Multi Objective Path Planning in Static Environment using Region of Sight

Hiba Hliwa[1], Bassam Atieh[2]
[1, 2] Mechatronics Department
[1]Tishreen University, [2] Manara University
[1, 2] Latakia, Syria
[1] hhelewa@gmail.com, [2] basa955@gmail.com

***Abstract*—Finding appropriate path, that avoids the hurdles and reach the goal in optimal way, has been an interesting challenge in recent years. In this paper, we present a new method for robot path planning in a static environment. The main aim; to tackle path planning in simple and fast method, seeks to satisfy many robot movement's requirements such as shortest path length, safety, smoothness with less time consuming. This method ensures that safety and smoothness criteria are already met, path length is then shortening the usage of environment details, where each point of this path can be seen only by its previous and next points. The obtained result is compared with the well-known methods and shows the superiority in simple and complex environments in term of time, smoothness and safety and the obtained path lengths are shorter than those obtained with PSO and ABC as well.**

Keywords—Robot Path Planning (RPP), Static Environment, Multi-Objective Optimization Algorithm (MOOA), Mobile Robot (MR), Region of Sight (RoS)

I. INTRODUCTION

Autonomous Navigation gives mobile robots the ability to help human in moving materials (in hospital, warehouses), or even replace him in dangerous or unreachable places for military and exploration purposes. This requires giving robot the ability to decide how to reach his desired goal efficiently.

The more freely robots intend to move, the smarter they must be. This includes their ability to generate feasible path between a start position and a goal position. Besides the obstacle avoidance, this path must also satisfy some other requirements or optimize certain performance criteria [1]. Hence the importance of path planning (PP) methods to determine the best movement to destination when working in a simple or complex environments [2].

Heuristic methods have been developed increasingly over the past two decades to cope with high computational costs and complexities of classic methods. When using heuristic algorithms, it is not guaranteed to find a solution, but if a solution is found, it will be done much faster than deterministic methods [3]. The main metaheuristic approaches employed in RPP are Simulated Annealing (SA)[4], Genetic Algorithms (GA)[5], and Tabu Search (TS) [8].

In most of the above-mentioned methods RPP problem has been dedicated to finding a feasible shortest path by formulating a single-objective optimization problem.

Finding the optimal path is never a single-objective problem, however it is essential to handle several objectives simultaneously with the aim of obtaining accurate solutions. To specify, three fundamental objectives can be taken into account: the path safety, the path length and the path smoothness [9]. Path safety is related to the minimal distance between path and nearby obstacles. The farther the path from the obstacle the safer it will be. The path length objective focus on obtaining paths as short as possible. This objective is mainly related to the robot operation time, since a shorter path will travel in less time. Finally, the path smoothness goal is to minimize the number of bends of the path. The path smoothness objective is related to the energy consumption, due to the influence of the number and magnitude of turns on the consumed energy [10].

In recent years, several studies proposed, multi objective optimization algorithms (MOOAs) are applied to tackle the PP problem. Davoodi et al. [11], Salmanpour et al.[12] and Elmi& Efe [13] have presented bi-objective optimization methods where the path length and the path safety are considered, and the path smoothness is ignored, (i.e. an objective related to the energy consumption). In the same manner, the optimization methods introduced by Ferariu and Cîmpanu [14], Zhu et al. [15] tackle the RPP problem through a bi-objective manner. The difference here is that these methods optimize the length and the smoothness of the path. Ahmed & Deb [10], Hidalgo-Paniagua et al. [9] and Thabit & Mohades [16] solve the RPP problem through the multi-objective optimization methods taking into account the three objectives: the path length, the path safety and the path smoothness. All these methods used optimization algorithms to find the optimal path, but Adding more than one objective to an optimization problem may cause two objectives conflict [17]. This increases the complexity and algorithm's execution time.

In this paper, we tackle the PP problem by using Region of Sight (RoS), which is based on environment guided techniques. The proposed method aims to minimize path desired parameters as possible, while preserving algorithm's execution time in its minimal level.

