International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
***Article Received:** 30 April 2023 **Revised:** 30 June 2023 **Accepted:** 20 July 2023*

Hybrid Swarm Algorithm for Mobile Robot Path Planning

Walaa Mahmood Hassen, Saman H. Amin, Ahmed Sabah Al-Araji

Computer Engineering Department, University of Technology, Iraq, Baghdad, Iraq.

Corresponding author: ce.21.09@grad.uotechnology.edu.iq

Abstract

The adoption of lightweight and effective swarm algorithms is required for low resource usage algorithms for mobile robot path planning crises. We present a hybrid swarm approach in this study that combines the best features of particle swarm optimization and river formation dynamics. This method looks for the shortest route while keeping the path as smooth as feasible. The best qualities of both approaches are combined and leveraged by the hybrid RFD-PSO methodology. While the RFD algorithm is well known for its smooth path discovery, it needs a lot of drops for good convergence and suffers from sinuosity problems. The generated hybrid RFD-PSO algorithm synergistically balances PSO's fast convergence with the river method's adaptive exploration and exploitation. Comparing the simulation results of the proposed method versus the Ant Colony Optimization (ACO), modified Ant Colony Optimization ACO*, PSO, RFD, A*, and Dijkstra's, Hybrid RFD-PSO have better results in creating optimal path.

Keywords: Mobile Robot, Path Planning, River Formation Dynamic, Static environment, Particle Swarm Optimization.

1. Introduction

Over the last ten years, mobile robots have been successfully modified to carry out a variety of activities in several industries. Machine learning technologies provide mobile robots additional intelligence so they can boost productivity with their navigational capabilities. [1]. It is applied in a variety of field, such as industrial, military, computer games, family services, education, mining, and security. in 2020, outbreak the Corona virus (COVID-19) virus outbreak, everyday life in the world has changed. The current epidemic is compelling us to reconsider the safety procedures of routine activities by prioritizing public health. In a number of areas, including production, logistics, and healthcare, robotics can be utilized to make working conditions safer and help combat this pervasive sickness. [2]. For the mobile robot to travel across a predetermined area without colliding with anything, Path planning is the process of determining the optimal path while avoiding static and moving obstacles between the source (starting point) and destination (target point) while considering several factors, including avoiding obstacles, finding the shortest possible path, and arriving at the destination in the least amount of time, using the least amount of energy, and achieving smoothness of the path [3]. The adoption of lightweight and effective swarm algorithms is required for low resource usage algorithms for mobile robot path planning crises. We present a hybrid swarm approach in this study that combines the best features of particle swarm optimization and river formation dynamics. This method looks for the shortest route while keeping the path as smooth as feasible. The best qualities of both approaches are combined and leveraged by the hybrid RFD-PSO methodology [4].

Many writers investigate a variety of solutions to the problem of path planning for mobile robots. There are, nevertheless, certain substantial hiatus and constraints that need to be processed. This work addresses two problems related to path planning for robots using a hybrid technique. Finding the quickest and smoothest route from the starting place to the objective point is the first obstacle, and avoiding collisions is the second. Using a suggested intelligent hybrid methodology, the River Formation Dynamic-Particle Swarm Optimization (RFD-PSO) method combines the RFD algorithm with particle swarm optimization to create the mobile robot's shortest and smoothest path.

This paper is structured as follows: The nonholonomic wheeled mobile robot system is depicted in Section (2). Section (3) focuses on the various types of route planning approaches, Section (4) shows the hybrid techniques, and Section (5) shows the simulated results. Finally, Section (6) represents the conclusion of the work.

2. The Kinematics Model of Mobile Robot Platform

An omnidirectional caster is placed at the front of the cart to support the mechanical structure and steady the platform, and two driving wheels oriented on a common axis make up the nonholonomic mobile robot seen in Figure 1 [5]. Two separate servo DC motors—one for the left wheel and one for the right—control the robot's motion and orientation. For both wheels, the distance, represented by the letters r and L, is the same. The mobile robot's center of mass is located at point c, which is also the wheel axle's center.

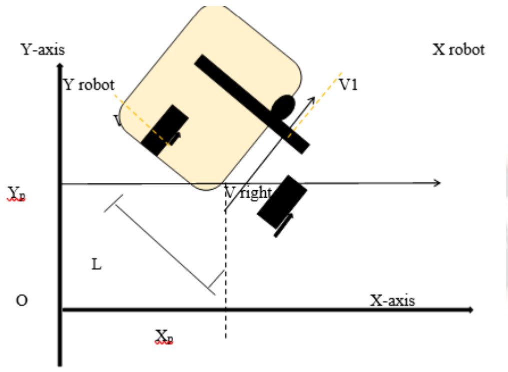


Figure 1. Platform of nonholonomic mobile robot [6].

The stance of the mobile robot in the surface's pose vector and in the global coordinates O, X-axis, and Y-axis is defined as in equation (1)[6]:

$$Q = [x, y, \theta]^T \tag{1}$$

Xp and Yp are the coordinates of point C, while θ is the robot's orientation angle with respect to the X-axis. The configuration of the mobile robot is characterized by these three generalized coordinates. Two prerequisites need to be met in order to analyze the motion and orientation of a wheeled mobile robot: the first is the condition of pure rolling, and the second is the condition of non-slipping, which means that the mobile robot's lateral motion must remain constant at zero, as stated in the following equation. (2)[6]:

$$-\dot{x}(k)\sin\theta(k) + \dot{y}(t)\cos\theta(k) = 0 \tag{2}$$

thus, the following are the computer simulation equations:

$$Xp(k) = [1/2(V\ left + V\ right) \times \cos(\theta(k)) \times Ts] + Xp\,(k-1) \tag{3}$$

$$Yp(k) = [1/2(V\ left + V\ right) \times \sin(\theta(k)) \times Ts] + Yp\,(k-1) \tag{4}$$

$$\theta(k) = [\,(V\ left - V\ right)/L \times Ts] + \theta\,(k-1) \tag{5}$$

Where V right is the platform's right wheel velocity. The platform's left wheel velocity is indicated by the symbol V left. L is the distance measured between the platform's driving wheels. The sampling time of the numerical computation is represented by the symbol Ts [7].

