Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 10, No. 1, March 2024, pp. 166-172
ISSN: 2338-3070, DOI: 10.26555/jiteki.v10i1.28513

Path Planning for Mobile Robots on Dynamic Environmental Obstacles Using PSO Optimization

Fahmizal[1], Innes Danarastri[1], Muhammad Arrofiq[1], Hari Maghfiroh[2], Henry Probo Santoso[3], Pinto Anugrah[4,5], Atinkut Molla[6]
[1]Department of Electrical Engineering and Informatics, Vocational College, Universitas Gadjah Mada, Indonesia
[2]Department of Electrical Engineering, Universitas Sebelas Maret, Surakarta, Indonesia
[3]Robotic and Autonomous Systems, University of Sussex, United Kingdom
[4]Department of Electrical Engineering, Universitas Andalas, Padang, Indonesia
[5]School of Electrical Engineering and Computer Science, University of Queensland, Brisbane, Australia
[6]Department of Information Technology, Injibara University, Injibara, Ethiopia

ARTICLE INFO

Article history:

Received March 07, 2024
Revised March 30, 2024
Published April 17, 2024

Keywords:

Mobile robot;
Path planning;
Optimization;
Particle swarm optimization (PSO)

ABSTRACT

The increasing integration of mobile robots in various industries necessitates efficient navigation strategies amidst dynamic environments. Path planning plays a crucial role in guiding mobile robots from their starting points to target destinations, contributing to automation and enhancing human-robot collaboration. This study focuses on devising a tailored path-planning approach for a fleet of mobile robots to navigate through dynamic obstacles and reach designated trajectories efficiently. Leveraging particle swarm optimization (PSO), our methodology optimizes the path while considering real-time environmental changes. We present a simulation-based implementation of the algorithm, where each robot maintains position, velocity, cost, and personal best information to converge towards the global optimal solution. Different obstacles consist of circles, squares, rectangles, and triangles with various colors and five handle-points used. Our findings demonstrate that PSO achieves a global best cost of 5.1017, indicative of the most efficient path, minimizing overall distance traveled.

Corresponding Author:

Hari Maghfiroh, Department of Electrical Engineering, Universitas Sebelas Maret, Surakarta, Indonesia
Email: hari.maghfiroh@staff.uns.ac.id

1. INTRODUCTION

Nowadays, robot demand is rising. Robots are becoming more and more commonplace in our daily lives. They have been essential in security, the services industry, humanitarian aid, drone photography, and security [1]-[4]. Wheeled robot is one type of robot which increased the popularity [5]-[6]. The use of mobile robots is currently increasing in several fields including industry, security, hygiene, medicine, and other fields [7]. Navigation is an important thing for mobile robots [8]. There are general problems of navigation which are perception, localization, motion control, and path planning [9]-[11]. Path planning is a fundamental thing that can help solve navigation problems, such as in mobile robots. The path planning algorithm is used to plan a path so that it can be followed by the mobile robot, which moves from the starting point to the target point. According to [12]-[14], path planning is the determination of a collision-free path in each environment.

Path planning is divided into local and global based on environmental knowledge [15]-[16]. Global path planning is for known environmental information. On the other hand, the local one is for unknown environmental information, the robot takes information during running using local sensors. Dynamic environments have multiple cases for each object and obstacle. Objects and obstacles can both move, and only one of them moves [17]. According to [18]-[19] path planning method is divided into traditional and heuristic methods. The traditional methods such as the Dijkstra [20], A* algorithm [21], and D* algorithm [22]. The heuristics such as genetic algorithm (GA) [23]-[24], ant colony optimization (ACO) [25], particle swarm optimization (PSO) [26]-[27], artificial potential field [28], and fuzzy logic [29].

Each path-planning approach has advantages and limitations. Table 1 shows the advantages and limitations of some path-planning approaches based on [30]. Each method has pros and cons. PSO is chosen in this study since it is simple, offers good performance, and fast convergence. Moreover, Gao *et al.* said that PSO is a widely used method in both theoretical and practical applications due to its strong searchability, fast convergence, and high efficiency [31]. Furthermore, the use of PSO in robotic path planning can give a high degree of precision at a rapid rate [32].

