



# A new path following scheme: safe distance from obstacles, smooth path, multi-robots

Ardashir Mohammadzadeh<sup>1</sup> · Behnam Firouzi<sup>2</sup>

Received: 1 March 2022 / Accepted: 8 February 2023 / Published online: 28 February 2023  
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

## Abstract

Robot routing is one of the most important topics in mobile robotics. The goal is to find a continuous path from an initial position to an end destination that is collision-free and optimal or near-optimal. Due to the growing trend of using automatic moving tools in industrial automation, and their application for various purposes such as transportation of goods, and service in industrial and hospital environments, many researchers have decided to conduct research in this field and route planning. The main challenge is to find a short route with a lack of collision with obstacles. This study examines path design for mobile robots and proposes a new and efficient idea for routing. Besides the short distance of the route and lack of collision with obstacles, the proposed method investigates other factors such as the safe distance from obstacles, path smoothness, and multiple robots. The results show the superior precision and speed of the proposed algorithm compared to similar algorithms. The suggested approach finds the shortest path with a safe distance from obstacles, in a minimum time. The major contributions of this method are summarized below: (1) a biogeographical algorithm is formulated for robot routing. (2) To improve the basic biogeographical algorithm, basic operations of the particle swarm optimization and genetic algorithm are integrated with it. (3) In addition to the shortest path problem, other problems such as path smoothness with a new idea and the safe distance from obstacles are included. Path smoothing is performed without involving it in the cost function, and merely through interpolation of the points found by the algorithm. (4) The proposed algorithm results in a good efficiency and finds the appropriate solution in a few iterations.

**Keywords** Biogeographical algorithm · Genetic algorithm · Particle swarm optimization · Path following · Robot

## Abbreviations

|           |                                     |
|-----------|-------------------------------------|
| GA        | Genetic algorithm                   |
| PSO       | Particle swarm optimization         |
| HHO       | Harris Hawk optimization            |
| BBO       | Biogeography-based optimization     |
| HSI       | Habitat suitability index           |
| SIV       | Ability index variable              |
| $P_{new}$ | New population                      |
| $P_h$     | h-th solution inside the population |

|                    |                              |
|--------------------|------------------------------|
| $\eta$             | Learning rate                |
| $\lambda_h$        | Emigration rate              |
| $\mu_h$            | Immigration rate             |
| $\Delta$           | Safe distance                |
| $r_{obs}(k)$       | Radii of k-th obstacle       |
| $x_{obs}, y_{obs}$ | x–y coordinates of obstacles |

## 1 Introduction

### 1.1 General overview and objectives

Automation has emerged in different sectors of the industry and production in recent years and is developing every day. Only a few decades have passed since the emergence of fully mechanized factories in which all processes are automatic and human force has no administrative role. We have recently witnessed the establishment of mechanized factories with amazing design, construction, and operation. The concept and knowledge of automatic control and the use of

✉ Ardashir Mohammadzadeh  
a.mzadeh@sut.edu.cn

Behnam Firouzi  
behnam.firouzi@ozi.edu.tr

<sup>1</sup> Multidisciplinary Center for Infrastructure Engineering, Shenyang University of Technology, Shenyang 110870, China

<sup>2</sup> Vibrations and Acoustics Laboratory (VAL), Mechanical Engineering Department, Ozyegin University, Istanbul, Turkey

mechanized systems in factories date back to World War II, but remarkable evolutions have occurred only recently. Robots are the most recent stage of human effort toward automatic industries. Robots are human-made machines that do not necessarily move like human beings but can make decisions and create and control pre-determined tasks (Chen and Gao 2020; Gao et al. 2020).

The robot motion planning problem has gained momentum as a major field of robotic science since the late 1970s. Today, mobile robots exist in all spheres of modern life. At the outset of robot industry design, robots used to be simple prototypes comprising several mechanical arms that could be controlled by an engine and performed their tasks in a static and very simple environment. In such environments, it was easy for robots to do their tasks without colliding with obstacles. Today, many engineering problems are related to robots and their motion planning directly or indirectly (Chen 2020; Chen et al. 2022). The routing problem necessitates robots to consider engineering constraints such as physical and transient constraints; therefore, path design methods are divided into two categories based on the conditions of the environment in which the robot is placed:

1. Static: Fixed conditions for the environment and obstacles.
2. Dynamic: Variable number of obstacles and placement.

In terms of available information, these two categories are divided into two main classes:

- Class 1: The robot has inadequate knowledge of its work environment, in which case asynchronous routing is performed.
- Class 2: The robot discovers the environment by using its sensors during motion, also known as synchronous planning.

Robot motion planning involves finding an optimal path from the origin towards the destination, without collision with the obstacles in the environment, considering some restrictions such as safe distance from obstacles, and the path smoothness. In designing a mobile robot path, concave obstacles are a major challenge due to the proximity of the path the robots find in the concave obstacle. This complicates the problem, and the robot may fall into an infinite loop inside the concave obstacle. To solve these problems, different heuristic methods have been developed. Among these, evolutionary algorithms try to overcome the deficits of other methods and have been increasingly used for this purpose (Petrović et al. 2022).

Robots can be divided into two categories in terms of mobility: (1) static-platform robots, (2) dynamic-platform robots. The former group involves series and parallel robots

that are used in closed environments due to the limited working space. Today, however, mobile robots (including flying, walking, wheeled, and swimming robots) that are free from the mentioned limitations of fixed-platform robots are used in different aspects of human society. Thus, these robots collision-free and safe routing is critical. Different constraints are regarded as criteria in the routing problem. These constraints can include constraints governing the kinematics or dynamics of the studied robot. Indices such as the shortest path traveled, least travel time, and minimum energy consumed for travel can also be regarded as the objective functions for routing (Yu et al. 2021). The existing methods can be divided into two groups of global and local routing problems.

Contrary to global routing, local routing often works based on the processing of data dispatched by sensors, which is why the second group is mostly used for online routing. Several methods have been proposed for static environments in the literature. A highly challenging topic in this classification is the collision-free routing of parallel robots in unknown and dynamic environments. During routing in dynamic environments, when the robot faces dynamic constraints, it will be much more difficult to follow the path when avoiding a collision (Gul et al. 2021, 2019).

## 1.2 Literature review

Collision-free routing algorithms have always received attention from researchers, and different algorithms have been proposed for this purpose, each of which has its own merits and demerits. Overall, they can be divided into offline and online algorithms. In offline approaches, all the specifications of the environment, including the placement of robots and obstacles, are known from the outset. In this approach, the environment is assumed to be constant, or its possible changes are neglected. This approach is divided into classic and advanced categories. The most important disadvantage of classic methods is their risk of entrapment in local minima. Numerous approaches based on evolutionary algorithms have been proposed to overcome this problem, which will be reviewed below (Chen et al. 2021; Mukherjee et al. 2021).

