



Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm

Milad Nazarahari^a, Esmaeel Khanmirza^{b,*}, Samira Doostie^a

^a Department of Mechanical Engineering, University of Alberta, Edmonton, Alberta T6G1H9, Canada

^b Department of Mechanical Engineering, Iran University of Science and Technology, Tehran 1684613114, Iran



ARTICLE INFO

Article history:

Received 20 October 2017

Revised 4 August 2018

Accepted 5 August 2018

Available online 7 August 2018

Keywords:

Mobile robot

Path planning

Genetic algorithm

Artificial potential field

Multi-robot path planning

Multi-objective path planning

ABSTRACT

This paper presents a hybrid approach for path planning of multiple mobile robots in continuous environments. For this purpose, first, an innovative Artificial Potential Field (APF) algorithm is presented to find all feasible paths between the start and destination locations in a discrete gridded environment. Next, an enhanced Genetic Algorithm (EGA) is developed to improve the initial paths in continuous space and find the optimal path between start and destination locations. The proposed APF works based on a time-efficient deterministic scheme to find a set of feasible initial paths and is guaranteed to find a feasible path if one exists. The EGA utilizes five customized crossover and mutation operators to improve the initial paths. In this paper, path length, smoothness, and safety are combined to form a multi-objective path planning problem. In addition, the proposed method is extended to deal with multiple mobile robot path planning problem. For this purpose, a new term is added to the objective function which measures the distance between robots and a collision removal operator is added to the EGA to remove possible collision between paths. To assess the efficiency of the proposed algorithm, 12 planar environments with different sizes and complexities were examined. Evaluations showed that the control parameters of the proposed algorithm do not affect the performance of the EGA considerably. Moreover, a comparative study has been made between the proposed algorithm, A*, PRM, B-RRT and Particle Swarm Optimization (PSO). The comparative study showed that the proposed algorithm outperforms PSO as well as well-recognized deterministic (A*) and probabilistic (PRM and B-RRT) path planning algorithms in terms of path length, run time, and success rate. Finally, simulations proved the efficiency of the proposed algorithm for a four-robot path planning problem. In this case, not only the proposed algorithm determined collision-free paths, but also it found near optimal solution for all robots.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

Path planning (PP) problem can be considered as one of the most active fields in several robotic related research areas. A diverse range of robotic applications in industry, medicine, and agriculture encourage researchers to carry out research works in the PP field. A path planner should find an optimal (or near optimal) collision-free path, in terms of a particular criterion, between the start and destination location of a robot in an environment filled with obstacles (Klančar, Zdešar, Blažič, & Škrjanc, 2017). The PP problem can be solved for robotic arms (Kunz, Reiser, Stilman, & Verl, 2010), mobile robots (Klančar et al., 2017; Masehian & Sedighizadeh, 2007; Saicharan, Tiwari, & Roberts, 2016; Thoa, Copot, Trung, & Keyser, 2016), and UAVs (Tisdale, Kim, & Hedrick,

2009). Considering the affecting factors on PP like different kinds of robots and environments, static or dynamic obstacles, and multiple robots, finding the shortest path with the highest degree of smoothness which avoids collision with other robots and obstacles is still a challenging problem.

Due to the complexity of the PP problem, usually, it is considered as an NP-hard problem (Canny, 1998). As such, heuristic or evolutionary algorithms were widely used to find an optimal solution for this problem, specifically, in large complex environments; genetic algorithm (GA) (Castillo, Trujillo, & Melin, 2007; Davoodi, Panahi, Mohades, Hashemi, & Naser, 2015; Hoshiar, Kianpour, Nazarahari, & Korayem, 2016; Kala, Shukla, & Tiwari, 2010; Korayem, Hoshiar, & Nazarahari, 2016; Qu, Xing, & Alexander, 2013; Tuncer & Yıldırım, 2012; Wang & Hedner, 2016), particle swarm optimization (PSO) (Doostie, Hoshiar, Nazarahari, Lee, & Choi, 2018; Han & Seo, 2017; Mac, Copot, Tran, & Keyser, 2017; Mo & Xu, 2015; Song, Wang, & Zou, 2017; Tang, Zhu, & Luo, 2016), ant colony optimization (Melin, Garcia, Montiel, Castillo, & Sepúlveda, 2009;

* Corresponding author.

E-mail addresses: nazaraha@ualberta.ca (M. Nazarahari), khanmirza@iust.ac.ir (E. Khanmirza), doostie@ualberta.ca (S. Doostie).

Zeng, Xi, & Xiao, 2016), artificial bee colony (Contreras-Cruz, Ayala-Ramirez, & Hernandez-Belmonte, 2015), bacterial forging optimization (Hossain & Ferdous, 2015), and shuffled frog leaping (Hidalgo-Paniagua, Vega-Rodríguez, & Ferruz, 2016; Hidalgo-Paniagua, Vega-Rodríguez, Ferruz, & Pavón, 2015) to name a few.

The first limitation of the abovementioned research works is that most of them modeled the environment as a discrete gridded space and tried to find the best set of grids to determine the optimal path (Castillo et al., 2007; Hidalgo-Paniagua et al., 2015; Qu et al., 2013; Tang et al., 2016; Tuncer & Yildirim, 2012; Wang & Hedner, 2016). The main limitation of this approach is that the position of the grids should be fixed beforehand, and thus, the flexibility of the PP will be limited by the grid size. Moreover, an efficient implementation of A* (Hart, Nilsson, & Raphael, 1968) or Dijkstra's algorithm (Dijkstra, 1959) can also be used to determine the optimal path in a discrete gridded environment. The main advantages of the A* and Dijkstra's algorithm are: (i) having a deterministic nature, unlike evolutionary algorithms, and thus, their performance does not depend on the initial solutions, and (ii) time-efficiency, specifically for 2-dimensional PP, in comparison to the most of evolutionary algorithms.

Feasible initial solutions can improve the performance of the evolutionary algorithms considerably; and even in some cases, using random initial solutions may result in not finding any feasible path after several iterations, specially in large complex environments. The second limitation of some previous works is that either they start with a random initial solution, or adopt a random search method to build the feasible initial solution (Castillo et al., 2007; Qu et al., 2013; Tuncer & Yildirim, 2012). Random search method usually works as follows: first, a random path is built, then, it is checked for feasibility. If the path is accepted in the feasibility check, it will be considered as an initial solution. Otherwise, a new random path will be built, and this process continues until a sufficient number of feasible initial solutions are generated. This approach is nor time-efficient, neither computationally-efficient. Using common operators of evolutionary algorithms, e.g., crossover and mutation in GA, is the third limitation of the mentioned research works. Although the performance of these operators has been proved in numerous engineering problems, the efficiency of an evolutionary algorithm can be considerably improved by designing specific operators which are compatible with the solution coding of the problem.

Single objective PP, commonly path length minimization, is the fourth limitation of the previous research works. In general, a multi-objective PP problem can find more proper solutions, particularly for practical applications. The three well-recognized objectives which typically were used for PP problem are: (i) path length: finding a path with shortest length between start and destination locations, (ii) path smoothness: a path with lowest degree of direction change, and (iii) path safety: a path which always satisfies a predefined safety margin (distance) with respect to obstacles. Aggregating these objectives will result in a multi-objective PP problem which can find accurate paths, specifically for practical applications. Finally, most of the mentioned works are only capable of dealing with single-robot PP problem. Nowadays, more powerful CPUs and parallel computing methods increased the applications of robotic teams. As such, an efficient PP algorithm should also be able to handle the multi-robot PP problem.

In this paper, we present a novel multi-objective enhanced Genetic Algorithm (EGA) to find the optimal path in terms of path length, smoothness, and safety in a continuous environment. Unlike most of the previous works, the proposed algorithm is able to find a path in a continuous environment which is not limited by the grids in a discrete gridded environment. Moreover, an innovative Artificial Potential Field (APF) method is presented to build a set of feasible initial paths in an efficient deterministic fashion.

Five improved crossover and mutation operators are introduced for EGA to modify the initial paths and generate an optimal solution in terms of the mentioned objectives. These operators are modified such that they can minimize the path length and maximize the path smoothness at the same time while altering the solutions minimally to avoid generating infeasible paths. Finally, a new version of the proposed algorithm is introduced to handle the multi-robot PP problem. For this purpose, a new collision term is added to the objective function which represents the possibility of collision between mobile robots, and a new collision-removal operator to remove the collision between two paths. The performance of the proposed method is compared with four well-recognized evolutionary, deterministic, and probabilistic PP approaches: (i) PSO, (ii) A*(with two different discretization size), (iii) Probabilistic Road Map (PRM), and (iv) Bi-directional Rapidly-exploring Random Tree (B-RRT) for 12 environments.

The rest of this paper is organized as follows. Section 2 presents the related works. The problem is defined in Section 3. Section 4 provides the description of the proposed method including environment modeling and solution structure, feasible initial solution planner, the EGA structure and improved operators, the objective function, and the multi-robot PP algorithm. The simulation results and discussion are presented in Section 5. Finally, the paper is concluded in Section 6.

2. Related works

So far, PP in a discrete gridded environment was the most common approach, and researchers developed deterministic (Dijkstra, 1959; Hart et al., 1968) and evolutionary algorithms for this purpose (Castillo et al., 2007; Hidalgo-Paniagua et al., 2015; Qu et al., 2013; Tang et al., 2016; Tuncer & Yildirim, 2012; Wang & Hedner, 2016). Tuncer and Yildirim (2012) introduced a new mutation operator to find the path with the shortest length in a discrete gridded environment using GA. A modified PSO-DE algorithm was introduced by Tang et al. (2016) for PP in a discrete environment with a combination of path length and safety as the objective function. The main flaw of hybridizing PSO and differential evolution is long runtimes, even for small environments.

A co-evolutionary GA was introduced in Qu et al. (2013) for PP of multiple robots by considering the linear combination of path length, smoothness, and safety as an objective function. The introduced co-evolutionary mechanism solved multiple PP problems in parallel, one for each robot, and evaluated the collision between robots at each iteration. The problem with this mechanism is that the slave GAs, which find the optimal path for each robot, will continuously modify the paths without any knowledge about the collision between robots. In fact, collision between robots must be considered as an additional constraint for PP problem, similar to obstacles; in other words, the PP problem must be solved as a PP problem in a dynamic environment (each robot can be considered as a moving obstacle for other robots). Liu, Liu, and Yang (2011) proposed an adaptive GA which adjusts the probabilities of the genetic operators based on the solution fitness. However, using a simple crossover or mutation operator with variable probability might not be the most efficient method to increase the population diversity and avoid premature convergence.

Sedaghat (2011) and Davoodi et al. (2013) proposed a novel method to create feasible initial solutions. They used the distance of the grids in a discrete environment from obstacles to provide a new map. Then, the provided map was used to select grids far from obstacles step by step and find a feasible path. The main flaw of this method is that the grids which are far from both obstacles and destination are desirable for selection; however, selecting these grids might lead to a path far from destination. The Voronoi diagram was used in Mo and Xu (2015) to generate initial feasible

paths which then fed to a Biogeography PSO for optimal PP. This research showed the advantage of using feasible initial paths instead of random ones. Han and Seo (2017) selected a set of points in the obstacles' neighborhood and used them as a guide to build an initial feasible path. Then, a modified PSO was adopted to optimize the position of the points and find the optimal piecewise linear path. They showed that the success rate of 100% could be achieved using a feasible initial path. Also, a modified operator could improve the path smoothness while the problem originally aimed at minimizing the path length. Contreras-Cruz et al. (2015) introduced a two-phase PP algorithm: first, a feasible path planner using artificial bee colony generates an initial path, then, the initial path is optimized using evolutionary computing with respect to measures like path length and smoothness. A hierarchical three-phase PP algorithm was proposed by Mac et al. (2017): (i) using triangular decomposition to divide the environment into obstacle and free configuration space, (ii) utilizing the Dijkstra's algorithm to find an initial collision-free path, and (iii) adopting a new constrained multi-objective PSO for global PP. However, the mentioned study did not provide any comparison between the proposed algorithm and other PP algorithms in terms of objectives like path length and runtime to prove the efficiency of the introduced algorithm.

