Waypoint-Based Path Planning for Autonomous Robots with PSO

Gustavo Manhenti Faustino
Center of Mathematics, Computing and Cognition
Universidade Federal do ABC
Santo Andre, Brazil
g.faustino@aluno.ufabc.edu.br

Mateus Coelho Silva
Center of Mathematics, Computing and Cognition
Universidade Federal do ABC
Santo Andre, Brazil
mateuscoelho.ccom@gmail.com

***Abstract*—Autonomous robots are essential for numerous applications requiring high precision and involving significant risks. However, planning and optimizing paths for a single robot in a dynamic environment remains a highly complex task. This challenge necessitates a balance between obstacle avoidance, distance optimization, and efficiency. Traditional strategies often utilize a wide range of path planning algorithms, such as local avoidance, A*, and ACO. Nevertheless, an approach employing waypoints and a "danger zone" in conjunction with a Particle Swarm Optimization (PSO) algorithm can be an effective method for path optimization. Here, we demonstrate that this path planning algorithm provides an efficient and rapid means to generate a path to a goal in an environment with a single obstacle. Through simulations conducted in IR-SIM, a Python-based robotics simulator, we show that a customized PSO, equipped with a multi-objective fitness function, effectively analyzes both intersections with an area around the obstacle (the "danger zone") and the total path distance. This approach yields paths with over a 97% success rate in guiding the robot to its goal and exhibits relatively low convergence times. Our results illustrate PSO's effectiveness as a path planning algorithm, highlighting its adaptability to various types of obstacles and positions within a 2D environment. This strategy represents an advancement in the use of heuristic algorithms for autonomous robot path planning, leading to a faster, less computationally demanding algorithm with a high success rate, capable of avoiding diverse obstacle types and easily adaptable to a broad range of single-robot environmental problems.**

***Index Terms*—autonomous robotics, PSO, path optimization, routing, path planning**

I. Introduction

Autonomous robots offer a wide range of applications in many areas of society, principally those that need the execution of tasks that require high precision in a high-risk environment, such as surgery [1], transport in traffic areas [2], and rescue [3]. An efficient method for robot control is indispensable to ensure that the objective is achieved while maintaining obstacle avoidance.

Optimizing the planning and routing of the path for an autonomous robot is a highly complex task that demands the integration of capabilities such as obstacle avoidance, collision prevention, and accurate goal reaching, while aiming to maintain the shortest route and minimal computational cost.

Consequently, numerous path planning algorithms have been proposed to generate paths and identify optimal solutions for robot navigation. Fu et al. [4], in their work, present a review of some path planning algorithms. Traditional algorithms like A*, Rapidly-exploring Random Tree (RRT), and Ant Colony Optimization (ACO) have been widely applied, but can present challenges such as high computational cost, sub-optimal paths, or slow convergence, respectively.

In this work, we used the Particle Swarm Optimization (PSO) algorithm, a centralized metaheuristic, which emerges as a promising solution. Unlike many other optimization strategies, the PSO is capable of solving some complex problems and can be easily modified for diverse environments through simple alterations to its primary function and parameters. One of the limitations of the PSO is its typically high computational cost, which we attempted to mitigate using our strategy [5].

In this context, the proposed work uses PSO to optimize the positions of intermediary points (waypoints), which form a path of sections between the start and goal points. For this, we employ a modified PSO that uses a position vector of the waypoints as the particles, optimizing all waypoints simultaneously to maximize convergence speed. These particles are evaluated by a multi-objective fitness function that checks not only the path's length but also if the lines between the points pass through a "dangerous zone" around the obstacle, to discard any path that does not avoid the obstacle. This paper aims to prove the efficiency and low computational power consumption of the proposed algorithm. Figure 1 displays a path generated by the modified PSO algorithm, demonstrating that the result is a short path that avoids an obstacle.

II. Theoretical References and Related Work

This section provides a theoretical explanation of the path planning and optimization as well as the application of PSO, in the autonomous robotics.

