RESEARCH ARTICLE OPEN ACCESS

Optimal Path Planning of Mobile Robot Using Hybrid Tabu Search- Firefly Algorithm

Hiba Hliwa[1], Mohsen Daoud[2], Nizar Abdulrahman[3], Bassam Atieh[4]
PhD Student [1], Professor [2], Assistant Professor [3], Assistant Professor [4]
Department of Mechanical and Electronic Engineering
Tishreen University
Syria

ABSTRACT

This paper introduces offline hybrid optimal path planning algorithm begins with combining GV and GD approach to broader representation of the environment. Tabu Search followed by Firefly Algorithm is then applied to get the optimal path between two given point. Finally, this path is smoothed with Bezier curve.

This proposed algorithm is used to plan a path for mobile robot in static environment. A set of maps were used to simulate and evaluate this algorithm. The results obtained are compared with those of the other algorithms for different sets of continuous maps. This comparison shows the superiority of TS/FA algorithm in almost all comparisons.

***Keywords* :**— Optimal Path Planning, Hybrid algorithms, Tabu Search, Firefly Algorithm, Bezier curve, Visibility Graph, Voronoi Diagram.

I. INTRODUCTION

Mobile robots have been widely applied in many fields, including industry, agriculture, architecture, and military[1]. An efficient path planning based on the parameters (e.g. cost, energy, time, and distance) for a motion robot plays an important role in mobile robot navigation. Finding an optimal path from start point to target point without colliding with the obstacle in an environment of the robot working space is the most necessary task for efficient route planning[2]. An optimization route of the motion robot has to satisfy goals such as shortest path, lowest energy consumption, or right time, without colliding with the obstacle on its paths [3].

The robot path planning problem can be divided into classical methods and heuristic methods[4]. The most important classical methods consist of cell decomposition method (CD), potential field method (PFM), sub-goal method (SG) and sampling-based methods [5].

Planning schemes based on sampling-based motion planning (SBP) algorithms have received considerable attention due to their capability in complex and/or time critical real-world planning problems. The two well-known road-map methods, namely, visibility graph (VG) [6] and Voronoi diagram (VD) [7] have achieved very good results with dramatically different types of roads. A visibility graph is a graph that comes as close as possible to obstacles. As a result, the shortest path is found by applying this method; however, the path touches obstacles at the vertices or edges and thus is dangerous for the robot. Contrary, Voronoi diagram creates a road that tends to maximize the distance between the robot and the obstacles. Therefore, the solution paths based on Voronoi diagram are not optimal with respect to path length [8].

In path planning problem, classic methods proved to be inefficient for high-dimension space, requiring considerably long time and huge storage memory [9]. Consequently, heuristic methods are developed to cope with the curse of dimensionality in path planning problem. Many efficient metaheuristic approaches such as genetic algorithm (GA), ant colony optimization (ACO), tabu search (TS), and particle swarm optimization (PSO) are applied to path planning problems.

Su et al. [10] used Particle Swarm Optimization (PSO) to optimize the path of a mobile robot through an environment containing static obstacles. Whereas, Haj Darwish et al. [11] presented a solution to plan a path using a new form of the Bees Algorithm (ABC) for a 2-Wheeled Differential Drive mobile robot moves in indoor environment. Wang et al.[12] improved ant colony algorithm (ACO) to resolve path planning problem, which can improve convergence rate by using this improved algorithm.

The PSO was hybridized with the Gravitational Search Algorithm (GSA) to minimize the maximum path length and hence minimize the arrival time of all robots to their destination in the environment[13]. Galvez et al.[14] proposed a new algorithm based on combining Tabu Search (TS) with the Bezier curve They conducted different experiments using 2D and 3D curves, and their proposed algorithm achieved competitive results. Firefly Algorithm (FA) and Bezier curve were used to locate the shortest feasible (collision-free) path, and the results of the proposed algorithm were compared with GA and adaptive inertia weight PSO (PSO-w) [15] The results proved that the PSO-w achieved the shortest optimal path and the FA achieved the highest success rates.

In this paper, Both VG and VD are combined to model mobile robot roadmap, TS is then used to calculate the appropriate path between source and goal. After optimizing this path using FA, it will be smoothed using Bezier curve.

The rest of this paper is organized as follows: A brief about th basic concepts of environment representation and both tabu search (TS) and Firefly Algorithm (FA) is described in section II, whereas section III presents the methodology of calculating and optimizing the path between source and goal. Section IV illustrates the effectiveness of the proposed algorithm through some simulations and results. Section V evaluates the proposed algorithm by comparing with other frequently used methods including ACO, ABC and PSO algorithms. Finally, section VI concludes the paper.

