Periodicals of Engineering and Natural Sciences ISSN 2303-4521
Vol. 9, No. 2, March 2021, pp.365-374

Path-planning in 3D space using butterfly optimization algorithm

Al Hamami Duaa Jaafar[1], Al Gburi Hussein Qahtan[2], Al Baiati Ali Emad[3]
[1]Ministry of Education, Administrative Affairs, Biometrics Dep.
[2]Ministry of Education, Administrative Affairs, ID's Dep.
[3]Ministry of Higher Education, University of Technology, Computer Engineering Dep.

ABSTRACT

The Butterfly Optimization Algorithm is one of the most recent nature-inspired algorithms that mimic the butterflies' behavior in mating and finding food, for solving the global optimization problems. The algorithm utilizes the sense of butterflies of smelling for determining the location of nectar and find mates, which is based on the foraging strategy of those insects. This paper represents a method of using the BOA algorithm for solving the problem of path planning in three-dimensional space. The proposed method finds a path from a particular starting point to any chosen goal, where the generated final path is completely safe and collision-free. The algorithm is based on 3 phases: the initial phase, the iteration phase, and the final phase. The movement of butterflies is based on two search moves, one of them is Local random search; where the butterfly moves randomly within the swarm, and the other is Global search; where the butterfly moves towards the best-fitted butterfly in the current population. The proposed method in this paper is able to find a collision-free path from the start point to the goal in all of the presented test environments in proximately well performance and the results were computed in terms of execution time and path length.

Keywords: Path planning, 3D space, BOA, Butterfly-Optimization-Algorithm, AI

Corresponding Author:

Al Baiati Ali Emad,
Computer Engineering Department
University of Technology
Al-sinaa, Street, Baghdad, Iraq
Email: 120098@uotechnology.edu.iq

1. Introduction

Path planning in three-dimensions is a complicated optimization problem that is a very important technology for any autonomous control module, it allows safe route from any start point to the problem goal [1,2,3]. 3D path planning has been applied for many subjects like unmanned aerial vehicles (UAV) [4], and uninhabited combat aerial vehicle (UCAV) [5] and so on. Where the autonomous module (robot) can dive in air or underwater or even in the outer space and in hazard areas, where it is hard for humans to reach and explore, while the robot can navigate and finish jobs autonomously without human involvement [6].
Many researchers were interested in this topic of path planning in 3D space, as in 2014 Wang Honglun et.al. represented a proposed method for 3-Dimensional path planning for UAV that follows the interfered fluid dynamical system in a terrain of different shapes of obstacles, where they first transformed the complex terrain into basic shapes and their combinations, Secondly; 3D streamlines are obtained then, solutions are derived for streamlines with multiple obstacles, and finally; the genetic algorithm was applied to find an optimal path within the constraints of the UAV [4].
In 2017 Bo Zhang and Haibin Duan proposed a method using a novel method of a nature-inspired optimization algorithm which is the predator-prey Pigeon algorithm for the UCAV (Uninhabited Combat Aerial Vehicle); that is mainly optimizing the route of flight while taking into consideration the different constraints' types in the complex combating dynamic environment. The prey-predator concept was adopted to enhance the convergence

speed and improve the global best properties where the characteristics of the final optimal path are represented in a cost function form [5].
Just like the prey pigeon optimization algorithm; this paper uses the butterfly optimization algorithm (BOA) [7], solving the problem of path planning, in which the algorithm runs finding a safe path from start to goal in 3D space avoiding any obstacle along the way. The BOA is a meta-heuristic swarm algorithm that exhibits improved performance that exceeds the traditional approaches. To this time, research has granted nature-inspired swarm algorithms for solving different searching problems [8,9].
In 2017 SanKalap Arora and Satvir Singh introduced the chaos into the BOA to improve the performance of the algorithm in terms of convergence speed and the local optima [10]. Later, in 2019 P.PRIYadharshini et al. presented an improved version of the BOA for capacitated vehicle routing problem using local search operator [11]. Also in 2019, another version of the algorithm was proposed to improve the performance of the BOA by K.M.Dhanya and S.Kanmani. the improved version presented a mutated butterfly optimization that is solving global optimization problems where the original method was combined with Canchy mutation to avoid entrapment in local optima [12].
The remainder of this paper is a series of the following: First, the original BOA is covered next section (2). Then section (3) describes the problem of path planning in general and path planning in 3D space in particular. Section (4) explains the proposed method of path planning using the BOA algorithm. Sections (5 and 6) discuss the testing environments and experimental results of applying the proposed method to the test environments, followed by Section (7) of conclusions and future works.