The remaining part of paper is organized as follows. Section II represents Multi-Objective path planning using RoS. Results and the conclusion are presented in Section III and Section IV, respectively.

978-1-7281-5655-2/20/$31.00 ©2020 IEEE

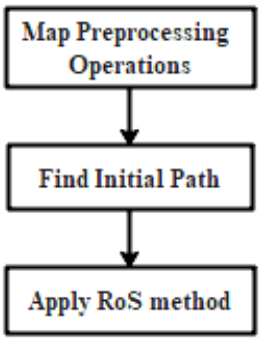

Fig. 1 proposed method's levels.

II. Multi-Objective Path Planning Using Region of Sight

This section introduces the proposed method, which is divided into two levels (as described in Fig .1), While the first level applies preprocessing operation on environment map image to assure safety and smoothness, the second level shortens the path as the processed environment allows.

A. Environment's Map Preprocessing Operations

Since almost all the shortest paths touch obstacle borders, especially at the edges, resulting unsafe and nonsmoothed paths, this paper assumes that expanding all obstacle boundaries in the environment map and smoothing their edges could satisfy safety and smoothness objectives without any further procedures.

To ensure that no collision between robot and obstacles could happen, the robot rotation center (RC) should be located outside obstacle boundaries at any point of the path. Which means that if obstacle boundaries expand with the max length between robot RC and its borders r_{safety}, safety will be guarantied. This expansion is applied on environment map image after normalizing r_{safety} according to map dimension as follows,

$$R_s = \left\lceil \frac{r_{safety}}{\min(D_{real})} \right\rceil * 100 * \min(D_{map}) \qquad (1)$$

Where, R_s is safety distance relative to map (measured in pixels), D_{map} is map dimension (measured in pixel), D_{real} is the real robot environment dimension (m) and r_{safety} is safety distance (m).

Since shortest paths usually pass along obstacle edges, softening these sharp edges removes resulted edges and smoothing the resulted path as well. This can be applied easily by using circular expansion tool to expand obstacle boundaries as shown in Fig. 2.

B. Find Initial Path

After preparing environment map, roadmap is calculated using Voronoi-Visibility points (presented in [18]). Initial Path can then be found between any two points in the

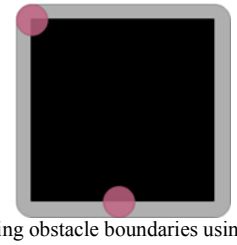

Fig. 2 Expanding obstacle boundaries using circular tool.

environment by connecting these points to the roadmap to get search graph, and using best first search (BFS) to find the best possible path.

C. Path Shortening using RoS:

Finding shortest path, according to Bhattacharya [19], happens when the clearance from obstacle set to zero. Although, this minimization of the clearance aggravates the chances of collision, the previous expanding of obstacle boundaries gives safety distance to prevent these chances.

This paper assumes that cutting all seen path segment (relative to test point of the path) and jumping directly to the last seen point of the path minimize the path length. And by many iterations the shortest path will be reached as described in pseudocode in Algorithm.1

Algorithm. 1: Path Shortening

Input: Map, P

Output: $P_{shortest}$

Initialize $P_{new} = [\ ], P_{old} = P$

while length $(P_{new}) \neq$ length (P_{old}) **do**

$P_{new} = (next) = P_{old}(start)$

while there is no safe path between $P_{new}(next)$ **and** $P_{old}(end)$ **do**

Build Robot Perspective from $P_{new}(next)$

$P_{new}(next)$ = farthest visible path point

endwhile

P_{old} = inverse (P_{new})

endwhile

return $P_{shortest} = P_{new}$

The region of sight is generated as a circular sector CR (is given in equation (2)) centered at the test point and is oriented to contain all the remaining path points

$$CR = re^{i\theta} \qquad (2)$$

Where, r is RoS radius and dedicated to the longest distance between the test point and all remaining path points:

$$r = \max_{i<j\leq end} \left| \overline{P_i P_j} \right| \qquad (2)$$

Where, P_i, P_j are test point and another path point respectively.

Sector range is chosen to cover the area where the path may be reduced, which ranges from the smallest to largest angles between test point and the others.