A. Path Planning

Path planning algorithms aim to determine an optimal, collision-free route from a start to a goal point [6]. While established strategies like Probabilistic Roadmaps (PRM) [7] and Artificial Potential Fields (APF) [8] are effective, they often focus on metrics like path quality or avoiding local minima. In contrast, this work employs Particle Swarm Optimization (PSO) to optimize waypoints using a "danger zone" concept

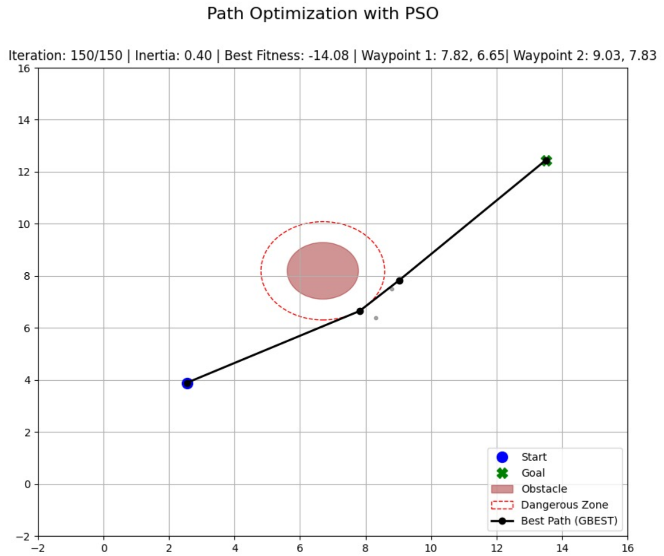

Fig. 1. A path generated and optimized by the proposed algorithm and strategy.

for obstacle avoidance. Our primary evaluation focuses on demonstrating high efficiency and rapid convergence, offering a distinct analysis centered on computational cost-effectiveness and success rate.

B. Particle Swarm Optimization Algorithm

Particle Swarm Optimization (PSO) is a swarm intelligence metaheuristic inspired by flocking behavior, where particles (candidate solutions) iteratively adjust their position based on their own best-known position and the swarm's global best [5]. Each particle's velocity and position are updated according to Equations 2 and 1, respectively.

PSO has been widely applied to path planning, often integrated into more complex frameworks. For instance, prior work has combined it with multi-objective functions and probabilistic roadmaps to find the shortest and smoothest paths [9] or has used multimodal strategies with Bézier curves for path smoothing [10]. Our approach, however, utilizes PSO in a more direct manner. By focusing solely on optimizing waypoint coordinates, we test the algorithm's raw performance in terms of convergence speed and success rate, demonstrating its effectiveness as a simpler, yet robust, solution for single-obstacle environments.

$$V_{i+1} = \omega \cdot V_i + c_1 r_1 (PBEST - X_i) + c_2 r_2 (GBEST - X_i) \tag{1}$$

$$X_{i+1} = X_i + V_{i+1} \tag{2}$$

III. Background and Methodology

In the previous sections, we discussed the importance and key concepts related to path planning within the context of this study. This section will delve into the path planning strategy, the applied PSO algorithm, the simulation environment, and the evaluation criteria for results.

A. Path Planning Strategy

The scenario is a 2D environment with a static obstacle and a goal. Our solution generates two waypoints between the start and goal to ensure obstacle avoidance. These two intermediate points are optimized by a PSO algorithm that aims to reduce the total distance and altogether avoid obstacle collision. The robot's behavior is a simple differential movement with a logic of going directly to the next "goal point" in a straight line.

This strategy is intended to be applied in different variations of the environment, generating the path for any format, size, or position of obstacle. For this purpose, it uses a "safe and danger zone" approach for obstacle recognition that guarantees avoidance and efficient paths, as shown below. This approach can be seen in Figure 1.

B. Implementing PSO in the Strategy

For this application of PSO, each particle is represented by a vector $P_i = (X_{i1}, Y_{i1}, X_{i2}, Y_{i2})$, where $(X_{i1}, Y_{i1}, ...)$ is the position of the first way-point and $(..., X_{i2}, Y_{i2})$the position of the second way-point. Correspondingly, GBEST and PBEST store the optimal positions found by the swarm and by individual particles and are represented by a similar vector. The position of the robot is a simple two-dimensional position vector, $R = (X_r, Y_r)$, as is the goal's position.