2. The butterfly optimization algorithm (BOA)

BOA is a recent nature-inspired algorithm for global optimization that falls in the meta-heuristic category [13], which is applied for solving the non-convex optimization problems in improved performance if compared with other conventional types of optimization techniques [7,14]. As modeled in the steps of the algorithm; BOA mimics the behavior of food foraging of butterflies and it has been proven to be one of the best recently used optimization algorithm [15,16,17].

2.1 Butterflies in nature

Up to this date; more than 18,000 species of them have been known, where their various senses have been found to be the reason behind their survival for millions of years. In the Linnaean system of Animal Kingdom; Butterflies belong in the Lepidoptera class [18,19]. They use different senses for mating partners and laying eggs in inappropriate places and finding food, migrating to different places, and escaping predators. These senses are hearing, smelling, taste, sight, and touch.
The smell sense is the most important for butterflies since it helps them to find their food (nectar) even from long distances. Butterflies sense smells thought their receptors which are actually nerve cells scattered over the butterfly's body such as antennas and legs. Those receptors are called Chemoreceptors and they guide butterflies for a better mating partner and such [20].

2.2 Fragrance [7,14,19]

The Butterflies' fragrance is a way for those species to share information among each other where they spread fragrances with different intensities attracting other butterflies to their location, the rest of the group will follow the strongest fragrance that they can sense in the arena and move accordingly and this is known as the "global search phase" of the BOA, otherwise; when the butterfly fail in sensing any scent within its arena then it will be moving randomly until a nearby scent is detected, and this is known as the "local search phase" of the algorithm. Fragrances in BOA have their unique scents which define their values and that distinguish this algorithm from any other meta-heuristics.
Butterflies' fragrances in BOA are affected by three basic parameters:
(1) Sensory-Modality (c): This refers to the measurement and process of the raw information input used by the sensor like light, sounds, smell, and temperature where modality is represented by fragrance.

(2) Stimulus-Intensity (I): is the magnitude of the solution which is represented by the butterfly in BOA, where the other butterflies can smell it and be attracted towards it.
(3) Power-exponent (a): it's a parameter where the intensity is raised to.
Increasing the intensity (I) will result in increasing the fragrance (f) increases faster, and that is a Response Expansion. While in Response Compression the fragrance (f) increases slower than (I) when (I) increase. The Linear Expansion is when both increase proportionally. In BOA the Response Compression is used predicting the value of (I), while (I) is associated with the objective function equation (1):

$$F=cIa \quad (1)$$

Where F is the fragrance magnitude (how the other butterflies sense it). (c) is the sensory-modality, and (I) is the stimulus-intensity powered by (a): the power-exponent showcasing the absorption degree. a and c fall in the range of [0,1]. a varies the degree of absorption, when a=1 shows that there is no absorption in the environment for the butterfly's fragrance, so the sensed value is at the same capacity of the source and it can be sensed the same anywhere in the environment space. And if a=0, the fragrance then is omitted and can't be sensed at all. Both a and c affect both the behavior and the speed of the algorithm and its' convergence. In maximization problems, the intensity is considered to be proportional to its objective function.

2.3 Butterfly movement [7,21]

The movement of butterflies in BOA is refined as the following:

1- For the butterflies to attract each other, they supposed to emit some fragrance.
2- Emitting more fragrance; the butterflies move towards the best (in the global phase) or randomly (in the local phase).
3- The environment of the problem defines and affects the stimulus-intensity of the butterflies.

2.4 The original butterfly optimization algorithm (BOA) [7,14]

This algorithm consists of three phases:

- The initialization phase: the parameters' setting and population initialization phase.
- The iteration phase: where the search is performed iteratively.
- The final phase: terminating the algorithm and finding the best solution.