Hidalgo-Paniagua et al. (2015, 2016) proposed a multi-objective (path length, smoothness, and safety) Shuffled Frog-Leaping algorithm with three refiner operators to find the optimal path in a gridded environment. The proposed method generates loops within the paths, and thus, the authors proposed a loop deletion operator which is time-consuming. Davoodi et al. (2015) presented a multi-objective GA for PP in a continuous environment. The proposed algorithm initiates with random paths and uses simple crossover and mutation operator to generate feasible paths and optimize them. To compensate for the drawback of random initial paths and common genetic operators, new operators were introduced to correct collision with obstacles and self-intersection. However, using a multi-objective algorithm with an infeasible initial solution and several operators could result in long runtimes.

Instead of connecting the center of the grids in a discrete environment with linear segments to build a piecewise linear path, Song et al. proposed the application of Bezier Curve (Song & Wang, 2016) and a multimodal delayed PSO (Song et al., 2017) to build smooth paths. Using this method, the mobile robot will be able to follow a curve, instead of a piecewise linear path, which is desirable for many applications. Zuo, Guo, Xu, and Fu (2015) used the A* algorithm to determine the best grids for building a piecewise linear path. Next, a least square policy iteration was adopted to optimize the position of the selected grids in a predefined radius and find a smooth path. Although the results obtained using this hybrid method are considerably better than the RRT, the least square optimization is limited to a predefined radius around the pre-selected points. In addition, the authors did not provide a comparison between A* and their proposed algorithm to prove its efficiency.

The main limitations of the previous research works and the major contributions of this paper to cope with these limitations can be summarized as follows:

- Most of the previous works used a discrete gridded environment and developed a method to find the best set of grids which builds the optimal path. The optimal solution in these approaches is limited by the grid size, which should be determined beforehand. In contrast, the presented algorithm in this paper deals with global PP in a continuous environment. Thus, its performance is not limited by an external parameter like discretization size.

- Running an evolutionary algorithm with random initial solution is the second limitation of the previous works. Although some works have been done to generate a feasible initial solution, they are not efficient neither in terms of time nor objective function. The introduced APF uses a deterministic approach to build a feasible initial population in a time-efficient manner. The presented APF is also guaranteed to find a feasible path if one exists.
- Utilizing conventional genetic operators is the third limitation of the past works. In this paper, five genetic operators are designed to improve the solutions at each iteration in terms of the defined objective function. The operators are designed not only to build new paths and maintain the diversity of the population (avoid premature convergence) but also to take advantage of available information such as the destination location.
- Single objective PP was the common approach in previous works. However, single objective PP limits the application of the previous works in practice, as they only consider a path with minimum length. Following such a path can increase the energy consumption of the robot (due to sharp direction changes) or make the robot vulnerable to potential risks in the environment. As such, in this research, three objectives, path length, smoothness, and safety, are combined to form a multi-objective PP problem.
- Most of the previous works were developed to handle a single-robot PP problem. However, the presented method is extended to the multi-robot case. For this purpose, a collision-avoidance term is added to the objective function to capture the dynamic nature of multi-robot PP problem, and a collision removal operator is introduced to remove possible collision between robots.
- Finally, in contrary to some previous works, the performance of the proposed algorithm is compared with four well-recognized algorithms: PSO, A* (with two different discretization size), PRM, and B-RRT, for 12 environments in terms of path length, runtime, and success rate.

3. Problem definition

The environment (or configuration space) of PP consists of free space and space filled with obstacles. The predefined start and desired destination of the robot are located in the free space. PP can be defined as finding a finite set of feasible movements (movements in the free space) to navigate the robot between start and destination. Usually, there is more than one path between start and destination; however, the PP algorithms are used to find the optimal path in terms of a predefined objective function, e.g., the path with the shortest length, the highest degree of smoothness, or the safest path.

This paper proposes an innovative PP algorithm to determine the optimal path in almost every complex environment between pre-defined start, S , and destination, D , locations. The quality of the paths is measured based on (i) path length, (ii) path smoothness, and (iii) path safety. In this study, we consider the PP problem in 2-dimensional environments filled with arbitrarily-shaped static obstacles, which have no interaction (positive or negative attributes) with the free space. Moreover, the robots are considered as points, and their size will be added as a confidence radius around obstacles. Also, in the multi-robot PP case, we assume that all robots move with constant speed.

4. Proposed path planning method

The proposed algorithm includes two phases: (i) find a suboptimal initial feasible path (solution) using the proposed APF algorithm, and (ii) improve the initial path using the EGA to achieve the optimal path in a continuous environment. The APF algorithm

generates the potential map of the environment which can be used to determine all feasible piecewise linear paths between start and destination locations in a gridded environment. The APF algorithm is guaranteed to find at least one feasible path if one exists. Then, the obtained paths, the position of the selected grids which will be called Path Bases (PBs) hereafter, are passed to the EGA. The EGA is customized with newly introduced crossover and mutation operators to find the optimal position of PBs in every complex environment. Then, the PBs can be connected to each other using a piecewise linear or cubic spline to build the optimal path. Fig. 1 shows the flowchart of the proposed EGA for optimal PP in a continuous environment.

4.1. Environment modeling and solution structure

PP should be solved in continuous space with arbitrary shape obstacles; however, in this situation, the problem becomes computationally intractable. As such, most researchers tried to simplify the PP by changing the problem to finding a finite set of steps in a discrete gridded environment between the start and destination locations. However, such algorithms are always limited by the discretization size.

The proposed algorithm benefits from a hybrid approach to model the environment. First, the environment is discretized using square shape grids. Fig. 2(a) shows a sample 10×10 environment which is discretized using square grids with length 1 (all dimensions are unitless). Thus, 121 nodes (blue circles) are distributed uniformly in the environment. The path which can be traveled by the robot will be defined by selecting a proper set of nodes in the gridded environment. For this purpose, the APF algorithm assigns a proper potential value to all nodes, and then, all feasible paths between start and destination locations will be determined based on the pre-calculated potentials. The dashed line in Fig. 2(a) shows the selected PBs (magenta pentagrams) and the feasible path between start and destination.

Next, the coordinates of PBs related to each feasible initial path is coded as a solution. Fig. 2(b) shows the solution structure (chromosomes of the EGA) corresponding to the initial path shown in Fig. 2(a). As the initial paths obtained by the proposed APF might have a different number of PBs, the mentioned solution coding will result in an EGA with variable length chromosome. Finally, five improved arithmetic crossover and mutation operators will be employed to find the optimal position of PBs. Using chromosomes with variable length could improve the flexibility of the path planner, as it can select the optimal number PBs automatically.

4.2. Feasible initial path planner using proposed APF

In the proposed APF, the largest potential is assigned to the node corresponding to the destination location, and then, by traveling to its adjacent nodes, the potential decreases gradually. Adjacent nodes of each node are those which lie in the free space with distance less than or equal to $d\sqrt{2}$ from the intended node (d is the discretization size). This leads to a new map of the environment which is called the potential map. All feasible paths between start and destination can be found using the potential map. Fig. 3(a) shows the potential map of the environment shown in Fig. 2(a).

Three lists of nodes will be used to build the potential map. The OPEN list contains the nodes which have not been assigned a potential, the TEMP list contains the nodes which have been allocated a potential, but their adjacent nodes have not, and the CLOSED list includes the nodes which themselves and their adjacent nodes have been assigned a potential. First, all nodes are inserted in the OPEN list. Then, (i) the destination node is removed from the OPEN list, (ii) the largest potential ξ_0 ($\xi_0 = 300$ in Fig. 3(a)) is assigned

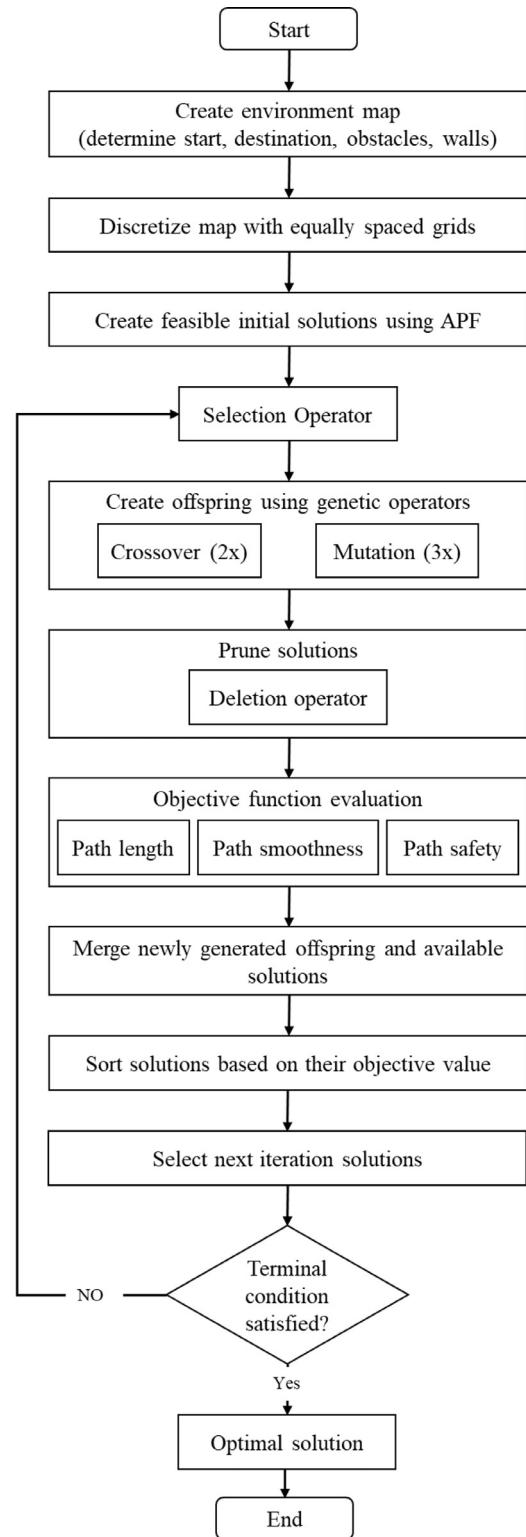
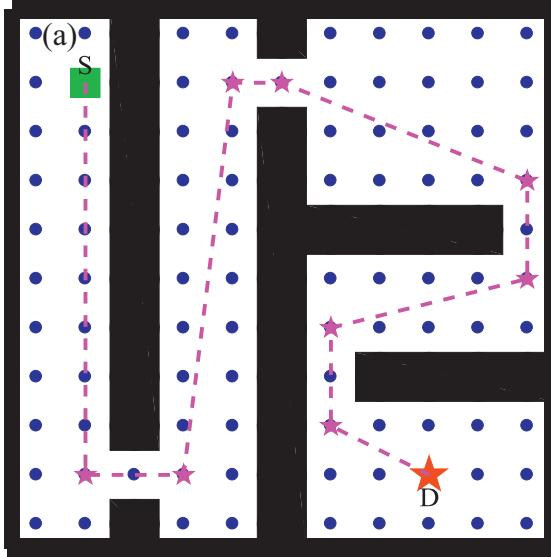


Fig. 1. The proposed Enhanced GA flowchart for optimal path planning in a continuous environment.

to it, and finally (iii) it is inserted in the TEMP list. Next, all obstacle nodes will be removed from the OPEN list, their potential set to $-\xi_0$, and inserted in the CLOSED list. Then, starting from the destination, (i) the potential of $\xi_1 = \xi_0 - \alpha$ (α is the decrement step and equals to 10 in Fig. 3(a)) will be assigned to all its adjacent nodes which are in the OPEN list, (ii) these adjacent nodes will be inserted in the TEMP list and removed from the OPEN list,



(b)

Solution Structure
Chromosome of IGA

x	1	3	4	5	10	10	6	6
y	1	1	9	9	7	5	4	2

Fig. 2. Environment modeling and solution structure; (a) a sample 10×10 gridded environment; blue circles show the equally-spaced distributed nodes, magenta dashed line shows an initial path, and magenta pentagrams show the PBs; (b) variable length solution coding. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).