In online methods, the specifications of the working environment are not necessarily available and can change with time. In this method, the specifications of the environment are identified momentarily by using the information of sensors. These approaches are also divided into classic and advanced categories. Classic approaches include the potential field, histogram, and dynamic window approaches. In the potential field method, by placing attraction and repulsion fields, efforts are made to have the robot reach the destination without collision. In the histogram method, the robot's path is determined based on the minimum field density and proximity to the destination. In the dynamic window

method, the linear velocity and practical angles are taken into account based on the robot's acceleration (Sandakalum and Ang Jr 2022). In the advanced methods of this group, several techniques have been introduced to reduce the risk of entrapment in local minima. For instance, some references have combined these techniques with evolutionary algorithms including the genetic algorithm (GA). Different methods of robot routing are reviewed in the rest of this chapter, with an emphasis on evolutionary methods, and the results are examined (Mukherjee et al. 2020; Bae et al. 2021).

The PointBug algorithm is proposed by Buniyamin et al. (2011) for the first time. This algorithm can detect the closest obstacles and uses their margin to reach the destination. The important point is the robot's reaction if the margin and edges of objects are not detectable. Thus, its working range must follow certain conditions. A novel method is proposed by Cherubini and Chaumette to perform robot navigation and avoidance of collision simultaneously by using a camera and estimating the risk of a collision via a potential vector field. This framework may interact with unavoidable obstacles, decelerate and, eventually, stop the robot (Cherubini and Chaumette 2011). In this reference, a novel method is presented based on machine vision to detect obstacles and prevent the mobile robot's collision with them in unknown smooth environments. This reference has designed and constructed a mobile robot equipped with a camera.

An algorithm for image processing to detect the obstacles in the environment and control the proposed motion path is also applied to the robot. Three colored lenses are installed on the robot that radiates light of certain angles along the robot's path. The received image of the robot contains the points of these colored lights. By processing the images and determining their coordinates, the position of the obstacle, if any, is detected, and the command for changing the path and avoiding the obstacle is issued. These obstacles can be static or dynamic. The test results showed that the proposed method can detect and prevent collision with obstacles of any shape or material with high reliability, while other methods have certain limitations in this regard (Liu and Liu 2022).

In Song et al. (2021), a method is implemented on the robot that required heavy calculations and processing of images. This method divided the image obtained from the camera into smaller blocks and, then, compared two consecutive images to find the block, in which the motion and change took place. If there was a significant motion between the blocks, it was assumed that there were moving objects inside the block, which could slow down the robot's reaction speed. In Yoo and Oh (2021), a mobile robot is studied that could move in the home environment using information extracted from vision. Moreover, a vision-based system was designed, which consisted of a type of machine that used an omnidirectional camera as the only sensor for home navigation and a stereo vision

system for detecting obstacles. In Lin et al. (2018), a method is proposed, based on which the robot navigated the environment by extracting the environment characteristics such as the color and geometry of the obstacles. However, since the proposed method was based on probabilities, there was the possibility of errors.

In Zhang et al. (2022), a novel algorithm, which is a combination of discrete programming, convex optimization, velocity obstacles, and a receding horizon is presented for motion planning and control of mobile robots. The environment is unknown, and the proposed algorithm is simulated for two separate scenarios, one for a static unknown environment and the other for a dynamic unknown environment, by using the CVX software package in the MATLAB simulator. Then, based on requirements for an actual robot, the algorithm is executed on an EPUCK robot in ROS. The Gurobi solver is used for implementation. To convexify the non-convex constraints, a linear mixed-integer programming method is used. All the constraints must retain their linearity in the next horizons. Thus, to avoid a collision, two types of constraints are utilized. In the first horizon, the velocity obstacle constraint is used to ensure a lack of collision, and in the next horizons, constraints based on the bug method concept are employed to ensure a lack of collision and partly model the unknowns in measuring positions and velocities. Results show the high reliability of this algorithm and its adherence to all safety requirements and lack of collision. The solution time is more than 0.004 s in each stage, suggesting the immediate nature of this routing method.

A reinforcement learning algorithm is employed by Lee and Jeong (2021) for the optimal location of intermediate destinations and found paths that lead to more precise maps of the environment. The innovation of this reference lies in its consideration of the location of intermediate points such that, without the expert observer having any information about the map, the map of the environment is constructed with better precision. By using the reinforcement learning algorithm, strategies are designed based on which the robot's motion reduces the a posteriori uncertainty of the robot. As noted before, a major method in robot path design is the artificial potential field method because it is based on simple mathematical calculations. The most important demerit of this method is entrapment in local minima. An approach to overcome this problem is the use of optimization methods for finding approach attraction and repulsion coefficients and step size that can pass local minima and consider path length in optimization. In Ding et al. (2020), Harris Hawk optimization (HHO) algorithm is used in order to enhance the patient coverage using mobile wireless sensors. HHO is a nature-inspired evolutionary algorithm that has shown superiority in a wide range of engineering problems due to its ability to maintain a balance between exploitation and exploration phases (Abualigah et al. 2022, 2021).

In Peng and Qiu (2022) and Barma et al. (2021), an improved GA is adopted for routing a flying robot. In addition to the length of the path, other parameters, such as path safety and smoothness, are also included in the design. The GA is combined with another algorithm for the routing problem, showing improved precision and speed of routing. These approaches include the combination of GA with the ant colony algorithm, Particle swarm optimization (PSO), and similar methods (Che et al. 2020). The other routing algorithms include: inference search algorithm (Tamizi and Ghaffari 2019), frog leaping (Jazebi and Ghaffari 2020), imperialist competitive algorithm (Mohammadnezhad and Ghaffari 2019), and fuzzy logic approach (Mottaghinia and Ghaffari 2018).

### 1.3 Motivations

The literature review shows that (see Table 1): (1) in most of the above studies, just the shortness of route, and lack of collision with obstacles are taken to account, and the restrictions such as safe distance from obstacles and path smoothness are neglected. (2) The improved optimization techniques have been rarely used for this problem. (3) The problem of multiple robots, considering have been rarely studied considering various restrictions have been seldom studied.

### 1.4 Contributions

Regarding the above discussion, in this paper, a new method is presented. The main contributions are: (1) a biogeographical algorithm is formulated for robot routing. (2) To improve the basic biogeographical algorithm, part of the Particle swarm optimization and genetic algorithm operations are integrated with it. (3) In addition to the shortest path problem, other problems including path smoothness with a new idea and the safe distance from obstacles are included. Path smoothing is performed without involving it in the cost function, and merely through interpolation of the points found by the algorithm. (4) The proposed algorithm results in a

good efficiency and finds the appropriate solution in a few iterations.

## 2 The proposed algorithm

Biogeography-based optimization (BBO) is a relatively new algorithm for smart optimization introduced by Dan Simon in 2008. This algorithm is inspired by the dissemination of species in multiple habitats. A mathematical model was extracted by proposing a probabilistic model of the migration of species in habitats, which led to the creation of a novel optimization model used in BBO. This algorithm enjoys the specifications of both previous algorithms (Taghizadeh et al. 2021; Mirshojaee et al. 2020).