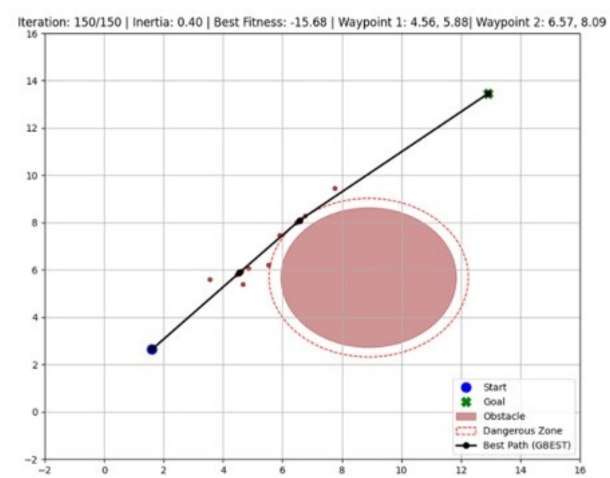

Fig. 2. Visual representation of the process of convergence of PSO.

To evaluate each particle's solution, we used a fitness function, shown in Equation 3. This function measures the total path length and determines if the planned path passes through a circular space surrounding the obstacle, represented for Ω, thereby delimiting a 'safe zone' from an unsafe one, a 'danger zone'. The total path length, L, is calculated using the euclidean distance, between all the four points of a complete path, and is represented by the Equation 4, where W_1 is the position of the first way-point, S is the start, G is the goal and W_2 is the second way-point.

$$
\begin{cases}
Fitness = -L + k1 + k2 + k3 \\
k1 = -999, \qquad \overline{SW_1} \quad \cap \quad \Omega + r \\
k2 = -999, \qquad \overline{W_2 W_1} \quad \cap \quad \Omega + r \\
k3 = -999, \qquad \overline{W_2 G} \quad \cap \quad \Omega + r
\end{cases} \tag{3}
$$

$$
L = ||W_1 - S|| + ||W_2 - W_1|| + ||G - W_2|| \tag{4}
$$

To determine if the path passes through the “danger zone” of obstacles of any format, we calculate the center point of each obstacle and create a circle around it. The radius of this circle is the largest distance from the obstacle’s limit to its center, plus a security factor r based on the robot’s size. Using this unsafe area, the fitness value undergoes a significant decrease if the path passes through an obstacle, ensuring maximum avoidance. The entire fitness function is shown in Equation 4.

A demonstration of the PSO’s convergence to its global optimum is shown in Figure 2, which displays the algorithm’s progress with 150 iteratinons. It’s possible to see that as the number of iterations increases, the particles converge more towards the “GBEST” position. This convergence can be observed in how the particles begin to occupy the same positions at the maximum iterations.

C. Simulations using IR-SIM Software

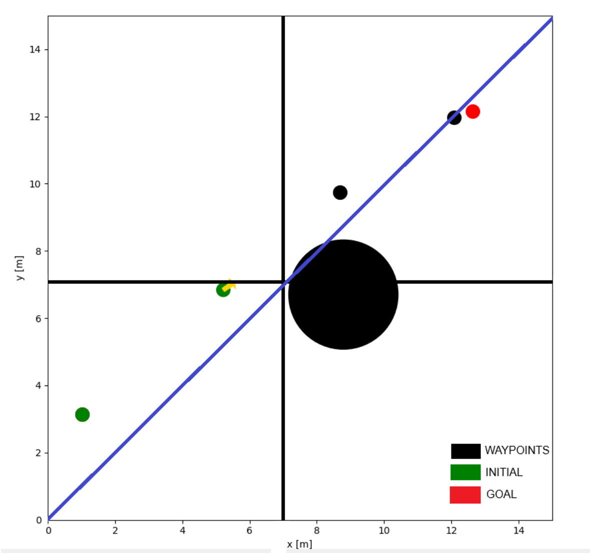

Fig. 3. Example of the generation of objects and points in the environment.

The application of PSO needs the initialization of a series of parameters to be applied in the Equation 3. The values utilized in tests are shown in the Table I.

The simulations are conducted with the randomness already cited and include both circular and rectangular obstacles that vary in size. We performed 40 simulations for each obstacle format. In every 10 simulations, the start point and the goal swap the opposite quadrants they occupy. This series of simulations was conducted to verify the relationship between

TABLE I
PARAMETERS USED IN TESTS

Iterations	150
r	0.8
c_1	1.5
c_2	2.5
Max Inertia	0.9
Minimum Inertia	0.4

the number of particles, convergence time, and the success of the algorithm by creating paths for 20, 10, and 1 particle. The hardware utilized is a Windows 11 notebook, with an Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz and 8GB of RAM. An example of 4 types of simulations is shown in Figure 4.