The BOA algorithm initializes at the first phase where it defines the parameters, the workspace, and the objective function. Also initiating the butterflies' population as the size of the population is still unchanged throughout the algorithm execution and to store their information; some memory fixed size is allocated. The startup position for each butterfly is also created in this phase by being chosen in a random matter in the solution space which decides the value of their initial fitness and fragrance.
The iteration phase starts from here where the search is performed by the created butterflies' population. With each iteration performed by this algorithm, all the butterflies in the population are moved in the space changing their fragrance's values and the new values are computed by equation (1).
This algorithm has two key steps: Global and Local searches. Equation (2) shows the movement of the butterfly in the global search phase towards the best (fitted) one in the current iteration (g*):

$$xii+1= xit + (r2 \times g^*- xit) \times fi \quad (2)$$

xit represents the value of x in the iteration t for the butterfly (i) in the population. g* is the current best butterfly in the population. fi is the fitness of the butterfly (i). And r is just a random number within the period [0,1].
The local search allows the butterfly to walk randomly within the population as shown in equation (3):

$$xii+1= xit + (r2 \times xjt – xkt) \times fi \quad (3)$$

xj and xk represent the butterflies (jth) and (kth)in the population of the current iteration t, and r is a number randomly chosen within the period [0,1].
For butterflies' behavior in mating and finding food; the search can occur at a global or local scale, then a switch-probability parameter (p) applied to changeover between the local and global searching phases.

The iteration-phase is continued until matching the given stopping criteria, where it can be reaching a specific number of iteration without a noticeable change in the solutions, or reaching a specific execution time interval, or a particular error rate is reached.

The final phase starts as the iteration phase is concluded; then the best solution is found according to its fitness and the results are represented. Algorithm (1) shows the steps of the original BOA.

3. The problem of path planning

In order for the robot or an autonomous vehicle to perform different tasks independently while avoiding colliding with obstacles along the way; it has to be able to plan its path from one point to another and this represents the problem of path planning, which is considered to be one of the most challenging difficulties in the field of automatic motion planning [22].

The main goal of any path planning algorithm is to compute (plan) a path or multiple paths for the robot(s) to follow, where the problem space is represented as the problem environment and which can be described as the static or dynamic, known or unknown environment. Each environment includes different objects that may block the robot's way that is called obstacles which can be static or dynamic. The algorithm is to plan a collision-free path from a given starting position called the start-point up to the desired position that is called the goal-point [23].

3.1 Path planning in 3D Space

Path planning in 3D space means increasing the scale of the environment to work on a 3-axis dimension instead of the traditional 2D of x and y dimensions in terms of the environment objects' positions and movement equations and that can be a harder task for some algorithms and methods to solve.

In 3D space, each point is represented by 3-axis parameters (x,y,z), which form the position of the robot (solution) in the space. That gives the robot more freedom to navigate in a different direction each time. 3D path planning gives the algorithm realistic perspective where the robot can move freely in all direction that makes it possible for the robot to float in space or dive under the water and opens the doors for many possible applications in the field of the robotic and autonomous vehicle to work on bigger and more challenging platforms [24,1]. The nature-inspired algorithms are scored to be the best in solving such complex difficult problems [25,26].

Algorithm (1)

Initial phase

```
objective function f(x), x=( x_1, x_2,…, x_dim), dim= number of dimensions
generate initial population of n butterflies x_i (i=1,2,…,n)
define the value of stimulus intensity I_i at x_i by f(x_i)
define sensor modality c, the power exponent a and switch probability p
```

Iteration phase

```
while stopping criteria is not met do
for each butterfly bf in population do
Calculate fragrance for bf using equation (1)
end for
find the best bf
for each butterfly in bf do
generate a random number r from [0,1]
if r < p then
move towards best butterfly using equation (2)
else
move randomly using equation (3)
end if
end for
update the value of a
end while
```

The final phase

```
output the best solution found
```

4. Path planning 3D space using BOA (proposed method)

The Butterfly Optimization Algorithm is a recent meta-heuristics that is nature inspires which mimics the butterflies' behavior; mainly, their high sense of fragrance to the food and each other which gives the algorithm the robustness of feeding on information from the environment by sensing the goal (food) position, and of each other's feedback by sensing each other's' fragrance that each of the butterflies has and can be sensed by others to help them move towards the right direction.

