Mobile Robot Path Planning Method Using Firefly Algorithm for 3D Sphere Dynamic & Partially Known Environment

Alia Karim Abdul Hassan **Duaa Jaafar Fadhil**

Computer Science Department, University of Technology, Baghdad, Iraq

hassanalia2000@yahoo.com *satomi.emoki111@gmail.com*

Keywords: Path Planning, 3D, sphere space, Firefly Algorithm, Swarm Intelligence, Static environment, Dynamic environment, Partially-known environment, Mobile Robot.

Abstract

In this paper new method is proposed to solve the problem of path planning for mobile robot in a dynamic-partially known three-dimensional sphere environment by using a modified version of the Firefly Algorithm that successfully finds near optimal and collision free path while maintaining quick, easy and completely safe navigation throughout the path to the goal.

I. Introduction

Path planning is a base problem for any mobile robot system [1], [2], where in order for a robot to move freely in a space by its own we need to professionally plan its path to ensure safe navigation till the robot reach the requested goal position[3] [4] [5]**.** Therefore; great interest have been given to this field and due to the development of technology aspects this field have been progressed as well to keep up this development Such as planning in 3D environment rather than the traditional old 2D planning. 3D environment considered as closer simulation to the real world from the other dimensions in terms of volume and axis which makes it more useful when applying the method to applications of path planning in the real world[2] thus this proposed method is designed to work in 3D space while taking into consideration static obstacles such as buildings (both known and unknown position) and the dynamic obstacles like humans whose constantly moving in the space of the real world. The robot then needs the ability to move in between finding a safe path that ensures reaching the goal as shortly and quickly as possible [6].

With the widely known Swarm Intelligence that is considered as a type of Artificial intelligence [7], one of the recent Swarm Intelligence methods is the

Firefly algorithm[8] which we've chosen with few modifications applied to suit the chosen 3D sphere environment while still delivering good performance.

II. Related Works

Remarkable work has been made in the firefly algorithm to apply it on different applications or improve on it, such works as a novel algorithm for global optimization presented by Zhang et al. when they combine the firefly algorithm with the Differential Evolution (DE) getting the benefits of both algorithms for better results [8], Xin-She Yang did the same previously combining the Firefly with the Levy Flights algorithms formulating new powerful metaheuristic algorithm [9]. Other authors on the other hand had modified on the firefly itself to suit their work aspect, such as Wang et al. proposing a nee modified firefly algorithm to solve the UCAV path planning problem [10], and Liu et al. who modified version of the firefly algorithm was proposed for AUVs path planning problem in three-dimensional form [11].

The proposed method in this paper present a version of the firefly algorithm that has been modified to work in both static and dynamic environments and the partially known environments, where different scenarios were tested to evaluate the algorithm quality while focusing on the 3D space challenges which was missing in the previously mentioned related works.

III. 3D Space

Working in a 3D space gives the algorithm more realistic view where the robot has got more freedom to move in all possible direction around as if it was floating in space, diving in the air or underwater, so the next step may be one of 26 directions that the robot can choose from or stay in the same spot which makes it a total of 27 position options instead of the traditional 9 position options that experienced in the 2D world (figure 1) [2].

Dealing with 3D space means adding an extra parameter representing the extra dimension (z) to the existing (x) and (y) so each element point in the space is represented by (x,y,z) parameter format which makes the 3D environment more challenging to plan on, the following are some of the challenges that the 3D space apply on the planner: [12] [1] [2]

5- Extra parameters: planning in a 3D space means adding an extra parameter that represents the 3rd axis to all the element definitions of the space, also all the equations and procedures that apply to the axis need to be increased for the extra added axis this affects the computation time and memory space used.

1- **Environment scale:** in 3D space, when scaling the environment to larger size means increasing the scale on 3 axis, which triples the effect of scaling the environment on the algorithm, making it harder for some algorithms to cover the problem space and reach the goal when the dimension of the space increase unlike scaling in 2D space which does not have a huge effect unless the scale was very large.