### 2.1 Background of the algorithm

The BBO algorithm is a relatively new algorithm whose good performance has been studied in different domains. The BBO algorithm is used in Jamuna and Swarup (2011) to optimize the design of a power system. In this problem, researchers investigated the performance of BBO along with several other algorithms, including the deterministic tabu search, stochastic tabu search, simulated annealing, redefined genetic algorithm, and hybrid genetic stochastic methods. The findings demonstrated the proper efficiency and performance of the BBO. The performance of BBO and GA is compared in Simon et al. (2011) based on 14 criterion functions. In this study, 14 criterion functions in five, 10, and 20 dimensions were regarded as the objective function. The results indicated the superior ability of BBO in convergence to the absolute optimal solution compared to GA.

By increasing the dimensions of the optimization problem, the success percentage of BBO in finding the optimal solution increased. In solving the five-dimensional problem, in 52% of the cases, the BBO had superior performance. In solving the 10- and 20-dimensional problems, it outperformed the GA in 90 and 92% of the cases, respectively. The BBO is used in Hadidi and Nazari (2013) to optimize the design of a heat exchanger pipe. The BBO algorithm

**Table 1** General comparison of various methods

| References                                                                     | Year | No collision | Safe distance | Path smoothness | Multi robots |
|--------------------------------------------------------------------------------|------|--------------|---------------|-----------------|--------------|
| Buniamin et al. (2011), Cherubini and Chaumette (2011)                         | 2011 | ✓            | ×             | ×               | ×            |
| Che et al. (2020), Chen and Gao (2020)                                         | 2020 | ✓            | ✓             | ×               | ×            |
| Gul et al. (2021), Song et al. (2021), Yoo and Oh (2021), Lee and Jeong (2021) | 2021 | ✓            | ×             | ×               | ×            |
| Yu et al. (2021)                                                               | 2021 | ✓            | ×             | ×               | ✓            |
| Peng and Qiu (2022)                                                            | 2022 | ✓            | ×             | ×               | ✓            |

was used to minimize the total costs of equipment, including investment and total annual costs of energy for the heat exchanger pipe, by altering different design variables, including pipe length, pipe outer diameter, etc. The comparison of the results of BBO with other researchers depicted the superiority of BBO in reducing costs in all examples with similar operational conditions and a reduced execution time, such that the investment cost was reduced by 14% (Goel 2022; Sarma et al. 2017).

### 2.2 Theoretical foundations

Biogeography is the science of studying the geographical distribution of creatures. Biogeographical mathematical simulations describe the migration of a species from one habitat to another, as well as the emergence or extinction of a species. Habitats that are more suitable for the residence of a species have a higher habitat suitability index (HSI). HSI depends on plant coverage, precipitation, area, temperature, etc. Criteria that specify the quality of the habitat are known as suitability index variable (SIV).

In fact, SIVs are independent variables, and HSI is a variable dependent on SIVs. Habitats with a high HSI accommodate more species; on the other hand, habitats with a lower HSI contain fewer species. As the HSI of habitat and the number of species rise, the tendency to emigration from that habitat to search for a habitat with better food sources and fewer population increases. On the other hand, habitats with less population show a greater tendency for immigration. Figure 1 depicts the effects of the number of species on the immigration rate  $\lambda$  and emigration rate  $\mu$ .

Based on Fig. 1, maximum migration to a habitat occurs when there are no species in the habitat. As the number of species rises, the habitat becomes more populous, and fewer species may migrate there; thus,  $\lambda$  declines. The point at which the habitat has a maximum number of cases of  $S_{max}$  will be a point with immigration of 0. Based on the

immigration diagram, if there are no species in the habitat,  $\mu$  will be 0, and as the number of species rises, this rate will increase. The maximum  $\mu$  is E and the maximum  $\lambda$  of I; the balance in the number of species occurs when  $\mu = \lambda$ .  $S_0$  shows the location of this balance. By considering  $n = S_{max}$ , and based on Fig. 1,  $\mu$  and  $\lambda$  can be expressed as follows when there are k species in the habitat:

$$\mu_k = E \frac{k}{n} \tag{1}$$

$$\lambda_k = I \left( 1 - \frac{k}{n} \right) \tag{2}$$

where  $\mu_k$  and  $\lambda_k$  are respectively the emigration and immigration rates of a habitat with k species. In certain conditions,  $\mu_k + \lambda_k = E = I$  can be assumed.

### 2.3 Mathematical formulation

Simon proposed biogeography-based optimization (BBO) algorithm using biogeography theory in nature. Suppose there is a problem and a set of possible solutions. Each solution can be considered as a habitat, where SIVs represent the decision variables. Habitats and SIVs are like chromosomes and genes in GA, respectively. SIVs determine HSI of a habitat. The higher the HSI, the more suitable the habitat will be. HSI is the same as the objective function in other similar algorithms. Suppose that there is a specific graph with  $E = I$  for each solution (habitat) and the number of species (S) that are directly associated with HSI is manifested by HSI value (Fig. 2)

As shown in Fig. 2, S1 and S2 represent solutions with low and high HSIs, respectively. S1 indicates a habitat with few species, while S2 includes a larger number of species.  $\lambda_1$  of S1 is greater than that of S2. Moreover,  $\mu_1$  of S1 is smaller than  $\mu_2$  of S2.

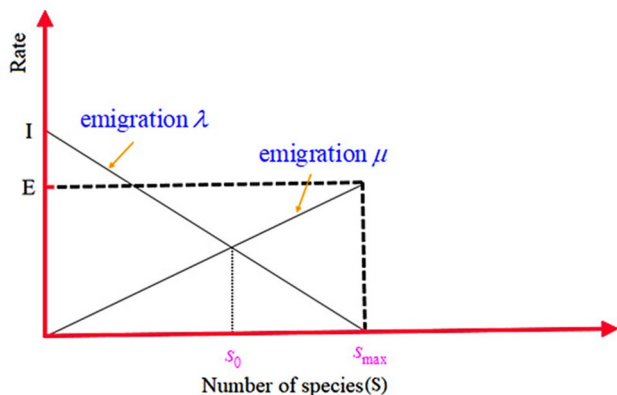


Fig. 1 Emigration and immigration in a habitat

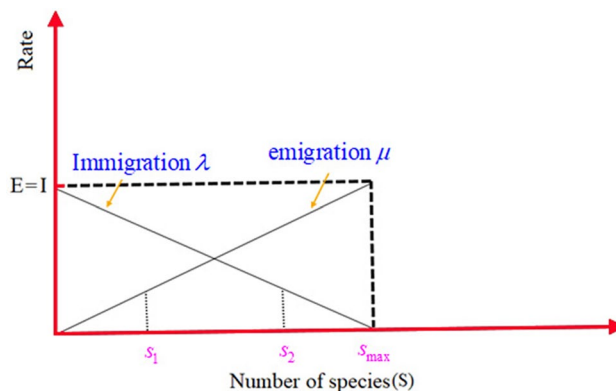


Fig. 2 Comparison of two possible solutions for one problem; S1 has a low HSI, and S2 has a high HSI