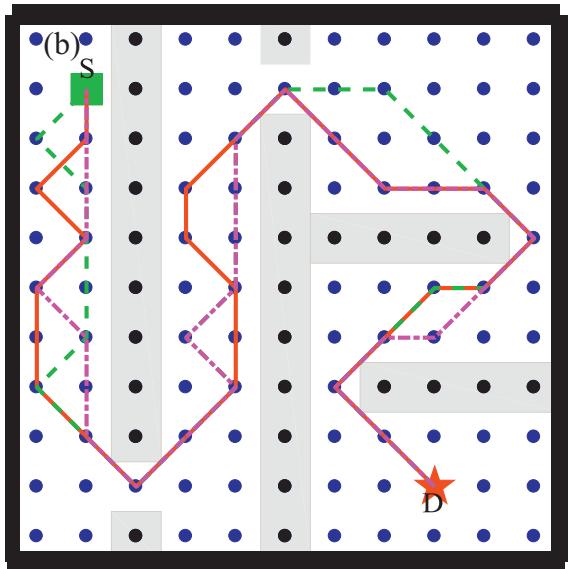
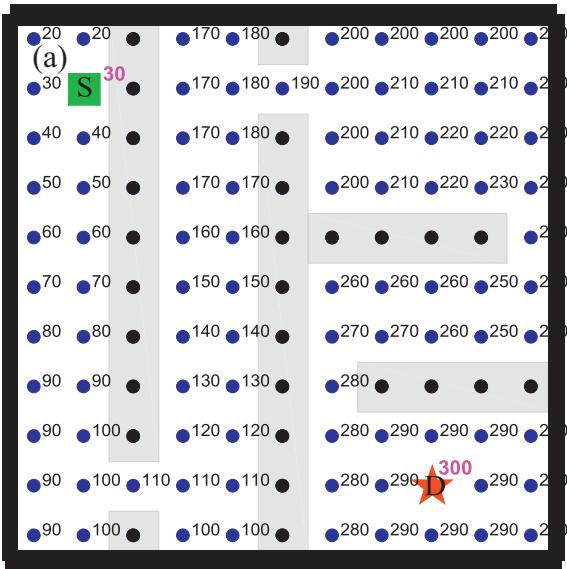


Fig. 3. Determining feasible initial paths using the proposed APF; (a) the potential map of the environment shown in Fig. 1(a); (b) three paths obtained using the APF.

and (iii) the destination will be removed from the TEMP list and inserted in the CLOSED list. Generally, in the i th iteration, (i) nodes on the TEMP list will be extracted and a potential of $\xi_i = \xi_{i-1} - \alpha$ is assigned to their adjacent nodes which are in the OPEN list, (ii) nodes in the TEMP list will be removed from it and inserted in the CLOSED list, and (iii) nodes which accept a potential value at this iteration will be removed from the OPEN list and inserted in the TEMP list. Repeating this procedure will lead to the potential map of the environment, as shown in Fig. 3(a). The pseudo-code of the APF algorithm is given in Algorithm 1. It should be noted that the parameters ξ_0 and α can be arbitrary selected based on the environment size.

To determine the feasible initial paths, starting from the start location, at each iteration, an adjacent node with the highest potential should be selected. Repeating this procedure, the potential of the path increases gradually till it meets the destination

location, the point with the largest potential in the map. At this point, a feasible path between start and destination locations is found. The proposed APF is guaranteed to find a feasible initial path if one exists. In some situations, the two adjacent nodes of a particular node have the same potential. At this point, the initial path will be divided into two subpaths, and each sub-path will continue its progress separately. Fig. 3(b) shows three different paths obtained by the proposed APF. In the case that the number of initial paths obtained by the APF is lower than the required population size of the EGA, new paths will be generated by selecting one of the APF paths and moving some of its PBs in a small radius randomly. Although the main idea of the APF is similar to some deterministic PP algorithms, the main feature which makes it an ideal choice for evolutionary PP is that it can determine all feasible paths in the environment, and thus, form the feasible initial population of the evolution algorithm efficiently.

Algorithm 1 Structure of the introduced APF for feasible initial PP.

```

OPEN: list of all nodes without potential value
TEMP: list of all nodes which have potential value, but their neighbors do not
have
CLOSED: list of all nodes which both themselves and their neighbors have a
potential value
Initialization
1. OPEN ← All nodes
2. Q ← all nodes associated with obstacles
3. OPEN{Q} = [] (remove all Qs from OPEN)
4.  $P_Q = -\xi_0$  (assign the property of  $-\xi_0$  to the potential of all nodes in Q)
5. CLOSED ← Q
6. Q ← destination node
7.  $P_Q = \xi_0$ 
8. TEMP ← Q
9. OPEN{Q} = []
Main Loop
10. While ~isempty(TEMP)
11. Q ← TEMP{1} (put the first element of TEMP in Q)
12. N ← OPEN{neighbors of Q}
13.  $P_N = P_Q - \alpha$ 
14. TEMP{Q} = []
15. TEMP ← N
16. OPEN{N} = []
17. CLOSED ← Q
18. End-While
12. the potential of all nodes is found
13. Return

```

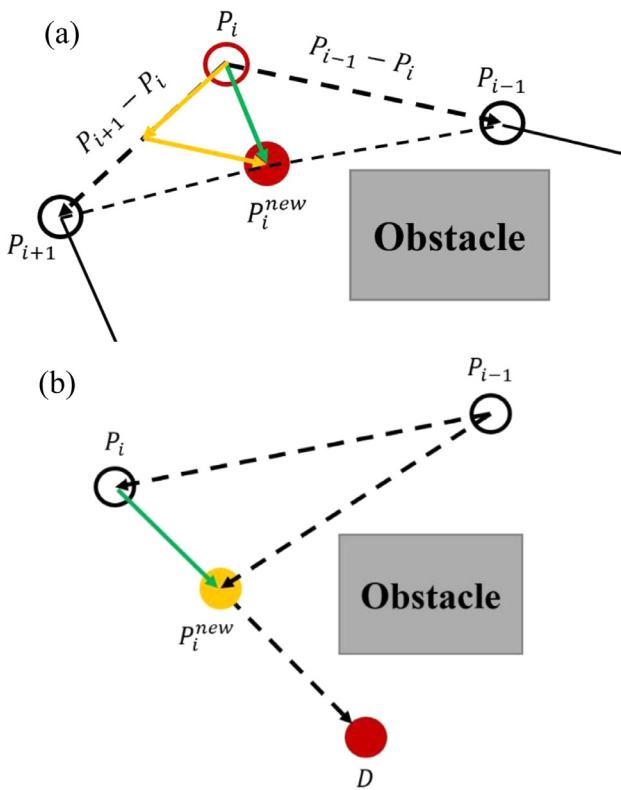


Fig. 4. Improved mutation operators; (a) integrating the position of three PBs to modify the position of the middle PB; (b) moving an arbitrarily selected PB toward the destination location.

4.3. Enhanced genetic algorithm

This paper introduces an EGA with modified genetic operators to determine the optimal position of PBs in a continuous environment. The main features of the EGA are (i) variable length chromosomes to add more flexibility in finding the optimal path, (ii) five arithmetic crossover and mutation operators to find the

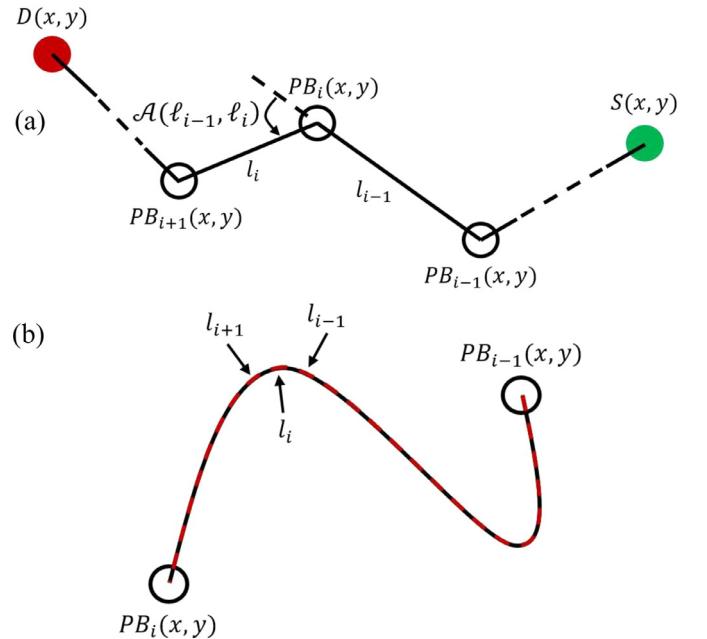


Fig. 5. Path representation in (a) piecewise linear case, and (b) spline case, PBs and line segments between interpolants are shown in the figure.

Table 1

Characteristics of the benchmark environments used for performance evaluation.

ID	Map	Size	ID	Map	Size
1	Random 1	30 × 30	7	Slits 1	20 × 20
2	Maze 1	50 × 50	8	Slits 2	20 × 20
3	Home	50 × 50	9	Maze 2	50 × 50
4	Corridor 1	30 × 30	10	Maze 3	50 × 50
5	BoxPile	30 × 30	11	Random 2	30 × 30
6	Corridor 2	30 × 30	12	PlankPile	30 × 30

optimal position of PBs, and (iii) an improvement operator to reduce the number of PBs automatically if needed. The evolution of the EGA is repeated until a termination condition has been reached. In this paper, reaching a fixed number of iterations is used as the terminal condition.

4.3.1. Selection operators

The selection operator selects some individuals from the population randomly or based on a predefined measure. In this paper, we used the Roulette Wheel Selection (RWS) to select parents for crossover operator. The selection probability of each individual has an inverse relationship with its quality, as such, better individuals have more chance to be selected by RWS. Also, the rank-based selection was employed to build the population of i th iteration from the pool consisting the population of $(i-1)$ th iteration and the new offspring obtained by genetic operators.

4.3.2. Crossover operators

The modified crossover operators combine two solutions, with the same or different number of PBs, to produce a new offspring which is probably better in terms of a predefined objective. The first crossover operator finds a new offspring by calculating the mean of the two parents:

$$x^f = \frac{x_1^p + x_2^p}{2} , \quad y^f = \frac{y_1^p + y_2^p}{2} \quad (1)$$

where x_1^p, y_1^p and x_2^p, y_2^p are the x and y coordinates of PBs of the two parents, and x^f, y^f are the coordinates of the offspring, respectively. When the number of PBs of the two parents are not

Table 2

Comparison of path length, path smoothness, runtime, and success rate of A*, PRM, B-RRT, PSO and the proposed EGA with piecewise linear and spline paths. The values are reported as mean \pm standard deviation for 50 executions.