2- **Mass obstacles:** in planning environments, the obstacles can be represented as points or they can have mass and occupy space in the environment, the mass obstacles can come in different shapes and sizes, making it challenging for the robot to avoid, and 3D mass obstacles are even more challenging for the planner and the robot to avoid.

3- **Dynamic environments:** path planning in dynamic environments is quite challenging and need high precision and quick algorithm to keep track with the quick changes of the space while navigating.

4- **Extra position choices:** navigating in 3D space mean more freedom for the robot to move up and down along with the other directions, but that also adds extra condition for the planner to find a good quality path it should have what it takes to choose the best location out of the available many positions in 3D space.

5- **Local minima problem:** weather the robot works in 2D or 3D space; there is always a chance of a local minima problem to occur, so a good planner should have the ability to avoid or solve that problem if it exist in the problem space.

To solve the problem of path planning in 3D space and all those challenges that comes with it; the Nature-inspired Firefly Algorithm in Sphere space was chosen to work with.

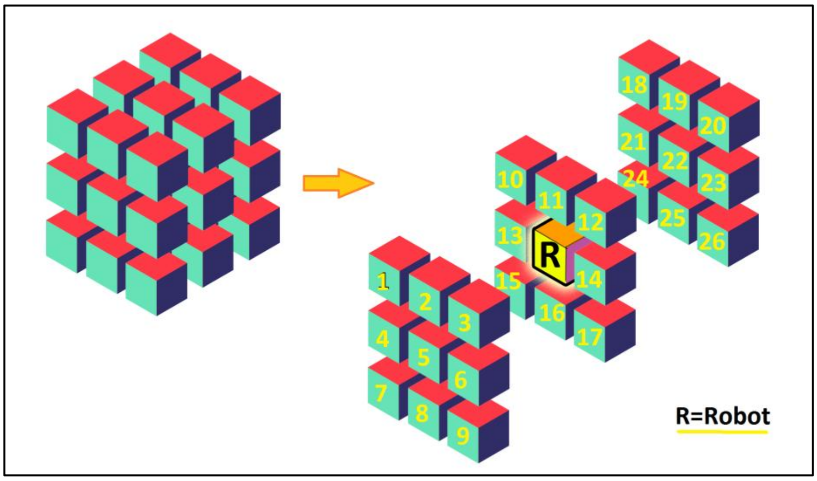

Figure (1): Robot possible positions in 3D space

IV. Firefly Algorithm (FA)

One of the recent Swarm Intelligence (SI) meta-heuristic algorithm which was proposed by Xin-She Yang in 2007 inspired biologically by the light flashing behavior of the tropical fireflies [8].

Behavior of Fireflies

There exist near two thousand of firefly species, most of them produce short rhythmic flashes that can be seen in tropical regions, giving beautiful seen to the summer sky in those areas. However, those flash lights are used by the fireflies basically for attraction purposes like attracting mates (communication) or attracting preys. Sometimes those lights can act as a warning sign for potential predators [13].

In the inspired Firefly Algorithm three rules have being used to idealize the behavior of the fireflies[11] [9] [8], those are:

- All the fireflies are considered as one gender, so that any firefly can be attracted to the other regardless of their gender.
- The Attractiveness of the fireflies is proportional to their brightness and they both decrease when the distance increase, this way one firefly will move towards the other depending on their brightness, and if there is no brighter one then the firefly will move randomly.
- The brightness of a firefly is determined or at least affected by the objective function.

Original Firefly Algorithm

In the standard Firefly Algorithm there are two important values that control the movement of the fireflies and the path of the algorithm which are: the Light Intensity and the changeable Attractiveness[13] [8].