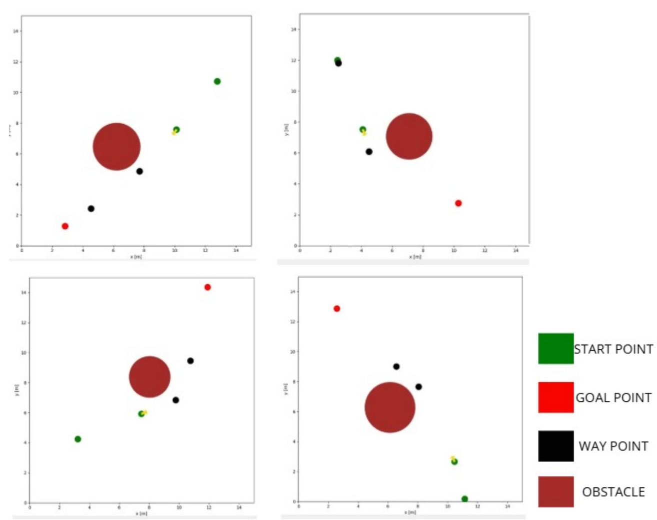

Fig. 4. 4 Simulations with Starts and Goals in opposite quadrants.

D. Evaluation of Results

Finally, for the analysis of the test results, the number of successful scenarios and the average convergence time of the algorithm in the simulations for each format of obstacle will be considered. These tests aim to demonstrate the efficiency, adaptability, and cost-effectiveness of the strategy and the proposed PSO algorithm.

IV. RESULTS

This section presents the results from 240 simulated scenarios conducted in IR-SIM, following the previously described methodology.

The graphic plots in Figure 5 demonstrate the percentage of times that the algorithm made a path guiding the robot to the obstacle without crashing into it. The results show a similarly high success rate for simulations with 10 and 20 particles across both obstacle formats, staying in over 97%, and a medium-to-low rate for tests with an individual particle, with around 60% average success rate. This result indicates that a reasonable range for the number of particles to maximize the success of the PSO is between 10 and 20, and the use of only

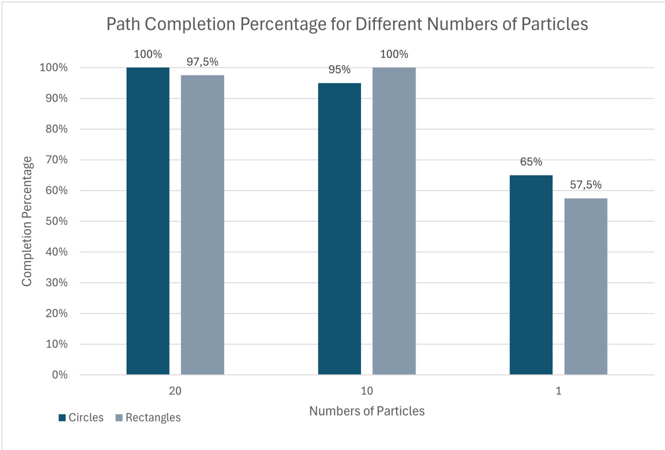

Fig. 5. Graphic of the Success Rate in Relation with the Number fo Particles.

one particle is entirely outside the expected behavior of the algorithm. It is important to note that the paths created with 10 particles exhibited fitness values significantly lower than those created with 20 particles, suggesting a loss of optimization. Furthermore, the results indicate that agent numbers around 20 are optimal for this application.

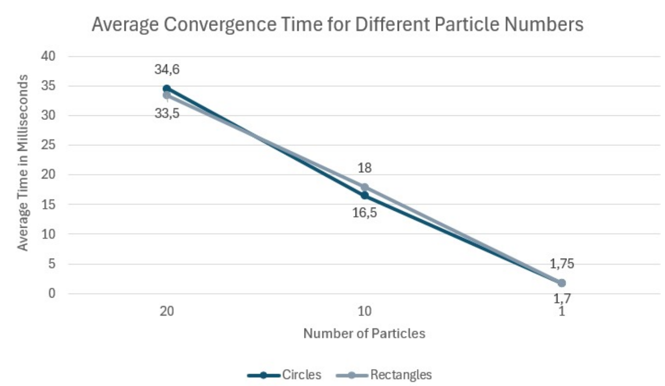

Fig. 6. Linear graphic of convergence time for different number of particles, values in milliseconds (mS).