In this paper, the problem environment is represented by 3D space that includes the start point (where all the butterflies launch from), the goal point (the final position need to be reached by one or more butterflies) and the obstacles which are different sizes of mass represented by sphere shapes for easier collision avoidance process and the environment itself is chosen to be sphere-shaped for the same reason.

The algorithm starts by initializing the butterflies' population which can be as big or small as the environment requires. Having a bigger population may slow down the algorithm, but allows finding a better solution and escaping local minima.

The other parameters are assigned like the stimulus intensity (I), and the sensor modality (c) which restrict and control the fitness of the butterfly (fragrance) according to environmental constraints. Another parameter to be assigned is the switching probability (p) that decides the movement of each butterfly to whether to follow the local or global search. The switch probability parameter can be decided by trial and error.

As mentioned; each butterfly has its own fragrance which can be computed by the fitness function equation (4), where the fragrance is affected by the distance of the butterfly to both the start and goal points, so each iteration as the butterfly change its position; the fitness (fragrance) of it will be changed accordingly and thus has to be computed again.

$$F = 3G + S \tag{4}$$

Where F is the fitness (fragrance) value, S represents the Euclidean distance from the start to the current placement of the butterfly, and G is Euclidean distance from the current position of the butterfly to the goal.

The best butterfly is assigned by comparing the fragrance of all the butterflies within each iteration and the butterfly with the smallest fitness is the best in minimization problems, and the butterfly with the largest fitness is considered the best in maximization problems.

Within each iteration, a number (r) is randomly generated for every butterfly, and then compared to the value of the switching-probability (p), where if the r is less than p then the butterfly must move in the global search phase equation (2), otherwise; the butterfly will move in the local search phase equation (3).

The movement of butterflies continues each iteration following the local or global search until the goal is reached by one or more butterflies. The algorithm then will be terminated and the best path of the butterflies that reaches the goal is chosen by computing the paths' length and grasping the shortest among them.

Obstacle avoiding mechanism is included within the butterflies' movement, where at each local or global move; the new position for the butterfly is found and checked to be obstacle-free before it can be assigned to the butterfly, otherwise if the position found falls in the space of an obstacle or outside the parameter of the environment; then the equation must be applied again with different random values and a new position for the butterfly can be found and checked again and so on until new safe position is acquired and assigned to the butterfly to move to. Thus the final path generated to be a completely safe obstacle-free path.

The proposed algorithm is shown in step in Algorithm (2).

5. Testing environments

Path planning is known to be applied in any scenario, where the robot can move indoors: like storage buildings, hotels, hospitals, and so on or outdoors where it can be flying in the air, diving in the outer space, or underwater. For this proposed algorithm we have chosen some basic and crowded sphere environments to test the performance in different scales of complexity. Table (1) shows the different environments and their descriptions. Fig. 1 shows the different testing environments, where the environments from 1-4 represent different cases that can cause local minima in other methods like APF (Artificial Potential Field), and the results show how the proposed method overcomes all the environment types and find proper path each time.

Algorithm (2)

Initial phase

```
environment modeling in 3-Dimensional space of (x,y,z) format
define the Start point and the Goal point
mapping the obstacles into a list obs_m (m=1,2,…,number of obstacles)
define the objective function f(x) by equation (4)
generating initial population of n butterflies x_i (i=1,2,…,n)
define the value of:
    -   Stimulus-intensity I
    -   Sensor-modality c
    -   Power-exponent a
    -   Switch-probability p
```

Iteration phase

```
while (iteration_number<1000 and flag≠1) do
    for all butterflies bf in the population do
            Calculate bf fragrances using the objective function f(x)
    end for
    find the best g* with the best fitness (minimum value)
    for all butterflies bf in the population do
            randomly generate a number r from [0,1]
            if r < p then
                    obs_flag=0
                    While obs_flag≠1
                    Generate new position using equation (2)
                    if (new_position ∩ obs_m =1) then
                            obs_flag=1
                    end if
                    end while
            else
                    obs_flag=0
                    While obs_flag≠1
                    Generate new position using equation (3)
                    if (new_position ∩ obs_m =1) then
                            obs_flag=1
                    end if
                    end while
            end if
    end for
end while
```