For simplicity, we may assume that the attractiveness of a firefly can be determined by the brightness which is associated with the objective function. Hence, the brightness I of a firefly at position x can be computed as $\mathbf{I_x} \ \boldsymbol{\alpha} \ \boldsymbol{f_{(x)}}$, where $\boldsymbol{f_{(x)}}$ is the chosen Objective function. The attractiveness β on the other hand, should be judged the eye of the beholder or the other firefly, so this value (β) will be changing as the distance $\mathbf{r_{ij}}$ between two fireflies (i,j) changes. The light tends to be absorbed in the media, resulting in less intensity also as the distance to the source increase, allowing the attractiveness to vary as well with the degree of absorption [13]. So for a given medium with fixed light absorption coefficient Ɣ, the light intensity change will be computed as

$$\boldsymbol{I = I_0 \, e^{-\text{Ɣ}r^2}} \qquad \dots . \quad \mathbf{(1)}$$

Where I_0 is the light intensity at the source.

And since is already known now that the brightness is proportional to the light intensity which makes computing the change in the value of the brightness be in the form

$$\boldsymbol{\beta = \beta_0 \, e^{-\text{Ɣ}r^2}} \qquad \dots . \quad \mathbf{(2)}$$

Where β_0 is the original brightness of the firefly (i.e. at distance r=0).

The distance r_{ij} between any two fireflies can be calculated as Euclidean Distance by the equation (3)

$$r_{ij} = |x_i - x_j| = \sqrt{\sum_{k=1}^{d}(x_{i,k} - x_{j,k})^2} \quad \ldots. \quad (3)$$

Where (d) represent the dimension of the problem.

Lastly, the movement of the firefly from position $\mathbf{x_i}$ to another towards the more attractive firefly $\mathbf{x_j}$ is calculated by equation (4)

$$x_{i+1} = x_i + \beta_0 e^{-\Upsilon r^2}(x_j - x_i) + \propto \varepsilon_i \quad \ldots. \quad (4)$$

Where ε_i is a random variable and ∝ is a scaling parameter added to control the step size [14] [9] [8].Algorithm 1 shows the main steps of the original firefly algorithm[14] [13].

V. Sphere Environment

To solve the third challenge (**Mass Obstacles**); spherical environment was chosen to work with for this algorithm. This type of environment means including the whole workspace in large sphere that covers the majority of it centered on the center point of the space, also all the obstacles are assigned each to a sphere surrounding it allowing them to be represented by the center point and the radius of the sphere that covers their space [12].

The Spherical workspace allows the environment and each obstacle inside to take that one specific sphere shape which enables ease, speed and more accurate computations and any shape of obstacle will not be a problem for the planner since all of them will be covered in one uniform sphere shape [5]

Algorithm 1: Original Firefly Algorithm

Input: Population size (number of fireflies)
Output: Optimal solution(s)

```
1: Define the objective function $f(x)$, $x=(x_1,\ldots x_d)^T$
        Where, d= dimension, T= population size
2: Define the light intensity $I_i$ at $x_i$
3: Define the absorbent coefficient ϒ value
4: while (t < Population size)
5: Repeat for all fireflies:
        If ( $I_i < I_j$ )
        Then
                Move $x_i$ towards $x_j$ one step by equation (4)
                Evaluate the new population by $f(x)$ and update values
        End if
6: Rank all fireflies according to their I value and find the current best
```

```
7: End while
8: Return best solution(s)
9: End
```

VI. The Modified Firefly Algorithm (MFA)

Applying the firefly behavior on a path planning problem can be achieved in many different ways and this proposed method is no exception where each firefly is considered as an agent on its own and the path that is generated during the navigation of each single firefly of the population represent one of the solutions. This method results in several possible solutions (paths) at the end of the algorithm run, those solutions are equal to the number of the fireflies' population.

To get the optimal solution among the rest, simply the best is picked up, and that is decided by the path's length and completeness since all the resulted solutions are collision free already. The chosen path at the end is the shortest, complete, collision free and near optimal path available.

The proposed modified firefly algorithm (Algorithm 2) can function in static, dynamic and mixed 3D sphere environment generating near-optimal collision free path, it also was designed to solve all the problems and challenges of the 3D environment that was explained earlier with good performance.