Table 1. Advantages and limitations of some path planning algorithms

Algorithms	Advantages	Limitations
Dijkstra	Easy to perform, Excellent performance	Do not consider the state of the path
A*	Simple, Low-cost, Efficient	Performance degrades in dynamic conditions
D*	The quickest route to an object	Excessively calculated when the map's grid number is too high
GA	Optimized route	High computation time
ACO	Good robustness, Fast convergence	Performs badly in vast search areas
PSO	Good robustness, Simple, Fast convergence	Poor local search capability

Planning a path for a swarm of robots is a problem that is still difficult because, in the same environment, robots must be able to work together to form a path. Each robot must be able to communicate in a swarm to provide information to each other when it finds obstacles and looks for other paths that are considered more optimal, in this case, the shortest distance and short time. One of the population-based search algorithms applied to the optimization function is particle swarm optimization (PSO). PSO was originally developed by Kennedy in 1995, inspired by the behavior of flocks of birds foraging in unknown locations [33]. Research by Nursena Baygin *et al.* in 2018 designed a robot so that it could reach the target with the shortest path and in the shortest time [34]. Xiangjun Liu *et al.* used the PSO method on a multi-cruise missile to avoid collisions, by updating new paths until finding the shortest one [35]. Bin Zhang *et al.* used the PSO method in optimization by updating the position and velocity of each particle [36]. Xu *et al.* used the PSO method for logistics robot navigation, by obtaining short path results thereby minimizing robot performance [37].

This research proposes path planning by creating a path planning algorithm for a mobile robot herd which is validated through simulation using MATLAB software to minimize costs and reach target points without hitting obstacles in a dynamic environment. The proposed research designs a path-planning simulation with obstacles in the form of circles and adds other shapes such as squares, rectangles, and triangles. The results of this research design are obtaining data related to the position, speed, cost, and best values of particles in a swarm of mobile robots, as well as obtaining global best values as the best position values among particles in a swarm of robots as in [38]. The contributions of this research include:

1) Customized Path Planning Algorithm: Development of a tailored path planning algorithm optimized through PSO to enable mobile robot herds to navigate efficiently amidst dynamic environments.
2) Validation through Simulation: Validation of the proposed algorithm through comprehensive simulations using MATLAB software, demonstrating its efficacy in minimizing costs and achieving obstacle-free navigation.

This paper consists of four parts. Section 2 discusses the research methodology. Section 3 discusses the results of research simulations using the particle swarm optimization method with the help of MATLAB simulation software. Meanwhile, section 4 is the conclusion regarding the research that has been carried out.

2. METHODS

Planning a path for a swarm of robots is still a difficult problem because, in the same environment, robots must be able to work together with each other [39]. Apart from that, each robot must be able to communicate to provide information to each other when it encounters obstacles along the path it passes. The expected route planning is safe, avoids obstacles, and has a short travel time. The path planning algorithm is used to plan a path so that the mobile robot can follow, which moves from the initial position to the target position. Path planning is used to help robots find the shortest or optimal path. Fig. 1 shows the path planning function.

The research method used is PSO to optimize the path planning. The PSO method is based on the behavior of flocks of animals such as birds, ants, termites, and bees by imitating the social behavior of these organisms [40]. Each particle behaves in a connected manner using its intelligence and is also influenced by its collective group. Each particle moves in a certain space and then has the best position that it has ever passed or found. Each particle will convey information from its best position to other particles in the swarm so that they can adjust their respective positions and speeds based on the information they have obtained. Fig. 2 illustrates the particle swarm optimization method.