With a certain probability like  $P_{mod}$ , each solution can be improved based on another solution. If solution  $S_i$  is selected for improvement, the migration rate  $\lambda$  is used to modify SIVs in the possible decision-making process. After selecting SIVs for modification, migration rate  $\mu$  of other solutions is used to select the improved solution, and SIVs of the selected solution are randomly replaced with those of solution  $S_i$ . It should be noted that all the solutions (in case of lack of elitism) are modified at each stage; however, the degree to which each solution is modified is inversely associated with the suitability of its HSI. The modifying solution is selected based on the probability of migration rate. For this purpose, the roulette wheel can be used. Transferring SIVs from one solution to another in a quite similar way may not be appropriate, as this would make it impossible to fully search the decision space. Thus, the following equation should be used to replace SIVs:

$$SIV_{i,k}^{new} = SIV_{i,k} + \alpha (SIV_{j,k} - SIV_{i,k}) \tag{3}$$

where  $SIV_{i,k}^{new}$  is the  $k$ th modified SIV for the  $i$ th solution,  $SIV_{i,k}$  is the  $k$ th SIV for the  $i$ th solution (modified solution),  $SIV_{j,k}$  is the  $k$ th SIV for the  $j$ th solution (modifying solution) and  $\alpha$  is an analogy between 0 and 1 that is specified by the user. Major hazards such as diseases, natural disasters, etc. can drastically change the HSI of the habitat. Therefore, the condition of a habitat may suddenly become favorable or unfavorable. This phenomenon is similar to the mutation in GA. After migration, this mutation can be randomly applied to solutions (previous step). Mutations can be applied to SIVs after migration based on the potential distribution such as uniform or bell-shaped distribution. Fig. 3 illustrates the flowchart of this algorithm.

### 3 Steps of proposed algorithm

In this section, the routing problem by using the proposed algorithm will be explained. The proposed algorithm has all the specifications of GA, PSO, and BBO algorithms. As mentioned earlier, BBO is a population-based evolutionary method that is inspired by the phenomenon of the migration of animals and birds between different places. In fact, BBO is the study of the geographical distribution of different animal species. In this method, islands that are suitable for habitat have high suitability for habitation, which follow the characteristics of rainfall, plant diversity, the soil of the region, temperature, and so on. To start, a number of random paths must be generated, the starting point of which is the current location of the robot and the end point of which is the final target. These initial solutions are called initial population.

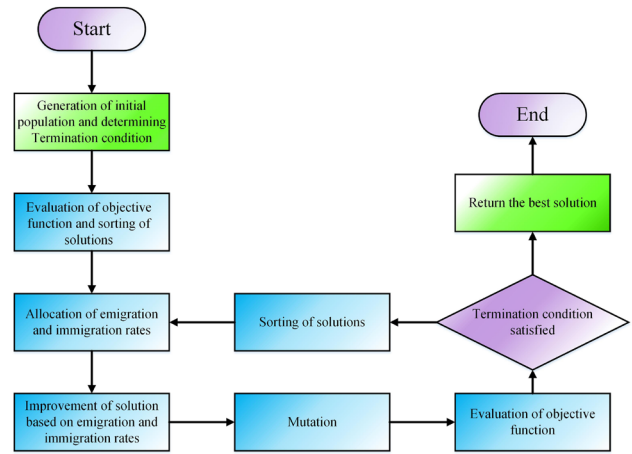


Fig. 3 Flowchart of the BBO algorithm

By applying the rules of the suggested algorithm, the best solution is selected as the final optimal path. The optimal path is better than others in terms of length, a safe distance from obstacles, no collision with other robots and obstacles, and smoothness. The details of this method are explained below step by step (see the general flowchart in Fig. 4).

1. The first step involved initialization. The population is denoted with  $P$ . The size of  $P$  is shown with  $N$ .

2. For each member of the population  $P_h, h = 1, \dots, N$ , the emigration and immigration rates  $\lambda_h$  and  $\mu_h$  are defined as:

$$\lambda_h = \frac{h-1}{N}, \mu_h = 1 - \frac{h-1}{N} \tag{4}$$

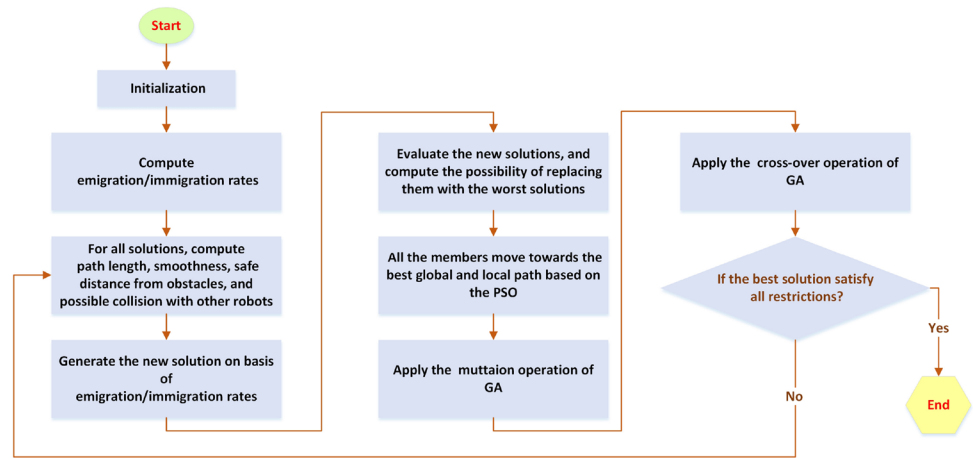
3. For each member of the population  $P_h, h = 1, \dots, N$ , the cost function is calculated. The basic cost function is the length of the path. In multi-objective optimization, other criteria can be added to the cost function. Other important criteria besides distance are path smoothness and safety. The smoothness and safety rates are defined as Mac et al. (2017); Ou et al. (2022):

$$S(P_h) = \sum_{i=2}^{N-1} A(l_{i-1}, l_i) \tag{5}$$

$$R(P_h) = \sum_{i=1}^{N-1} D_i \tag{6}$$

where  $A(l_{i-1}, l_i)$  denotes a change in the robot's direction from line  $l_{i-1}$  to line  $l_i$ . Parameter  $D_i$  is the distance of the  $i$ th line segment from the closest obstacle. The overall cost function can be assumed as a weighted function of the three mentioned factors. In the proposed method, the cost function is a combination of lack of collision and distance. Distance

**Fig. 4** Flowchart of the suggested algorithm



from the obstacle can also be added to the algorithm. The distance of the path from all the obstacles is calculated and added to the cost function as a coefficient. To smooth the path, the idea of interpolation between specified points has been put forward. Particles are given by the algorithm turn into a smoother path by using interpolation.

4. Members of the population  $P_h, h = 1, \dots, N$  are ordered from the worst case to the best case.

5. For each member  $P_h, h = 1, \dots, N$ , a random number between 0 and 1 is generated. If the random number is  $< \lambda_h$ , another member is selected and the migration operation is applied as:

$$P'_h = P_h + \eta(P_{h'} - P_h) \tag{7}$$

where  $\eta$  is a number between 0 and 1. The newly generated member is stored as  $P'_h$ . To select  $P'_h$ , the proposed method is as follows: First,  $\beta$  members are randomly selected and among them, the best case with the highest  $\mu_h$  is regarded as  $P'_h$ .

6. The newly generated member  $P'_h$  is evaluated.

7. The new population  $P_{new}$  is generated by adding  $n_{new}$  members of the newly generated members  $P'_h$ . If the new member is better than the previous one, it replaces it.