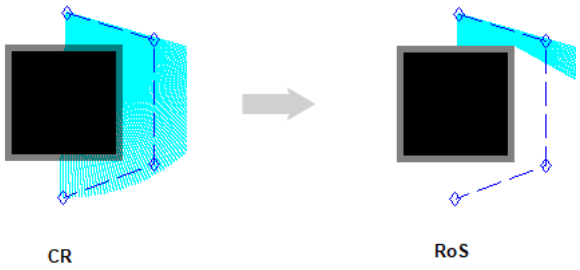

Fig. 3 Generating RoS around test point.

$$\min_{i<j\le end}(\angle P_iP_j) \le \theta \le \max_{i<j\le end}(\angle P_iP_j) \tag{3}$$

The resulted CR is intersected with the environment map for each vector of the CR to generate the RoS.

$$RoS = \bigcup_{\theta=\theta_{\min}}^{\theta_{\max}} CR_\theta \cap Map \tag{4}$$

As shown in Fig. 3, RoS shows only the feasible area where the robot can move safely to any point of it, therefore path can be shortened by selecting the intersection point between path and RoS as the next point of the path. This selected point turns to be the next test point, where the new RoS is generated around. This procedure continues until it reaches a point, which has a direct and a safe connection to the end point (as shown in Fig. 4).

Although 1st iteration has shortened the path length, seeing path from different viewpoints can show details and shortening possibilities can't be seen from the first viewpoint. So, 2nd iteration find RoS for test points from different orientation by search back from end to start point.

As for consecutive modifications in path shape, addition details can result and can be eliminated by subsequent iterations. So, this method stops only when the same path results from two successive iterations.

D. Path Evaluation

The evaluation process seeks to investigate the efficiency of the proposed method in terms of algorithm execution time, shortness, safety and smoothness.

Algorithm's execution time is directly related to the number of iterations (N) needed to find optimal path.

Whilst, Shortness is determined by algorithm's ability to find the shortest path between two given points, path length (measured in pixels) is defined as the total Euclidean distance roamed by the robot. It is obtained through summing out the lengths of path segments using

$$PL = \sum_{k=1}^{n-1} \left|\overline{P_kP_{k+1}}\right| \tag{5}$$

Where, $\left|\overline{P_kP_{k+1}}\right|$ is the segment length between the k_{th} and $(k+1)_{th}$ coordinates of the path list.

On the other hand, safety is measured (in pixels) by the minimal distance between the resulted path and nearby obstacles.

Finally, smoothness is concerned with method's efficiency in generating a gradient and smoothly changing path. Curvature criterion κ (measured in (m^{-1}) gives details of velocity and acceleration derivative changes along the tested path, which measures smoothness and energy consumption as well.

$$\kappa = \frac{\left|x'y'' - y'x''\right|}{\left[(x')^2 + (y')^2\right]^{3/2}} \tag{6}$$

Where, $(x', y'), (x'', y'')$ are first and second derivative of path points coordination (x, y) respectively.

III. Results and Discussion

This section investigates our proposed method efficiency on different environments, this method is then compared with well-known methods (PSO, ACO, ABC) used for path planning. Since this simulation is applied regardless of the robot model or dimension, the safety