This modified version has the following modifications to ensure good quality performance while keeping the benefits of the original algorithm and solving the 3D space challenges:

- **Collision avoidance:** A basic important constrain when it comes to path planning problem and that is the collision avoidance constrain and since this proposed method works on both static and dynamic environments where the obstacles may be visible or invisible to the robot while moving, the obstacles' data are fed to the robot online when the robot starts navigating in the space (by using sensors or cameras), so a check for the next position validity always take place each time pulse as the robot is assumed to move at the same speed as the dynamic obstacles. This procedure will apply to all of the fireflies so all the final generated paths will be obstacle free paths which solve the Dynamic environments challenge (no.4).
- **Scaled step size**: Because of the random value included in the step movement of the fireflies in the algorithm, the step size could vary from tiny steps to large ones. In this proposed method the step size should not be larger than a specific value which represents the smallest size allowed for an obstacle, and that ensures that the path won't pass through the obstacle to the

next position behind it, this modification provides safety while plans which is essential in any path planning algorithm.

- **Directed randomized move**: Luckily the randomization effect is already scaled in the original FA version, applied to our method makes the movement of the firefly somewhat seized, but still random enough to move in different sometimes undesired directions, and as a path planner or at least in our method the agents need to always move toward the goal to get that optimized path feature. Hence, an additional condition is added to the mFA to ensure that the next step is always better than the previous one which was done by comparing the brightness of the current and the next position to be to ensure that the next position is always better (i.e. in the direction of the goal).

- **The brightness**: Each firefly move towards the brighter firefly it can spot and according to the attractiveness value of the comparision fireflies the move is made. In this modified version the attractiveness of the firefly is represented by the objective function f(x) which is computed by the equation **5:**

$$F_{(x)} = H_{(x)} + 3G_{(x)} \qquad \dots . \quad \mathbf{(5)}$$

Where $H_{(x)}$ is the distance between the start point and the current position (x), and $G_{(x)}$ is the distance between the current position (x) and the goal point. Both $H_{(x)}$ and $G_{(x)}$ can be calculated as Euclidean Distance by the equation (3). The value of $G_{(x)}$ is tripled to increase the effect of the goal position on the result of the function which is a key to make sure that the robot always heads towards the goal in each step in a 3D space; unless no available better position can found then in this case the robot will choose a least better position to translate to, that solves the challenge of extra position choices (no.5), where the planner is able to choose the best next position of the available choices by the help of the f(x) function.

Other Features of the mFA

Along with the above modifications; the mFA has extra features that helped solving the 3D space challenges, those features exist in the original version and they were maintained in the modified version for better results, the features are:

1- This algorithm is quick and easy to implement, and doesn't require large space in the memory, which makes it suitable for large scale environment, it also is simple since it uses one main equation to move the fireflies which solves the challenge no.1 that concerns extra parameters and memory space.

2- The algorithm is based on random movement of the fireflies in the space which gives them ability to cover more of the problem space, also having more than one solution possibility represented by the multiple fireflies which provide different solution each; helps with the optimality of the final path and avoiding any possible local minima problem. That solves the challenges 2 and 6 which concern the environment scale and the local minima challenges.

3- The algorithm was modified to work smoothly with sphere space which provides more accuracy and safety and gives a solution for challenge no.3 that is avoiding mass obstacles.

Algorithm 2: Modified Firefly Algorithm

```
1- Define Start point and Goal point.
2- Mapping obstacles in the space.
3- Define the Objective function $f_{(x)}$ , the absorbent coefficient Υ and the convergence allowed value.
___________ Offline Phase _________
4- Generate an initial population of fireflies $x_i$ ,(i=1,2,...,n).
5- Compute the light intensity value I for each firefly.
6- Move all fireflies one step according to their light intensity value:
•  For each firefly do..
    -  Generate next step according to equation 4
    -  Check the new position if it was obstacle free
    -  Check if the new position within the given scale of step size
    -  Check for the direction as if the new position is better intensity than the previous or not
•  Keep generating new position to each firefly until they meet all the conditions then apply them as the new position for that firefly.
7- Check if any firefly reached the goal point, if none, then go to (step 4), otherwise continue.
8- Compare all the resulted paths and return the best path found.
___________ Online Phase _________
9- Follow one step on the found path.
10- Check if any firefly reaches the goal, then terminate the execution and return the results as best path found, the execution time and path length; otherwise continue.
11- Check the next step ahead on the path if contains any obstacle then go to (step 4), else go to (step 9).
```

VII. Simulation and Results

The proposed modified Firefly algorithm was simulated using Matlab programming language in a three-dimensional sphere environment with a 3D sphere shaped obstacles (static & dynamic). The simulation was run on a laptop with Intel® Core™ i7-3517U CPU@1.90GHz , 15.9GB memory and Windows 8.1 Pro.