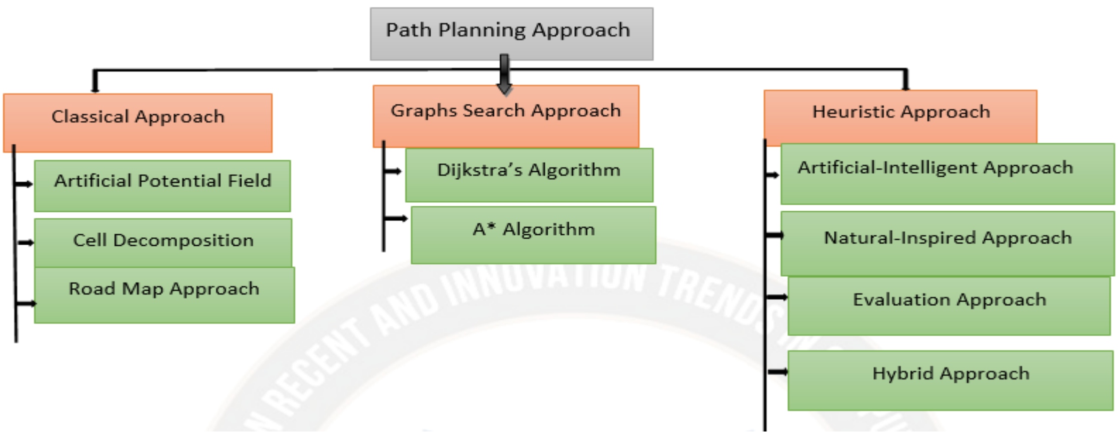


Figure. 2 Categorization of the path planning algorithm [12]

3. Types of Path Planning Approaches

In most cases, path planning aims to produce a direct path towards a target while taking kinematic and/or dynamic restrictions into account, avoiding hazards or collisions, and optimizing a specified objective function. Different planning objectives were met by a variety of technologies, including risk minimization, real-time planning, collision avoidance, and performance optimization. Several algorithms for robotic systems have been developed [8] Including:

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
***Article Received:** 30 April 2023 **Revised:** 30 June 2023 **Accepted:** 20 July 2023*

3.1 Classical Approaches

Initially, classical methods were implemented to solve robot navigation challenges since methods for artificial intelligence had not yet been discovered. When employing traditional methods to complete a task, it is noticed that either a result is acquired, or it is proven that a result does not exist [9]. Methods such as Cell Decomposition, Road Map, Artificial Potential Field are commonly utilized in motion mapping problems within the classical approach.

_Cell Decomposition:

This approach divides the mobile robot's workspace into several simple sections, each of which is referred to as a cell. The path from the beginning grid to the target grid is looked for in this connected network of grids. Typically, the path is represented by the cell's ordinal number and based on any obstacles in the environment [10].

_Artificial Potential Field:

Khatib [11] created the first artificial potential field for mobile robot navigation. According to this approach, the environment contains both attracting and repulsive forces. The target will provide an attractive force that drives the mobile robot to proceed in its direction. while the impediments produce a repelling force that comes from them. These fictitious forces pull the robot in the desired direction while avoiding any impediments. By following the negative gradients, the robot will arrive at the desired spot. The technique is frequently used in mobile robot navigation, including real-time robot path planning. However, one drawback is the potential for being stuck in local minima.

_ Road Map Approach:

It is a way of moving between two points and is illustrated by a collection of curves that have one dimension and that join the open spaces [12]. The primary drawback of the classical approach is its increased computational cost and inability to respond to environmental unpredictability; therefore, it is less suitable for real-time execution [9].

3.2 Graph-Search Approaches:

Graph search techniques have been widely applied in the past to energy-efficient path planning. These methods primarily determine a route from the starting point to the end points by assessing particular nodes or states.One disadvantage of this approach is that it will fail if no path exists [13]. These methods have many types, including:

_ Dijkstra algorithm: This systematic search technique was introduced by Dijkstra in 1959 to determine, based on traversing costs, the best path from the starting location to all subsequent locations. This approach's inability to determine the best distance for each node to reach the goal in the absence of any prior knowledge about the graph is a limitation. It also uses a blind search, which wastes resources and results in lengthy processing times.[13].

_ The A-star(A*) algorithm: Hart et al. first presented this strategy in 1968, and A* might be seen of as an advancement of the Dijkstra algorithm. The A* approach's primary feature is that it evaluates the found solution using a fitness function. Based on the fitness rating, A* starts at a specific point and chooses the next one. Although the A* technique is well-known for its simplicity, flexibility, and optimization in path planning, it also has several disadvantages, such as redundant points, a large overhead memory need, and a long computation time. [14].

3.3 Heuristic Approaches:

Due to developments in the field of artificial intelligence, mobile robots may now function more efficiently in dynamic situations with barriers that are both stationary and moving. These modern approaches are currently more common than traditional ones because they can manage decision-making and environmental uncertainty [9]. Several tactics in this field are being examined at the moment, such as:

_ Artificial intelligence techniques have been extensively used in a variety of research and development fields. One such technique is the Fuzzy Logic System, which Zadeh first presented in 1965 [15]. A fuzzy logic-based navigation system for omnidirectional mobile robots has been described by Zavlangas et al. [16]. Furthermore, Janglova [17] shown how to use an artificial neural network (ANN) to guide a wheeled mobile robot through a mostly uncharted area.