8. All the members move towards the best global and local path based on the PSO.

9. For some of the best and worst paths, the cross-over genetic operation is applied.

10.  $m$  members of the worst solutions are selected and replaced with random solutions.

The simple pseudo-code for the suggested algorithm is given Algorithm 1.

---

**Algorithm 1** Simple pseudo code for suggested algorithm.

---

- 1: get number of robots ( $K$ )
  - 2: **for**  $i \leq K$  **do**
  - 3:     Initialize the population, (Population size= $N$ )
  - 4:     **for**  $j \leq N$  **do**
  - 5:         Evaluate  $j$ -th solution,
  - 6:     **for**  $j \leq N$  **do**
  - 7:         Generate the new solution by (7),
  - 8:     Evaluate new solutions
  - 9:     Replace the better new solutions with worst ones
  - 10:    **for**  $j \leq N$  **do**
  - 11:         $j$ -th solution moves towards the best global and local solution based on the PSO
  - 12:    **for** 20% of best solutions **do**
  - 13:        Apply the cross-over operation of GA,
  - 14:    **for** 20% of worst solutions, **do**
  - 15:        Apply the mutation operation of GA,
  - 16:    **if** The best solution satisfy all restrictions **then**
  - 17:        Stop the algorithm,
  - 18:    **else**
  - 19:        Repeat the algorithm,
-

**Table 2** Simulation environment

|             |                             |
|-------------|-----------------------------|
| Matlab      | 2018a                       |
| Processor   | Intel(R) Core(TM) i7-4720HQ |
| CPU         | 2.6 GH                      |
| RAM         | 8 GB                        |
| System type | 64-bit operating system     |

**Table 3** Simulation condition

|                  |           |
|------------------|-----------|
| Population size  | 100       |
| Number of robots | 3         |
| Size of room     | 20 × 20 m |
| $\eta$           | 0.1       |
| $\Delta$         | 0.5       |

## 4 Simulation and results

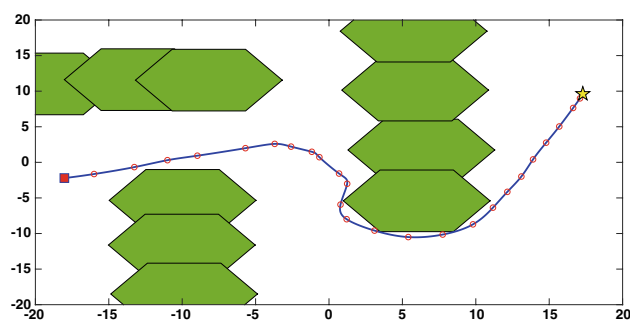
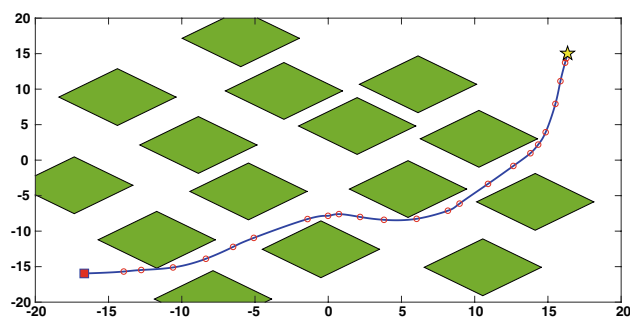
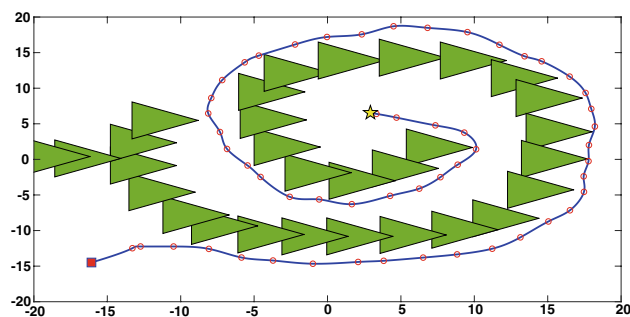
In this section, the proposed method is simulated in MATLAB and the results are examined, see Tables 2–3 for simulation environment and conditions. There is no such big restriction in experimental configurations. The only point that we need to take care of that, is the size of the initial population. With the big initial population, the accuracy is improved, but the computational cost is increased. So, we need to have a trade-off between the accuracy and speed of the algorithm. Please note that there is no data on this problem. The performance of the suggested method is examined under various conditions. Various difficult maps, various obstacles, and a various number of robots are taken to account. First, we deal with the shortest path problem. The results are as follows.

### 4.1 Finding the shortest path

The performance of the proposed algorithm is assessed on different maps with different and rigid obstacles. Here, only distance and lack of collision are included in the cost function. The maps are given in Figs. 5, 6 and 7. Evidently, the proposed algorithm has managed to find proper paths without collision. The proposed algorithm has a very high speed and finds the best path in most maps in less than 10 iterations.

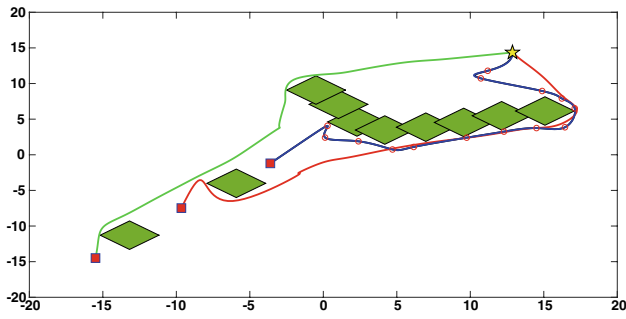
### 4.2 Multi robots

In this section, the problem of multi-robots is investigated. The problem is that, in addition to find the shortest path, the robots should not collide. The speed of all robots is the same. Then the points that the robots may collide,

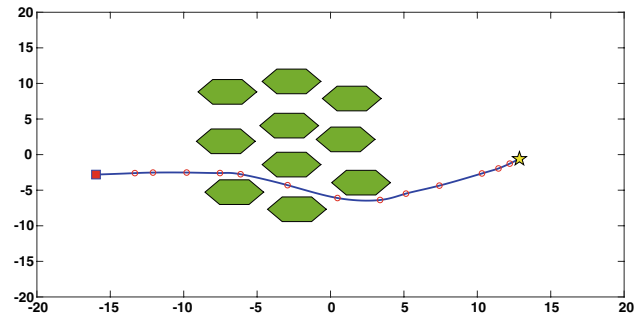
**Fig. 5** Map 1: diagram of the path found by the proposed algorithm**Fig. 6** Map 2: diagram of the path found by the proposed algorithm**Fig. 7** Map 3: diagram of the path found by the proposed algorithm

are considered obstacles. In this problem, it is assumed that the map of robots and obstacles is known, and all the robots have the same motion velocity; thus, their paths are determined one by one. First, the path is found for the first robot without collision with obstacles and other robots. Then, the proper path for the second robot is found similarly. If these two paths meet, the distance of the point of confluence from each robot is determined. Based on the robots' constant velocity, a collision occurs if this distance is the same for both robots. If this distance is equal, the confluence point is regarded as an obstacle for the second robot, and a new routing algorithm and path are found for the second robot. This algorithm continues up to a certain iteration. The proposed algorithm effectively found the