The final phase

```
calculate paths' length for all found solutions
determine the best path according to the path length
output of the results
```

Table 1. Testing Environments

Title	Number of obstacles	Description
Map 1	1	These environments test the general performance of the algorithm with simple environments
Map 2	2	
Map 3	3	These environments with local minima possibility
Map 4	4	
Map 5	10	this environment test the performance of the algorithm with crowded obstacles

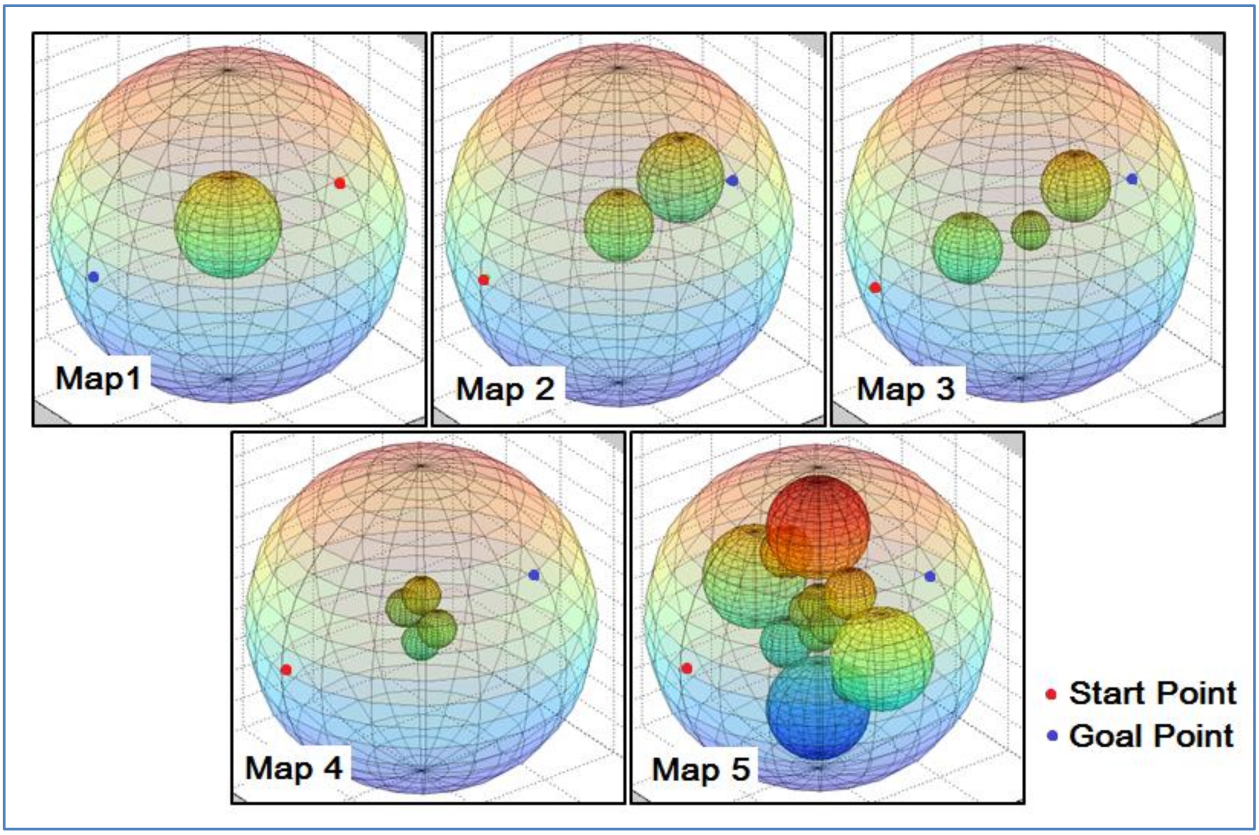

Figure 1. Shows the different test environments

6. Experimental results

The previous section explains the various environments where the proposed method was applied and tested. The best path was determined by the butterfly that reaches the goal first and if more than one butterfly finds the goal at the same time then the best solution is selected by the shorter path among them.
The parameters chosen and used in the algorithm are shown in Table (2). And the results of applying the algorithm to each of the test environments are shown in Table (3) in terms of execution time (in seconds) and path length (in length units), and the figures of each environment (2-4) are directed in that table.
The results are showing that the proposed algorithm was able to find a path from the start across to the goal without colliding with any of the obstacles along the way.
Also, the algorithm has a great deal of randomness in the movement of butterflies, represented by the random number in the local and global moving equations. This allows for the more random path wherein one way it is a longer path than needed, but on the other hand, it allows escaping the local minima problems by trying a new position with a rate of randomness in the direction each time.