The executed environment (figure 2) is basically a cube of size (20×20×20) unit. Where the actual arena is a sphere shape space of a radius =

10. The center of the sphere is at the center of the environment at the coordinates (0,0,0) as (x,y,z). The proposed modified Firefly Algorithm was tested to find a path from a specified start point to a specified goal point in different configurations of that environment, the results however are changeable due to the random fireflies' steps and the random movement of the obstacles, so the configuration changes differently each time the work is executed (table1).

Execution parameters used:

- Start point = (-9.5 , 0.01 , 0.01)
- Goal point = (8 ,0 ,0)
- Population = 5
- Absorbent coefficient = 0.3
- Step Size = 1
- Convergence value = 1

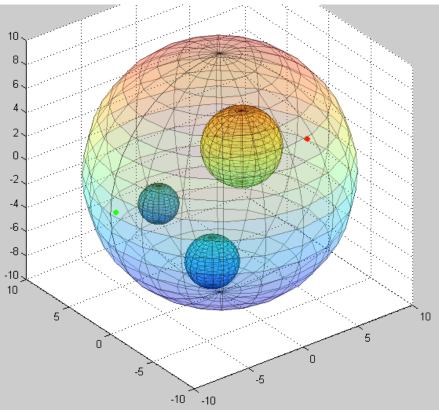

Figure 2: 3D sphere environment

The execution parameters were chosen by trial and error.

Table 1 shows the execution time and path length in the running of the proposed mFA algorithm for the environments discussed earlier. Note that movement of the dynamic obstacles in this simulation is random in all directions so the execution time and path length will vary from an execution to another in maps (2,3) but it's all around the given time.

Table 1: Simulation results

Modified Firefly Algorithm (MFA)

Map No.	Execution time (seconds)	Path Length (length units)	Number of obstacles	Map explanation
1	0.4	25	8	Static obstacles
2	0.7	27	24	Static obstacles
3	0.6	24	12	Dynamic obstacles
4	0.7	25	8	Mixed obstacles (4-static , 4-dynamic)
The random movement of the dynamic obstacles in the last two maps causes some difference in the execution time of the same environment each run, depending on how many times the robot faces an obstacle that run and thus had to change its direction and pick new step.				

The proposed mFA can solve most of the problems that occur with path planning in 3D environment and find a path if one exists and in less time. it also adds a Safety rim around each obstacle to avoid the case when the distance to the obstacle that is against the robot's current position is smaller than the robot's step, then the robot will check the next position which passes through the obstacle's edge and move towards it ignoring the obstacle that's in the way and avoid collision with obstacles. The planned path with proposed mFA is near optimal path; also it uses float values to indicate positions allowing for smoother and more accurate path representation. Figure 3 shows a path generated using the mFA in two different views.

