Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 133 (2018) 230–239

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Robotics and Smart Manufacturing (RoSMa2018)

Three Dimensional Path Planning for UAVs in Dynamic Environment using Glow-worm Swarm Optimization

Utkarsh Goel, Shubham Varshney, Anshul Jain, Saumil Maheshwari, Anupam Shukla

Advanced Computing Lab, ABV - Indian Institute of Information Technology and Management, Gwalior 474015, India

Abstract

We propose an efficient solution for finding a collision-free path in a Three-Dimensional environment with dynamic obstacles for Unmanned Aerial Vehicles (UAVs). Path Planning for Unmanned Aerial Vehicles (UAVs) in Three Dimensional Dynamic Environment is considered a challenging task in the field of robotics. During their mission, UAVs have to maneuver in an environment which can have obstacles of varying size and random motion. The aim of the proposed algorithm is to traverse an optimal flight route in real world environment with no collision with environmental elements. This paper proposes use of a Glow-worm Swarm Optimization (GSO) for Path-Planning of Unmanned Aerial Vehicles (UAVs). It provides improved convergence rate and accuracy than the other Meta Heuristic optimization algorithms. The simulation is modelled in a real world environment. A swarm of particles is made to co-ordinate with each other for optimal path planning. The simulation is run in Python and the viability of the algorithm according to path-cost, time and number of expanded nodes is measured.

Keywords: Unmanned Aerial Vehicles (UAVs); 3D-Path Planning; Dynamic Environment; Nature Inspired Algorithms; Glow-worm Swarm Optimization (GSO)

1. Introduction

Unmanned Aerial Vehicles (UAVs) are mobile robots which can be controlled remotely. Due to their ability to work in complex environments, they are gaining attention in various areas such as military, scientific research, tracking operations etc. They are more versatile than autonomous land vehicle (ALV) which work in 2D environment. UAVs can work in complex terrain, plan path of their own while avoiding obstacles in between. Though they require comparatively large computation time but their immense use has over shadowed it. In present scenario, the use of UAVs has been confined to a limited number of applications. They are operated in areas that are free of any other obstacle outside the control of the authority in charge of the UAV. If the use of UAVs has to be extended to general

* Corresponding author. Tel.: +91-9540018654 ;
E-mail address: ugoel96@gmail.com

1877-0509
10.1016/j.procs.2018.07.028

use, UAVs must be able to not only avoid fixed obstacles but also deal with dynamic obstacles. While they are flying with high velocity, they should be capable of reacting to these types of variable changes quickly. The autonomous control modules for Unmanned Aerial Vehicles (UAVs) use three dimensional path planners in order to generate their path and make the vehicle fly in the respective terrain. The path planner is responsible for traversing an optimal path from a source to goal while avoiding the obstacles/no-fly zones in between. The UAVs unlike the commercial airlines which fly in a constant and pre-specified trajectory, fly to different terrains with different trajectories according to the situation and the conditions of the flight. UAV path-planning is considered a more complex task to solve due to its nature and contains several complexities which have only recently being started to be considered. These factors can range from differential constraints like wind speed, atmospheric irregularities, wind pressure, etc. which affect and hinder the normal trajectory follow-up of a UAV, and limit the information gathering from the sensors present in the vehicle.

The resulting gap between finding the exact solution and generating viable approximate solution is far from being completely addressed. Hence, it has been a challenge to efficiently decide a guidance system. Path planning is to be done in real time with zero degree of negligence and hence it falls under the category of hard optimization problems. For these types of problems, the most logical solution would be to use algorithms which are not deterministic but their solutions are near optimal and not too hypothetical to be implemented. A strategy needs to be devised by a robot to plan path while avoiding the obstacles and thus reach the desired goal safely. There might be various other complexities like the goals may be in motion, they may disappear at run-time, etc. While Land Robots / Autonomous Control Vehicles (ACVs) are confined to Two Dimensions, water and air robots have to deal in three dimensional space. The UAVs are fitted with sensors and high-end programmed Path Planners which have the ability to change their behavior by devising an optimal low cost path. Other features like coordination, task assignment, etc. are fitted along with path planner to aid in path planning. The path planning can be done in two ways, Static Path Planning a.k.a. Offline Path Planning which deals with planning the trajectory beforehand. The pre-generated trajectory is then traversed at run-time. Other type of path planning is Dynamic Path planning a.k.a. Online Path Planning which deals with planning the trajectory on the go. As three dimensional path planning is to be done in real time considering the dynamic nature of the environment, it can be defined as a NP-Hard problem. The path is in the form continuous connected points which offers an optimal route from source to destination. Path is smoothened to spline curves by applying various algorithms. Existing solutions for the problem include the use of visibility graphs, randomly exploring algorithms, Probabilistic Road Maps, Dijkstra Algorithm which is a deterministic search algorithm, A* and D* which are Heuristic based search algorithms and various other meta heuristic algorithms. [4] [17]