7. Conclusion and future works

In this paper, a method of path planning in 3D space is represented where a recent nature-inspired algorithm is used and known to be the Butterfly Optimization Algorithm (BOA). The proposed method has successfully found a path in all of the testing environments. The random movement of the butterflies in the population gave the algorithm robustness in finding the path and avoiding local minima problems but, the generated path has extra length and rigid edges. Also, the population size has a great effect on the execution time and path length, since a smaller population can't find the goal easily and that adds to the time and path length, a large population can help find the path easier but it will add extra computation time expense.
For future works; the final path can be optimized in terms of path length, also a mechanism can be added to smooth the final path like the B-Spline method, and the general performance can be improved for better convergence speed.

Table 2. Algorithm Parameters

Seq.	Parameter	Value
1	Dimension	3D
3	Butterfly population size	10
4	Stimulus intensity I	Fitness (f)
5	Sensor modality c	3
6	Power Exponent a	1
7	Switch probability p	0.8
8	Random number r	[0.1]
9	Convergence	0.05 (length unit)

Table 3. Execution results

Map Title	Execution time (sec.)	Path length (units)	Start point position	Goal point position	Figure number
Map 1	0.9	33	(8,1,0)	(-9.5,1,0)	2
Map 2	32	35	(8,0,0)	(-9.5,1,0)	
Map 3	1.2	31	(8,0,0)	(-9,1.5,-1.5)	3
Map 4	1.3	32	(8,0,0)	(-9.5,1,1)	
Map 5	4	30	(8,0,0)	(-9.5,1,1)	4

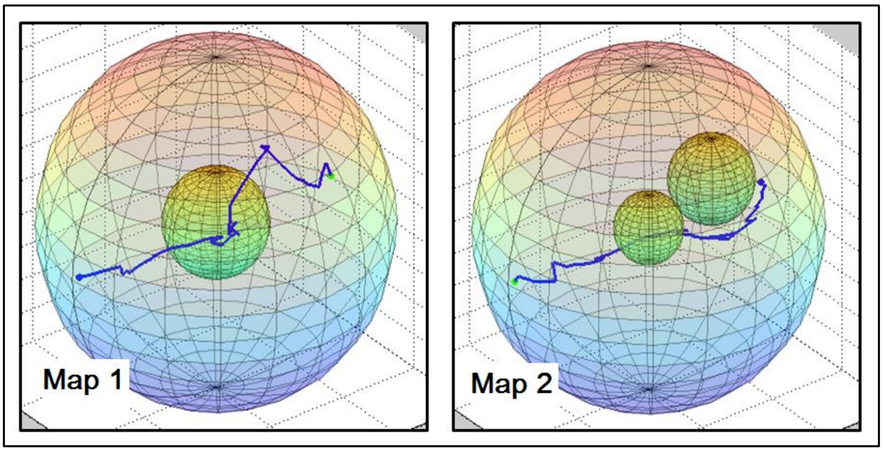

Figure 2. Algorithm Execution in Maps (1,2)