		Path length	Path smoothness	Runtime (s)	Success rate (%)
Map 1	A* (grid size = 1)	35.79 \pm 0.00	360.00 \pm 0.00	0.10 \pm 0.00	100
	A* (grid size = 0.25)	35.30 \pm 0.00	360.00 \pm 0.00	6.50 \pm 0.00	100
	PRM	36.83 \pm 0.74	238.64 \pm 50.31	2.20 \pm 0.13	96
	B-RRT	36.88 \pm 1.37	316.53 \pm 75.65	0.76 \pm 0.36	100
	PSO Piecewise Linear	34.84 \pm 0.04	239.55 \pm 5.49	1.68 \pm 0.02	100
	EGA Piecewise Linear	34.06 \pm 0.10	232.39 \pm 7.16	1.37 \pm 0.01	100
	EGA Spline	34.31 \pm 0.17	275.00 \pm 18.24	1.41 \pm 0.02	100
		—	—	—	0
Map 2	A* (grid size = 1)	251.52 \pm 0.00	1665.00 \pm 0.00	0.66 \pm 0.00	100
	A* (grid size = 0.25)	245.51 \pm 0.00	1575.00 \pm 0.00	66.10 \pm 0.00	100
	PRM	259.45 \pm 1.14	1526.88 \pm 20.11	74.61 \pm 2.63	60
	B-RRT	—	—	—	0
	PSO Piecewise Linear	254.22 \pm 0.00	1502.65 \pm 8.43	2.95 \pm 0.05	100
	EGA Piecewise Linear	250.71 \pm 0.26	1505.97 \pm 21.87	2.57 \pm 0.01	100
	EGA Spline	253.41 \pm 0.46	1528.52 \pm 9.61	2.59 \pm 0.01	100
		—	—	—	0
Map 3	A* (grid size = 1)	132.22 \pm 0.00	1800.00 \pm 0.00	0.44 \pm 0.00	100
	A* (grid size = 0.25)	127.77 \pm 0.00	2070.00 \pm 0.00	41.24 \pm 0.00	100
	PRM	134.43 \pm 1.37	1436.76 \pm 73.81	63.65 \pm 0.80	66
	B-RRT	—	—	—	0
	PSO Piecewise Linear	127.43 \pm 0.12	1359.60 \pm 9.53	3.13 \pm 0.04	100
	EGA Piecewise Linear	126.38 \pm 0.32	1324.45 \pm 24.57	2.68 \pm 0.01	100
	EGA Spline	128.12 \pm 0.32	1482.05 \pm 37.95	2.75 \pm 0.00	100
		—	—	—	0
Map 4	A* (grid size = 1)	33.96 \pm 0.00	360.00 \pm 0.00	0.04 \pm 0.00	100
	A* (grid size = 0.25)	33.17 \pm 0.00	405.01 \pm 0.00	0.87 \pm 0.00	100
	PRM	36.15 \pm 1.31	463.30 \pm 88.62	7.15 \pm 0.39	100
	B-RRT	36.37 \pm 0.98	538.48 \pm 82.55	0.83 \pm 0.47	100
	PSO Piecewise Linear	34.15 \pm 0.02	360.00 \pm 7.82	1.47 \pm 0.01	100
	EGA Piecewise Linear	33.34 \pm 0.12	303.65 \pm 25.31	1.30 \pm 0.00	100
	EGA Spline	33.57 \pm 0.2	367.98 \pm 34.78	1.35 \pm 0.01	100
		—	—	—	0
Map 5	A* (grid size = 1)	48.20 \pm 0.00	855.00 \pm 0.00	0.06 \pm 0.00	100
	A* (grid size = 0.25)	46.48 \pm 0.00	1035.00 \pm 0.00	4.07 \pm 0.00	100
	PRM	48.75 \pm 0.48	478.28 \pm 84.49	14.13 \pm 0.58	98
	B-RRT	50.13 \pm 0.93	504.73 \pm 93.90	4.42 \pm 1.91	100
	PSO Piecewise Linear	46.93 \pm 0.06	572.67 \pm 10.11	1.84 \pm 0.01	100
	EGA Piecewise Linear	46.07 \pm 0.20	536.43 \pm 28.54	1.55 \pm 0.01	100
	EGA Spline	46.67 \pm 0.29	600.97 \pm 31.00	1.59 \pm 0.00	100
		—	—	—	0
Map 6	A* (grid size = 1)	68.24 \pm 0.00	855.00 \pm 0.00	0.13 \pm 0.00	100
	A* (grid size = 0.25)	66.25 \pm 0.00	855.01 \pm 0.00	8.91 \pm 0.00	100
	PRM	70.88 \pm 0.99	900.82 \pm 29.64	14.11 \pm 0.15	82
	B-RRT	86.55 \pm 1.62	656.33 \pm 95.01	5.04 \pm 1.87	100
	PSO Piecewise Linear	67.12 \pm 0.04	642.34 \pm 7.95	1.48 \pm 0.01	100
	EGA Piecewise Linear	66.09 \pm 0.19	619.11 \pm 33.06	1.29 \pm 0.00	100
	EGA Spline	66.94 \pm 0.31	768.58 \pm 41.16	1.34 \pm 0.01	100
		—	—	—	0
Map 7	A* (grid size = 1)	68.35 \pm 0.00	1350.00 \pm 0.00	0.05 \pm 0.00	100
	A* (grid size = 0.25)	64.50 \pm 0.00	1440.00 \pm 0.00	2.01 \pm 0.00	100
	PRM	68.03 \pm 0.74	1140.57 \pm 32.96	4.05 \pm 0.14	92
	B-RRT	77.76 \pm 2.67	1291.86 \pm 102.96	4.09 \pm 1.32	100
	PSO Piecewise Linear	65.13 \pm 0.20	1058.76 \pm 21.34	1.32 \pm 0.00	100
	EGA Piecewise Linear	64.16 \pm 0.22	1086.76 \pm 16.77	1.02 \pm 0.00	100
	EGA Spline	65.93 \pm 0.56	1214.93 \pm 30.34	1.04 \pm 0.00	100
		—	—	—	0
Map 8	A* (grid size = 1)	41.45 \pm 0.00	945.00 \pm 0.00	0.04 \pm 0.00	100
	A* (grid size = 0.25)	39.33 \pm 0.00	990.01 \pm 0.00	1.32 \pm 0.00	100
	PRM	40.97 \pm 0.69	714.70 \pm 81.70	11.87 \pm 0.47	70
	B-RRT	46.18 \pm 1.76	881.95 \pm 80.97	3.97 \pm 3.16	100
	PSO Piecewise Linear	39.91 \pm 0.11	719.42 \pm 12.28	1.34 \pm 0.01	100
	EGA Piecewise Linear	39.30 \pm 0.23	678.92 \pm 34.28	1.01 \pm 0.00	100
	EGA Spline	39.82 \pm 0.38	756.21 \pm 39.50	1.04 \pm 0.01	100
		—	—	—	0
Map 9	A* (grid size = 1)	197.20 \pm 0.00	1710.00 \pm 0.00	0.60 \pm 0.00	100
	A* (grid size = 0.25)	192.40 \pm 0.00	2025.00 \pm 0.00	61.20 \pm 0.00	100
	PRM	199.79 \pm 1.25	1443.66 \pm 26.75	46.71 \pm 0.77	32
	B-RRT	—	—	—	0
	PSO Piecewise Linear	195.88 \pm 0.02	1441.48 \pm 6.86	2.80 \pm 0.01	100
	EGA Piecewise Linear	194.28 \pm 0.52	1487.62 \pm 20.31	2.68 \pm 0.01	100
	EGA Spline	197.09 \pm 0.38	1578.16 \pm 37.65	2.54 \pm 0.01	100
		—	—	—	0
Map 10	A* (grid size = 1)	157.09 \pm 0.00	1575.00 \pm 0.00	0.17 \pm 0.00	100
	A* (grid size = 0.25)	152.58 \pm 0.00	1530.00 \pm 0.00	8.10 \pm 0.00	100
	PRM	160.26 \pm 1.78	1407.34 \pm 80.44	53.25 \pm 1.19	18
	B-RRT	—	—	—	0
	PSO Linear	157.71 \pm 0.11	1367.00 \pm 4.69	1.91 \pm 0.01	100
	EGA Piecewise Linear	155.05 \pm 0.29	1350.41 \pm 17.14	1.70 \pm 0.01	100
	EGA Spline	156.42 \pm 0.65	1508.47 \pm 41.41	1.63 \pm 0.01	100
		—	—	—	0

(continued on next page)

Table 2 (continued)

		Path length	Path smoothness	Runtime (s)	Success rate (%)
Map 11	A* (grid size = 1)	43.86 ± 0.00	855.00 ± 0.00	0.10 ± 0.00	100
	A* (grid size = 0.25)	42.67 ± 0.00	1035.00 ± 0.00	5.22 ± 0.00	100
	PRM	43.39 ± 0.37	262.49 ± 104.87	26.12 ± 1.10	100
	B-RRT	47.56 ± 1.82	565.36 ± 166.22	2.65 ± 2.39	100
	PSO Piecewise Linear	42.78 ± 0.16	601.22 ± 15.06	1.83 ± 0.01	100
	EGA Piecewise Linear	42.23 ± 0.16	534.97 ± 59.55	1.53 ± 0.01	100
	EGA Spline	42.97 ± 0.43	682.08 ± 87.53	1.57 ± 0.02	100
Map 12	A* (grid size = 1)	51.76 ± 0.00	1620.00 ± 0.00	0.08 ± 0.00	100
	A* (grid size = 0.25)	49.47 ± 0.00	1845.00 ± 0.00	6.00 ± 0.00	100
	PRM	59.49 ± 3.19	1326.63 ± 125.19	27.78 ± 0.71	92
	B-RRT	57.51 ± 3.08	1519.08 ± 92.40	10.17 ± 9.23	100
	PSO Piecewise Linear	51.54 ± 0.01	1620.02 ± 13.91	1.92 ± 0.01	100
	EGA Piecewise Linear	49.72 ± 0.20	1302.49 ± 92.38	1.59 ± 0.01	100
	EGA Spline	51.98 ± 0.38	1530.81 ± 73.83	1.60 ± 0.01	100

equal, first, the parent with the smaller number of PBs is selected, P_s . Then, PBs of the offspring (x^f, y^f) are calculated by taking the mean of each PB of the P_s and the nearest PB of the other parent. In the second crossover operator, the two parents are combined in a random fashion to increase the diversity of the population and search the whole free space:

$$x^f = \alpha x_1^p + (1 - \alpha)x_2^p, \quad y^f = \alpha y_1^p + (1 - \alpha)y_2^p \quad (2)$$

where α is a vector of random numbers in the range of $[-1, +1]$. In the case that the number of PBs of the two parents are not equal, similar to the first crossover operator, the PBs with minimum distance are combined together. While the first crossover operator is designed to generate shorter and smoother paths, the second crossover operator is adopted to search the environment in a random fashion and avoid premature convergence.

4.3.3. Mutation operators

The mutation operators are useful in terms of increasing the diversity of the population, searching all areas of the environment and moving a set of PBs in a particular direction to increase the path smoothness and decrease the path length. The first mutation operator is an arithmetic mutation operator which changes the position of a pre-defined number of PBs in random directions. This operator is considerably helpful for increasing the diversity of the population and avoiding premature convergence.

