Optimization of Mobile Robot Navigation Using Hybrid Dragonfly-Cuckoo Search Algorithm

Muthukumaran S, Sivaramakrishnan R
Department of Production Technology
Madras Institute of Technology, Anna University
Chennai, India
smkumaran90@gmail.com

Abstract

Mobile robot navigation is considered one of the most complex tasks in the mobile robotics application space. Mobile robot navigation is regarded as a multi-objective optimization problem since several parameters such as path length, bending energy, and time are to be optimized simultaneously. A novel approach for navigation of a mobile robot in a complex environment is presented in this work. Two well-known optimization algorithms, namely Dragonfly and Cuckoo search algorithms, are hybridized to perform the mobile robot's path planning task. Objective functions for minimizing the path length and bending energy are designed and implemented. Obstacle avoidance is also an essential criterion considered here. Simulation results prove that the developed hybrid algorithm performs better than the standalone individual algorithms. Simulations using Turtlebot and ROS are performed to evaluate the path planning feasibility. The proposed algorithm also proves its superiority in selected benchmark functions.

Keywords: *Metaheuristic optimization algorithms, Dragonfly algorithm, Cuckoo search algorithm, Multi-objective optimization, Obstacle avoidance, Robot navigation.*

1. Introduction

Any mobile robot's primary goal is to find the optimal and feasible navigation path between the start and target points without hitting the obstacles. Mobile robot navigational strategies are classified based on the amount of available information regarding the robot workspace. In this work, environment details are primarily unknown and also the robot can react to the sudden changes in the environment. Path planning algorithms are also classified based on the number of robots and the moving targets' nature.

Several researchers developed and implemented several optimization algorithms over the decades, but it remains a challenging task. Several pieces of literature prove that heuristic and meta-heuristic optimization algorithms perform better in robot navigation tasks [13].

Several heuristic/metaheuristic algorithms such as memetic algorithms [1], differential evolution algorithm [6], NSGA-II algorithm [5], generalized wavefront algorithm [15] and grasshopper optimization algorithm [16], are implemented by scientists to solve multi-objective mobile robot navigation problem. Each algorithm is based on the different swarm characteristics and possesses distinct advantages and disadvantages, and there is still scope for improvement.

Dragonfly algorithm is a popular optimization algorithm with a proven record of accomplishment in various multi-objective optimization tasks [4]. Several complex optimization problems such as tuning machine learning algorithms, colour image segmentation, and wireless networking are successfully optimized using Dragonfly algorithm. Though Dragonfly optimization algorithm is successful in several optimization spaces, it has not been explored much in robotics optimization.

1.1 Novelty of the proposed work

In this present research, a novel Hybrid Dragonfly-Cuckoo optimization algorithm is proposed. The proposed Hybrid Dragonfly-Cuckoo Optimization Algorithm is implemented to solve mobile robots' navigation problem in unknown or partially known environments. From several pieces of literature, it is marked that the hybridization of Dragonfly algorithm with the help of Cuckoo search algorithm is also entirely novel.

1.2 Hybrid Dragonfly-Cuckoo Search Algorithm

Dragonfly Algorithm (DA) algorithm is an optimization algorithm stimulated by the different static and dynamic swarming characteristics of Dragonflies, representing the exploration and exploitation phases of metaheuristic optimization[14]. Direct actions of insects such as separation from the neighbour, alignment towards the goal, cohesion among neighbours, attraction to the food source, a distraction from enemies are considered.

These principles define the moving behaviour of dragonflies during its exploration and exploitative phases. During exploration, the high alignment and low cohesion of dragonflies are maintained, and during exploitation, poor alignment and high cohesion are held between the dragonflies. Levy random parameter is employed here to improve the stochastic nature of the algorithm.