1.1. Swarm Intelligence

Swarm Intelligence, which is collective behaviour of self organized, decentralized particles, is commonly observed in nature. The swarm intelligence algorithms deploy a search based strategy which use a distributed approach where every agent operates autonomously [17]. They in turn collaborate with surrounding neighbours to explore the environment. In this strategy, the agents operate in following two phases: First stage deals with the agents performing an exploration behaviour. While exploring, they seek data and check if it is above a fixed pre-defined threshold. These agents broadcast the sensed data to their neighbors by means of different communication channels (luciferin in case of glow-worm, pheromone in case of ants, etc.) available to them. This is received by other agents of the swarm in the neighborhood range. If the agent detects a value higher than the pre-defined threshold, it changes its state to search state from exploration state. If not, agent continues with the exploration behaviour until it receives or senses a data value beyond the pre-defined threshold.

In the Search phase, to find the optimal data source, the agents start to collaborate with their neighbors. In order to continue the search, each agent uses its own data and data received from its neighbors to find a promising direction to move. If it is greater than its own sensing value, the agent switches its search direction towards the position of the agent which provides the maximum and hence more promising value in its neighborhood. If not, the agent continues the search in its current path. The data of an agent is refreshed in every T seconds, that is, the agent keeps on following its current search path for at least T seconds. Once the value is refreshed, it decides whether to continue on existing path or switch to a more promising path. On collaborating with surrounding agents, after detecting data values above the threshold, the agent becomes a member of a virtual team that is exploring a particular promising area of the environment which in turn leads to the autonomous emergence of different teams of cooperating agents, which is the main philosophy behind swarm intelligence.

1.2. Glow-worm Swarm Optimization

Glow-worm Swarm Optimization (GSO) is a newly introduced Nature Inspired Algorithm based on Swarm Intelligence, developed by K.N. Krishnand and D. Ghose [5]. This algorithm is inspired by the foraging behaviour of lightning worms know as Glow-worms. These insects have a special property of emitting light using a chemical luciferin, (a light-emitting compound that generate bio luminescence) present in them. They are able to adjust the quantity of luciferin and thus can control the amount of light emitted by them and use it for attracting their prey, moving in swarms, etc. Unlike other algorithms which locate global solution in the objective domain, GSO algorithm finds multiple optimal solutions having same or different objective function values, which adds to its importance. The GSO algorithm works by dividing its swarm of population in disjoint groups which converge at different points containing the local optima to find multiple solutions.

The algorithm and its variants are observed to be highly efficient considering speed, convergence rate, quality of solution etc. The algorithm has comparatively lower chance of local or premature optima convergence. A glow-worm is attracted towards a brighter glowing glow-worm in its neighborhood which has luciferin content higher than its own.

A glow-worm identifies another glow-worm as a neighbor, when it is located within its current local-decision domain. The higher the intensity of luciferin, the better is the location of glow-worm in the search space. Each iteration is characterized by changing position of glow-worms and an update in luciferin value. The algorithm can be divided into a total of four phases: -

1.2.1. ***Glow-worm Distribution Phase***

This phase deals with randomly dispersing glow-worms in the environment. Each glow-worm is initialized with an initial luciferin value l_0 and a range of r_0.

1.2.2. ***Luciferin update Phase***

Each iteration is characterized by updating the luciferin value of each glow-worm according to luciferin decay constant and an objective function. The luciferin update phase can be stated as follows:

$$l_i(t) = (1-\rho)l_i(t-1) + \gamma J(t) \tag{1}$$

Here, the luciferin value of the i^{th} glow-worm is denoted by $l_i(t)$, the luciferin value of i^{th} glow-worm at time (t-1) is denoted by $l_i(t-1)$, γ denotes the luciferin enhancement constant, ρ denotes luciferin decay constant and $J(t)$ denotes the objective function value of i^{th} glow-worm at time t.

1.2.3. ***Movement Phase***

A glow-worm is attracted to a neighboring glow-worm with luciferin content higher than luciferin content of its own. The probabilistic mechanism is used to choose a neighbor in the direction of which the glow-worm moves. The movement of glow-worms is decided according to the probability equation which is stated as follows:

$$p_i j(t) = (l_j(t) - l_i(t))/(\sum\nolimits_{k\in N_t}(l_k(t) - l_i(t)) \tag{2}$$

Here, j belongs to a set $N_i(t)$ given as,

$$j \in N_i(t), N_i(t) = j : d_{ij}(t) < r_d^i(t); l_i(t) < l_j(t) \tag{3}$$

Here,the distance between glow-worm i and glow-worm j at time t is denoted by $d_{ij}(t)$ and the decision/search range of the glow-worm is denoted by r_d. The following path update equation by which glow-worm moves in direction of its selected neighbor is:

$$x_i(t+1) = x_i(t) + st * ((x_j(t) - x_i(t))/ \parallel x_j(t) - x_i(t) \parallel) \tag{4}$$

Here,the location of the glow-worm i during the t^{th} iteration is denoted by $x_i(t)$ which is a real number, st denotes the step size and the euclidean distance operator is denoted by $\parallel . \parallel$.