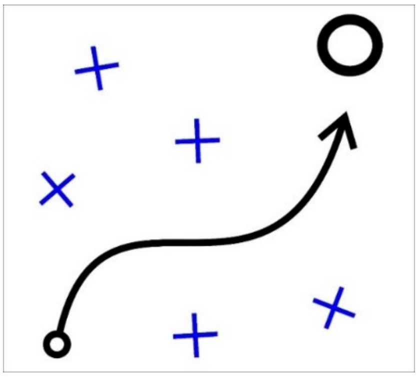

Fig. 1. Path Planning

The formula used to update the speed and position are presented in (1) and (2), respectively [41]. Where k = 1,2, …, K, and i = 1,2, …, is the population size and iteration number, w is inertia weight, r_1, and r_2 are random values between 0-1, c_1 is a local learning factor, c_2 is a global learning factor. Commonly, c_2 is bigger than c_1. Velocity updates are carried out to determine the direction of movement of particle positions in a population. The speed limit used is based on the maximum value of the particle position. After calculating the speed, the position is updated to obtain the fitness value. If the fitness value of each particle is better than the personal best value, then the personal best is used as the current position. The fitness value will be compared with the global best value. If the fitness value is better, then it will update the global best value. This process will repeat itself until it stops when the desired results are obtained or the maximum iteration has been reached. The data obtained includes position, speed, cost, and personal best for each particle according to the existing population size. The value of each personal best is used to obtain the global best value as the best position of all particles in the population.

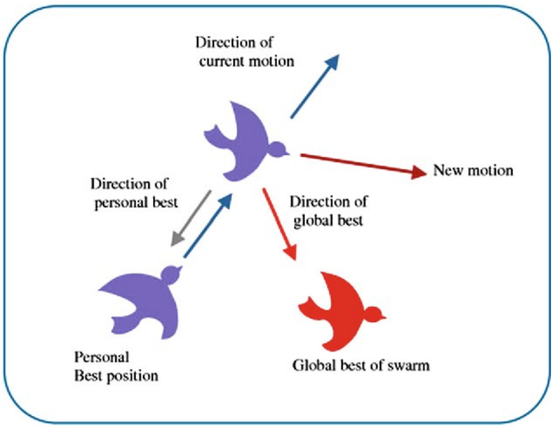

Fig. 2. Particle Swarm Optimization

$$v_i{}^{t+1} = wv_i{}^t + c_1r_1(xBest_i^t - x_i^t) + c_2r_2(gBest_i^t - x_i^t) \quad (1)$$

$$x_i{}^{t+1} = x_i{}^t + v_i{}^t \quad (2)$$

Fig. 3 shows the flow of the research methods carried out. The research began by initializing the parameters used, such as population size, size, and position of obstacles, up to the maximum number of iterations used. The values of the parameters are determined from the start. In the population, apart from the initial position and velocity of the particles, the parameters used are the inertial load (w) which functions to change the previous position to a new position for each particle. Apart from that, we use acceleration coefficients c_1 and c_2 with positive values ranging from 0-4, as well as random values r_1 and r_2 with values ranging between 0-1. Then initialize the initial position and velocity values of the particles randomly. Then calculate the suitability value (fitness function). The fitness function is a suitability value that functions to evaluate how close the solution obtained is to the optimal solution of a problem. After that, it updates the speed and position values of each particle. When each particle moves to a new position from its initial position, it will be marked as a personal best [42].