The second mutation operator modifies the position of a PB to improve the path in terms of both length and smoothness. This operator improves the position of a PB by integrating its current position (P_i) and the directions toward the PBs in its either side, P_{i-1} and P_{i+1} ,

$$\begin{aligned} x_i^f &= x_i^p + \alpha(x_{i-1}^p - x_i^p) + \beta(x_{i+1}^p - x_i^p) \\ y_i^f &= y_i^p + \alpha(y_{i-1}^p - y_i^p) + \beta(y_{i+1}^p - y_i^p) \end{aligned} \quad (3)$$

where α and β are small positive coefficients. Fig. 4(a) shows the mechanism of the second mutation operator. This operator produces paths with higher smoothness and shorter length. The third mutation operator moves arbitrarily selected PBs towards the destination (see Fig. 4(b)). The combination of these three mutation operators provides a powerful tool which improves both the exploration and exploitation abilities of the EGA.

4.3.4. Deletion operator

By choosing a small enough grid size, the proposed APF will produce initial feasible paths with too many PBs. In other words, the APF begins from the start location and select appropriate nodes in the environment, using the mechanism mentioned in Section 4.2, to reach the destination. As such, in almost all cases, too many PBs will be selected by APF. The more the number of

PBs, the less flexible the fitted piecewise linear/spline path on the PBs. Also, using too many PBs is computationally inefficient. On the other hand, by decreasing the number of PBs, the probability of making an infeasible path will increase. Thus, the number of PBs should be tuned intelligently. To make the algorithm performance robust, independent of the number of PBs in the initial paths, a deletion operator is used in the EGA. Deletion operator deletes redundant PBs of a path. For this purpose, first, an individual is selected from the mating pool. Then the deletion operator assesses each PB sequentially. If the deletion of any chosen PB is beneficial (turns an infeasible path to a feasible one, reduces the path length, increases the path smoothness), the deletion operator deletes it.

4.4. Objective function

The objective function determines the quality of a path based on predefined measures. In this paper, two different sets of objective functions are used: (i) single objective: path length, and (ii) multi-objective: path length, path smoothness, path safety. The single objective function (path length) is used to compare the performance of the proposed algorithm with A*, PRM, B-RRT, and PSO. Next, a multi-objective function is presented to determine the optimal path (the path with the shortest length, the highest degree of smoothness, and maximum safety) in complex environments. For this purpose, a weighted linear combination of the mentioned objectives is considered as the objective function.

When a piecewise linear path is built using the PBs (connecting the PBs using straight lines as shown in Fig. 5(a)), the path length, $\mathcal{L}(p)$, is calculated by summing the length of the all line segments in the path p . Moreover, for the piecewise linear path p with N segments, the path smoothness and safety can be calculated using:

$$\mathcal{S}(p) = \sum_{i=2}^{N-1} \mathcal{A}(\ell_{i-1}, \ell_i) \quad (4)$$

$$\mathcal{R}(p) = \sum_{i=1}^{N-1} \mathcal{D}_i \quad (5)$$

where $\mathcal{S}(p)$ and $\mathcal{R}(p)$ are the smoothness and safety of the path p , respectively, $\mathcal{A}(\ell_{i-1}, \ell_i)$ is the change in the direction of the robot to follow the line ℓ_i after line ℓ_{i-1} , Fig. 5(a), and can be calculated using the cosine rule, \mathcal{D}_i is the distance of the PB_i from the closest obstacle, and N is the number of piecewise linear segments of path p . Then, the objective function is calculated as the weighted linear sum of the three objectives:

$$\mathcal{C}(p) = \mathcal{W}_{\mathcal{L}} \times \mathcal{L}(p) + \mathcal{W}_s \times \mathcal{S}(p) + \frac{\mathcal{W}_{\mathcal{R}}}{\mathcal{R}(p)} \quad (6)$$

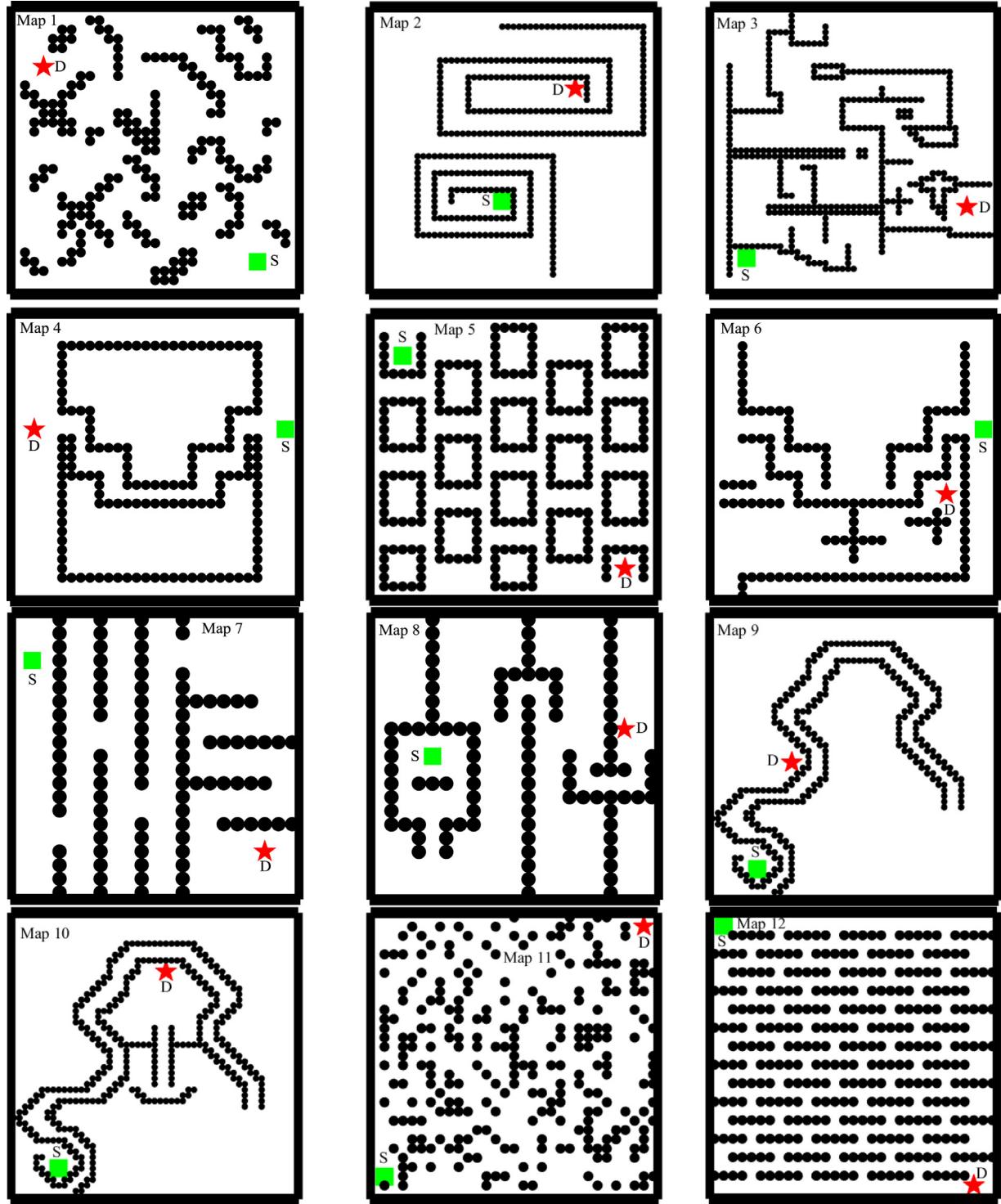


Fig. 6. The set of benchmark maps used for performance evaluation.

where \mathcal{W}_L , \mathcal{W}_S , and \mathcal{W}_R are the weights of the path length, path smoothness, and path safety. For infeasible paths, a large penalty will be added to the objective value of the path.

When the end-user is interested in a path with a gradual direction change, a spline path can be built by connecting the PBs using a cubic spline as shown in Fig. 5(b). In this case, the smooth line segment between PB_i and PB_{i-1} , Fig. 5(b), is comprised of several short linear segments and the objectives can be calculated similar to the piecewise linear case.

4.5. Multi-robot path planning

Optimal PP of multiple mobile robots which move simultaneously is a challenging problem. In this situation, the PP problem can be categorized as a dynamic PP problem with moving obstacles; each robot is a moving obstacle for all other robots. Thus, three tasks should be handled simultaneously: (i) the optimal path for each robot should be determined based on the position of static obstacles, (ii) the determined paths should be checked for possible

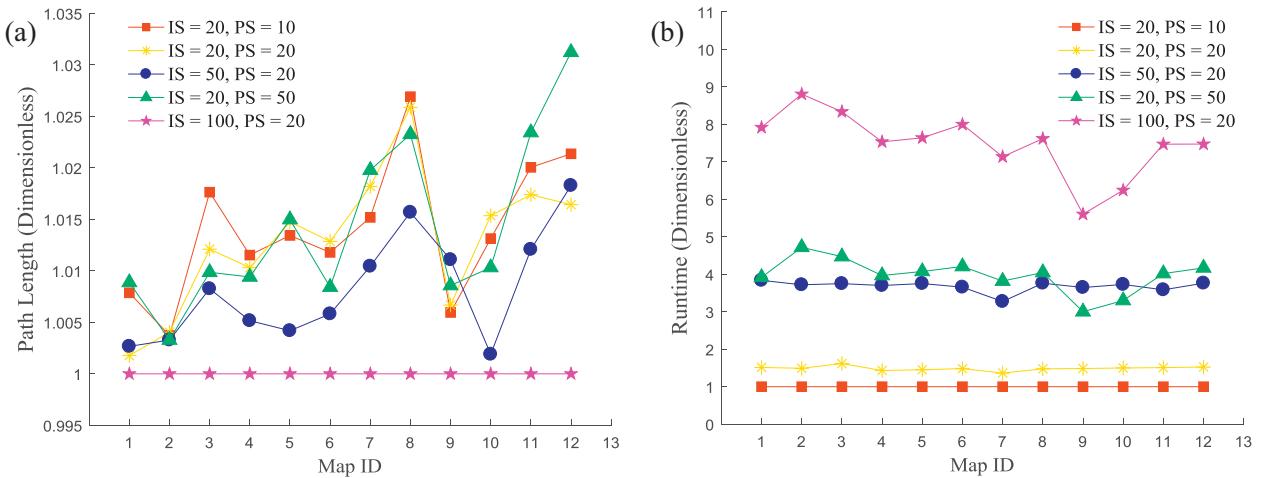


Fig. 7. Normalized mean (a) path length and (b) runtime obtained for 50 executions of the piecewise linear case. For each environment, all path lengths/runtimes were normalized based on the minimum value obtained among control parameters. IS: number of iterations of the EGA, and PS: population size of the EGA.