*1.2.4. **Neighborhood Range Update Phase***

This phase comprises of updating the neighborhood range of the corresponding glow-worms. The range update equation is given as:

$$r_d^i(t+1) = min\{r_s, max\{0, r_d^i(t) + \beta(n_t - N_i(t))\}\} \tag{5}$$

where the range of the glow-worm during the $(t+1)^{th}$ iteration is given by $r_d^i(t+1)$, r_s defines the minimum sensory radius, β denotes a constant and n(t) denotes the parameter to control neighborhoods' quantity for the glowworm.

2. RELATED WORK

Path Planning for Unmanned Aerial Vehicles (UAVs) is similar to solving the well-known combinatorial optimization Traveling Salesman Problem (TSP). 2 Lots of work has been done in path planning in 2D and 3D environment for static obstacles. Krishnand and Ghose who proposed the GSO algorithm 5, tested it rigorously on several multi-modal functions and observed its performance improvement over other meta-heuristics. Aljarah and Ludwig did research on adjusting GSO to solve the data clustering problem and in turn locate multiple optimal points.1 UAV path planning in two dimensional environment using modified GSO algorithm was proposed by Tang and Zhou.12 Hybrid Cuckoo Search algorithm has been used alongside GA operators to prevent falling into local optimum and aid in path planning.16 Gai-Ge Wang proposed an improved bat algorithm for 3Dimensional path planning algorithm in combating environment by using mutation operations of DE.14 A parallel genetic algorithm was proposed by Sanci and Isler which uses a single UAV to find out a solution.11 A research proposes neural network based controllers to train and guide robot through static obstacles. Ants pheromone update and tour construction phase can be used on processing elements for path planning purpose. An approach proposes use of evolutionary dynamic navigation planning algorithm, which is quite efficient in responding to concurrent constraint updates in a short period of time.13 Behavior of ants has been modelled and used as heuristic information which helps in searching the map quite efficiently. 15 For solving the Two Dimensional Robot Navigation problem, fuzzy based controllers are used along with an evolutionary procedure to optimize the parameters involved. A novel algorithm based on Trajectory propagation and disturbed fluid based novel algorithm also has been proposed to solve Three-Dimensional path planning problem in static environment. A research also presents a hyper-heuristic approach to develop a 3D online path planning for UAV navigation using on-board sensors. 3 Glow-worm Swarm Optimization has an edge over other Nature Inspired algorithms as it converges faster and is hence more efficient considered to PSO, BBO, ACO, etc. 10 GA has been widely used for Robot path planning. Mathematical analysis of GA has been described in 6, 8,7, 9 We propose to solve the path planning problem for Three Dimensional Environment with Dynamic Obstacles using Glow-worm Swarm Optimization which is Nature Inspired Algorithm based on Swarm Intelligence.

3. RESEARCH GAPS

The research gaps fulfilled to justify the novelty of the research are describer here. Even the best possible solution obtained using Deterministic Algorithms dont guarantee feasibility and solution obtained using meta-heuristic algorithms is sub optimal. Though they perform quite well for path planning problem and other situations, there has not been an extensive research on the use of nature inspired algorithms and evolutionary algorithms. Path planning for UAVs is normally performed in two dimensions with static obstacles and more research is needed to be done in the case of the three dimensional environment with dynamic obstacles to model the problem in real world. Though some other algorithms have been used to solve this problem, Nature Inspired Algorithms still need to be implemented in this area.

4. METHODOLOGY

4.1. Cost Function

The main aim of path planning problem is to minimize the cost taken by UAV to reach from source to goal.The cost of the potential solution depends on numerous factors. It may be possible that there is no path between the source and the goal because of UAV getting stuck in an obstacle. In this case, the cost is infinite. The cost depends on path-length and altitude. The cost factor for traversing in x, y and z direction is Cx, Cy, Cz respectively. We calculate a scale factor 'sf' from Cx, Cy and Cz, which in turn is then used to generate path with optimal cost. Various parameters used in cost function are:-

1. cx, cy, cz : Directional cost factors for the x-, y-, and z- directions. These three factors are used to scale the cost of travel by the value provided beforehand.
2. t_max: It is a soft limit on the path - planning time. Path refinement is terminated when calculated value exceeds t_max.
3. total_cost: Net cost UAV incurs while traversing from source to the goal node. (The shorter the path traversed, lesser is the cost and thus less is the number of nodes expanded), time required for reaching goal, and fuel consumption from source to goal.