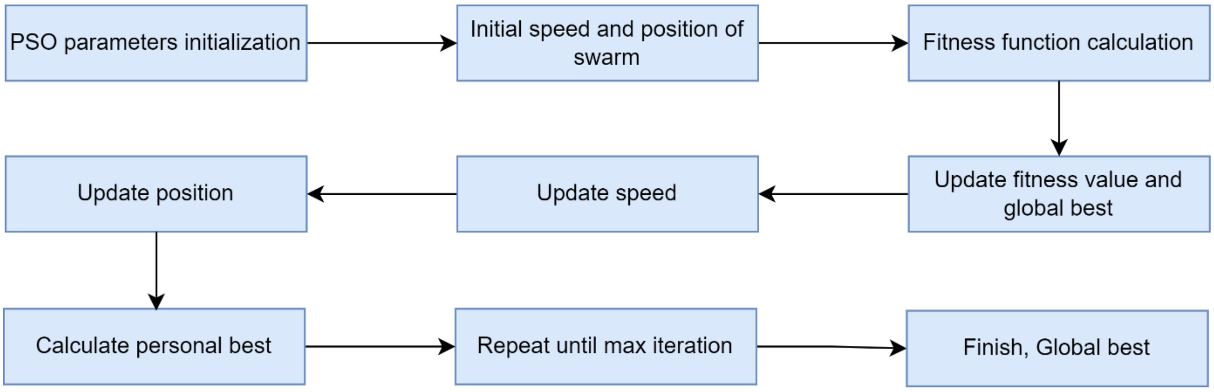

Fig. 3. Research flow process

The population used in this research is 150, so it will produce 150 data related to the position, speed, cost, and personal best by repeating the loop (iteration) 150 times. The iteration will stop when it has reached the maximum iteration, or the value has reached convergence. Other parameters used are w inertia weight of 1, *w-damp* (inertia weight damping) of 0.98, and acceleration coefficient also called learning rates (c_1 and c_2) of 1.5 each. The fitness function formula used in this research is presented in (3) where z, is the fitness function, d is the diameter of the obstacle, v is a violation, when the object hits an obstacle, and L is the length, to determine the length/shortness of the path taken, and β, namely the weighting value for the violation, in this study we set as 100.

$$z = L * (1 + \beta * v) \tag{3}$$

$$L = sum\left(\sqrt{(dx^\wedge 2 + dy^\wedge 2)}\right) \tag{4}$$

$$v = max\left(1 - \frac{d}{r^2}, 0\right) \tag{5}$$

$$d = \sqrt{(x - x_0)^2 (y - y_0)^2} \tag{6}$$

After creating the algorithm using the PSO method, then create the program code using MATLAB software. The simulation is created by initializing the number of particles in the swarm and determining the number of obstacles used. In this research, simulation is needed to show the algorithm's ability to replan a path when it encounters an obstacle from the original path. This simulation simulates how a mobile robot moves following a set of waypoints from the starting point to the target point.

3. RESULTS AND DISCUSSION

Each mobile robot as an object uses the fitness function value to determine the appropriate path and to calculate the next position. After that, the speed and position values of each mobile robot particle are obtained. Each mobile robot will have a personal best value from the fitness function calculation which consists of position and speed values as used by [43]. This process will repeat itself until it stops when the desired results are obtained, or the maximum iteration has been reached.

Based on Fig. 4, the starting point is shown by a red box, while the target point is shown by a green star. This image is the result of a simulation of a population of 150 mobile robots moving from the starting point to the target point. In this task, five handle points are used to navigate the mobile robot from the starting to the end position; hence, in Fig. 4 there are five small circles in the path. From the picture, the object can find a path to reach its goal by avoiding existing obstacles. The shape of the obstacles consists of circles, squares, rectangles, and triangles with various colors. The starting point used as the initial position of the object is (0,0) with the target point being the object's destination (4,3). Using MATLAB software, detailed data is obtained on both speed and position values for the x-axis and y-axis, as well as cost values. The best cost value for the 150 mobile robots is also displayed. Cost shows the value obtained from the fitness function calculation, while best cost shows the personal best value. Then a graphic plot, Fig. 5, is obtained showing the iterations on the x-axis and the best cost value on the y-axis with the following blue line.

When each particle moves from the initial position to the target position, it will be marked as a personal best. Personal best is very helpful in finding optimal solutions. Each particle moves randomly and can remember the best position which will later share information and can survive from generation to generation (iteration). The simulation results also display detailed position, speed, and best cost values for a population of 150 mobile robots. Personal best is the best value that every mobile robot has to date. Meanwhile, the global best value is the best value that can be achieved by all mobile robots. Personal best updates are carried out by comparing the previous personal best with the results of the current position update. Apart from that, by comparing the fitness values of the values that have been obtained, the lowest value will be the latest personal best. The newest personal best with the lowest fitness will become the global best.