_ Nature-Inspired Computation-Based Approaches: these approaches include Particles Swarm Optimization (PSO) algorithm proposed by Kennedy and Eberhart in 1995, It is a swarm intelligent inspired by the social behavior of animals such as birds and fish [18]. The authors propose several algorithms PSO algorithm to improve particle properties and increase their efficiency. Another algorithm Ant Colony Optimization (ACO) was provided in [19]. The ACO method, noted for its resilience, global optimization, and parallelism, is commonly used in path planning due to its ability to incorporate with numerous heuristic algorithms for improved efficiency.

_Hybrid Approaches: hybrid approach. In this case, these algorithms combine a few path planning strategies to offer safe and useful navigation on a local and worldwide level.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
***Article Received:** 30 April 2023 **Revised:** 30 June 2023 **Accepted:** 20 July 2023*

These algorithms often compensate for each other's faults, thus even if they are occasionally more complex, the outcome is usually better than when the combined procedures are carried out independently [20].

4. Algorithm for hybrid path planning

This paper aims to tackle two important path-planning issues: Avoiding colliding with obstacles and finding the optimal path for a robot in a static environment. A suggested River Formation Dynamic algorithm and the Particle Swarm Optimization technique were combined to create a hybrid approach that addresses these two issues.

4.1 River Formation Dynamic Algorithm (RFD):

The *River Formation Dynamic Algorithm* is a heuristic optimization algorithm based on a swarm population method. RFD suggested for solving optimization problems in 2008 by Rabanal et al [21]. RFD is a gradient version of ant colony optimization (ACO). The RFD algorithm's fundamental idea is to replicate how riverbeds emerge. Gravitational forces cause a collection of dropped objects to be drawn toward the earth's center from their initial position. As a result, these droplets are dispersed across their surroundings in search of the sea, which is the lowest point. This results in the formation of riverbeds with many meanders. In graph theory issues, the RFD makes use of this concept. The optimal answer is found by examining an environment with a group of agents-drops that are formed and travel on edges connecting nodes. This is achieved through soil erosion and sedimentation processes that are related to shifts in each node's allocated height. Drops alter node elevations along their path as they move across an environment [22].

The RFD algorithm works like this: when nodes move along, they either deposit or remove sediment from their routes, causing each node to receive a certain quantity of dirt in turn, increasing the heights of the nodes as they go. The declining gradient, which is directly proportional to the height difference between the droplet's current node and its neighboring node, determines the selection probability of the subsequent node. Except for the target node, which maintains a height of zero during the whole process, the constructed environment is initially flat, with nodes' elevations being equal. Drops are added to the original node to facilitate additional site exploration and the discovery of the ideal route [23]. A group of drops moves around the area in a set order at each step, visiting nodes in turn and then performing erosion on them. According to the following pseudo-code, the modified RFD algorithm [23] scheme is presented:

InitializeMaps ()

InitializeDrops ()

while (conditionMet does not end ())

    moveDrops ()

    analyzePaths ()

    erodePaths ()

    depositSediments ()

end while

The initialization of the algorithm's maps, where each node additionally has information about whether it has an obstacle and extra data, such as the amount of time needed to go across and how far away it is from the target, is the first stage. The first node in the algorithm receives the necessary number of drops to start. Until the designated termination condition—which indicates that every drop is traveling in the same direction—is met, the procedure keeps going. An upper limit on iterations is also included, along with a condition that checks to see whether the answer hasn't improved over the last n loops, both of which help shorten calculation times. Droplets advance one by one until they reach the target or are unable to proceed [23]. In such instances, they evaporate, and the cycle begins over. The possibility that the drop k in the node i chooses the subsequent node j is:

$$Pk(i,j) = \left\{ \begin{cases} \frac{gradient(i,j)}{total} & \text{for } j \in Vk(i) \\ \frac{\omega/|gradient(i,j)|}{total} & \text{for } j \in Uk(i) \\ \frac{\delta}{total} & \text{for } j \in Fk(i) \end{cases} \right\} \quad (6)$$

$$gradient(i,j) = \frac{altitude(i) - altitude(j)}{distance(i,j)} \quad (7)$$

$$total = \left(\sum_{j\in v_k(i)} ient\ t(i,j)\right) + \left(\sum_{j\in v_k(i)} \frac{\omega}{|ient(i\cdot j)|}\right) + \left(\sum_{j\in f_k(i)} \delta\right) \quad (8)$$

Article Received: *30 April 2023* ***Revised:*** *30 June 2023* ***Accepted:*** *20 July 2023*

$F_k(i)$ is the name of a group of neighbors with a flat gradient; $U_k(i)$ is the name of a group of neighbors with a negative gradient (Node j is higher than Node i); and $V_k(i)$ is the name of a group of neighbors with a positive gradient (Node i is higher than Node j). ω and δ, the two coefficients, are both constant, tiny values.

Once every drop has been tracked, an erosion process is applied to the paths that have been followed, whereby node heights are decreased with each subsequent node in proportion to their gradients. The relationship between the erosion amount for each node pair (I and j) and variables like the total number of drops used (D), the total number of nodes in the network (N), and an assigned erosion coefficient (E) is established by equation (9). A drop deposits some of the carrying sediment and evaporates for the duration of the algorithm iteration if it is unable to select a new node for transition. This lessens the possibility of turning down a dead end, weakening harmful pathways [22].

$$erosion\ (i, j) = \frac{E}{(N-1) \cdot D} \cdot gradient\ (i, j) \tag{9}$$

The sediment is introduced to all nodes after each iteration in a specific, tiny quantity in order to prevent a situation in which all heights are close to zero, which would render slopes insignificant, and wreck all created pathways [24]. The equation calculates the required amount of sediment (10).

$$\text{sediment} = \frac{erosionProduced}{N-1} \tag{10}$$

4.2 Particle Swarm Optimization algorithm

Particle Swarm Optimization (PSO) is a community of experimentation that makes use of several research technology points to mimic a flock of birds' communal behavior [24]. Team leadership is not necessary for the PSO to achieve its goals.; rather, it aims to mimic the behavior of a social animal. The flock of birds just follows one of the individuals who are closest to the food when it is time to go in search of nourishment. Through this kind of efficient communication with the other elements, the group of birds may accomplish its desired goal. When using PSO, each swarm particle serves as a potential solution to an optimization issue that specifies the dimensions of its speed and position vectors. Investigating the optimization problem's solution space can lead to the finding of an optimum solution.