II. BASIC CONCEPTS

A. *Map Representation:*

Robot navigation rely on occupancy grid maps to encode the obstacles of the area surrounding a robot. These maps can be learned from sensor data, they are well suited to solve problems like path planning, collision avoidance, or localization.

several grid-based spatial representations have been developed that can be derived from grid maps, e.g., distance maps, Voronoi diagrams, and configuration space maps. These representations are important building blocks for many different robotic applications, since they can be used to speed-up algorithms that solve the aforementioned problems.

Voronoi graph is popular cell decomposition method for solving navigation tasks. Their application as roadmaps is an appealing technique for path planning, since they are "sparse" in the sense that different paths on the graph correspond to topologically different routes with respect to obstacles, which significantly reduces the search problem. Also, moving along the edges of a Voronoi graph ensures the greatest possible clearance when passing between obstacles [16]. On the other hand, planning motion paths using the visibility graph yields the shortest collision-free possible path between a start and a goal configuration [17]. However, shortest paths are in general tangent to obstacles, so a path computed from a visibility graph usually contains semi-free configurations (the robot is in contact with an obstacle, but their interiors do not intersect) and therefore does not have any clearance [18].

In this paper, hybrid of these two latter approaches is used. It evolves from the visibility graph to the Voronoi diagram as c grows from 0 to ∞ , where c is the preferred amount of clearance. This map contains also the visibility graph of the obstacles dilated with a disc of radius r. This guarantees that the paths in the diagram are not only short but also narrow passages in the scene may disappear, which implies that it is not possible to pass through these narrow passages keeping a distance of at least r from the obstacles as shown in Fig. 1.

B. *Tabu Search (TS):*

TS is a metaheuristic that is used to manage a local method to search the solution space without entrapping into a local optimum by means of some strategies [19].

TS is a local search method used for mathematical optimization. A local search takes a potential solution to a problem and checks its immediate neighbours in the hope of finding an improved solution. Local search methods tend to become stuck in suboptimal regions or on plateaus where many solutions are equally fit. Tabu Search enhances the performance of these techniques by using memory structure that describes the visited solutions or user-provided sets of rules. If a potential solution has been previously visited within a certain short-term period or if it has violated a rule, it is marked as "taboo" so that the algorithm does not consider that possibility repeatedly[20].

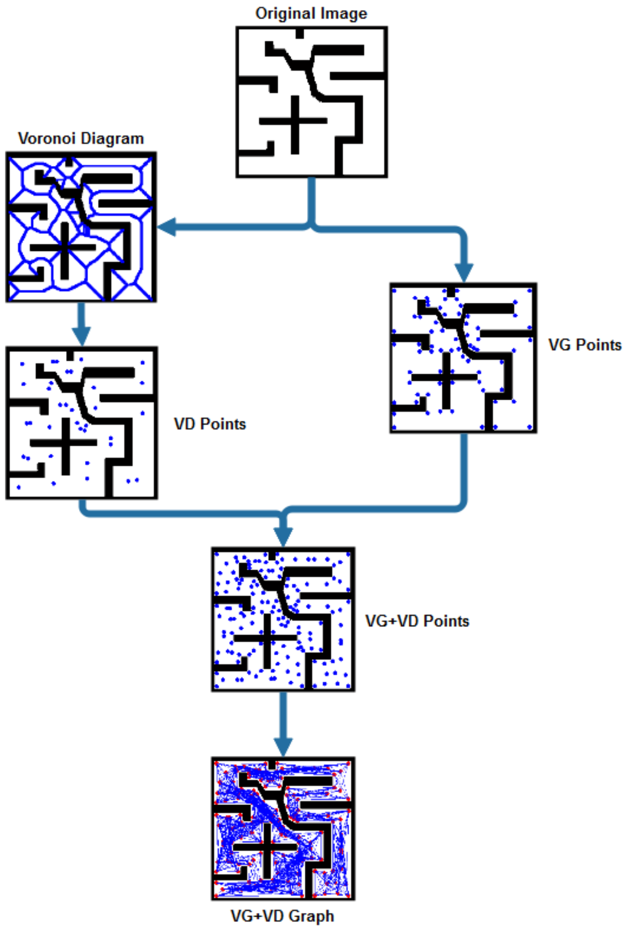


Fig. 1 Map Representation

C. *Firefly Algorithm (FA):*

FA is a metaheuristic approach for optimization problems. The search strategy in FA comes from imitation of the social behaviour of fireflies flying in the tropical summer sky. Fireflies communicate, search for pray and find mates using bioluminescence with varied flashing patterns. By mimicking nature, various metaheuristic algorithms can be designed. Some of the flashing characteristics of fireflies were idealized so as to design a firefly-inspired algorithm. For simplicity, only three rules were followed [21]:

1) All fireflies are unisex so that one firefly will be attracted by other fireflies regardless of their sex.
2) Attractiveness is proportional to firefly brightness. For any pair of flashing fireflies, the less bright one will move towards the brighter one. Attractiveness is proportional to the brightness which decreases with increasing distance between fireflies. If there are no

brighter fireflies than a particular firefly, this individual will move at random in the space.

3) The brightness of a firefly is influenced or determined by the objective function.

In this paper, brightness is simply proportional to the value of the cost function.

D. *Bezier Curve:*

Bezier curve has good geometric properties and has been widely used in computer graphics applications. One of the biggest advantages of the Bezier curve is that the Bezier curve is also convex if the control point is a convex polygon, that is, the feature polygon is convex. In addition, it can describe and express free curves and surfaces succinctly and perfectly [22].

A Bezier Curve of degree n is a parametric curve composed of Bernestein basis polynomials of degree n and it can be defined as [23]:

$$\sum_{i=0}^{N} P_i B_{i,N}(t) \qquad t \in [0,1] \tag{1}$$

Where $B_{i,N}(t)$ is a Bernstein polynomial represented by:

$$B_{i,N}(t) = C_N^i t^k (1-t)^{N-i} \quad (i = 0,1,\dots,N) \tag{2}$$

For the curve C_v in the two dimensions, the coordinates of these points on it can be calculated by:

$$\begin{cases} P_{C_v}(x,t) = \sum_{i}^{N} P_i(x) B_{i,N}(t) \\ P_{C_v}(y,t) = \sum_{i}^{N} P_i(y) B_{i,N}(t) \end{cases} \quad t \in [0,1] \tag{3}$$

The length of the Bezier curve C_v can be defined as follows:

$$f_{len}(c_v) = \sum_{k=2}^{m} \sqrt{a+b}$$
$$a = \left(P_{C_v}\left(x,\frac{k}{m}\right) - P_{C_v}\left(x,\frac{k-1}{m}\right)\right)^2 \tag{4}$$
$$b = \left(P_{C_v}\left(y,\frac{k}{m}\right) - P_{C_v}\left(y,\frac{k-1}{m}\right)\right)^2$$

Where m is path points' number.

III. PROPOSED HYBRID APPROACH

The proposed algorithm combines the properties of the two metaheuristic algorithms to handle obstacles presence and to find the optimal path from source point to destination point in a static environment. The main goal to use the meta-heuristic TS and FA is to obtain the optimized path with minimum possible iteration and execution time.

In this hybridized approach shown in Fig. 2, TS is used to get suitable path according to the VG-VD map. This path is then considered as initial solution of the FA, which find the optimal path in the continuous roadmap representation. Bezier curve is then used to smooth the result path.

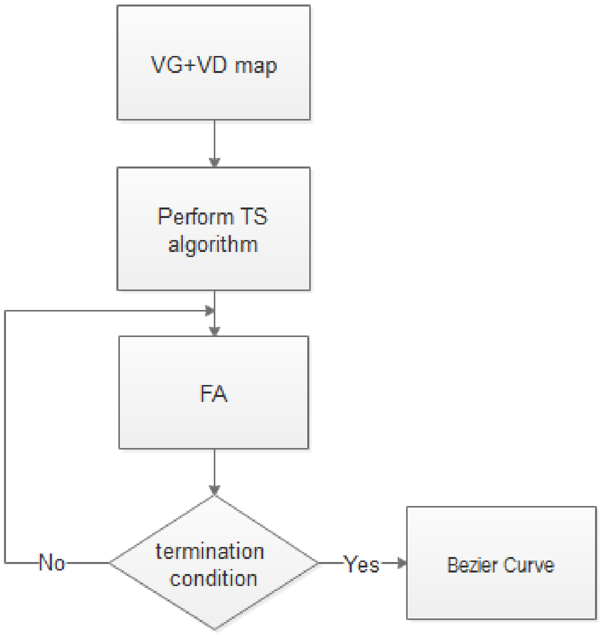


Fig. 2 Proposed Hybrid Algorithm

After determining map points and finding all possible connection between them, cost between the connected points is calculated using Euclidean distance:

$$c = \sqrt{(x_{p1} - x_{p2})^2 + (y_{p1} - y_{p2})^2} \tag{5}$$

Where (x_{p1}, y_{p1}) the coordinates of the first point and (x_{p2}, y_{p2}) is the coordination of the second one.