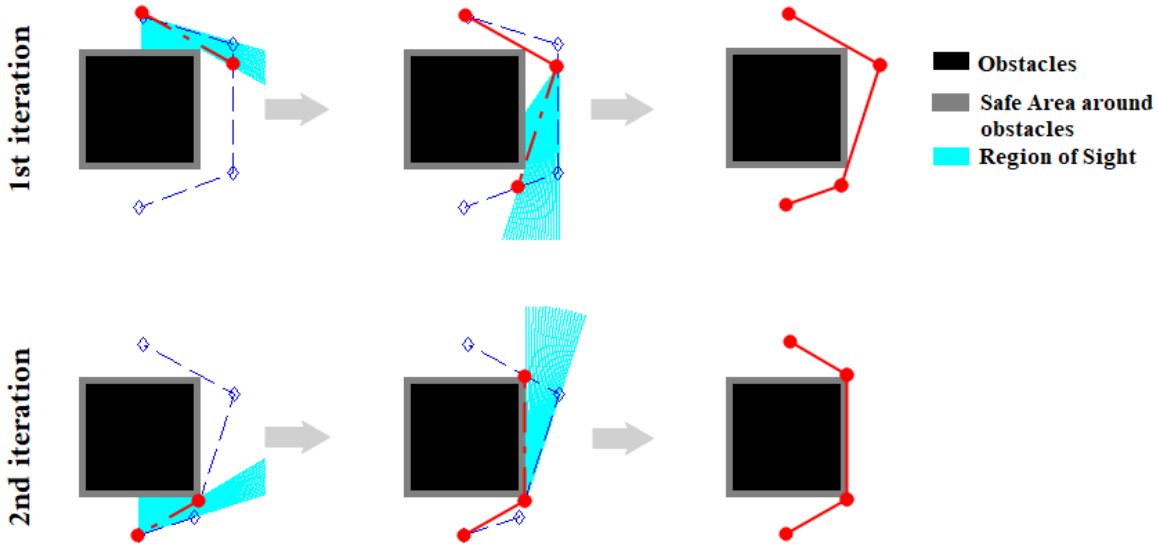

Fig. 4 Shortening Initial Path by two iterations of the proposed method

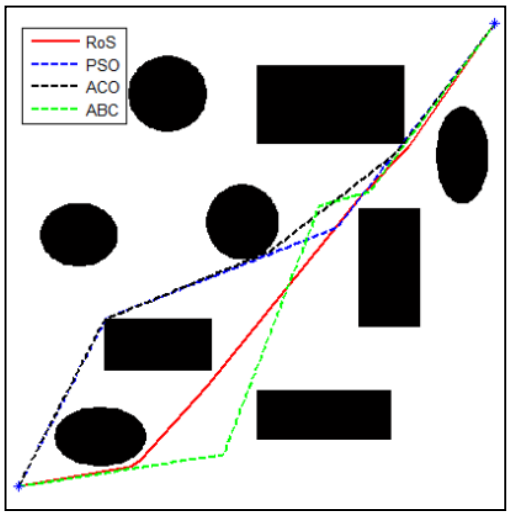

Fig. 5 simple environment experimental results.

radius is considered as

$$R_s = \lceil 0.01 * \min(D_{map}) \rceil \quad (7)$$

Three different maps belong to three different types used to evaluate proposed method

A. Simple Environment

The environment considered in this section consists of nine dense convex obstacles. The total area of this map is $500 \times 500\, pixel^2$.

All tested methods used the same initial path and they optimize it then according to their own fitness functions. All the resulted paths are shown in Fig. 5, while the quantitative results are stated in TABLE 1. They all show that our proposed method achieves the best results in all objective criterion with the least number of iterations. It is followed by ACO in both shortness and smoothness at the expense of safety with the largest number of iterations compared with the others.

TABLE I. OBJECTIVE VALUES IN SIMPLE ENVIRONMNET TEST

Method	Shortness (pixels)	Safety (pixels)	Smoothness (m^{-1})	No. of Iterations
RoS	**677.49**	**5**	**0.005**	**7**
PSO	687.194±12.5	1	0.2304	72±20
ACO	681.06±9.25	1	0.1856	100±25
ABC	726.745±18.6	3±2	0.2639	36±12

B. Non-Convex Obstacles Environment

Although, previous environment deals with sparse convex obstacle, non-convex obstacles apply more challenge to path planning algorithms. This section uses $400 \times 400\, pixel^2$ environment map [20] with 16 non-convex obstacles.