Table 1 parameters, each of which has a PSO definition.

Parameters	Definitions
$Vi\ (j)$	i^{th} the iteration's particle velocity j
$Xi\ (j)$	i^{th} vector positions in iteration j
$P_{iBest}\ (j)$	The best possible state of fitness for the i^{th} particle
$G_{i\ Best}\ (j)$	The whole swarm's optimal global fitness value
W	Weight of the velocity's inertia 0.5
c_1 , c_2	The individual and social cognitive (0.75, 0.75)
r_1 , r_2	a uniform distribution of random numbers between 0 and 1.

1. Determine number of iterations that can be performed.

2. initialize all the particles.

3. Verify the fitness score for each particle; if it is greater than the best fitness value (P_{ibest}), adjust the current value to (new P_{ibest}).

4. For each particle, locate the particle in the particle neighborhood that has the greatest global fitness (G_{best}).

- Using Equation (4), compute the particle velocity$V_i(j)$.
- Use the updated velocity value.
- Using Equation (5), compute the new particle location $X_i(j)$.
- Use the updated position value.

5. Continue 3 until the greater number of repetitions is reached.

Figure 3. PSO algorithm's pseudo-program [12]

The pseudo-program for the PSO method is shown in Figure 3. The update functions of the locations and velocities vectors at the j^{th} iteration are expressed as follows [25]:

$$V_i\ (j+1) = wV_i\ (j) + c_1 r_1\ (P_i\ be(j) - X_i\ (j)) + c_2\ r_2\ (Gbest(j) - X_i\ (j)) \tag{11}$$

$$X_i\ (j+1) = X_i\ (j) + V_i\ (j+1) \tag{12}$$

c_1, and c_2 must be chosen for the requirements in Equation 13 [26].

$$(c_1 + c_2) < 4 \tag{13}$$

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
Article Received: *30 April 2023* ***Revised:*** *30 June 2023* ***Accepted:*** *20 July 2023*

As though the particle swarm optimization method can successfully resolve the path-planning issue and generate a smoothest path, it can easily slip into local optimal in many optimization scenarios. Additionally, there isn't a general definition of convergence that can be used in practice for PSO, and the convergence interval for multidimensional issues is mostly unknown [27]. Furthermore, in a complicated context, this algorithm is unable to ensure that it will produce the best answer.

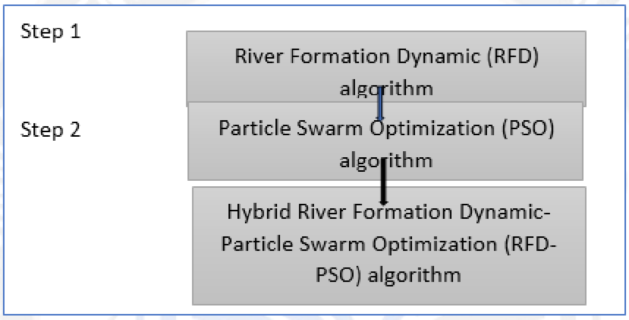


Figure. 4 The suggested Methodology for Path Planning

4.3 Hybrid algorithm

The River Formation Dynamic technique can find a collision-free path, but because it takes a long time, there is no certainty that it will find the best way. On the other hand, the particle swarm optimization method offers a smooth path but can, especially in limited pathways, enter a local optimum that results in an ineffective solution.

To create the RFD-PSO hybrid algorithm, this study employs these two approaches. This proposed method uses the PSO algorithm along with the River Formation Dynamic algorithm to construct the optimal path with avoiding collisions.

The following is a hybridization approach applied in this algorithm:

1. In the beginning an estimated path to the target is found using the RFD technique. Subsequently, one of the particles in the PSO algorithm starts from this RFD channel.

2. To fine-tune the route, the PSO algorithm is then repeated through several rounds. Every particle is a possible solution (that is, a possible route the robot might take).

3. The PSO method modifies the particle placements by taking into account both the particle's personal best position thus far and the best position discovered globally by all particles combined.

4. To keep the particles from traveling too far in a single iteration, the velocities are also limited by a maximum and minimum value. A particle's velocity is mirrored if its new position is outside of the specified range.

5. The code determines if the current solution is practical at the conclusion of each iteration, and if it is, it is designated as the best option overall.

6. The last path that the robot should take is the global best solution that the PSO algorithm, which was initially directed by the RFD path, discovered.

5. Simulation results

Figure 5 illustrates the utilization of a static environment with a space of work measuring (700 x 700) cm to assess the efficacy of the suggested technique. Utilizing the MATLAB 2022a package, the computer hardware specs were Intel Core i7-1165G7, 2.80GHz CPU, 8.00 GB RAM, and other specifications. Finding an optimal path will be accomplished by using three different algorithms: the proposed Hybrid RFD-PSO algorithm, the PSO algorithm, and the suggested RFD algorithm. The outcomes of these algorithms will be compared to determine the optimal path while keeping the robot and obstacles at a safe distance.