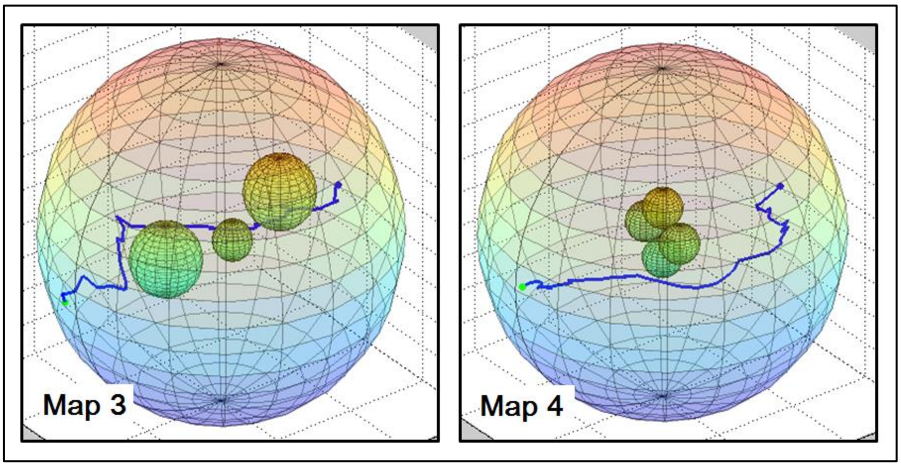

Figure 3. Algorithm Execution in Maps (3,4)

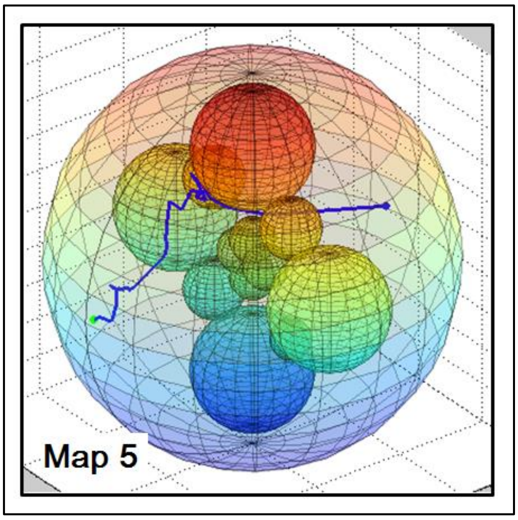

Figure 4. Algorithm Execution in Map (5)

References

[1] Pan J., Chu S., “A parallel compact cuckoo search algorithm for three-dimensional path planning”, Applied Soft Computing, Vol 94, DOI: 106443,September 2020.

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

[3] Zhang H., Lin W. and Chen A., "Path Planning for the Mobile Robot: A Review", MDPI journal, Symmetry, 10, 450; doi:10.3390/sym10100450, 2018.

[4] Honglun W., Wentao L., Peng Y., Xiao L. and Chang L., "Three-dimensional path planning for unmanned aerial vehicle based on interfered fluid dynamical system", Chinese Journal of Aeronautics,28(1): 229–239, 2015.

[5] Zhang B. and Duan H., "Three-Dimensional Path Planning for Uninhabited Combat Aerial Vehicle Based on Predator-Prey Pigeon-Inspired Optimizationin Dynamic Environment", IEEE/ACM Transactions on computational biology and bioinformatics, VOL. 14, NO. 1, January/February 2017.

[6] Weifeng S., Min T., Lijun Z., Zhiqiang H. and Lei S., “A Survey of Using Swarm Intelligence Algorithms in IoT”, MDPI-journal-sensors, doi:10.3390/s20051420, March 2020.

[7] Arora S., and Singh S., "Butterfly optimization algorithm: a novel approach for global Optimization",Springer-Verlag GmbH Germany, part of Springer Nature 2018.

[8] Arora S., Singh S., “Node Localization in Wireless Sensor Networks Using Butterfly Optimization Algorithm”. Arab J Sci Eng 42, 3325–3335, 2017.

[9] Fathy H., “Butterfly optimization algorithm based methodology for enhancing the shaded photovoltaic array extracted power via reconfiguration process“, Energy Conversion and Management, Volume 220, 15 September 2020, 113115.

[10] Arora S. and Singh S., "An improved butterfly optimization algorithm with chaos", Journal of Intelligent & Fuzzy Systems 32, 2017, 1079–1088, DOI:10.3233/JIFS-16798, IOS Press.