The global best value shows the best position value and speed value from a population of 150 mobile robots. Fig. 5 informs that the fitness value convergence at the 83rd iteration, from the total of 150 iterations, with the cost function value of 5.1017. The detailed position and velocity are shown in Table 2. This value was obtained from the 45th population. There are five values for both the x and y axis and both position and velocity which indicate the handle points number one to five.

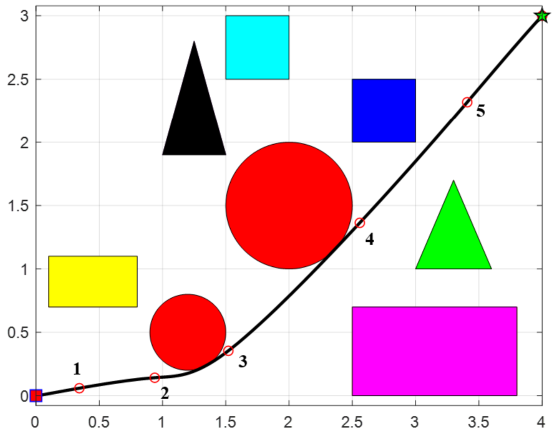

Fig. 4. Path planning simulation result

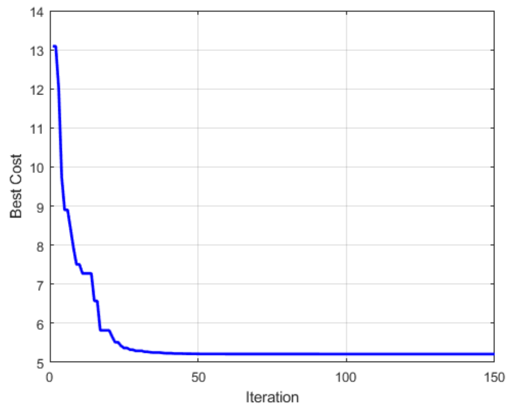

Fig 5. PSO fitness value

Table 2. Global Best

Position	Velocity	*Cost*
x: [0.5848 1.0030 1.4960 1.7429 2.5135]	x: [-9.3145e-^{07} -1.8658e-^{06} -4.0577e-^{04} 2.4113e-^{05} 1.4848e-^{06}]	5.1017
y: [0.6594 1.1295 1.6727 1.9288 2.3097]	y: [1.5766e-^{06} -4.5241e-^{05} 7.7371e-^{07} 1.2252e-^{07} 7.8556e-^{08}]	

4. CONCLUSION

In conclusion, the utilization of particle swarm optimization (PSO) for path planning in mobile robot swarms has been demonstrated as an effective approach for navigating dynamic environments while minimizing path length. Through simulation validation using MATLAB, our study successfully guided a swarm of 150 mobile robots from the starting point (0.0) to the target point (4.3) without encountering obstacles. The PSO algorithm facilitated the identification of the optimal path, resulting in the shortest distance traveled, with a minimum cost value of 5.1017 achieved in the 83rd iteration. Factors such as varying terrain complexity, sensor limitations, and real-world uncertainties may impact the performance of PSO-based path-planning algorithms and warrant further investigation. For future work, researchers are encouraged to explore the adaptability and robustness of PSO-based path-planning methods across diverse scenarios. Investigating the scalability of the algorithm for larger robot swarms, incorporating real-time sensor data for dynamic obstacle avoidance, and enhancing computational efficiency are promising avenues for advancement in this field.

REFERENCES

[1] H. M. Gross, A. Scheidig, S. Müller, B. Schütz, C. Fricke, and S. Meyer, "Living with a mobile companion robot in your own apartment - Final implementation and results of a 20-week field study with 20 seniors," *in Proc. of IEEE Int. Conf. Robot. Autom*., pp. 2253–2259, 2019, https://doi.org/10.1109/ICRA.2019.8793693.

[2] C. J. Lai and C. P. Tsai, "Design of introducing service robot into catering services," in *Proc. of ACM Int. Conf. Proceeding Ser*. pp. 62–66, 2018, https://doi.org/10.1145/3208833.3208837.

