1, 151–158. 2019.

20. https://mapf.info/index.php/Data/Data.