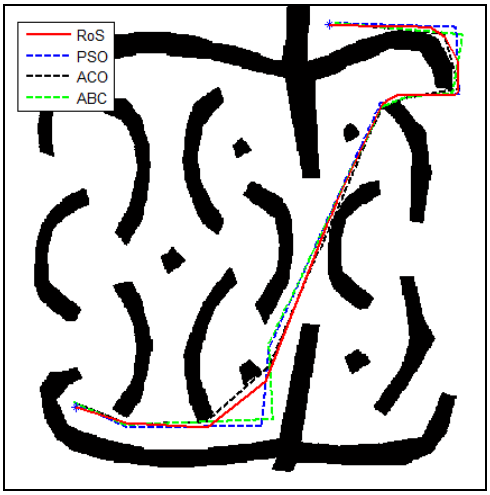

Fig. 6 Non-Convex Obstacles Environment experimental results.

TABLE II. OBJECTIVE VALUES IN NON-CONVEX OBSTACLE ENVIRONMENT

Method	Shortness (pixels)	Safety (pixels)	Smoothness (m^{-1})	No. of Iterations
RoS	623.74	**4**	**0.1123**	**4**
PSO	660.554±20.58	2±1	1.315	120±25
ACO	616.30±8.69	1	1.5465	75±20
ABC	654.745±17.26	2±1	1.3626	51±17

C. Complex Environment:

The final environment is a maze-like [21], with total area $230 \times 200\, pixel^2$. The current environment was not included for simulating real-life applications, but to determine the capability of each algorithm to perform well in space where the path towards the goal is not obvious.

TABLE III. OBJECTIVE VALUES IN MAZE-LIKE ENVIRONMENT

Method	Shortness (pixels)	Safety (pixels)	Smoothness (m^{-1})	Iter. No.
RoS	356.772	**3**	**0.1806**	**7**
PSO	349.87±5.3	2±1	3.5806	100±25
ACO	344.4±4.3	1	3.202	79±25
ABC	415.22±8.2	4±2	2.7572	66±32

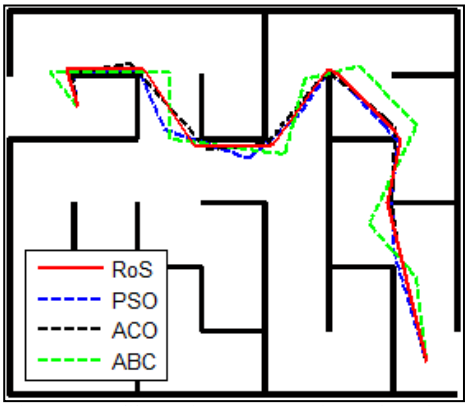

Fig. 7 Maze-like environment experimental results.

As appears in both Fig.6 and Fig. 7, RoS method reverses here between safety and shortness, while it maintains the best objective value in smoothness with the least number of iterations.

The previous results show that expanding obstacles boundaries using circular tool has maintained both safety and smoothness within acceptable limits (outperformed all other compared methods). It happened due to the aligns of the resulted path with the expanded smoothed edges of the nearby obstacles.

The environmental guided technique used to shorten the initial path led to speedup path length convergence, and, consequently it reduces the algorithm execution time.

IV. Conclusion

This paper has presented multi-objective path planning method using environmental guided techniques. Minimal safety and smoothness limits were firstly preserved, path length was then shortened as much as the environment details allowed.

The obtained results show the superiority of the proposed method in simple and complex environments in comparison with the state-of-the-art PP methods, especially in its short algorithm's execution time. This allows to test it later on dynamic environments, apply it in real world scenarios with real robots and extend the safety distance concept to include robot specifications parameters.

The simplicity of RoS can be also expanded to deal with different traffic preferences and 3D environments.

References

[1] G. Klančar, A. Zdešar, S. Blažič, and I. àkrjanc, Wheeled Mobile Robotics. From Fundamentals Towards Autonomous Systems. Butterworth-Heinemann, 2017.

[2] B. K. Patle, G. Babu L, A. Pandey, D. R. K. Parhi, and A. Jagadeesh, "A review: On path planning strategies for navigation of mobile robot," Defence Technology, Apr. 2019.

[3] E. Masehian and D. Sedighizadeh, "Classic and Heuristic Approaches in Robot Motion Planning A Chronological Review," vol. 1, no. 5, p. 6, 2007.

[4] S. Hayat and Z. Kausar, "Mobile robot path planning for circular shaped obstacles using simulated annealing," in 2015 International Conference on Control, Automation and Robotics, 2015, pp. 69–73.