[3] S. Meghana, T. V. Nikhil, R. Murali, S. Sanjana, R. Vidhya, and K. J. Mohammed, "Design and implementation of surveillance robot for outdoor security," in *Proc. of IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol. Proc.*, pp. 1679–1682, 2017, https://doi.org/10.1109/RTEICT.2017.8256885.

[4] W. Rahmaniar and A. E. Rakhmania, "Online digital image stabilization for an unmanned aerial vehicle (UAV)," *J. Robot. Control*, vol. 2, no. 4, pp. 234–239, 2021, https://doi.org/10.18196/jrc.2484.

[5] M. S. Kaiser, S. Al Mamun, M. Mahmud, and M. H. Tania, "Healthcare robots to combat COVID-19," *COVID-19: Prediction, decision-making, and its impacts*, pp.83-97, 2021, https://doi.org/10.1007/978-981-15-9682-7_10.

[6] C.C. Chang, Y.-H. Juan, C.L. Huang, and H.J. Chen, "Scenario Analysis for Road Following Using JetBot," in *IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE),* pp. 403–406, 2020, https://doi.org/10.1109/ECICE50847.2020.9302017.

[7] M. Auzan, R. M. Hujja, M. R. Fuadin, and D. Lelono, "Path Tracking and Position Control of Nonholonomic Differential Drive Wheeled Mobile Robot," *J. Ilm. Tek. Elektro Komput. dan Inform*., vol. 7, no. 3, p. 368, 2021, https://doi.org/10.26555/jiteki.v7i3.21017.

[8] M. Wahdan, "Motion planning for autonomous mobile robots," *AEJ - Alexandria Eng. J.,* vol. 44, no. 1, pp. 51–57, 2005.

[9] C. Zheng, R. Green, "Vision-based autonomous navigation in indoor environments," in *Proceedings of the 2010 25th International Conference of Image and Vision Computing,* pp. 1-7, 2010, https://doi.org/10.1109/IVCNZ.2010.6148876.

[10] N. Sariff, N. Buniyamin, "Ant colony system for robot path planning in a global static environment," in *Proceedings of the 9th WSEAS International Conference on System Science and Simulation in Engineering (ICOSSSE'10),* pp. 192–197, 2010, https://dl.acm.org/doi/10.5555/1938907.1938939.

[11] E. A. Oyekanlu et. al., "A review of recent advances in automated guided vehicle technologies: Integration challenges and research areas for 5G-based smart manufacturing applications," *IEEE Access*, vol. 8, pp. 202312–202353, 2020, https://doi.org/10.1109/ACCESS.2020.3035729.

[12] T. Mac, C. Copot, D. Tran, R. Keyser, "Heuristic approaches in robot path planning: A survey," *Robot. Auton. Syst.,* vol. 86, pp. 13-28, 2016, https://doi.org/10.1016/j.robot.2016.08.001.

[13] S. Nurmaini and B. Tutuko, "Intelligent robotics navigation system: Problems, methods, and algorithm," *Int. J. Electr. Comput. Eng*., vol. 7, no. 6, pp. 3711–3726, 2017, https://doi.org/10.11591/ijece.v7i6.pp3711-3726.

[14] H. Y. Zhang, W. M. Lin, and A. X. Chen, "Path planning for the mobile robot: A review," *Symmetry (Basel).,* vol. 10, no. 10, 2018, https://doi.org/10.3390/sym10100450.

[15] K. Karur, N. Sharma, C. Dharmatti, and J. E. Siegel, "A Survey of Path Planning Algorithms for Mobile Robots," *Vehicles*, vol. 3, no. 3, pp. 448–468, 2021, https://doi.org/10.3390/vehicles3030027.

[16] I. Hassani, I. Ergui and C. Rekik, "Turning Point and Free Segments Strategies for Navigation of Wheeled Mobile Robot," *International Journal of Robotics and Control Systems*, vol. 2, no. 1, pp. 172-186, 2022, https://doi.org/10.31763/ijrcs.v2i1.586.

[17] X. Liang, C. Liu, X. Song, and C. Hao, "Research on mobile robot path planning in a dynamic environment", *Proceedings Chinese Automation Congress*, pp. 3890–3894, 2017, https://doi.org/10.1109/CAC.2017.8243459.