The linear graphic in Figure 6 represents the algorithm's convergence time, which is a good indicator of the processing power it requires. The results show that the number of particles and the PSO's convergence time are directly related, and that the most efficient number of particles stays around 10 particles, with less time to converge and an acceptable success rate. This condition is primarily caused by the simulator's operation and the application of the PSO logic, which cannot process more than one particle simultaneously, leading to an increase in the number of loops in the main logic.

These results show that the proposed strategy is a robust and low-cost way to generate paths, avoiding a single obstacle. The "danger zone" approach successfully guarantees that the optimization avoids any obstacle format or position. The use of two waypoints enabled the PSO to encounter secure paths even when the obstacle was between the robot and the goal, and the delimitation of a non-dangerous area lowered the risks of crashes if the robot had some failure, showing its application for real scenarios. For future works, the tests can be made with more static obstacles, and with more formats, to show even more the adaptability of the strategy.

V. Conclusion and Future Works

In this paper, a path planning and optimization strategy for autonomous robots is proposed. We use Particle Swarm Optimization (PSO) to generate intermediary points between the start and the goal of the robot, called waypoints, and a "danger zone" limitation to avoid obstacles securely.

The proposed variation of PSO uses a vector of n Euclidean coordinates for each particle, simultaneously processing all waypoints to avoid a large consumption of power. Meanwhile, the "danger zone" and the fitness function, which rewards suitable solutions, prevent paths from passing through the obstacle. The proposed method demonstrated efficient avoidance for both obstacle formats, achieving a success rate of over 97%. The fast convergence times confirm the algorithm is computationally efficient.

Acknowledgment

The authors would like to acknowledge CAPES, CNPq, the Federal University of ABC and Lakehead University, Canada, for supporting this research. This work was developed at the laboratory of artificial intelligence, robotics and algorithms (IARA++), at the Federal University of ABC, Santo André, Brazil.

References

[1] Y. Rivero-Moreno, M. Rodriguez, P. Losada-Muñoz, S. Redden, S. Lopez-Lezama, A. Vidal-Gallardo, D. Machado-Paled, J. C. Guilarte, S. Teran-Quintero, Y. Rivero *et al.*, "Autonomous robotic surgery: has the future arrived?" *Cureus*, vol. 16, no. 1, 2024.

[2] Y. Lu, "Autonomous robots for transport applications." *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 7, 2024.

[3] S. S. Alam, T. Ahmed, M. S. Islam, and M. M. F. Chowdhury, "A smart approach for human rescue and environment monitoring autonomous robot," *International Journal of Mechanical Engineering and Robotics Research*, vol. 10, no. 4, pp. 209–215, 2021.

[4] S. Fu, D. Yang, Z. Mei, and W. Zheng, "Progress in construction robot path-planning algorithms," *Applied Sciences*, vol. 15, no. 3, p. 1165, 2025.

[5] M. Jain, V. Saihjpal, N. Singh, and S. B. Singh, "An overview of variants and advancements of pso algorithm," *Applied Sciences*, vol. 12, no. 17, p. 8392, 2022.

[6] H. Qin, S. Shao, T. Wang, X. Yu, Y. Jiang, and Z. Cao, "Review of autonomous path planning algorithms for mobile robots," *Drones*, vol. 7, no. 3, p. 211, 2023.

[7] S. Alarabi, C. Luo, and M. Santora, "A prm approach to path planning with obstacle avoidance of an autonomous robot," in *2022 8th international conference on automation, robotics and applications (ICARA)*. IEEE, 2022, pp. 76–80.

[8] T. A. Teli and M. A. Wani, "A fuzzy based local minima avoidance path planning in autonomous robots," *International Journal of Information Technology*, vol. 13, no. 1, pp. 33–40, 2021.

[9] B. Song, Z. Wang, and L. Zou, "On global smooth path planning for mobile robots using a novel multimodal delayed pso algorithm," *Cognitive Computation*, vol. 9, no. 1, pp. 5–17, 2017.

[10] E. Masehian and D. Sedighizadeh, "A multi-objective pso-based algorithm for robot path planning," in *2010 IEEE international conference on industrial technology*. IEEE, 2010, pp. 465–470.