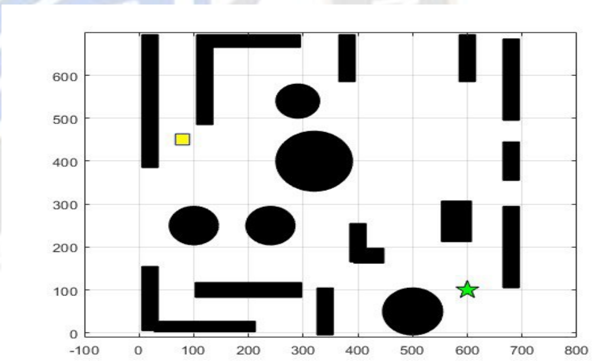


Figure. 5 The Proposed work environment

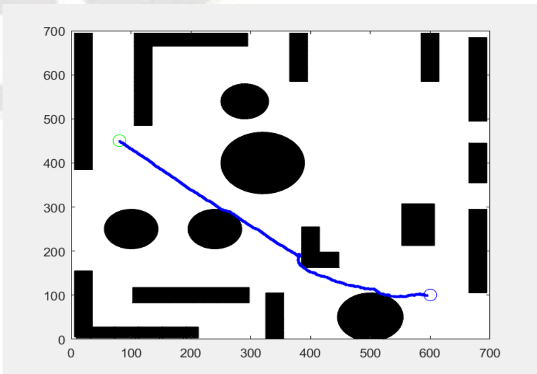


Figure. 6 simulated outcomes of the path length of the RFD method

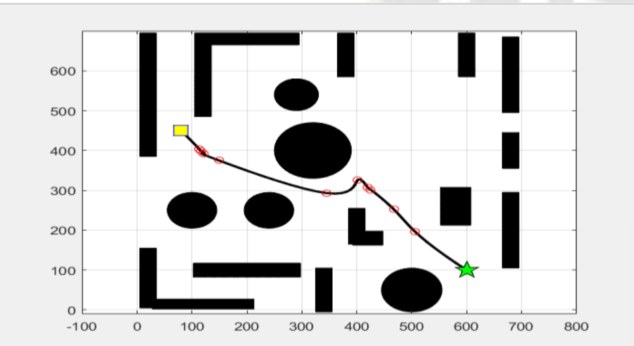
International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
***Article Received:** 30 April 2023 **Revised:** 30 June 2023 **Accepted:** 20 July 2023*

A Case study:

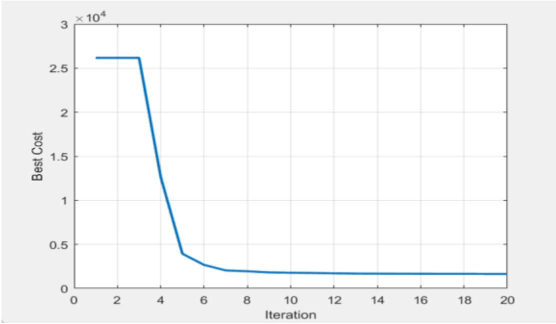
The robot starts at location (80, 450) cm and has a destination set at coordinates (600, 100) cm. An ideal path of 596.69 cm was obtained by using the River Formation Dynamic approach, as shown in Figure 6. The best route inside the assigned workspace was then found using the PSO approach, as shown in Figure 7 [a].The maximum possible number of repetitions in Figure 7 [b] is 50, and repetition number 20 yielded the best cost option. The value of the PSO function for costs is 626.8174 centimeters. The recommended hybrid strategy was applied in Figure 8 [c] to determine the optimal path through the indicated workspace. As shown in Figure 8[d], the optimal cost solution was identified at repeat number 20, with a maximum repetition number of equal to 50 repetitions. The distance function of the hybrid approaches that are suggested is 628.8554 cm. Table 2 shows that when compared to the river formation dynamic and PSO algorithms, the hybrid algorithm's suggested route was the fastest and most efficient way to get from the starting point to the destination.

Table 2. Comparison of the path lengths

Algorithm's	lengths of the shortest paths	Repetition
The suggested RFD	**626.8174 cm**	**20**
The PSO	**628.69 cm**	**20(no. of drops)**
The suggested hybrid RFD-PSO	**596.8554 cm**	**20**

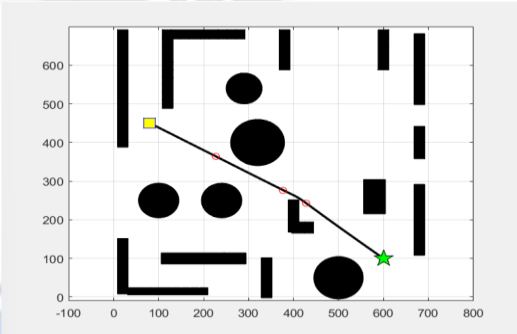


[a]

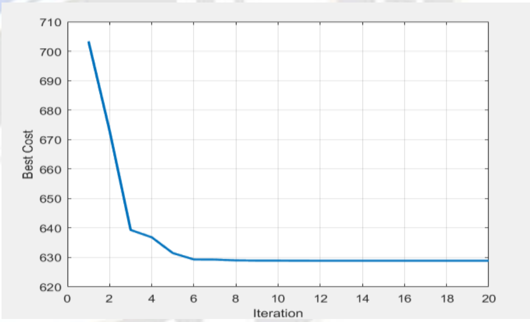


[b]

Figure. 7 The PSO algorithm: [a] path-planning and [b] the cost function.



[c]



[d]

Figure. 8 The proposed hybrid algorithm (RFD-PSO): [a] path-planning and [b] the cost function

To demonstrate that the suggested hybrid algorithm, the RFD-PSO, offers the shortest path, a comparison with other studies using various path planning algorithms in a static environment has been carried out. First, using a complicated environmental map and a [20 by 20] m area, the suggested RFD-PSO was contrasted with the [SAACO] and the [FACO], as presented in [28,29]. Fig. 9 displays the outcomes of the River Formation Dynamic algorithm simulation procedure. The result is a route length of 32.6793 m.