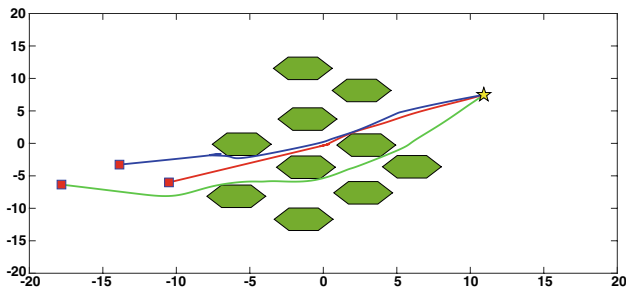




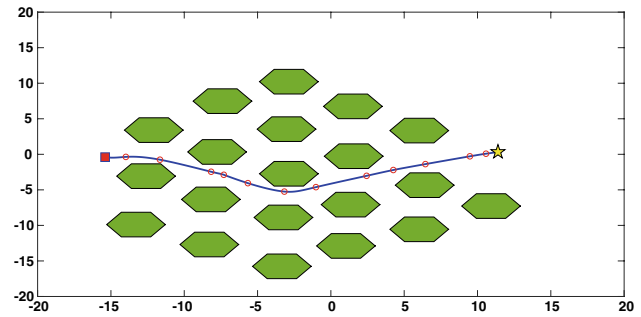
**Fig. 8** Map 1: multi robots; The rectangles are robots and the star is the target



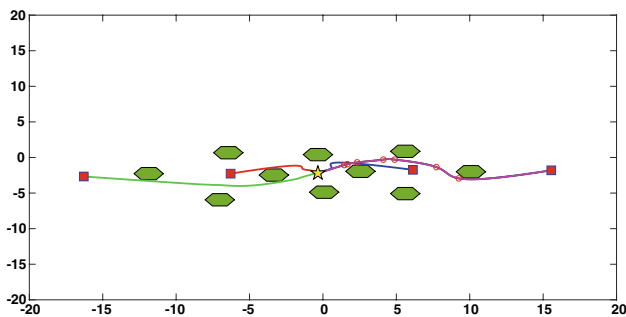
**Fig. 11** Map 1: safe distance from the obstacles



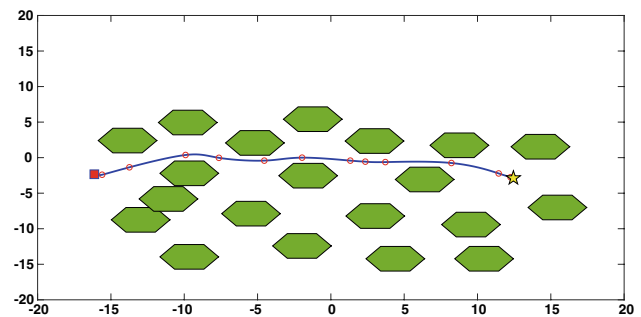
**Fig. 9** Map 2: multi robots; The rectangles are robots and the star is the target



**Fig. 12** Map 2: safe distance from the obstacles



**Fig. 10** Map 3: multi robots; The rectangles are robots and the star is the target



**Fig. 13** Map 3: safe distance from the obstacles

optimal path without collision, with the path being the best.

The results for the path of multi robots are depicted in Figs. 8, 9 and 10. It is seen that the robots successfully find the best paths with no colliding with obstacles.

### 4.3 Safe distance from the obstacles

To add the safe distance from the obstacles the cost function  $v$  is revised as follows:

$$v = \max(1 - d/(r_{obs}(k) + \Delta), 0) \tag{8}$$

where,

$$d = \sqrt{((x - x_{obs}(k))^2 + (y - y_{obs}(k))^2)} \tag{9}$$

where  $\Delta$  represents the safe distance from the obstacle.  $x_{obs}$  and  $y_{obs}$  denote the x-y coordinates of obstacles.  $r_{obs}(k)$  is the radius of the k-th obstacle. The results are given in Figs. 11, 12 and 13. It is seen that the robot successfully finds a path of safe distance from obstacles.



Fig. 14 Designed setup

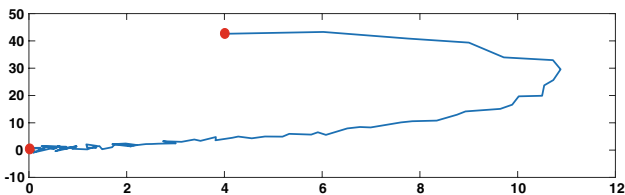


Fig. 15 Experimental results

### 4.4 Experimental verification

To further verify the feasibility of the suggested approach, an experimental setup is designed as shown in Fig. 14. The robot is connected to the computer by wireless communication. The commands for the robot are sent from Matlab, and the position of the robot at each sample time is sent for Matlab for monitoring. The speed of the robot is constant. The coordinates of the next point are determined by the suggested algorithm. The robot is supposed to reach the destination based on the suggested algorithm. The performance is given in Fig. 15. It is seen that the robot well reached the destination in the presence of obstacles. This examination demonstrates the feasibility of the suggested algorithm in practice.

## 5 Comparison of results

To better demonstrate the abilities of the proposed algorithm, the value of the cost function for similar algorithms after 50 iterations are compared with the proposed algorithm. Several complex maps are used for comparison. The simulation condition is assumed to be the same for all the algorithms. The initial population is similar for all the algorithms and the number of iterations is also the same. The results are given in Table 4. All methods are repeated 10 times, and the mean results are provided. Evidently, the proposed algorithm yielded better results.

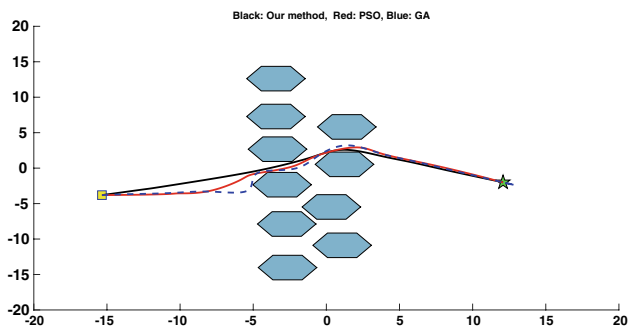
The reasons for its superiority are summarized below:

- The proposed algorithm is optimized based on the BBO idea and the features of GA and PSO are added to it.
- Besides cross-over and mutation, a percentage of less optimal solutions are also moved towards the more optimal solution by the PSO algorithm.
- A small percentage of less optimal solutions are replaced with random solutions.
- The cost function is multi-objective, and in addition to path shortness, can support a safe distance from the obstacles and path smoothness. This ability was demonstrated in the above simulations.