The appropriate path between two points (source and goal) could be specified using TS, which its basic steps are summarized by the pseudo code shown in Algorithm 1

Algorithm 1: Tabu Search

```
Generate the initial feasible path.
Current path= initial path
repeat
   for each cell in the current path do
      if Move(current cell) is not tabu then
         Generate the new path after applying the move:
         exchanging, inserting, removing the current cell
         Calculate the new path cost
         if new path cost ≤ current path cost then
            Make the move tabu
            Current path= new path
            Add the new path to the set of  best paths
         end if
      else
         if the move is tabu and new path cost ≤ current
            path cost (aspiration criteria) then
            Add the new path to the set of best paths
         end if
      end if
   end for
Update the tabu lists
```

TS algorithm is applied on each segment to generate new segments that improve its cost. At the end of an iteration, the best segments are recombined to constitute the best path. The segmentation contributes to reducing th search time, which was confirmed by simulations.

The resulted path could be optimal for the points representing the roadmap. But, it may not be optimal for the real roadmap. Therefore, FA is used to optimize this path according to the continuous giving map.

For simplicity, several characteristics of fireflies are idealized into three rules described in [24]. Based on these three rules, the FA can be described in Algorithm 2.

For two fireflies $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, they can be updated as follows:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}\left(x_i^t - x_j^t\right) + \alpha \varepsilon_i^t \qquad \textbf{(6)}$$

where $\boldsymbol{\alpha}$ is the step size, $\boldsymbol{\beta_0}$ is the attractiveness at $\boldsymbol{r} = \mathbf{0}$, the second part is the attraction, while the third is rand.

In our present work, we take $\boldsymbol{\beta_0} = \mathbf{1}$, $\boldsymbol{\alpha} \in [\mathbf{0}, \mathbf{1}]$, and $\boldsymbol{\gamma} = \mathbf{1}$

Algorithm 2: Firefly Algorithm

```
 Initialization. Set G = 1; define γ; set step size α and β0
 at r = 0.
Evaluate the light intensity I determined by f(x)
While G < MaxGeneration do
   for i = 1 : NP (all NP fireflies) do
      for j = 1 : NP (NP fireflies) do
        if (Ij < Ii),
          move firefly i towards j;
        end if
      Update attractiveness;
      Update light intensity;
      end for j
   end for i
  G = G + 1;
end while
```

The resulted path is defined as the straight lines obtained by joining two successive points of FA solution.

This paper uses Bezier curve in each and every intersection point of two straight segments of the path in order to gradually change the direction, and remain the longest possible line between these curves as shown in Fig. 3[25].

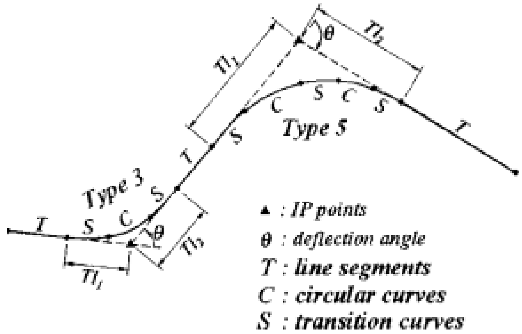


Fig. 3 proposed smoothed path using Bezier curve.

In this paper, 5 control points ($PC1, PC, P1, PT, PT1$) are used to perform Bezier curve as shown in Fig.4

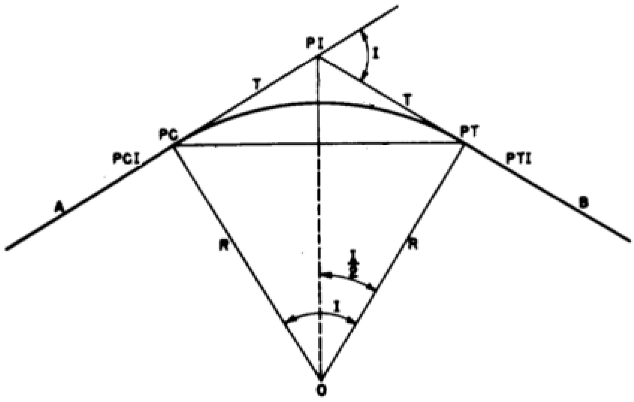
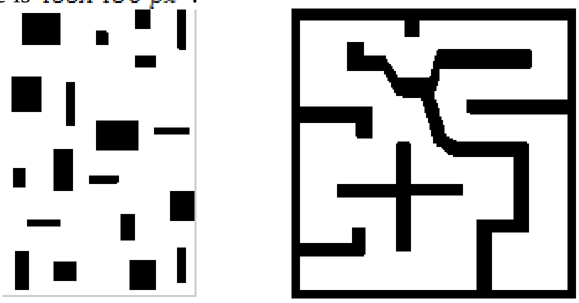