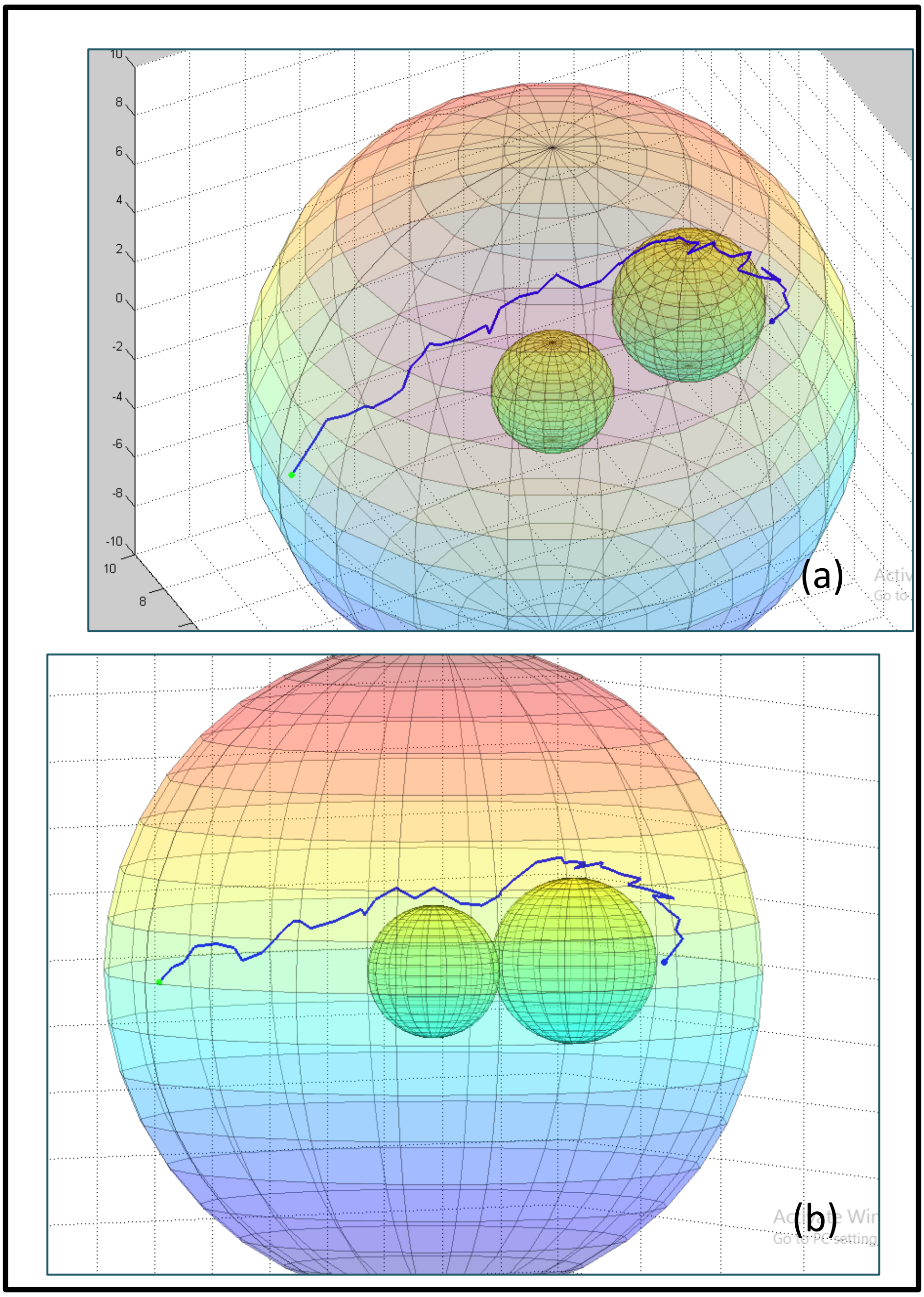

Figure 3: Path generated using mFA, **(a)** 3D view, **(b)** side view

VIII.Conclusion

A proposed modified version for the well-known Firefly Algorithm was developed in this paper to solve the problem of robot path planning. Where the proposed method works in 3-dimensional of both static and dynamic where it has the ability to avoid known, partially known and unknown 3D sphere obstacles reaching the goal point. The proposed modified version is near optimal, of simple, complete and collision free with extra safety constrains provided by the safety rim, it also allows more accurate path with high speed execution time, it also solves the problems of the 3D environments and works through all the challenges with good performance.

As a future work, the algorithm can be developed further by adding more conditions to strict the path even more and more efforts can be added to accurate the convergence condition.

References

[1] Latombe, J. C. **(1991)**. *Robot Motion Planning*, second printing, Kluwer Academic Publishers, NewYork, USA.