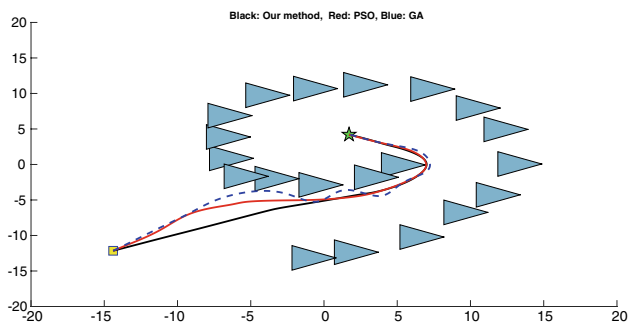
Several diagrams are provided below to graphically depict the superiority of the proposed algorithm. To this end, in an equal map and the same iteration, the paths found by different algorithms are compared. Results of the comparison are given in Figs. 16, 17 and 18. At the same iteration, the path found by the proposed algorithm is better than the rest of the algorithms. Note that this comparison was provided only to show the good speed of the proposed algorithm. Naturally, if the number of iterations increases, other algorithms will also approach the more optimal path.

Table 4 Comparison of cost-function for different algorithms

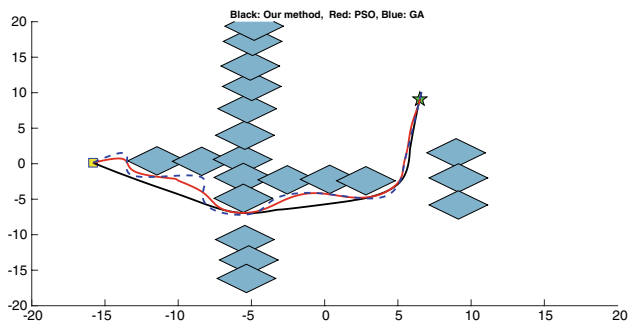
| Population size | Number of map | GA       | PSO     | BBO      | Proposed algorithm |
|-----------------|---------------|----------|---------|----------|--------------------|
| 100             | 1             | 77.3015  | 75.6478 | 76.1214  | 71.0145            |
|                 | 2             | 46.0154  | 44.5479 | 43.6217  | 35.0147            |
|                 | 3             | 56.0140  | 55.4527 | 56.0124  | 53.4572            |
| 50              | 1             | 85.0147  | 83.2170 | 82.5321  | 75.5470            |
|                 | 2             | 49.3174  | 53.0214 | 440.2171 | 38.757             |
|                 | 3             | 580.6572 | 59.5487 | 590.2176 | 54.0137            |



**Fig. 16** Map 1: comparison of the path found by the proposed algorithm (black) with GA (blue) and basic PSO (red)



**Fig. 17** Map 2: comparison of the path found by the proposed algorithm (black) with GA (blue) and basic PSO (red)



**Fig. 18** Map 3: comparison of the path found by the proposed algorithm (black) with GA (blue) and basic PSO (red)

**Remark 1** To make the algorithm to be more robust, we propose to generate a solution with other classical methods and put it inside the initial population. This cause the initial solutions to be improved fast. Also, we propose, to keep the best solution at each iteration. The operations such as mutation should not be applied to the best solution.

**Remark 2** The main limitation of the suggested algorithm is that the speed of various robots is considered to be the

same. So, for future studies, the algorithm can be developed for multi-robot problems with different speeds. For practical applications, a new scheme can be added to the algorithm to predict the speed of various robots and obstacles.

**Remark 3** The computational cost of the suggested scheme is less than PSO and GA. Because the suggested algorithm benefits the best features and operations of both PSO and GA. So, the convergence speed of the proposed algorithm has been improved. To verify this property, in the Table 4, it is seen that by decreasing the population size, the suggested scheme's accuracy does not decrease significantly.

**Remark 4** Since in the suggested algorithm, the position of obstacles is known (measured by the sensors at each sample time); So, in a dynamic problem, if the positions are changed, it is obvious that the performance will not change. To verify this, the suggested algorithm is applied to a multi-robot problem. When multi-robots are moved to the destination, the position of one robot at each sample time is considered an obstacle for other robots. The results (Figs. 8, 9, 10) show that the suggested algorithm results in desired efficiency and there are no collision between robots and obstacles.

**Remark 5** The obstacles in maps are determined, randomly. We want to show that the suggested algorithm can handle various maps with complex obstacles.

## 6 Technical aspects

Recently, the remote sensing technologies such as laser scanners provide high accuracy and resolution in distance measuring. So, it is essential to improve the monitoring systems by the use of novel technologies. This suggested system reduces the damage caused by collisions, increase safety, as well as reduce manpower and operator costs, due to obstacle detection and navigation without using the operator. The efficiency is also increased due to the online detection of obstacles and the continuous operation.

Installation of some sensors, converters, and intelligent hardware and data integration by improved algorithms for navigation, guidance, and control is one of the principles of the suggested approach. The installed sensors to detect obstacles during navigation operations commonly include: radar, laser scanners (such as UXM-30LXH-EWA produced by Hokuyo company), binoculars, monoculars, infrared cameras, and ultrasonic sensors, or a combination of some of these sensors. In this context, the detection distance is usually limited to a few hundred meters. Some other expensive high accuracy positioning systems such

as Real-Time Kinematic and Differential Global Positioning Systems (RTK-DGPSs) can also be used. The central control system receives all the data and information, and after analyzing the data, activates the left and right engines in appropriate ways.

## 7 Conclusion

The multi-robot routing system in a complex environment is the core and key component of the development of intelligent navigation systems. It has many applications. For example, it is widely used in civil intelligent ships, unmanned ships, and unmanned merchant ships. Its core technology involves multiple constraints and complex problems. However, in most navigation systems, the data acquisition, and processing are done manually and are always prone to human error. In most conventional methods, only the collision-free problem is investigated, and the other key restrictions such as multi-robots problem, safe distance, high accuracy, and the smoothness of the path, are neglected.

In this paper, a new path-planning scheme was designed. By using different maps in the presence of different obstacles, it was shown that the proposed algorithm is highly effective and yields very good results. In the proposed algorithm, besides the shortest path, the path smoothness and safe distance from the obstacles are also included, demonstrating the success of the proposed algorithm. Finally, its performance was compared with similar algorithms, demonstrating its better precision and speed.

The following directions are suggested for future research:

1. Practical implementation of the proposed algorithm.
2. Development of this algorithm by considering moving obstacles.
3. Development of this algorithm by considering robot dynamics and technical issues.