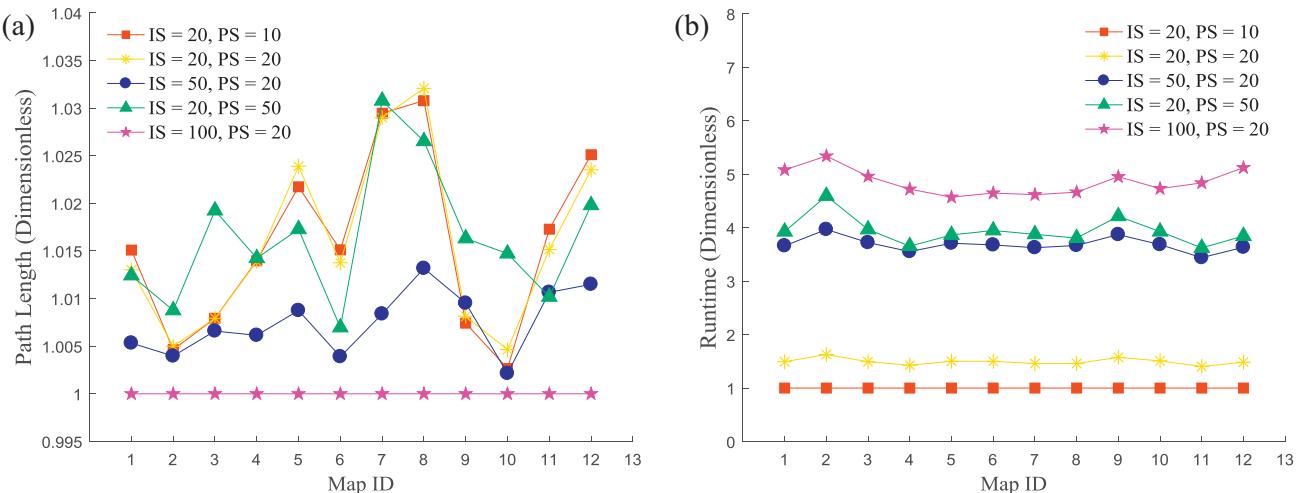


Fig. 8. Normalized mean (a) path length and (b) runtime obtained for 50 executions of the spline case. For each environment, all path lengths/runtimes were normalized based on the minimum value obtained among control parameters. IS: number of iterations of the EGA, and PS: population size of the EGA.

collisions among robots, and (iii) the optimal paths should be modified based on the collisions. In this paper, mobile robots with constant speed are considered. In this case, all robots travel the same distance in a particular period of time, and this distance can be used to detect possible collisions between robots. For PP of mobile robots with variable speed, only the distance calculation should be modified based on different speed values, and the rest of the algorithm can be applied in the same way.

Two mechanisms are introduced in this section to deal with the multi-robot PP problem. The first one measures the closest distance between every two robots and increases the objective function value associated with their paths if they are closer than a predefined safety distance. The penalty of the objective function value is inversely related to the distance between two robots; if their distance is larger than the predefined safety distance, the penalty is zero, otherwise, the closer the distance, the greater the penalty. In this paper, a linear penalty function is used, however, in more conservative cases, exponential penalty functions can also be used. This penalty mechanism ensures that the collision between robots will receive a higher priority in comparison to other objectives in Eq. (6), and path improvement through genetic operators will not end in a collision with other robots.

The second mechanism modifies the paths which collide with each other. For this purpose, one of the paths is selected randomly, and one of its PBs is selected and moved in a random fashion. This modification will remove the detected collision; however, the new path should be checked for possible collision with all other paths again. If the new path is collision-free, it will be replaced by the old one. Otherwise, the collision removal process will be repeated.

5. Results and discussion

To evaluate the performance of the proposed EGA, 12 planar environments were built. The environments were different in terms of size, and shape and number of obstacles. This set of environments provided the chance to conduct a comprehensive study on the effectiveness of the proposed EGA in terms of objective value, runtime, and rate of success. Fig. 6 shows the environments and Table 1 presents the characteristics of each map. It should be noted that the size of the environments and discretization size (for PP in the discrete gridded environment) are dimensionless.

5.1. Proposed algorithm control parameters

The control parameters of the proposed APF are (i) the discretization size, (ii) the highest assigned potential (ξ_0), and (iii) the

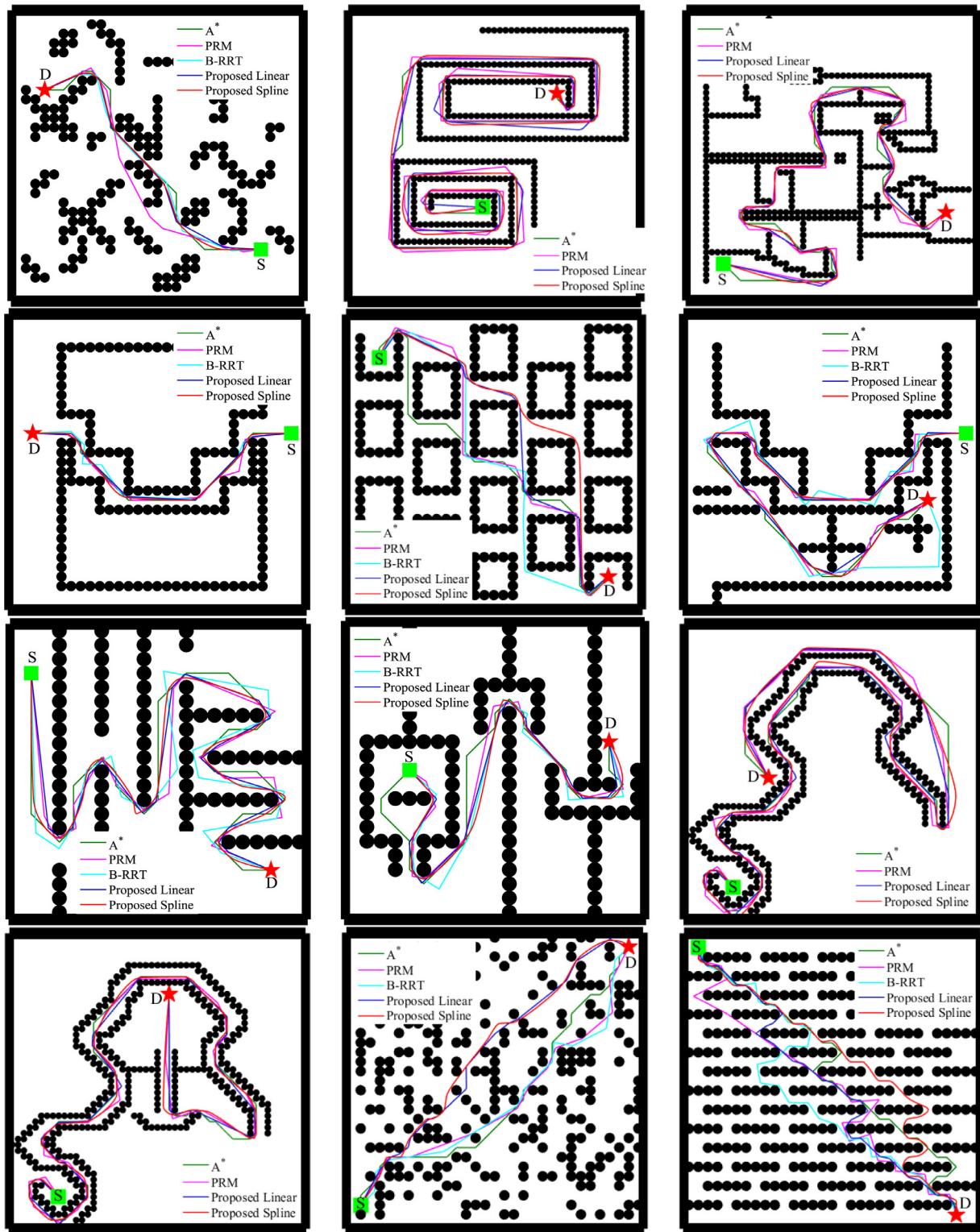


Fig. 9. The best paths obtained by A*, PRM, B-RRT, and the proposed EGA with piecewise linear and spline paths.

decrement step (α). The discretization size always was considered as 1 unit, and both ξ_0 and α were selected arbitrarily such that the lowest potential ($-\xi_0$) always belonged to obstacles. The control parameters of the EGA are population size, number of iterations, and crossover and mutation probabilities. The crossover and mutation probabilities were set to 0.35 and 0.15, respectively, and the

mutation rate (the number of genes in a chromosome which were affected by mutation) was set to 0.2.

The population size (PS) and number of iterations (IS) directly affect the quality of the solutions and runtime. Thus, a comparison has been made between several PSs and ISs to find the best value for each parameter. For this purpose, EGA was executed 50 times

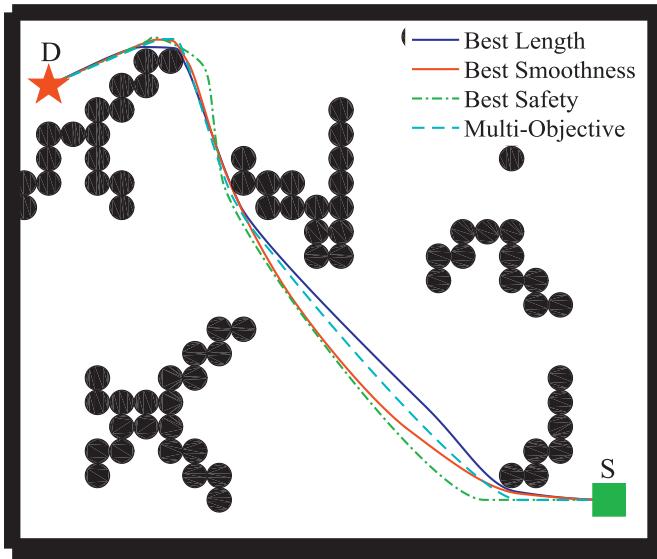


Fig. 10. Single-objective vs. multi-objective PP problem. A weighted linear combination of path length, path smoothness, and path safety was used with adjusted coefficients as the objective function.

for each environment, and the mean value of the objective function (path length) and runtime were calculated for these executions. Figs. 7 and 8 show the dimensionless path lengths and runtimes for piecewise linear and spline paths, respectively. To provide a meaningful comparison between path lengths and runtimes associated with five (IS, PS) pairs, the mean values were normalized according to the minimum value among all five (IS, PS) pairs.

As it was expected, Figs. 7(a) and 8(a) illustrate that the shortest paths were obtained by the pair (IS = 100, PS = 20). However, in the worst-case scenario, the length of the paths obtained by the pair (IS = 20, PS = 10) for piecewise linear and spline cases were 1.026 and 1.031 times longer than the minimum length obtained by the pair (IS = 100, PS = 20). At the same time, the fastest runtimes were always achieved by the pair (IS = 20, PS = 10), and except for the pair (IS = 20, PS = 20), the other pairs lasted at least three times longer than the pair (IS = 20, PS = 10). The shortest path lengths obtained for piecewise linear case by the pair (IS = 100, PS = 20) were 33.80, 249.78, 124.74, 32.96, 45.40, 65.32, 63.20, 38.27, 193.13, 153.04, 41.40, and 48.68 for environments 1 to 12, respectively. Also, the smallest runtimes obtained for piecewise linear case by (IS = 20, PS = 10) were 1.37, 2.57, 1.68, 1.30, 1.53, 1.29, 1.22, 1.21, 2.68, 1.70, 1.53, 1.59 seconds for environments 1 to 12, respectively. Both Figs. 7 and 8 show that although the shortest paths were obtained by the pair (IS = 100, PS = 20), however, the improvements in the path lengths were negligible in comparison to the ones obtained by the pair (IS = 20, PS = 10), considering the fact that (IS = 20, PS = 10) is at least five times faster than (IS = 100, PS = 20). Moreover, Figs. 7(a) and 8(a) show the robustness of the EGA, such that its performance was not highly affected by the control parameters. Another interesting observation from Figs. 7 and 8 is that although the runtimes for both (IS = 50, PS = 20) and (IS = 20, PS = 50) is almost the same, the pair with a higher number of iterations outperforms the other in terms of path length.

5.2. Comparative study

This section provides a performance comparison between the proposed EGA and A* (with discretization sizes of 1 unit and 0.25 units), PRM, B-RRT, and PSO. As the A*, PRM, and B-RRT were

mainly used to find the shortest path, the path length is considered as the objective for the EGA and PSO in the comparative study.