Similar to the Particle Swarm Optimization algorithm, step vector and position vector track dragonflies' movement directions/speed and position [12]. The step vector and position vector equation are mentioned below.

$$\Delta F_{(t+1)} = \left(s * S_{(i)} + a * A_{(i)} + c * C_{(i)} + f * F_{(i)} + e * E_{(i)}\right) + w * \Delta F_{(t)} \qquad (1)$$

$$F_{(t+1)} = F_{(t)} + \Delta F_{(t+1)} \qquad (2)$$

Cuckoo Search (CS) algorithm is based on cuckoo birds' reproduction behaviour (laying their eggs in the nests of other species) [2]. If the host bird discovers that the eggs in their nests do not belong to them, then the alien eggs are thrown out of the nests. Sometimes the host birds are forced to discard the whole nest itself. This worst nest replacement strategy highly improves the survival and productivity of the cuckoo breeds. The rules of CS algorithms are based on the following steps

Initially, each cuckoo bird will lay one egg at a time in a randomly chosen nest. Then the nests possessing high-quality eggs will be carried to the next generation. A host bird can discover an alien egg with a probability $p_a \in (0,1)$. Finally, the chosen alien egg will be discarded, or the whole nest will be abandoned.

The following equation is used to produce a new solution$X^{(t+1)}$, for a cuckoo I, by a levy flight where $\propto$ is the step size and $\oplus$ is the entry wise product operator.

$$X_i^{t+1} = X_i^t + \propto \oplus \, levy\ (\lambda) \qquad (3)$$

The proposed approach the Dragonfly - Cuckoo Algorithm (DA-CS) combines the behaviours of Dragonflies with the worst nest replacing a strategy of the CS algorithm. After each iteration, a fraction of the worst-performing particle is chosen in terms of the fitness function. The selected particles should be abandoned, and then the replacement of randomly generated ones is undertaken within the specified search space.

2. Objective Function Formulation

In this paper, the proposed Hybrid DA-CS algorithm is used to perform path planning of mobile robots. Each dragonfly in the search space is considered as a solution in the search space. So after iterating Dragonflies in the entire search space, several optimal and sub-optimal paths can be identified easily. One significant advantage of using metaheuristic algorithms is that several optimal and sub-optimal solutions are obtained easily in the search space. So if the robot encounters an obstacle in the middle, the robot will be redirected with suboptimal solutions. The critical parameters considered here for optimization are path length and the turning angle.

The entire environment can be modelled as a collection of node points and are represented by (x,y) coordinates. The shortest distance is defined as the sum of all points between the start to the target. Here the distance between two points is calculated based on the Euclidian distance. The Euclidean distance between two consecutive points $n_i = (x_i, y_i)$ and $n_{i+1} = (x_{i+1}, y_{i+1})$ are represented as

$$Path\ Lenth = \sum_{i=0}^{n} \sqrt{(x_i - x_{i+1})^{2} + (y_i - y_{i+1})^{2}} \qquad (4)$$

Path smoothness is directly related to the energy loss of the robot. The angle between two consecutive lines $n_1 n_2$ and $n_2 n_3$ are represented as

$$Path\ Smoothness = \frac{1}{n} \sum_{i=0}^{n} \pi - \cos^{-1} \frac{(x_2 - x_1)(x_3 - x_2) + (y_2 - y_1)(y_3 - y_2)}{dis(n_1, n_2) \times dis(n_2, n_3)} \qquad (5)$$

3. Path Planning Procedure

The steps to be following in the path planning are described as follows

Step 1: Model of the environment. All the starting point, target point, obstacle information are to be characterized for the robot modelling.

Step 2: All the related parameters such as population, separation, alignment and other dragonfly parameters are initialized.

Step 3: Dragonflies are initialized with the parameters and explored to find the optimal and suboptimal points.

Step 4: After each iteration, the worst set of dragonflies are terminated, and new dragonflies are introduced. This worst nest replacement strategy highly improves the quality of the solution.

Step 5: Optimal path nodes that satisfy the path length, and path smoothness objective function are selected and sent to the robot for tracking.