Fig. 4 principles of choosing control points

Where $\overline{PC1\ P1} = \overline{PT1\ P1}$, $\overline{PC\ P1} = \overline{PT\ P1}$, and there is collision-free path between PC and PT.

IV. EXPERIMENTAL RESULTS

In this section, we test the performance of the proposed metaheuristic TS/FA. Two maps shown in Fig. 5, which are ranged between simple and complex, are used to run experiments. The first Map [26] is $561X380\ px^2$ environment consists of dense rectangle obstacles in different sizes as depicted in (Fig. 5,a), whereas the second map is maze-like environment depicted in (Fig.5,b)[11]. The total area of the space is $433X430\ px^2$.

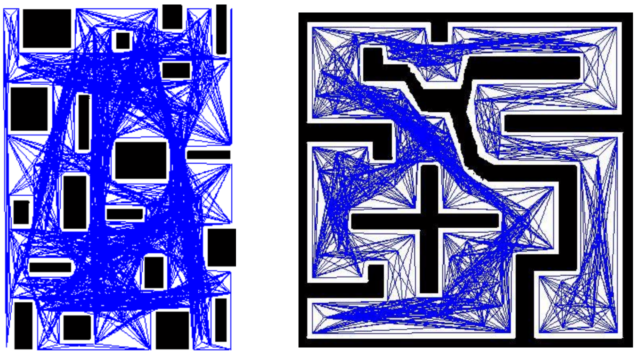


(a) (b)

Fig. 5 Test Maps.

All computations in this paper have been performed on a 1.8 GHz. Intel Core i5 processor with 4GB. of RAM. The source code has been implemented in Matlab, version 2014a.

VG+VD Map representation is applied on these two maps as shown in Fig. 6.

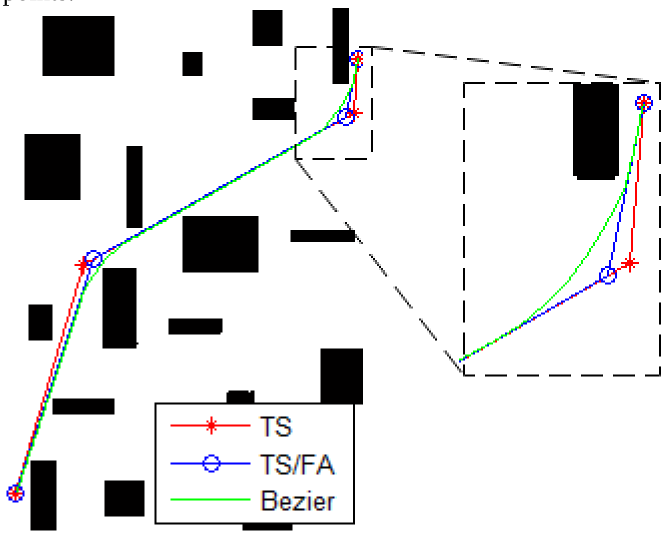


(a) (b)

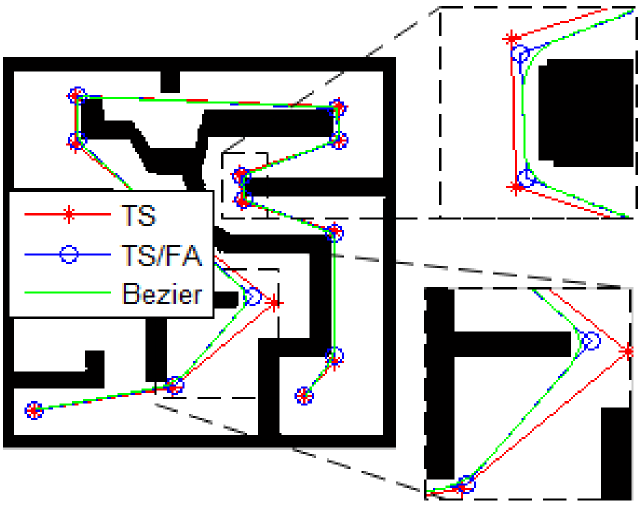
Fig. 6 VG+VD occupied grid maps.