[11] Priyadharshini P., Prakashraj K. and Padmapriya S.,"Improved Butterfly Optimization Algorithm Using Local Search Operator For Capacitated Vehicle Routing Problem", International Research Journal of Engineering and Technology (IRJET), e-ISSN: 2395-0056, p-ISSN: 2395-0072, Volume: 06 Issue: 04 | Apr 2019.

[12] Dhanya K. M., and Kanmani S., "Mutated Butterfly Optimization Algorithm", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-8 Issue-3, February 2019.

[13] Wang Z., Luo Q. and Zhou Y. “Hybrid metaheuristic algorithm using butterfly and flower pollination base on mutualism mechanism for global optimization problems”. Engineering with Computers, 2020. https://doi.org/10.1007/s00366-020-01025-8.

[14] Arora S. and Singh S., “An Effective Hybrid Butterfly Optimization Algorithm with Artificial Bee Colony for Numerical Optimization”, International Journal of Interactive Multimedia and Artificial Intelligence, Vol. 4, Nº4, DOI: 10.9781/ijimai.2017.442, March 2018.

[15] Yıldız B. S., Yıldız A. R., Albak E. I., Abderazek H., Sait S. M., Bureerat S., “Butterfly optimization algorithm for optimum shape design of automobile suspension components”, Materials Testing,Vol 62,Carl Hanser Verlag, München, April 2020.
[16] El-Hajjami L., Mellouli El-M., Berrada M., ” Optimal PID control of an autonomous vehicle using Butterfly Optimization Algorithm BOA”, BDIoT'19: Proceedings of the 4th International Conference on Big Data and Internet of Things, ISBN:9781450372404, DOI:10.1145/3372938, October 2019.
[17] Arora S. and Anand P., “Learning automata-based butterfly optimization algorithm for engineering design problems”, International Journal of Computational Materials Science and EngineeringVol. 07, No. 04, 1850021 (2018).
[18] Saccheri I., Kuussaari M., Kankare M., Vikman P., Fortelius W. and Hanski I., "Inbreeding and extinction in a butterfly metapopulation", NATURE |VOL 392 | 2 APRIL 1998.
[19] Zhi Y., Weiqing W., Haiyun W. and Khodaei H., “Improved butterfly optimization algorithm for CCHP driven by PEMFC”, Applied Thermal Engineering, Vol 173,114766, 5 June 2020.
[20] Pollard E. and Yates T., "Monitoring Butterflies for Ecology and Conservation: the British butterfly monitoring scheme", Springer, Berlin,Chapman & Hall, London 1993.
[21] Singh B. and Anand P., “A novel adaptive butterfly optimization algorithm”, International Journal of Computational Materials Science and Engineering.
[22] Sharayu Y. Ghangrekar, " A path planning and obstacle avoidance algorithm for an autonomous robotic vehicle", Conference Paper, DOI: 10.1109/MASCOT.2009.5366693 · Source: DBLP, September 2009.
[23] Song G.," A Motion Planning Approach To Protein Folding", Ph.D. Thesis, Texas A&M University, December 2003.
[24] Van den Berg J. P.," Path Planning in Dynamic Environments", Ph.D. Thesis, Netherlands, Drukkerij Bariet, Ruinen, ISBN: 978-90-393-4480-4, 2007.
[25] Mohammadi F.G., Amini M.H. and Arabnia H.R. “Applications of Nature-Inspired Algorithms for Dimension Reduction: Enabling Efficient Data Analytics”. In: Amini M. (eds) Optimization, Learning, and Control for Interdependent Complex Networks. Advances in Intelligent Systems and Computing, vol 1123. Springer, Cham. https://doi.org/10.1007/978-3-030-34094-0_4, 2020.
[26] Al Gburi Hussein Qahtan and AL-Baiati Ali Emad "Path-Planning Dynamic 3D Space Using Modified A* Algorithm", 2020 IOP Conf. Ser.: Mater. Sci. Eng. 928 032016, doi.org/10.1088/1757-899x/928/3/032016.