[2] Yang, L., Qi, J., Song, D., Xiao, J., Han, J., and Xia, Y. **(2016)**. *Survey of Robot 3D Path Planning Algorithms*, Hindawi Publishing Corporation, Journal of Control Science and Engineering, Article ID 7426913.

[3] Asl, S. V., Davarzani, Z., Staji, S. **(2015)**. *Planning Flying Robot Navigation in a Three-dimensional Space by Optimaztion Combining Q-learning and Monte Carlo Algorithms*, International Journal of Hybrid Information Technology, Vol.8, No.11 , pp.297-306

[4] Ducho ,F. ,Babinec ,A. ,Kajan ,M. ,Beno ,P. ,Florek ,M. ,Fico ,T. ,Jurišica ,L. **(2014)**. *Path planning with modified A star algorithm for a mobile robot*, ScienceDirect, Procedia Engineering 96, 59 – 69.

[5] Jaleel, T. A. , Hassan, A. K. A. **(2016)**. *Collision Avoidance Using Cat Swarm Algorithm for Multi Mobile Robot Path Planning in Dynamic Environment*, Iraqi Journal of Science, Vol. 57, No.3C, pp:2348-2359.

[6] Shwail, S., Karim, A., **(2014)** " Probabilistic Roadmap, A*, and GA for Proposed Decoupled Multi-Robot Path Planning", IRAQI JOURNAL OF APPLIED PHYSICS, Vol. 10, No. 2, April-June.

[7] Hassan, A., **(2014)**, "Path Planning Method for Single Mobile Robot in Dynamic Environment Based on Artificial Fish Swarm Algorithm", Eng. &Tech. Journal, Vol. 32,Part (B), No.2.

[8] Zhang, L., Liu, L., Yang, X-S., Dai,Y. **(2016)** *A Novel Hybrid Firefly Algorithm for Global Optimization*. PLoS ONE 11(9): e0163230. doi:10.1371/journal.pone.0163230.

[9] Yang, X-S. (**2010).** *Firefly Algorithm, L´evy Flights and Global Optimization*, Research and Development in Intelligent Systems XXVI (Eds M. Bramer, R. Ellis, M. Petridis), Springer London, pp. 209-218.

[10] wang, G. ,Guo, L. ,Duan, H. ,Liu, L. ,Wang, H. **(2012)**. A Modified Firefly Algorithm for UCAV Path Planning, International Journal of Hybrid Information Technology Vol. 5, No. 3, July.

[11] Liu, C. , Zhao, Y. ,Gao, F. ,Liu, L. **(2015).** *Three-Dimensional Path Planning Method for Autonomous Underwater Vehicle Based on Modified Firefly Algorithm*. Hindawi Publishing Corporation Mathematical Problems in Engineering, Article ID 561394.

[12] Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L. and Thrun, S. **(2005).** *Principles of robot motion: Theory, Algorithms, and Implementation,* A Bradford Book, The MIT Press, Cambridge, England.

[13] Yang, X-S. **(2010).** *Nature-Inspired Metaheuristic Algorithms* Second Edition, Published by Luniver Press, Frome, BA11 6TT, United Kingdom.

[14] [Yang X-S. 09] Yang, X-S. **(2009).** *Firefly Algorithms for Multimodal Optimization*, SAGA, LNCS 5792, pp. 169–178, c_Springer-Verlag Berlin Heidelberg.

الخلاصة

اذا البحث يقترح طريقة لحل مشكلة تخطيط مسار الروبوت المتحرك في ضمن بيئة شبه معروفة ثلاثية الابعاد كروية الشكل باستخدام نسخة معدلة من خوارزمية الحشرات المضيئة Firefly Algorithm والتي تمكنت بنجاح من ايجاد طريق شبه مثالي خالي من التصادم مع العوائق بسرعة وسهولة وملاحة آمنة على طول الطريق حتى الوصول للهدف.

الكلمات المفتاحية: تخطيط مسار, بيئة ثلاثية الابعاد، بيئة كروية، خوارزمية الحشرات المضيئة، ذكاء السرب، بيئة ثابتة، بيئة متحركة، بيئة شبه معروفة، روبوت متحرك.