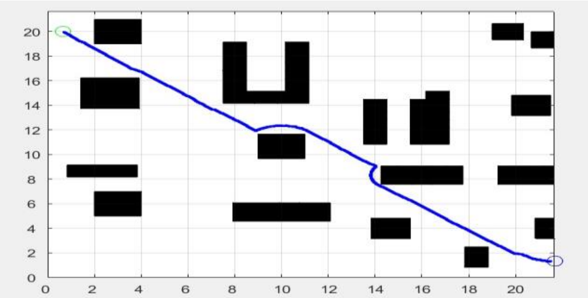
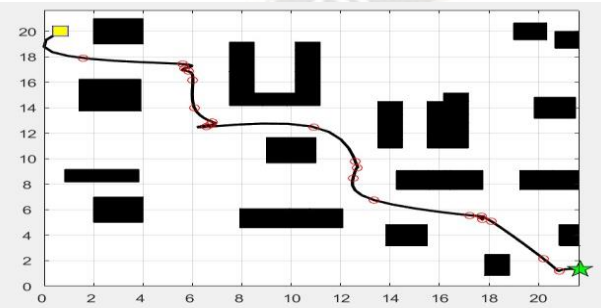
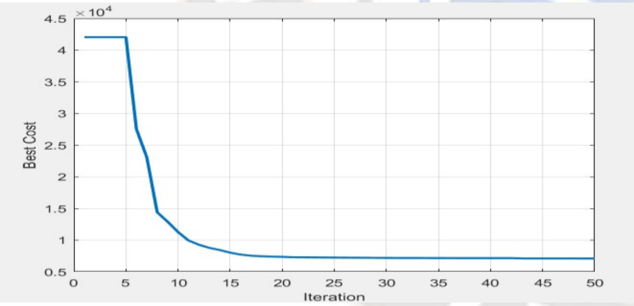


Figure. 9 simulated outcomes of the path length of the RFD algorithm.

Using the same environment as in Figure. 10 [1], the PSO method yields a optimal path of 44.9569 m and a repetition number of 48, as in Figure. 10 [2], with the greatest repetition number of equal to 50 repetitions.



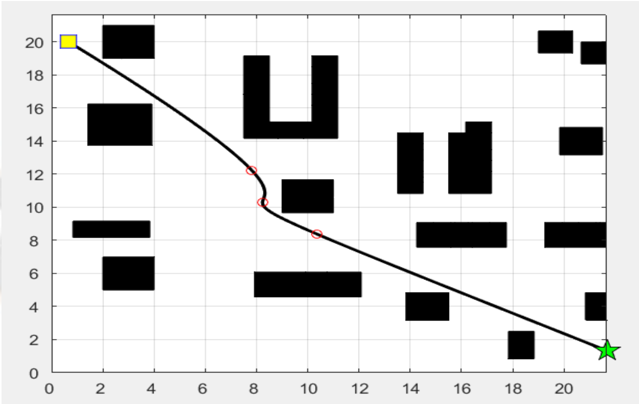
[1]



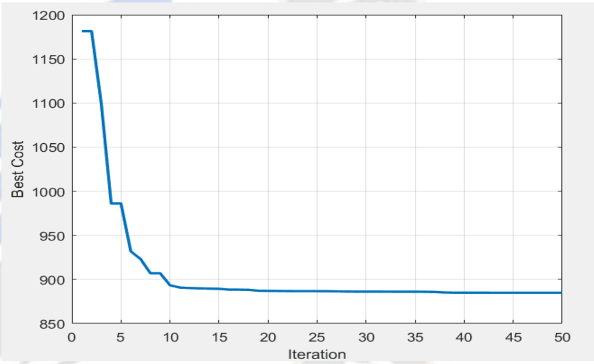
[2]

Figure. 10 The PSO algorithm: [1] path-planning and [2] the best cost function

Furthermore, as seen in Figure. 11[b] with a greatest repetition number equal to 50 repetitions, the simulation process utilizing the hybrid [RFD-PSO] method is presented in Figure. 11[a] yields a route length equal to 29.5753 m with a repetition number equal to 39.



[a]



[b]

Figure. 11 [a] simulated outcomes of the path length using the hybrid [RFD-PSO] Algorithm and [b] the hybrid RFD-PSO algorithm best cost function

Table 3 presents the outcomes of the comparison procedure and demonstrates how the hybrid method can create an optimal path with a smooth steering function by generating a small distance between the start and goal points. This attests to the suggested hybrid algorithm's effectiveness.

Table 3. Comparisons of the path lengths with the literary works [28,29]

Algorithm's	lengths of the shortest paths	Repetition
SAACO [28]	(29.796) m	(25)
FACO [29]	(29.3848) m	(23)
RFD	(32.6793) m	20(no. of drops)
PSO	(44.9569) m	(48)
hybrid RFD-PSO	(29.5753) m	(39)

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
***Article Received:** 30 April 2023 **Revised:** 30 June 2023 **Accepted:** 20 July 2023*

Second, using a static workspace with the area of [20 x 20] m alley map, the proposed RFD-PSO was compared with Dijkstra's method, Ant Colony Optimization, A* (a comprehensive approach and a heuristic variant), and ACO* (a customized version of Ant Colony Optimization) [22]. Figure. 12 displays the output of the River Formation Dynamic algorithm simulation procedure, which yielded a route with a length of 117.12 cm. Furthermore, using the PSO method in the same environment, as seen in Figure. 13, the simulation procedure yielded a route length of 45.20 cm. In contrast, the hybrid QOPSO algorithm, as seen in Figure. 14, generates a route with a length equal to 31.65 cm.