[18] X. Li, D. Wu, J. He, M. Bashir, and M. Liping, "An Improved Method of Particle Swarm Optimization for Path Planning of Mobile Robot," *J. Control Sci. Eng*., pp. 1-12, 2020, https://doi.org/10.1155/2020/3857894

[19] S. Campbell, N. O'Mahony, A. Carvalho, L. Krpalkova, D. Riordan, and J. Walsh, "Path Planning Techniques for Mobile Robots A Review," in *6th Int. Conf. Mechatronics Robot. Eng. ICMRE*, pp. 12–16, 2020, https://doi.org/10.1109/ICMRE49073.2020.9065187.

[20] A. Sedeño-noda and M. Colebrook, "A biobjective Dijkstra algorithm," *European Journal of Operational Research*, vol. 276, no. 1, pp. 106–118, 2019, https://doi.org/10.1016/j.ejor.2019.01.007.

[21] J. Votion and Y. Cao, "Diversity-based cooperative multi-vehicle path planning for risk management in cost map environments," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 8, pp. 6117–6127, 2019, https://doi.org/10.1109/TIE.2018.2874587.

[22] H. Shin and J. Chae, "A Performance Review of Collision-Free Path Planning Algorithms," *Electronics*, vol. 9, no. 2, p. 316, 2020, https://doi.org/10.3390/electronics9020316.

[23] J. W. Wu, D. Bin, X. B. Feng *et al.*, "GA based adaptive singularity-robust path planning of space robot for on-orbit detection," *Complexity*, 2018, https://doi.org/10.1155/2018/3702916.

[24] W. Rahmaniar and A.E. Rakhmania, "Mobile robot path planning in a trajectory with multiple obstacles using genetic algorithms," *Journal of Robotics and Control (JRC)*, vol. 3, no. 1, pp. 1-7, 2022, https://doi.org/10.18196/jrc.v3i1.11024.

[25] C. Xiong, D. Chen, D. Lu, Z. Zeng, and L. Lian, "Path planning of multiple autonomous marine vehicles for adaptive sampling using Voronoi-based ant colony optimization," *Robotics and Autonomous Systems*, vol. 115, pp. 90–103, 2019, https://doi.org/10.1016/j.robot.2019.02.002.

[26] L. Xun, D. Wu, Z. Zhao *et al.*, "Path planning method for indoor robot based on improved PSO," *Computer Measurement &Control*, vol. 28, no. 3, pp. 206–211, 2020.

[27] B. Al-Khlidi, A.T. Abdulsadda, and A. Al Bakri, "Optimal Robotic Path Planning Using Intelligent Search Algorithms", *Journal of Robotics and Control (JRC),* vol. 2, no. 6, pp. 519-526, 2021, https://doi.org/10.18196/26132.

[28] M.M.J. Samodro, R.D. Puriyanto, and W. Caesarendra, "Artificial potential field path planning algorithm in differential drive mobile robot platform for the dynamic environment," *International Journal of Robotics and Control Systems*, vol. 3, no. 2, pp.161-170, 2023, https://doi.org/10.31763/ijrcs.v3i2.944.

[29] N. Rinanto, I. Marzuqi, A. Khumaidi, and S. T. Sarena, "Obstacle Avoidance using Fuzzy Logic Controller on Wheeled Soccer Robot," *J. Ilm. Tek. Elektro Komput. dan Inform*., vol. 5, no. 1, pp. 26–35, 2019, https://doi.org/10.26555/jiteki.v5i1.13298.

[30] N.S. Abu, W.M. Bukhari, M.H. Adli, S.N. Omar, and S.A. Sohaimeh, "A Comprehensive Overview of Classical and Modern Route Planning Algorithms for Self-Driving Mobile Robots," *Journal of Robotics and Control* (JRC), vol. 3, no. 5, pp. 666-678, 2022, https://doi.org/10.18196/jrc.v3i5.14683.