It can simply realize that this proposed approach could cover all possible connection in these roadmaps taking into consideration safety, by determining minimum distance between these connections and any obstacle.

Fig. 7 shows the results of applying the proposed hybrid algorithms to find the optimal path between two definite points.



(a)



(b)

Fig. 7 proposed algorithm implementation

Table (1) shown the execution time Consumed in each step of this implementation.

Table (1) Time consumed in each level [sec]

	$VG + VD$	TS	FA	Bezier	all
Map 1	2.852	0.424	0.169	0.235	3.68
Map 2	1.4615	0.524	0.337	0.2238	2.546

Although map representation consumed the biggest part of execution time, it's not considered as problem because this step is a multiple-query pre-processing phase. This phase is done only one time to initialize the environment for further procedures. All other step takes in total less than one second to find smoothed optimal solution.

On the other hand, Table 2 shows path length changes among algorithm steps

Table 2 Path length variation in each level of the proposed algorithm

	TS	TS/FA	Bezier
Map 1	642.1477	629.56212	611.8079
Map 2	1374.3339	1323.8177	1298.3

Path length Patently decrease in each step Compared with its predecessor. And the overall path length improvement is approximately equivalent to 5% of the total path length.

V. COMPARISON AND DISCUSSION

The proposed method is compared with three common optimization algorithms, which are ACO [12], [27], PSO[10], [28]and ABC [2], [11].

This comparison is applied in two stages. The first stage focus on the algorithm's level, execution time and path length are the important comparison parameter. The second stage consists of testing the resulted paths using v-rep simulator. Robot travel time and Energy consumption will be the comparison parameter in this stage.

A. *Stage 1:*

The previous two maps would be the comparison base in this stage.

- **Execution time** required to reach the optimal path separated in to two level. Table 3 shows the nonsmoothed execution time for all four algorithms

Table 3 Execution time for nonsmoothed algorithm

	TS/FA	ACO	PSO	ABC
Map 1	**3.445**	5.336	7.839	5.83
Map 2	**2.322**	4.998	11.254	3.202

Proposed hybrid algorithm takes less time (although the time VG+VD consumed is taking into consideration) to find optimal solution compared with the other algorithms, because it needs less iterations (generally between 5 to 7 iteration) to stable the total cost. Whereas other algorithm took (25 to 75 iteration in average) to reach stable cost, which reflected on overall execution time.

Because of the maze-like environment in map 2 and the randomized initial solution of these algorithms, Both ACO, PSO and ABC face high failure percentage to find feasible path or took long time to find it. Therefore, 75% of the initial population of these algorithms is produced randomly and 25% depend on TS solution.

Table 4 Execution time for smoothed algorithm

	TS/FA	ACO	PSO	ABC
Map 1	**3.68**	9.514	12.462	9.547
Map 2	**2.546**	7.256	14.883	9.137

Table 4 reflects the influence of smoothing on the execution time. While Bezier smoothing procedure applied once in our proposed algorithm, spline curve applied every iteration in the other algorithms to smooth and test the feasible path. And this clearly reflected on the execution time.

- Path length: non-smoothed and smoothed paths are shown in table 5 and table 6 respectively

Table 5 Path length for nonsmoothed algorithm[cm]

	TS/FA	ACO	PSO	ABC
Map 1	629.5621	624.1477	**614.0054**	616.864
Map 2	1323.817	**1319.333**	1321.713	1341.949

Table 6 Path length for smoothed algorithm [cm]

	TS/FA	ACO	PSO	ABC
Map 1	**611.8079**	632.978	619.6238	626.1402
Map 2	**1298.3**	1355.7	1371.25	1397.6984

Although ACO and PSO have the shortest nonsmoothed path in map 1 and map 2 respectively, TS/FA has the shortest smoothed path in both maps.

Preserving the longest possible line and using Bezier curve only in corners was the main reason for the superiority of the proposed algorithm over others as shown in table 6.