Step 6: The procedure is continued until the robot reaches the target.

4. Simulation Results

The proposed path-planning algorithm is implemented and tested in both Matlab and ROS based simulator. The environment considered here is a two-dimensional continuous workspace with start, goal and obstacles. The entire setting is modelled in Gazebo simulator, and a Turtlebot is placed in the environment. Graphical simulation proves that the robot can reach the target in the shortest, smoothest optimal path.

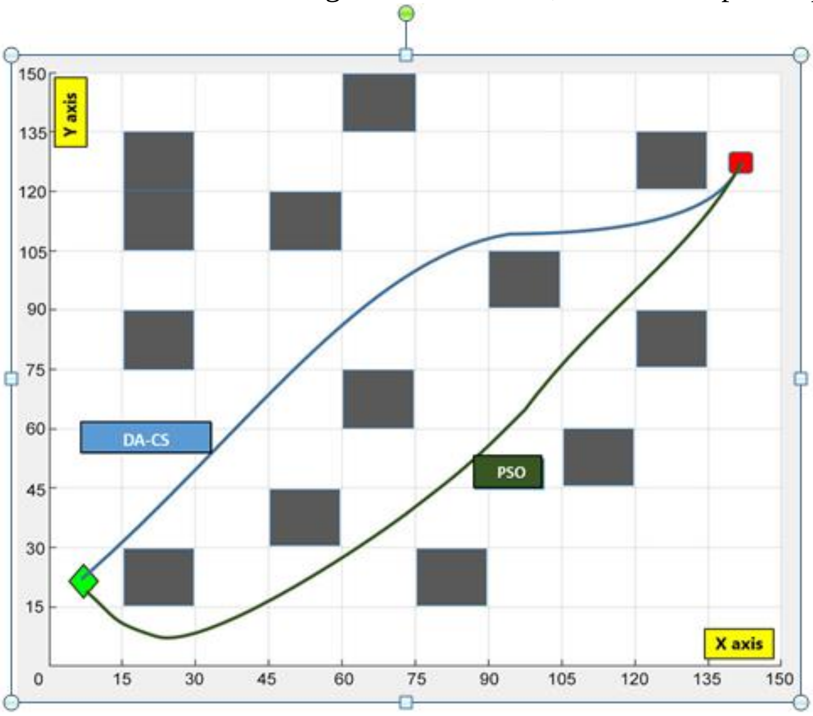

Fig 1 - Simulated Path comparison between Hybrid DA-CS and PSO

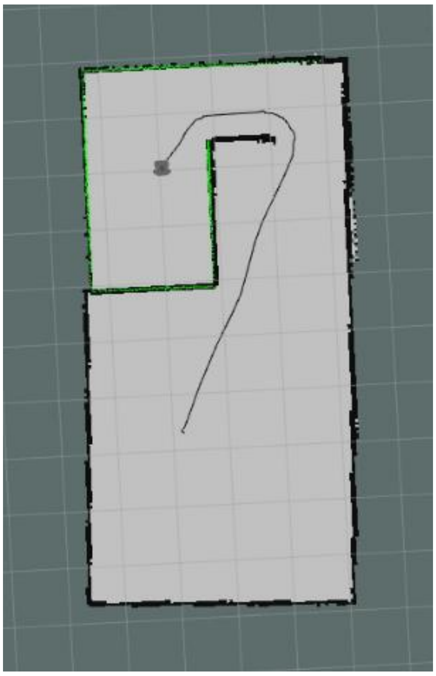

Fig 2 - Simulated Path traced by the Turtlebot using Hybrid DA-CS in ROS

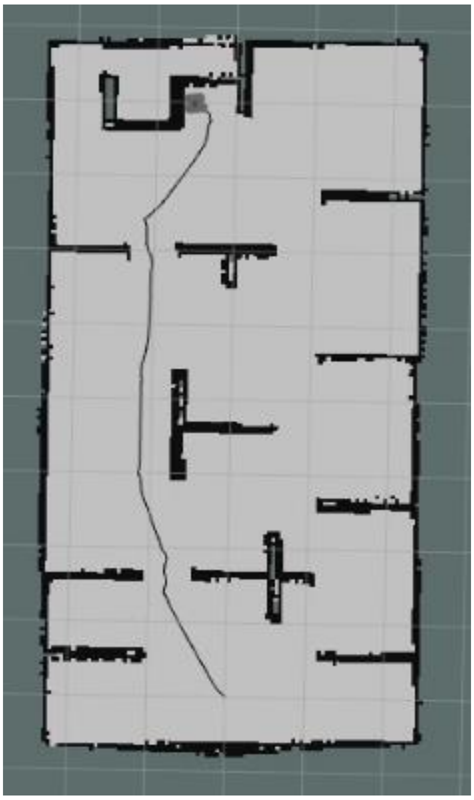

Fig 3 - Simulated Path traced by the Turtlebot using Hybrid DA-CS in ROS

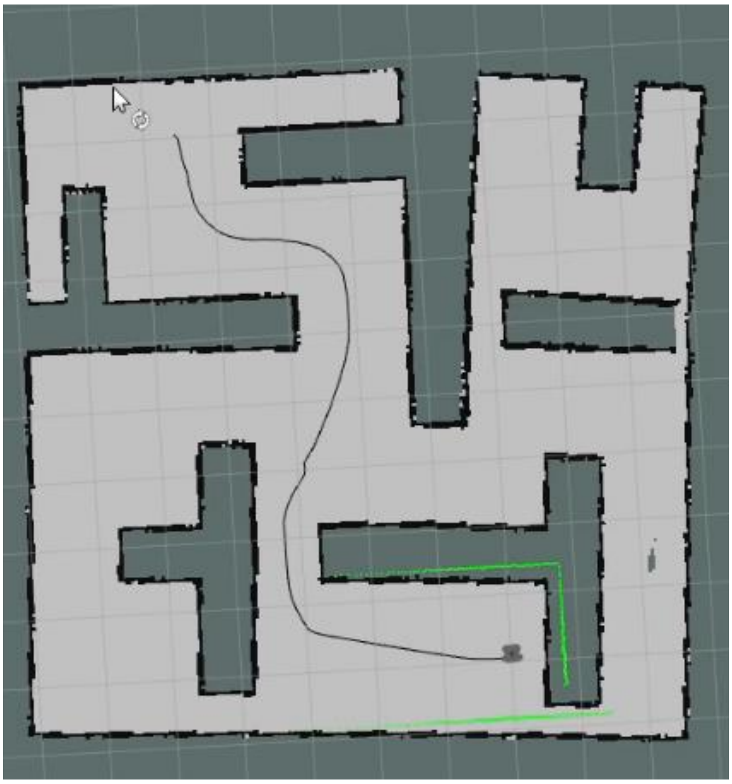

Fig 4 - Simulated Path traced by the Turtlebot using Hybrid DA-CS in ROS

5. Benchmark Function Evaluation

To evaluate the advantages of using the Cuckoo search algorithm with the Dragonfly algorithm, the proposed algorithm is evaluated using standard benchmark functions. The benchmark functions are selected to possess all the crucial properties like peaks, valleys and channels. The proposed Hybrid Dragonfly – Cuckoo search algorithm is tested on benchmark functions compared with conventional standalone Dragonfly and Cuckoo Search Algorithm. The benchmark functions considered for evaluation are

Table 1: Benchmark functions

Test Function	Function	f_{min}
F-1	$\sum_{i=1}^{n} x_i^2$	0
F-2	$f(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	0
F-3	$f(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos \frac{x_i}{\sqrt{i}} + 1$	0

All the iterations are set to 500 and are experimented over 30 times, and the average results are presented here. The above experiment proves that the proposed Hybrid DA-CS algorithm achieves solutions in lesser iterations compared to conventional standalone Dragonfly and Cuckoo search algorithm.