In contrast to Dijkstra's algorithm, A* benefits from a heuristic cost, which estimates the path length between current and destination locations for PP in a discrete gridded environment. The A* is guaranteed to find the optimal path if the heuristic function is admissible, i.e., the estimated distance from every location to the destination is smaller or equal to the actual distance. In this paper, the discretization size of 1 unit and 0.25 units were used for A* and Euclidean distance was used as the heuristic function.

PRM is a popular sampling-based PP algorithm which consists of two steps: (1) learning, where the roadmap of the free space is constructed, and (2) searching for a path in the constructed roadmap using Dijkstra's algorithms. Similar to PRM, B-RRT is a sampling-based PP algorithm which is working based on two independent trees, which grow simultaneously from start and destination locations. For both PRM and B-RRT, a deletion operator (the same as Section 4.3.4) was applied to the obtained path to remove additional nodes. Also, a preliminary investigation was performed to determine the proper number of sampled-points and effective radius (sampled-points neighboring in the effective radius will be connected) of the PRM for each environment. The selected number of sampled-points and effective radius are (100,9), (900,7.5), (1100,5), (300,6), (300,6), (300,9), (300,4), (350,6), (1100,5), (1100,5), (300,9) and (300,9) for environments 1 to 12, respectively. Also, the step size of the B-RRT (the distance at which the branches grow at each iteration) is set to 2 in all environments.

Both EGA and PSO were executed using control parameters of IS = 20 and PS = 10. Also, a feasible initial population obtained from the proposed APF was used to initiate the PSO. For PSO, the control parameters were set as follows: inertia weight = 0.5, inertia weight damping ratio = 0.99, personal learning coefficient = 2, and global learning coefficient = 2. Considering the stochastic behavior of the proposed EGA, PSO, PRM, and B-RRT, these algorithms were executed 50 times for each environment and the mean and standard deviation values of the results were presented for comparison.

Table 2 shows the mean value and standard deviation of the path length, smoothness, and runtime for 50 executions of the A* (with discretization sizes of 1 unit and 0.25 units), PRM, B-RRT, PSO, and the proposed EGA with piecewise linear and spline paths. The success rates of the mentioned algorithms in finding a feasible path in 50 executions are also presented in the last column of Table 2. Considering the deterministic nature of the A*, it always finds the same path for an environment. Thus, the standard deviation value and success rate of the A* are always zero and 100%, respectively.

Table 2 shows that for seven environments (out of 12), the shortest path was obtained by the proposed EGA, while for the others, A* with discretization size of 0.25 units obtained shortest paths. According to Table 2, paths obtained by the proposed EGA were 3.5% and 1% shorter on average than those obtained by A* with discretization sizes of 1 and 0.25 units, respectively. At the same time, although the path length was considered as the objective function, the introduced crossover and mutation operators enabled the proposed EGA to achieve smoothest paths in all environments, except maps 5, 7, 9, and 11. Particularly, while PRM showed the ability to obtain smooth paths, the path lengths were considerably longer than the EGA for all environments. As it was expected, A* with discretization size of 1 and PRM are the fastest and slowest PP algorithms, respectively. However, considering the path length, path smoothness, and runtime simultaneously, the proposed EGA showed the best performance for all environments.

An important feature of the proposed EGA is that it is guaranteed to find a feasible solution (similar to A*) if one exists. The simulations showed that the PRM and B-RRT face difficulty to find

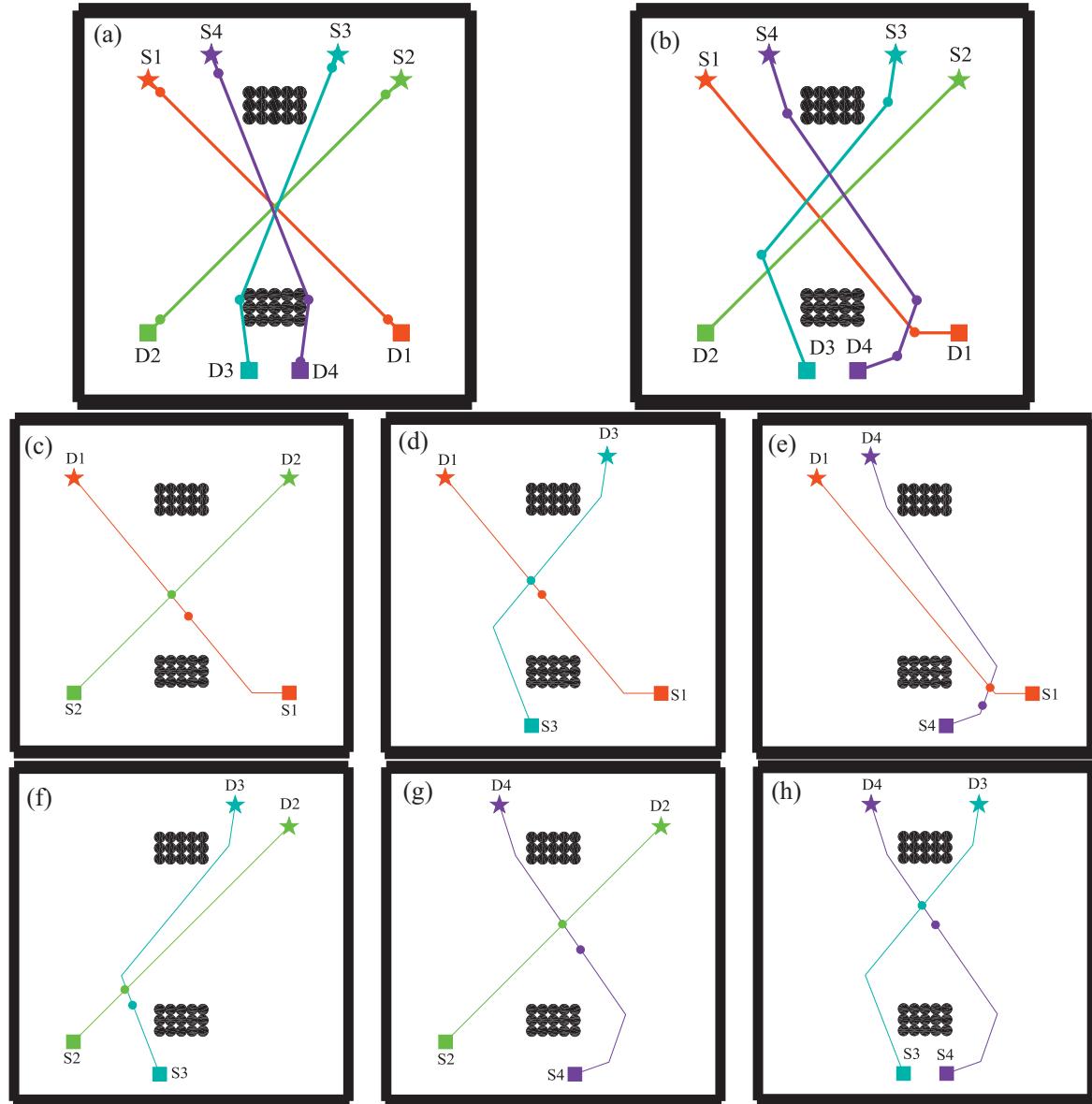


Fig. 11. Multi-robot PP using the proposed EGA. (a) Mobile robots will collide together if they move in their optimal paths; (b) the collision removal operator modifies the position of PBs to remove collisions; (c) to (h) the position of each two robots with respect to each other.

Table 3

Objective function evaluation in the multi-objective PP problem.

	$\mathcal{W}_L = 1$	$\mathcal{W}_L = 0$	$\mathcal{W}_L = 0$	$\mathcal{W}_L = 1.000$
	$\mathcal{W}_S = 0$	$\mathcal{W}_S = 1$	$\mathcal{W}_S = 0$	$\mathcal{W}_S = 0.187$
	$\mathcal{W}_R = 0$	$\mathcal{W}_R = 0$	$\mathcal{W}_R = 1$	$\mathcal{W}_R = 0.201$
Best length	32.71	0	0	33.49
Mean length	32.99	0	0	33.56
STD length	0.120	0	0	0.278
Best smoothness	0	174.81	0	32.40
Mean smoothness	0	189.93	0	36.16
STD smoothness	0	6.3098	0	2.247
Best safety	0	0	154.76	34.17
Mean safety	0	0	157.28	35.81
STD safety	0	0	1.5162	1.103

stop growing trees for run times longer than 120 seconds. Thus, for large complex environments, maps 2, 3, 9, and 10, the B-RRT could not find any feasible solution in 50 executions.

Low standard deviation values for path length, path smoothness, and runtime show that not only the proposed algorithm is able to find the best solution in every environment, but also, its performance is not highly affected by the initial solutions and the stochastic behavior of the EGA's operators. In contrast, both PRM and B-RRT show significant standard deviation values, which indicates their random sampling nature. Finally, Fig. 9 shows the best path obtained by each algorithm for 12 environments.

Considering the fact that PSO was developed for solving continuous problems, it will be an ideal choice for solving PP problem in the continuous environment. Nonetheless, Table 2 shows that the proposed EGA outperformed the PSO in terms of path length and runtime for all environments. This comparison shows the importance of utilizing customized operators for PP.

a feasible path in large complex environments, like maps 2 and 3. Specifically, an additional terminal condition was added to B-RRT;

5.3. Multi-objective path planning

For multi-objective PP, a weighted linear combination of the path length ($\mathcal{L}(p)$), path smoothness ($\mathcal{S}(p)$), and path safety ($\mathcal{R}(p)$) was defined as the objective function according to Eq. (6). As the objective function terms might have different orders of magnitude, e.g., in Table 2, the path length is sometimes 10 times smaller than the path smoothness, minimal improvement of an objective will affect the total objective value more than the considerable improvement of other objectives. Thus, a simple approach is introduced to find the proper objective function coefficients, $\mathcal{W}_{\mathcal{L}}$, $\mathcal{W}_{\mathcal{S}}$, and $\mathcal{W}_{\mathcal{R}}$, to normalize the value of all objectives with respect to each other. The following paragraphs describes this procedure for multi-objective PP problem shown in Fig. 10.

First three simulations should be performed by setting the coefficient associated with one term of the objective function to 1 and the other two to 0. For each case, the best objective value among 50 executions, and the mean and standard deviation of 50 executions were tabulated in Table 3. The first column of Table 3 and the blue path in Fig. 10 show the PP with path length as an objective. In this case, the proposed EGA tried to find a piecewise linear path with sharp direction changes in the vicinity of obstacles. When the smoothness coefficient was set to 1 and the other two to 0, the red path in Fig. 10 was obtained. In this case, the optimal path was a curve with the largest radius to minimize sharp direction changes. Finally, setting the safety coefficient to 1 and the other two to 0 resulted in the green path in Fig. 10. As it was expected, the path maintained a safety margin with respect to obstacles.

The first three columns of Table 3 illustrate that $\mathcal{L}(p)$, $\mathcal{S}(p)$, and $\mathcal{R}(p)$ have different orders of magnitudes. To make the orders of magnitude comparable, the objective function coefficients must be adjusted properly. To determine the adjusted coefficients, first the optimal value of $\mathcal{L}(p)$, $\mathcal{S}(p)$, and $\mathcal{R}(p)$ should be calculated, similar to those in the first three columns of Table 3. Next, the adjusted coefficients can be found as: $\mathcal{W}_{\mathcal{L}} = 1$, $\mathcal{W}_{\mathcal{S}} = \mathcal{L}_{\min}/\mathcal{S}_{\min}$, and $\mathcal{W}_{\mathcal{R}} = \mathcal{L}_{\min}/\mathcal{R}_{\min}$. Using these adjusted coefficients, the objective function terms will be in the same order of magnitude. The dashed cyan path in Fig. 10 and the last column in Table 3 show the optimal path obtained using adjusted coefficients. It can be seen from the last column of Table 3 that the three objective terms have almost the same magnitude, and therefore, have the same contribution in the total objective value.