**Data availability** This paper report no data.

## Declarations

**Conflict of interest** The authors declare no conflict of interest.

## References

Abualigah L, Yousri D, Abd Elaziz M, Ewees AA, Al-Qaness MA, Gandomi AH (2021) Aquila optimizer: a novel meta-heuristic optimization algorithm. *Comput Ind Eng* 157:107250

- Abualigah L, Diabat A, Altalhi M, Elaziz MA (2022) Improved gradual change-based harris hawks optimization for real-world engineering design problems. *Eng Comput* 20:1–41
- Bae J, Park S, Yim M, Seo T (2021) Polygon-based random tree search algorithm for a size-changing robot. *IEEE Robot Autom Lett* 20:20
- Barma PS, Dutta J, Mukherjee A, Kar S (2021) A hybrid ga-bfo algorithm for the profit-maximizing capacitated vehicle routing problem under uncertain paradigm. *J Intell Fuzzy Syst* 20:1–17
- Buniamin N, Ngah WW, Sariff N, Mohamad Z et al (2011) A simple local path planning algorithm for autonomous mobile robots. *Int J Syst Appl Eng Dev* 5:151–159
- Che G, Liu L, Yu Z (2020) An improved ant colony optimization algorithm based on particle swarm optimization algorithm for path planning of autonomous underwater vehicle. *J Ambient Intell Humaniz Comput* 11:3349–3354
- Chen D (2020) Fuzzy obstacle avoidance optimization of soccer robot based on an improved genetic algorithm. *J Ambient Intell Humaniz Comput* 11:6187–6198
- Chen X, Gao P (2020) Path planning and control of soccer robot based on genetic algorithm. *J Ambient Intell Humaniz Comput* 11:6177–6186
- Chen B, Zhang L, Chen H, Liang K, Chen X (2021) A novel extended kalman filter with support vector machine based method for the automatic diagnosis and segmentation of brain tumors. *Comput Methods Programs Biomed* 200:105797
- Chen D, Wang M, Li W, Li X, Hou H, Shan H (2022) Computed tomography image: guided needle biopsy in the diagnosis of lung malignant tumors under artificial intelligence algorithm. *Sci Program* 20:22
- Cherubini A, Chaumette F (2011) Visual navigation with obstacle avoidance. In: 2011 IEEE/RSJ international conference on intelligent robots and systems, IEEE, pp 1593–1598
- Ding W, Abdel-Basset M, Eldrandaly KA, Abdel-Fatah L, de Albuquerque VHC (2020) Smart supervision of cardiomyopathy based on fuzzy harris hawks optimizer and wearable sensing data optimization: a new model. *IEEE Trans Cybern* 51:4944–4958
- Gao P, Zhao D, Chen X (2020) Multi-dimensional data modelling of video image action recognition and motion capture in deep learning framework. *IET Image Proc* 14:1257–1264
- Goel L (2022) A novel approach for face recognition using biogeography based optimization with extinction and evolution. *Multimed Tools Appl* 20:1–28
- Gul F, Rahiman W, Nazli Alhady SS (2019) A comprehensive study for robot navigation techniques. *Cogent Eng* 6:1632046
- Gul F, Rahiman W, Alhady S, Ali A, Mir I, Jalil A (2021) Meta-heuristic approach for solving multi-objective path planning for autonomous guided robot using PSO-GWO optimization algorithm with evolutionary programming. *J Ambient Intell Humaniz Comput* 12:7873–7890
- Hadidi A, Nazari A (2013) Design and economic optimization of shell-and-tube heat exchangers using biogeography-based (bbo) algorithm. *Appl Therm Eng* 51:1263–1272
- Jamuna K, Swarup K (2011) Biogeography based optimization for optimal meter placement for security constrained state estimation. *Swarm Evol Comput* 1:89–96
- Jazebi SJ, Ghaffari A (2020) Risa: routing scheme for internet of things using shuffled frog leaping optimization algorithm. *J Ambient Intell Humaniz Comput* 11:4273–4283
- Lee H, Jeong J (2021) Mobile robot path optimization technique based on reinforcement learning algorithm in warehouse environment. *Appl Sci* 11:1209
- Lin HY, Yao CW, Cheng KS, Tran VL (2018) Topological map construction and scene recognition for vehicle localization. *Auton Robot* 42:65–81

- Liu S, Liu P (2022) Benchmarking and optimization of robot motion planning with motion planning pipeline. *Int J Adv Manuf Technol* 118:949–961
- Mac TT, Copot C, Tran DT, De Keyser R (2017) A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization. *Appl Soft Comput* 59:68–76
- Mirshojaei SH, Masoumi B, Zeinali E (2020) Mamhoa: a multi-agent meta-heuristic optimization algorithm with an approach for document summarization issues. *J Ambient Intell Humaniz Comput* 11:4967–4982
- Mohammadnezhad M, Ghaffari A (2019) Hybrid routing scheme using imperialist competitive algorithm and rbf neural networks for vansets. *Wirel Netw* 25:2831–2849
- Mottaghinia Z, Ghaffari A (2018) Fuzzy logic based distance and energy-aware routing protocol in delay-tolerant mobile sensor networks. *Wirel Pers Commun* 100:957–976
- Mukherjee A, Barma PS, Dutta J, Panigrahi G, Kar S, Maiti M (2020) A modified discrete antlion optimizer for the ring star problem with secondary sub-depots. *Neural Comput Appl* 32:8143–8156
- Mukherjee A, Barma PS, Dutta J, Panigrahi G, Kar S, Maiti M (2021) A multi-objective antlion optimizer for the ring tree problem with secondary sub-depots. *Oper Res* 20:1–39
- Ou J, Hong SH, Ziehl P, Wang Y (2022) GPU-based Global Path Planning Using Genetic Algorithm with Near Corner Initialization. *J Intell Robot Syst* 104:1–17
- Peng C, Qiu S (2022) A decomposition-based constrained multi-objective evolutionary algorithm with a local infeasibility utilization mechanism for uav path planning. *Appl Soft Comput* 20:108495
- Petrović L, Marković I, Petrović I (2022) Mixtures of gaussian processes for robot motion planning using stochastic trajectory optimization. *IEEE Trans Syst Man Cybern Syst* 20:20
- Sandakalum T, Ang MH Jr (2022) Motion planning for mobile manipulators—a systematic review. *Machines* 10:97
- Sarma A, Bhutani A, Goel L (2017) Hybridization of moth flame optimization and gravitational search algorithm and its application to detection of food quality. In: 2017 intelligent systems conference (IntelliSys), IEEE, pp 52–60
- Simon D, Rarick R, Ergezer M, Du D (2011) Analytical and numerical comparisons of biogeography-based optimization and genetic algorithms. *Inf Sci* 181:1224–1248
- Song H, Li A, Wang T, Wang M (2021) Multimodal deep reinforcement learning with auxiliary task for obstacle avoidance of indoor mobile robot. *Sensors* 21:1363
- Taghizadeh J, Ghobaei-Arani M, Shahidinejad A (2021) An efficient data replica placement mechanism using biogeography-based optimization technique in the fog computing environment. *J Ambient Intell Humaniz Comput* 20:1–21
- Tamizi S, Ghaffari A (2019) Inference search algorithm for optimizing scheduling and minimizing mean tardiness in parallel joint robots. *J Ambient Intell Humaniz Comput* 10:4771–4783
- Yoo H, Oh S (2021) Localizability-based topological local object occupancy map for homing navigation. In: 2021 18th international conference on ubiquitous robots (UR), IEEE, pp 22–25
- Yu D, Hu X, Liang K, Ying J (2021) A parallel algorithm for multi-agv systems. *J Ambient Intell Human Comput* 20:1–15
- Zhang Z, Li J, Liu Y, Xiao Q (2022) Design launch vehicle vertical landing guidance law using a gauss point discrete convex programming. *Advances in guidance, navigation and control*. Springer, Berlin, pp 3571–3584

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.