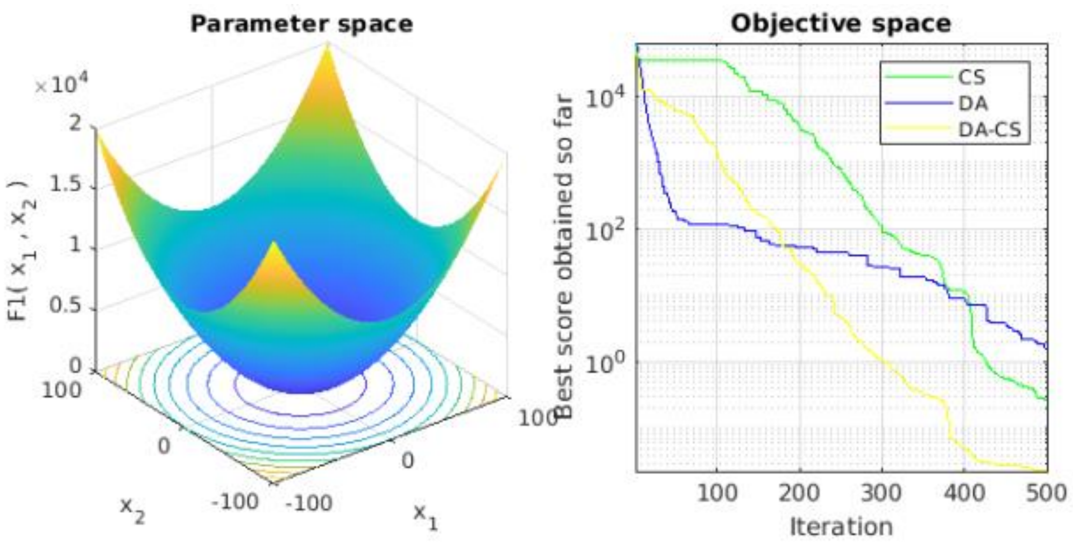

Fig 5 - Convergence graph of F-1

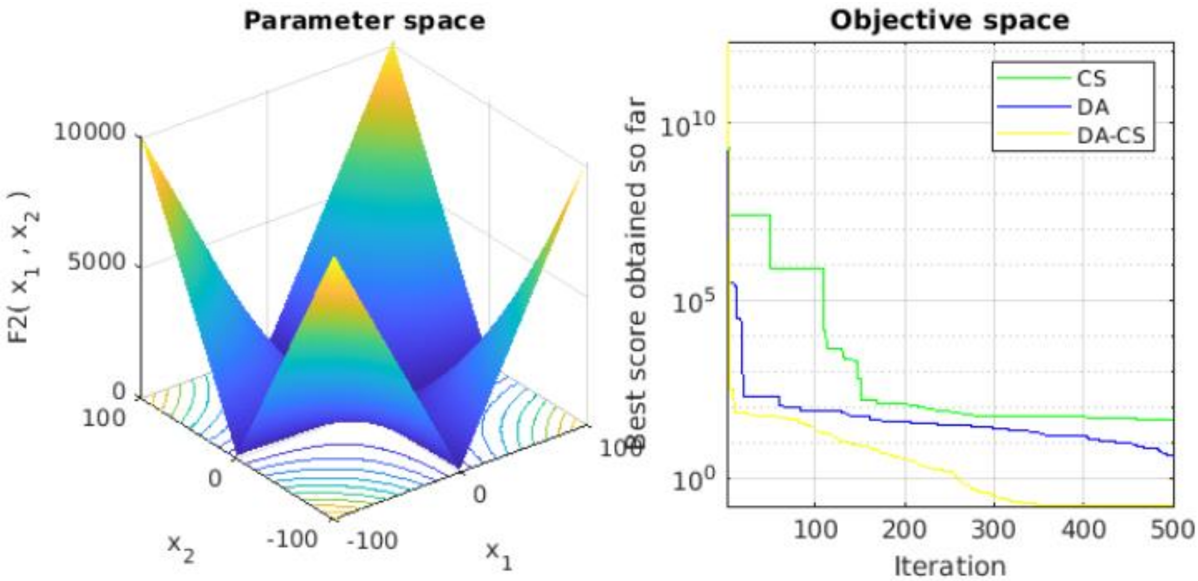

Fig 6 - Convergence graph of F-2

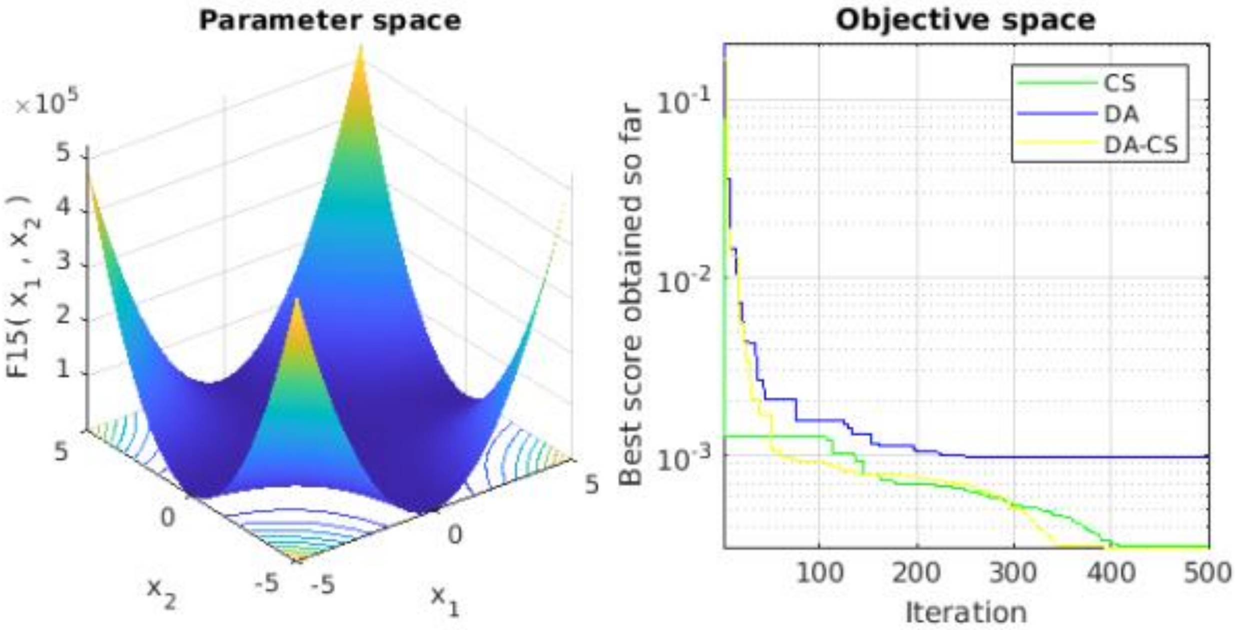

Fig 7 - Convergence graph of F-3

6. Conclusion and Future work

In this proposed work, a new hybrid optimization algorithm called Hybrid Dragonfly-Cuckoo Search algorithm is proposed. The proposed Hybrid Dragonfly-Cuckoo Search algorithm is implemented in Matlab and ROS to evaluate its performance in mobile robot path planning applications. The simulated results prove that the proposed algorithm can generate a short, smooth optimal path for the robot to navigate. All the simulated results also demonstrate that the proposed algorithm is beneficial in providing obstacle avoidance. The proposed Hybrid Dragonfly-Cuckoo search algorithm is also tested with the standard benchmark functions. In benchmark function evaluation, the proposed hybridized algorithm is compared with the conventional standalone algorithm. The benchmark function evaluation proves that the proposed hybridization is capable of converging earlier than the standalone algorithms.