[5] C. Lamini, S. Benhlima, and A. Elbekri, "Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning," Procedia Computer Science, vol. 127, pp. 180–189, 2018.

[6] P. I. Adamu, J. T. Jegede, H. I. Okagbue, and P. E. Oguntunde, "Shortest Path Planning Algorithm – A Particle Swarm Optimization (PSO) Approach," Proceedings of the World Congress on Engineering 2018, vol. 1, p. 6, 2018.

[7] R. Rashid, N. Perumal, I. Elamvazuthi, and M. K. Tageldeen, "Mobile Robot Path Planning Using Ant Colony Optimization," p. 6, 2016.

[8] W. Khaksar, T. S. Hong, K. S. M. Sahari, M. Khaksar, and J. Torresen, "Sampling-based online motion planning for mobile robots: utilization of Tabu search and adaptive neuro-fuzzy inference system," Neural Computing and Applications, Jun. 2017.

[9] A. Hidalgo-Paniagua, M. A. Vega-Rodríguez, J. Ferruz, and N. Pavón, "Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach," Soft Computing, vol. 21, no. 4, pp. 949–964, Feb. 2017.

[10] F. Ahmed and K. Deb, "Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms," Soft Computing, vol. 17, no. 7, pp. 1283–1299, Jul. 2013.

[11] M. Davoodi, F. Panahi, A. Mohades, and S. N. Hashemi, "Multi-objective path planning in discrete space," Applied Soft Computing, vol. 13, no. 1, pp. 709–720, Jan. 2013.

[12] S. Salmanpour, H. Monfared, and H. Omranpour, "Solving robot path planning problem by using a new elitist multi-objective IWD algorithm based on coefficient of variation," Soft Computing, vol. 21, no. 11, pp. 3063–3079, Jun. 2017.

[13] Z. Elmi and M. O. Efe, "Multi-objective grasshopper optimization algorithm for robot path planning in static environments," in 2018 IEEE International Conference on Industrial Technology (ICIT), 2018, pp. 244–249.

[14] L. Ferariu and C. Cimpanu, "Multiobjective hybrid evolutionary path planning with adaptive pareto ranking of variable-length chromosomes," in 2014 IEEE 12th International Symposium on Applied Machine Intelligence and Informatics (SAMI), 2014, pp. 73–78.

[15] Z. Zhu, J. Xiao, J.-Q. Li, F. Wang, and Q. Zhang, "Global path planning of wheeled robots using multi-objective memetic algorithms," Integrated Computer-Aided Engineering, vol. 22, no. 4, pp. 387–404, Aug. 2015.

[16] S. Thabit and A. Mohades, "Multi-Robot Path Planning Based on Multi-Objective Particle Swarm Optimization," IEEE Access, vol. 7, pp. 2138–2147, 2019.

[17] X. Chen and Y. Li, "Smooth Path Planning of a Mobile Robot Using Stochastic Particle Swarm Optimization," in 2006 International Conference on Mechatronics and Automation, 2006, pp. 1722–1727.

[18] H. Hliwa, M. Daoud, N. Abdulrahman, and B. Atieh, "Optimal Path Planning of Mobile Robot Using Hybrid Tabu Search- Firefly Algorithm," International Journal of Computer Science Trends and Technology (IJCST), vol. 6, no. 6, p. 7, 2018.

[19] P. Bhattacharya and M. L. Gavrilova, "Density-Based Clustering Based on Topological Properties of the Data Set," in Generalized Voronoi Diagram: A Geometry-Based Approach to Computational Intelligence, vol. 158, J. Kacprzyk and M. L. Gavrilova, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 197–214.

[20] A. Bhatia, L. E. Kavraki, and M. Y. Vardi, "Sampling-based motion planning with temporal goals," in 2010 IEEE International Conference on Robotics and Automation, 2010, pp. 2689–2696.

[21] J. Atmadja, "Implementasi Intuitif Penyelesaian Labirin dengan Algoritma Depth-First Search," Makalah IF2211, p. 7, 2016.