B. ***Stage 2:***

Fig. 8 depicts $5X5\ m$ simulation environment in V-rep simulator, which both robot and goal point is installed

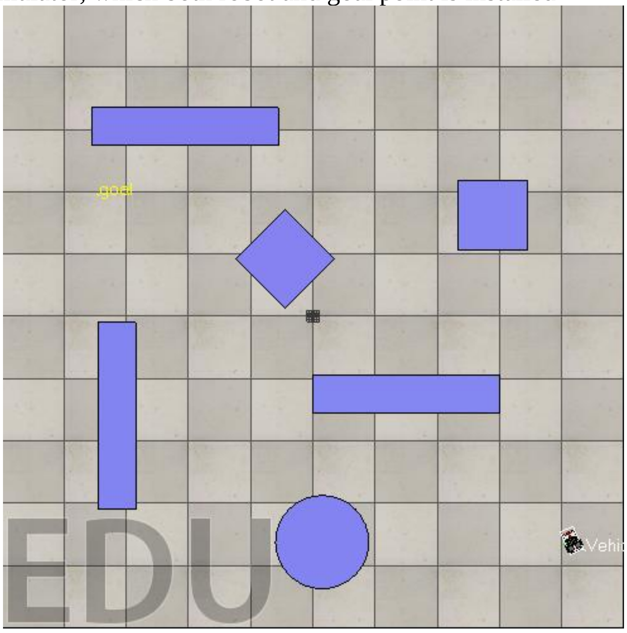


Fig. 8 Test environment on V-rep simulator

This stage tests the efficiency of the previous algorithms by making the robot tracking the paths they produced. Both travel time and Energy consumption as following:

- Travel Time: is the time that the robot needs to track the given path and reach goal point, which is shown in Table 7.

Table 7 Travel Time [sec]

TS/FA	ACO	PSO	ABC
48.5	49.6499977	49.149	49.2

Our proposed algorithm took little less time than other optimization algorithms.

-Energy consumption: shown in Table 8.

Table 8 Energy consumption [W]

TS/FA	ACO	PSO	ABC
3.499934	7.19626	7.101461	5.030248

TS/FA succeed in saving the greatest amount of energy than the others.

From all above we can observe that:

- The success rate of TS/FA is 100% in all test maps and environment used in this paper, while the others have some drawbacks in the maze-like map.
- Execution time and Path length are fixed in all test trial unlike the rest algorithms.
- Keeping the longest possible straight lines and choosing Optimal control point of Bezier curve leads to gradually change of the direction, which ensure shortest path and high performance regarding speed, safety, efficiency, and comfort [29].
- There is no obvious connection between path length and travel time, because the travel time is relative to speed more than distance.
- Without taking map representation in to consideration, TS/FA took less than 1second to find solution in big and complex environment, therefore it could be considered as real time algorithm.

VI. CONCLUSION

In this paper, we introduced offline hybrid optimal path planning algorithm, which guaranteed to find path in different map types. The comparison showed the efficiency of this algorithm in many aspects compared with the other optimization algorithms.

As execution time is less than 1 second in all test maps, it opens the door to use this algorithm to find optimal paths for the mobile robot in real time dynamic environments.

VII. REFERENCES

[1] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*, 2. ed. Cambridge, Mass.: MIT Press, 2011.

[2] T.-K. Dao, J.-S. Pan, T.-S. Pan, and T.-T. Nguyen, “Optimal path planning for motion robots based on bees pollen optimization algorithm,” *Journal of Information and Telecommunication*, vol. 1, no. 4, pp. 351–366, Oct. 2017.

[3] P. Raja, "Optimal path planning of mobile robots: A review," *International Journal of the Physical Sciences*, vol. 7, no. 9, Feb. 2012.

[4] P. Mohanty and D. Parhi, "Controlling the Motion of an Autonomous Mobile Robot Using Various Techniques: a Review," *Journal of Advanced Mechanical Engineering*, 2013.

[5] L. N and S. K.K., "A survey on path planning techniques for autonomous mobile robots," *IOSR J ournal of Mechanical and Civil Engineering (IOSR - JMC E)*, pp. 76–79, 2014.

[6] J. Cardinal and U. Hoffmann, "Recognition and Complexity of Point Visibility Graphs," *arXiv preprint arXiv:1503.07082*, 2015.

[7] M. P. Marbate and M. P. Jaini, "Role Of Voronoi Diagram Approach In Path Planning," *International Journal of Engineering Science and Technology*, vol. 5, no. 3, p. 527, 2013.

[8] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, "Heuristic approaches in robot path planning: A survey," *Robotics and Autonomous Systems*, vol. 86, pp. 13–28, Dec. 2016.

[9] Y. Ma, M. Zamirian, Y. Yang, Y. Xu, and J. Zhang, "Path Planning for Mobile Objects in Four-Dimension Based on Particle Swarm Optimization Method with Penalty Function," *Mathematical Problems in Engineering*, vol. 2013, pp. 1–9, 2013.