In this work, the only environment with static obstacles is considered. Further, the result can be extended to the environment with dynamic obstacles. The scope of research can also be extended to auto-tuning of parameters. The proposed algorithm can also be integrated with neural networks to introduce the algorithm learning capabilities.

References

1. Marco A. Contreras-Cruz, Victor Ayala-Ramirez, Uriel H. Hernandez-Belmonte, "Mobile robot path planning using artificial bee colony and evolutionary programming", Applied Soft Computing 30 (2015) pp.319–328
2. P. K. Mohanty, and D. R. Parhi, "Cuckoo search algorithm for the mobile robot navigation", In International Conference on Swarm, Evolutionary, and Memetic Computing, pp. 527-536, Springer, Cham, 2013.
3. P. K. Das, H. S. Behera, and B. K. Panigrahi, "A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning", Swarm and Evolutionary Computation, vol. 28, pp. 14-28, 2016
4. Fatin H Ajeil, Ibraheem Kasim Ibraheem, Mouayad A.Sahib, Amjad J.Humaidi, "Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm", Applied Soft Computing Volume 89, April 2020.
5. Seyedali Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems", Neural Computing and Applications, volume 27, pages 1053–1073, 2016.
6. Iztok Fister Jr, Dušan Fister and Iztok Fister, "A comprehensive review of cuckoo search: variants and hybrids", International Journal Mathematical Modelling and Numerical Optimisation, Vol. 4, No. 4, 2013
7. Kashif Hussain, Mohd Najib Mohd Salleh, Yuli Adam Prasetyo, Shi Cheng, "Personal Best Cuckoo Search Algorithm for Global Optimization", International Journal on Advanced Science Engineering Information Technology

8. Y-Q. Qin, D-B. Sun, N. Li, and Y-G. Cen, "Path planning for mobile robot using the particle swarm optimization with mutation operator", In Proceedings of 2004 International Conference on Machine Learning and Cybernetics, 2004, vol. 4, pp. 2473-2478, IEEE, Chine 2004.
9. C. Liu, Z. Gao, and W. Zhao, "A new path planning method based on firefly algorithm", In 2012 Fifth International Joint Conference on Computational Sciences and Optimization (CSO), pp. 775-778, IEEE, 2012.
10. Alejandro Hidalgo-Paniagua, Miguel A. Vega-Rodríguez, JoaquínFerruz, Nieves Pavón, "Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach", Soft Computing volume 21, pages 949–964 (2017)
11. Azzeddine Bakdi, Abdelfetah Hentout, Hakim Boutami, Abderraouf Maoudj, Ouarda Hachour, Brahim Bouzouia, "Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control", Robotics and Autonomous Systems 89 (2017) 95–109
12. Evan Krell, AlaaSheta, Arun Prashanth Ramaswamy Balasubramanian, Scott A. King, "Collision-Free Autonomous Robot Navigation In Unknown Environments Utilizing PSO For Path Planning", JAISCR, Sciendo, Vol. 9, No. 4, pp. 267-282,2019.
13. Mohd Nadhir, Ab Wahab, "A comparative review on mobile robot path planning: Classical or meta-heuristic methods?", Annual Reviews in Control, 2020
14. Yassine Meraihi, "Dragonfly algorithm: a comprehensive review and applications", Neural Computing and Applications, 2020
15. Sifan Wu, Yu Du, and Yonghua Zhang, "Mobile Robot Path Planning Based on a Generalized Wavefront Algorithm", Mathematical Problems in Engineering, Volume 2020, https://doi.org/10.1155/2020/6798798
16. Zahra Elmi and Mehmet Önder Efe, "Online path planning of mobile robot using grasshopper algorithm in a dynamic and unknown environment", Journal of Experimental & Theoretical Artificial Intelligence, 2020, https://doi.org/10.1080/0952813X.2020.1764631