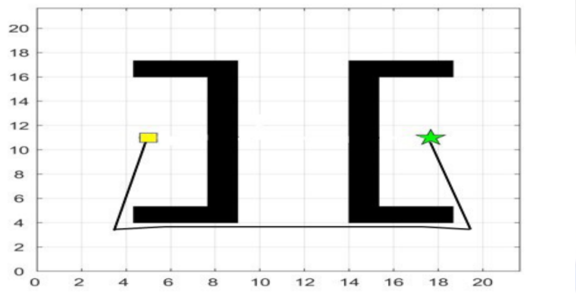


Figure. 12 Simulated outcomes of the path length of the RFD method.

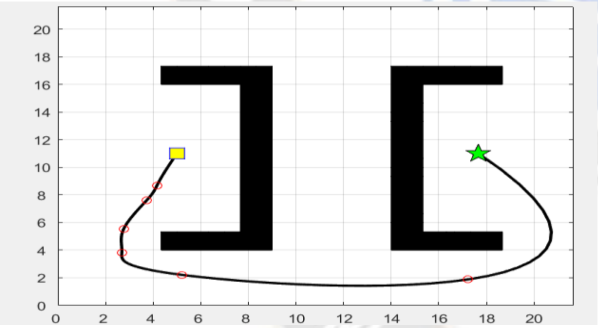


Figure. 13 Simulated outcomes of the path length of the PSO method.

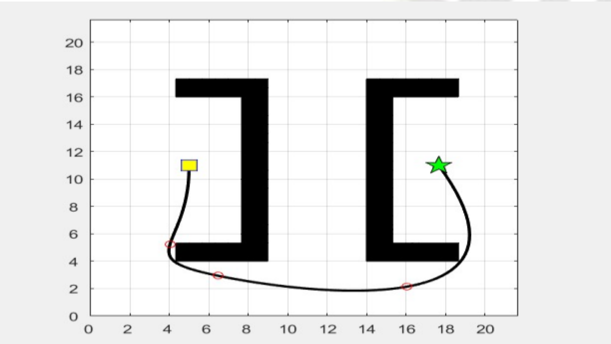


Figure. 14 Simulated outcomes of the path length of the hybrid RFD-PSO method.

Comparing the hybrid RFD-PSO method with Dijkstra's, A*, ACO, and ACO*, the simulation results show that it can effectively construct the shortest path.

Table 4. Comparisons of the path lengths with the literary works [22]

Algorithm's	lengths of the shortest paths	Repetition
Dijkstra's	33.48 cm	100
A*	33.48 cm	100
ACO	138.98 cm	100
ACO*	87.49 cm	100
RFD	117.12 cm	100
PSO	45.20 cm	100
hybrid RFD-PSO	31.65 cm	100

6. Conclusion

This paper suggests a hybrid strategy to handle two problems: avoiding obstacles and determining the mobile robot's smoothest and shortest path. The hybrid method employs both the River Formation Dynamic (RFD) and the PSO algorithms. The outcomes of the simulation show that by guiding the mobile robot toward the goal location, the suggested hybrid (RFD-PSO) strategy enhances the path length. A comparative study was conducted between the River Formation Dynamic (RFD) algorithm and multiple other algorithms in two different situations. While Ant Colony Optimization (ACO) and a modified version known as ACO* were included in the second environment, the first environment contained Self-Adaptive Ant Colony Optimization (SAACO) and fuzzy Ant Colony Optimization (FACO).

The first environment's comparison findings demonstrate that the RFD algorithm offers a path length increase of 27.43% when comparing to the SAACO approach and 23.25% when comparing to the FACO approach. Additionally, the second environment's comparative findings demonstrate that the RFD algorithm improves the path by 24.48% when compared to the ACO algorithm and 7% when comparing to the ACO* method. Comparing the suggested river creation dynamic method to the hybrid RFD-PSO algorithm, the latter offers a smoother course with the least distance. In comparison to the RFD algorithm, the hybrid RFD-PSO algorithm yields a 0.34% improvement in path length. Meanwhile, the hybrid RFD-PSO algorithm that was suggested proved to be more successful in handling complex environments than the PSO algorithm. When utilizing a complex map to compare the hybrid RFD-PSO method with the PSO algorithm, the RFD-PSO offers a path length improvement of 13.2% over the PSO approach. The suggested hybrid RFD-PSO method offered the optimal path for avoiding collisions when the mobile robot's path length was compared to several research papers.

References

[1] M. Dirik and A. F. Kocamaz, "RRT- Dijkstra: An Improved Path Planning Algorithm for Mobile Robot", *Journal of Soft Computing and Artificial Intelligent*, pp:69-77, 2020.

[2] M. Fuad, T. Agustinah, and D. Purwanto, "Modified Headed Social Force Model based on Hybrid Velocity Obstacles for Mobile Robot to Avoid Disturbed Groups of Pedestrians", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 3, pp. 222-241, 2021.

[3] A. Montazeri, A. Can, I. H. Imran, "Unmanned Aerial Systems: Autonomy, Cognition and Control", *Preprint submitted to Journal of LATEX Templates*, 2020.

[4] M. N. Zafara, J. C. Mohantab," Methodology for Path Planning and Optimization of Mobile Robots: A Review", *Procedia Computer Science*, pp:141-152, 2018.

[5] A. S. A. Araji, A. K. Ahmed, and M. K. Hamzah, "Development of a Path Planning Algorithms and Controller Design for Mobile Robot", In: *Proc. of Scientific Conference of Electrical Engineering*, pp. 72-77, 2018.

[6] A. S. A. Araji, M. F. Abbod, and H. S. A. Raweshidy, "Design of a Neural Predictive Controller for Nonholonomic Mobile Robot based on Posture Identifier", In: *Proc. of the Lasted International Conf. on Intelligent Systems and Control*, pp. 198-207, 2011.

[7] O. A. R. A. Wahhab, A. S. Al-Araji," Path Planning and Control Strategy Design for Mobile Robot Based on Hybrid Swarm Optimization Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol.14, No.3, 2021.