[31] T. T. Gao, L. Zhang, B. D. Li *et al.*, "Study on path planning of soccer robot based on improved particle swarm algorithm," *Journal of Xi'an Polytechnic University*, vol. 30, no. 5, pp. 609–615, 2016.

[32] C. S. Tan, R. Mohd-Mokhtar, and M. R. Arshad, "A Comprehensive Review of Coverage Path Planning in Robotics Using Classical and Heuristic Algorithms," *IEEE Access*, vol. 9, pp. 119310-119342, 2021, https://doi.org/10.1109/ACCESS.2021.3108177.

[33] H. Maghfiroh, O. Wahyunggoro, and A. Imam, "Novel iterative Ragone plot-based optimization of low pass filter for hybrid power sources electric vehicles," *e-Prime - Adv. Electr. Eng. Electron. Energy*, vol. 7, p. 100389, 2024, https://doi.org/10.1016/j.prime.2023.100389.

[34] N. Baygin, M. Baygin, and M. Karakose, "PSO Based Path Planning Approach for Multi-Service Robots in Dynamic Environments", in *International Conference on Artificial Intelligence and Data Processing*, IDAP, pp. 6–10, 2019, https://doi.org/10.1109/IDAP.2018.8620801.

[35] X. Liu, Q. Gu, and C. Yang, "Path planning of multi-cruise missile based on particle swarm optimization," in *Proceedings International Conference on Sensing, Diagnostics, Prognostics, and Control*, SDPC, pp. 910–912, 2019, https://doi.org/10.1109/SDPC.2019.00173.

[36] X. Yang, M. Jiang, X. Hu, and Z. Zhu, "Particle swarm optimization algorithm based on beetle antenna search optimization," *Journal of Physics: Conference Series*, vol. 1550, no. 2, pp. 1586–1589, 2020, https://doi.org/10.1088/1742-6596/1550/2/022034.

[37] R. Xu, Z. Qi, and G. Fen, "A Path Planning Method of Logistics Robot Based on Improved Ant Colony Algorithm," in *Proceedings International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering*, ICBAIE, pp. 450–453, 2020, https://doi.org/10.1109/ICBAIE49996.2020.00101.

[38] C.H. Chen, S.N. Yeh, "Personal best-oriented constriction type particle swarm optimization," in *IEEE Conference on Cybernetics and Intelligent Systems*, pp. 9–12, 2006, https://doi.org/10.1109/ICCIS.2006.252300.

[39] A. Ayari and S. Bouamama, "A new multi-robot path planning algorithm: Dynamic distributed particle swarm optimization," in *IEEE International Conference on Real-Time Computing and Robotics*, RCAR, pp. 437–442., 2018, https://doi.org/10.1109/RCAR.2017.8311901.

[40] S. Praptodiyono, H. Maghfiroh, and C. Hermanu, "BLDC Motor Control Optimization Using Optimal Adaptive PI Algorithm," *J. Elektron. dan Telekomun*., vol. 20, no. 2, p. 47, 2020, https://doi.org/10.14203/jet.v20.47-52.

[41] A. Ma'Arif, H. Nabila, I. Iswanto, and O. Wahyunggoro, "Application of Intelligent Search Algorithms in Proportional-Integral-Derivative Control of Direct-Current Motor System," *J. Phys. Conf. Ser*., vol. 1373, no. 1, 2019, https://doi.org/10.1088/1742-6596/1373/1/012039.

[42] S. Chen, I. Abdulselam, N. Yadollahpour, and Y. Gonzalez-Fernandez, "Particle Swarm Optimization with best Perturbations," in *IEEE Congress on Evolutionary Computation*, CEC, 2020, https://doi.org/10.1109/CEC48606.2020.9185801.

[43] M. Adharsh, A.G. Iyer, A. G. Athrey, V. Ashwin, and V.G. Venkatesh, "Particle Swarm Optimization Algorithm for finding the brightest spot in an arena," in *Proceedings of the 3rd International Conference on Electronics and Communication and Aerospace Technology*, ICECA, 484–489, 2019, https://doi.org/10.1109/ICECA.2019.8822219.