[10] K. Su, Y. Wang, and X. Hu, "Robot path planning based on random coding particle swarm optimization," *International Journal Of Advanced Computer Science And Applications*, vol. 6, no. 4, pp. 58–64, 2015.

[11] A. Haj Darwish, A. Joukhadar, and M. Kashkash, "Using the Bees Algorithm for wheeled mobile robot path planning in an indoor dynamic environment," *Cogent Engineering*, vol. 5, no. 1, Jan. 2018.

[12] R. Wang and H. Jiang, "Two-Dimension Path Planning Method Based on Improved Ant Colony Algorithm," *Advances in Pure Mathematics*, vol. 05, no. 09, pp. 571–578, 2015.

[13] P. K. Das, H. S. Behera, P. K. Jena, and B. K. Panigrahi, "Multi-robot path planning in a dynamic environment using improved gravitational search algorithm," *Journal of Electrical Systems and Information Technology*, vol. 3, no. 2, pp. 295–313, Sep. 2016.

[14] A. Galvez, A. Iglesias, and L. Cabellos, "Tabu Search-Based Method for Bezier Curve Parameterization," *International Journal of Software Engineering and Its Applications*, vol. 7, no. 5, pp. 283–296, Sep. 2013.

[15] B. Li, L. Liu, Q. Zhang, D. Lv, Y. Zhang, J. Zhang, and X. Shi, "Path planning based on firefly algorithm and Bezier curve," p. 4.

[16] B. Lau, C. Sprunk, and W. Burgard, "Efficient grid-based spatial representations for robot navigation in dynamic environments," *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1116–1130, Oct. 2013.

[17] M. de Berg, Ed., *Computational geometry: algorithms and applications*, 3rd ed. Berlin: Springer, 2008.

[18] R. Wein, J. P. van den Berg, and D. Halperin, "The visibility–Voronoi complex and its applications," *Computational Geometry*, vol. 36, no. 1, pp. 66–87, Jan. 2007.

[19] B. Naama, H. Bouzeboudja, M. Lahdeb, and Y. Ramdani, "A Hybrid Tabu Search and Algorithm Genetic for Solving the Economic Dispatch Problem," no. 22, p. 8, 2013.

[20] I. Châari, A. Koubâa, H. Bennaceur, A. Ammar, S. Trigui, M. Tounsi, E. Shakshuki, and H. Youssef, "On the Adequacy of Tabu Search for Global Robot Path Planning Problem in Grid Environments," *Procedia Computer Science*, vol. 32, pp. 604–613, 2014.

[21] L. Guo, G.-G. Wang, H. Wang, and D. Wang, "An Effective Hybrid Firefly Algorithm with Harmony Search for Global Numerical Optimization," *The Scientific World Journal*, vol. 2013, pp. 1–9, 2013.

[22] M. Elhoseny, A. Tharwat, and A. E. Hassanien, "Bezier Curve Based Path Planning in a Dynamic Field using Modified Genetic Algorithm," *Journal of Computational Science*, Aug. 2017.

[23] B. Li, L. Liu, Q. Zhang, D. Lv, Y. Zhang, J. Zhang, and X. Shi, "Path planning based on firefly algorithm and Bezier curve," p. 4, 2014.

[24] G. Wang, L. Guo, H. Duan, L. Liu, and H. Wang, "A Modified Firefly Algorithm for UCAV Path Planning," *International Journal of Hybrid Information Technology*, vol. 5, no. 3, p. 22, 2012.

[25] J.-F. Cheng and Y. Lee, "Model for Three-Dimensional Highway Alignment," *Journal of Transportation Engineering*, vol. 132, no. 12, pp. 913–920, Dec. 2006.

[26] M. Otte and N. Correll, "C-FOREST: Parallel Shortest Path Planning With Superlinear Speedup," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 798–806, Jun. 2013.

[27] T. Mohanraj, S. Arunkumar, M. Raghunath, and M. Anand, "Mobile robot path planning using ant colony optimization," *International Journal of Research in Engineering and Technology*, vol. 3, no. 11, pp. 1–6, 2014.

[28] T. T. Mac, C. Copot, D. T. Tran, and R. D. Keyser, "A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization," *Applied Soft Computing*, vol. 59, pp. 68–76, Oct. 2017.

[29] K. Wolhuter, *Geometric Design of Roads Handbook*. CRC Press, 2015.