[8] L. D. Filippis, G. Guglieri," Advanced Graph Search Algorithms for Path Planning of Flight Vehicles", *Recent Advances in Aircraft Technology*, pp:157-192, 2012.

[9] N. Adzhar, Y. Yusof, M. A. Ahmad," A Review on Autonomous Mobile Robot Path Planning Algorithms", *Advances in Science, Technology and Engineering Systems Journal*, Vol. 5, No. 3, pp: 236-240, 2020.

[10] S. K. Debnath, R. Omar, S. Bagchi, S. E. N, M.H. A. S. Kandar, K. Foyso, and T. K. Chakraborty, "Different Cell Decomposition Path Planning Methods for Unmanned Air Vehicles - A Review", In: *Proc. of the 11th National Technical Seminar on Unmanned System Technology*, pp. 99-111, 2021.

[11] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", In: *Proc. Of the Autonomous Robot Vehicles*, pp. 396-404,1985.

[12] Z. E. Kanoon, A. S. Al-Araji, and M. N. Abdullah," Enhancement of Cell Decomposition Path-Planning Algorithm for Autonomous Mobile Robot Based on an Intelligent Hybrid Optimization Method", *International Journal of Intelligent Engineering and Systems*, Vol.15, No.3, pp: 161-175, 2022.

[13] S. K. Debnath, R. Omar, N. B. Abdul Latip, S. Shelyna, E. Nadira, C. K. N.C. K. Melor, T. K. Chakraborty and E. Natarajan, "A review on graph search algorithms for optimal energy efficient path planning for an unmanned air vehicle", *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 15, No. 2, pp. 743-749, 2019.

[14] M. S. Abed, O. F. Lutfy, Q. F. Al-Doori, "A Review on Path Planning Algorithms for Mobile Robots", *Engineering and Technology Journal*, Vol. 39, No. 05, pp: 804-820, 2021.

[15] L. A. Zadeh," Fuzzy sets*", Information and Control*, vol.8, pp:338-353, 1965.

[16] P. G. Zavlangas and S. G. Tzafestas, "Motion Control for Mobile Robot Obstacle Avoidance and Navigation: A Fuzzy Logic-based Approach", *Systems Analysis Modelling Simulation*, Vol. 43, No. 12, pp. 1625-1637, 2003.

[17] D. Janglová, "Neural Networks in Mobile Robot Motion", *International Journal of Advanced Robotic Systems*, Vol. 1, No. 1, pp. 15-22, 2004.

[18] K. Menghour, L. Souici-Meslati," Hybrid ACO-PSO Based Approaches for Feature Selection", *International Journal of Intelligent Engineering and Systems*, Vol.9, No.3, 2016.

[19] B. Beklisi, R. Chellali, A. Yaw Appiah, F. Kyeremeh, "An Overview of Nature-Inspired, Conventional, and Hybrid Methods of Autonomous Vehicle Path Planning", *Journal of Advanced Transportation*, p.27, 2018.

[20] A. Vagale, R. Oucheikh, R. T. Bye, O. L. Osen, T. I. Fossen, "Path planning and collision avoidance

for autonomous surface vehicles I: a review", *Journal of Marine Science and Technology*, pp:1292–1306, 2021.
[21] P. Rabanal, I. Rodriguez, and F. Rubio, "Solving Dynamic TSP by Using River Formation Dynamics", *Fourth International Conference on Natural Computation,* pp. 246-250, 2008.
[22] G. Redlarski, M. Dabkowski, and A. Palkowski," Generating optimal paths in dynamic environments using river formation dynamics algorithm", *Journal of Computational Science*, 20:8-16, 2017.
[23] G. Redlarski, A. Palkowski, M. Dabkowski, "Using River formation dynamics algorithm in mobile robot navigation", *Journal of Solid State Phenomena*, Vol. 198, pp. 138–143, 2013.
[24] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", In*: Proc. of International Conference on Neural Networks*, pp. 1942-1948, 1995.
[25] R. H. A. Rubayi, M. K. Abd, and F. M. Flaih, "A New Enhancement on PSO Algorithm for Combined Economic-Emission Load Dispatch Issues", *International Journal of Intelligent Engineering and Systems,* Vol. 13, No. 1, pp. 77- 85, 2020.
[26] F. V. D. Bergh and A. P. Engelbrecht, "A Study of Particle Swarm Optimization Particle Trajectories", *Information Sciences*, Vol. 176, No. 8, pp. 937-971, 2006.
[27] S. K. Debnath, R. Omar, S. Bagchi, S. E. N, M. H. A. S. Kandar, K. Foyso, and T. K. Chakraborty, "Different Cell Decomposition Path Planning Methods for Unmanned Air Vehicles - A Review", In: *Proc. of The 11th National Technical Seminar on Unmanned System Technology,* pp. 99-111, 2021.
[28] Y. Peng, Z. Wang, J. Jin, P. Y. Huang, and W. Lu, "The Global Trajectory Programming of Robot based on Adaptive Ant Colony Algorithm", In: *Proc. of International Conference on Intelligent Human-Machine Systems and Cybernetics,* Vol. 1, pp. 108-111, 2013.
[29] C. Yen and M. Cheng, "A Study of Fuzzy Control with Ant Colony Algorithm Used in Mobile Robot for Shortest Path Planning and Obstacle Avoidance", *Microsystem Technologies*, Vol. 24, No. 1, pp. 125-135, 2018.
[30] Balasubramanian, S., Raparthi, M., Dodda, S. B., Maruthi, S., Kumar, N., & Dongari, S. (2023). AI-Enabled Mental Health Assessment and Intervention: Bridging Gaps in Access and Quality of Care. PowerTech Journal, 47(3), 81. https://doi.org/10.52783/pst.159