5.4. Multi-robot path planning

This section presents the application of the proposed EGA for multi-robot PP. To assess the performance of the proposed EGA, a four-robot problem, as shown in Fig. 11, was designed. The start and destination locations of the mobile robots are located in positions that ensure collision between robots if they move in their optimal path, as it is shown in Fig. 11(a). By taking advantage of the collision term in the objective function and collision removal operator, the solution shown in Fig. 11(b) was obtained. In Fig. 11(b), the position of PBs, shown with filled circles, before collision point was changed to avoid collision between agents.

Fig. 11(c)–(h) illustrate the path of each two mobile robots and their position when one of them reaches the collision point. For instance, in Fig. 11(c), as the second robot (green line), is closer to the collision point, it passes the collision point before the first robot reaches to this point. The same issue can be seen for every other two robots in Figs. 11(d) to (h). Thus, in the multi-robot case, the proposed EGA modifies the position of PBs to optimize the objectives like path length and smoothness using the proposed genetic operators and avoid the collision by adding a distance term to the objective function and removing the possible collisions by the introduced collision removal operator.

6. Conclusion

This paper presents a novel algorithm for PP of multiple mobile robots in a continuous environment. The proposed algorithm is guaranteed to find an optimal (or near-optimal) path in every complex environment. For this purpose, first, the proposed APF was used to build all feasible paths between any arbitrarily selected start and destination locations in a discrete gridded environment. The APF concept is guaranteed to find at least one feasible path if one exists. Next, the purposed EGA modifies the position of PBs obtained from initial paths to find the optimal piecewise linear or spline path between start and destination locations. Five customized crossover and mutation operators were introduced in EGA to improve the position of PBs. Also, a deletion operator was used to tune the number of PBs during the EGA evolution automatically. Finally, the proposed algorithm was extended to handle the multi-robot PP problem. In this case, the distance between mobile robots was used as a criterion to avoid a collision. Moreover, a collision removal operator was introduced to remove the possible collision between paths obtained by genetic operators.

Path length, smoothness, and safety were considered as objectives to evaluate the quality of the obtained paths. Such a composite objective function enables us to adjust the importance of each term (path length, smoothness, and safety) for any particular application. In addition, the simple approach introduced in this paper can be used to equalize the effect of the three terms for a multi-objective PP problem. A comparative study between the proposed EGA and A*, PRM, B-RRT, and PSO showed the superiority of proposed EGA in comparison with the mentioned well-recognized PP algorithms in terms of path length, path smoothness, runtime, and success rate. The comparative study was performed for 12 environments with various sizes and complexities. The comparative study showed that the success rate of the proposed EGA is always 100%, no matter how large and complex the environment is.

Simulation results showed that the control parameters of the EGA, including the population size and number of iterations, do not affect the performance of the EGA significantly. In addition, selecting a small population and few iterations will result in better solutions in comparison with A*, PRM, and B-RRT. At the same time, the variation in the best solutions obtained for several separate executions of the EGA is considerably smaller than probabilistic algorithms, PRM and B-RRT. The EGA has also shown a great potential for multi-robot PP problem. Adding a new objective term which represents the distance between mobile robots and the collision removal operator could handle the dynamic nature of the multi-robot PP problem and avoid collision between paths.

As a future work, the proposed EGA should be extended for PP in an environment with moving obstacles, since the present work only considered static obstacles. Moreover, a multi-objective optimization framework, like NSGA-II, can be used to combine the objectives, instead of the weighted linear combination. In this way, the weights will be removed from the objective function, and the optimal value for all objectives will be obtained at the same time. Finally, it is worth investigating the effect of practical motion planning constraints, such as constraints on velocity or steering angle of the robot, in the future works.

The proposed EGA can be considered as a time-efficient PP algorithm in a continuous environment, which guarantees to find optimal (or near-optimal) path in every complex environment. In comparison to deterministic PP algorithms like A* or Dijkstra's algorithm, it could be said that the proposed algorithm is not limited to the grid size of a discrete environment and is able to find the best piecewise linear or spline path. The two advantages of the proposed EGA in comparison to probabilistic PP algorithms like PRM or B-RRT, or evolutionary algorithms like PSO are time efficiency and the ability to find a near-optimal path in any complex

environment. As such, the proposed EGA can be considered as a significant contribution to the field of expert and intelligent systems for mobile robot PP.

Author contribution

Milad Nazarahari: Conceptualization, Methodology, Software, Formal Analysis, Writing – Original Draft.

Esmaael Khanmirza: Conceptualization, Methodology, Validation, Writing – Review & Editing, Supervision, Project Administration.

Samira Doostie: Methodology, Software, Validation, Resources, Formal Analysis, Writing – Original Draft.

Conflict of interest statement

None.

References

- Canny, J. F. (1998). *The complexity of robot motion planning*. Cambridge, Massachusetts: MIT Press.
- Castillo, O., Trujillo, L., & Melin, P. (2007). Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots. *Soft Computing*, 11(3), 269–279.
- Contreras-Cruz, M. A., Ayala-Ramirez, V., & Hernandez-Belmonte, U. H. (2015). Mobile robot path planning using artificial bee colony and evolutionary programming. *Applied Soft Computing Journal*, 30, 319–328. <https://doi.org/10.1016/j.asoc.2015.01.067>.
- Davoodi, M., Panahi, F., Mohades, A., Hashemi, S. N., & Naser, S. (2015). Clear and smooth path planning. *Applied Soft Computing Journal*, 32, 568–579. <https://doi.org/10.1016/j.asoc.2015.04.017>.
- Davoodi, M., Panahi, F., Mohades, A., Hashemi, S. N. S., Naser, S., & Hashemi, S. N. S. (2013). Multi-objective path planning in discrete space. *Applied Soft Computing Journal*, 13(1), 709–720.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269–271.
- Doostie, S., Hoshiar, A. K., Nazarahari, M., Lee, S., & Choi, H. (2018). Optimal path planning of multiple nanoparticles in continuous environment using a novel Adaptive Genetic Algorithm. *Precision Engineering*, 53, 65–78.
- Han, J., & Seo, Y. (2017). Mobile robot path planning with surrounding point set and path improvement. *Applied Soft Computing Journal*, 57, 35–47.
- Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics SSC4*, 4(2), 100–107.
- Hidalgo-Paniagua, A., Vega-Rodríguez, M. A., & Ferruz, J. (2016). Applying the MOVNS (Multi-Objective Variable Neighborhood Search) Algorithm to solve the path planning problem in mobile robotics. *Expert Systems With Applications*, 58, 20–35.
- Hidalgo-Paniagua, A., Vega-Rodríguez, M. A., Ferruz, J., & Pavón, N. (2015). MOSFLA-MRPP: multi-objective shuffled frog-leaping algorithm applied to mobile robot path planning. *Engineering Applications of Artificial Intelligence*, 44, 123–136. doi:10.1016/j.engappai.2015.05.011.
- Hoshiar, A. K., Kianpour, M., Nazarahari, M., & Korayem, M. H. (2016). Path planning in AFM nano-manipulation of multiple spherical nano particles using a co-evolutionary Genetic Algorithm. In *International conference on manipulation, automation and robotics at small scales (MARSS)*. Paris.
- Hossain, M. A., & Ferdous, I. (2015). Autonomous robot path planning in dynamic environment using a new optimization technique inspired by bacterial foraging technique. *Robotics and Autonomous Systems*, 64, 137–141.
- Kala, R., Shukla, A., & Tiwari, R. (2010). Dynamic environment robot path planning using Hierarchical Evolutionary Algorithms. *Cybernetics and Systems: An International Journal*, 41, 435–454.
- Klančar, G., Zdešar, A., Blažič, S., & Škrjanc, I. (2017). Chapter 4 – Path planning. In *Wheeled mobile robotics* (pp. 161–206).
- Korayem, M. H. H., Hoshiar, A. K. K., & Nazarahari, M. (2016). A hybrid co-evolutionary genetic algorithm for multiple nanoparticle assembly task path planning. *The International Journal of Advanced Manufacturing Technology*, 87(9–12), 3527–3543. doi:10.1007/s00170-016-8683-4.
- Kunz, T., Reiser, U., Stilmann, M., & Verl, A. (2010). Real-time path planning for a robot arm in changing environments. In *2010 IEEE/RSJ international conference on intelligent robots and systems* (pp. 5906–5911).
- Liu, C., Liu, H., & Yang, J. (2011). A path planning method based on adaptive genetic algorithm for mobile robot. *Journal of Information and Computational Science*, 8(5), 808–814.
- Mac, T. T., Copot, C., Tran, D. T., & Keyser, R. De. (2017). A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization. *Applied Soft Computing Journal*, 59, 68–76.
- Masehian, E., & Sedighzadeh, D. (2007). Classic and heuristic approaches in robot motion planning – A chronological review. *International Journal of Mechanical and Mechatronics Engineering*, 1(5), 249–254.
- Melin, P., García, M. A. P., Montiel, O., Castillo, O., & Sepúlveda, R. (2009). Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. *Applied Soft Computing*, 9, 1102–1110.
- Mo, H., & Xu, L. (2015). Research of biogeography particle swarm optimization for robot path planning. *Neurocomputing*, 148, 91–99.
- Qu, H., Xing, K., & Alexander, T. (2013). An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. *Neurocomputing*, 120, 509–517.
- Saicharan, B., Tiwari, R., & Roberts, N. (2016). Multi objective optimization based path planning in robotics using nature inspired algorithms: a survey. In *IEEE international conference on power electronics, intelligent control and energy systems*. Delhi.
- Sedaghat, N. (2011). Mobile robot path planning by new structured multi-objective genetic algorithm. In *2011 International conference of soft computing and pattern recognition (SoCPaR)* (pp. 79–83).
- Song, B., & Wang, Z. (2016). A new genetic algorithm approach to smooth path planning for mobile robots. *Assembly Automation*, 36, 138–145.
- Song, B., Wang, Z., & Zou, L. (2017). On global smooth path planning for mobile robots using a novel multimodal delayed PSO algorithm. *Cognitive Computation*, 9, 5–17.
- Tang, B., Zhu, Z., & Luo, J. (2016). Hybridizing particle swarm optimization and differential evolution for the mobile robot global path planning. *International Journal of Advanced Robotic Systems*.
- Thoa, T., Copot, C., Trung, D., & Keyser, R. De. (2016). Heuristic approaches in robot path planning: A survey. *Robotics and Autonomous Systems*, 86, 13–28.
- Tisdale, J., Kim, Z., & Hedrick, J. K. (2009). Autonomous UAV path planning and estimation. *IEEE Robotics & Automation Magazine*, 16(2), 35–42.
- Tuncer, A., & Yıldırım, M. (2012). Dynamic path planning of mobile robots with improved genetic algorithm. *Computers & Electrical Engineering*, 38, 1564–1572.
- Wang, Q., & Hedner, J. (2016). Path planning research for mobile robot based on immune Genetic Algorithm. In *Proceedings of the 17th world congress the international federation of automatic control, Seoul, Korea*: 39 (pp. 164–172).
- Zeng, M., Xi, L., & Xiao, A. (2016). The free step length ant colony algorithm in mobile robot path planning. *Advanced Robotics*, 30(23), 1509–1514.
- Zuo, L., Guo, Q., Xu, X., & Fu, H. (2015). A hierarchical path planning approach based on A* and least-squares policy iteration for mobile robots. *Neurocomputing*, 170, 257–266.