The solution to the path planning problem is in the form of collision free networked points connecting source 'S' to destination 'G'. The solution is denoted as L(x;y;z) such that:

$$L(x;y;z) = S, P1, P2 , P3 , P4 ,....Pn/G$$

where S, P1, P2 , P3 , P4 ,....Pn/G are the intermediate points in the collision free path traversed from source to destination.

The cost function for the algorithm is given by the equation:

$$cost = minimum\_of\_all\_particles(scale\_factor * sqrt(dx * dx + dy * dy + dz * dz)) \tag{6}$$

where dx, dy and dz are the distances moved in the x, y, z-direction respectively. Scale_factor, or sf, is calculated according to cX , cY and cZ.

The aim of the algorithm is to choose a minimal cost path. Every agent calculates cost of reaching the goal, and the swarm changes its direction to the direction of agent with minimum cost.

4.2. Path Generation for the Glow-Worms

The solution of the path planning problem is a set of networked collision free points from source goal. For each agent, there are six potential directions in the 3-Dimensional space where it can move. Figure 1 shows agent behaviour using swarm strategy. Technically, a grid/node is the shortest distance/point defined on map. It is smallest distance the agent can move. Thus, the search space can be considered to be divided into a grid based structure. Algorithm 1 defines the path generation algorithm using GSO.

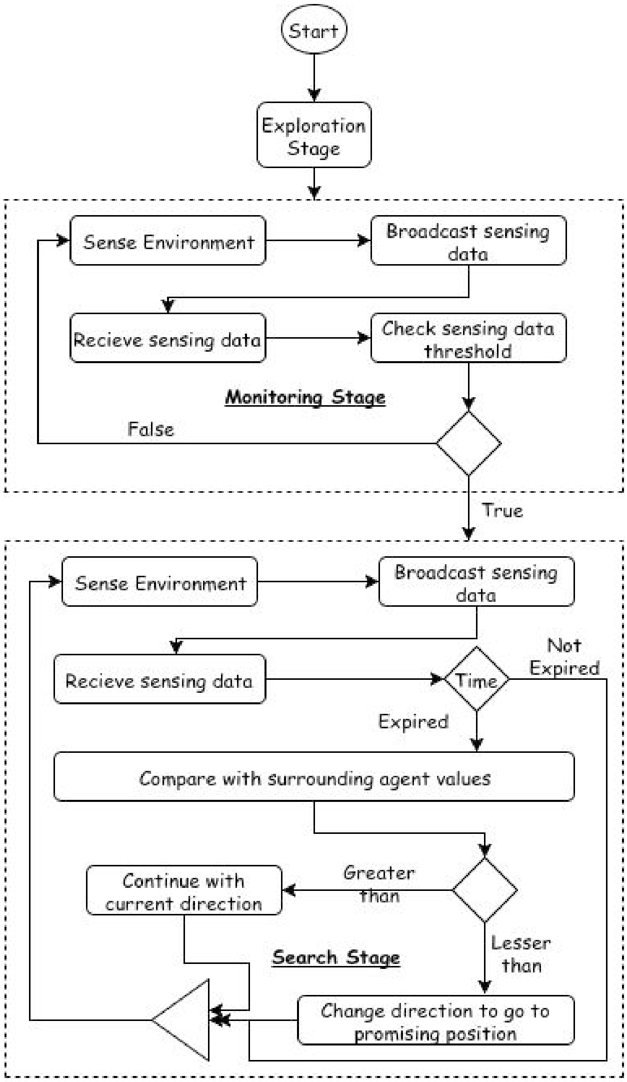

Fig. 1: Agent behavior using swarm strategy

Algorithm1: Path Generation using GSO

Step 1: Initialization. Initialise the environment, path solution matrix P, source node 'S', the initial luciferin value of the glow-worms l_o, luciferin decay constant β and luciferin enhancement value γ.

Step 2: Search Phase. Each agent marks the nodes occupied by obstacle as blocked in its search range in 'blkdNode' matrix and the cost of these nodes is set to infinity. Luciferin content of each particle is updated according to luciferin enhancement factor, luciferin decay constant and the cost (calculated from the cost function) of reaching goal from current node to goal node *Equation (6)*.

Step 3: The algorithm chooses a glow-worm with the maximum luciferin content and the swarm changes its direction to the direction of chosen glow-worm.

Step 4: If the current node is goal, simulation is successful, else jump to Step 2.

5. EXPERIMENT AND RESULTS

5.1. ENVIRONMENT

An environment needs to be set up, for path planning of UAV where the simulation will be carried out. To deal with the high costs of performing real test flights, we have performed simulation which offers to test different approaches under the same constraints and conditions. The environment comprises of many regions where the motion of the UAV is restricted. The UAV is supposed to avoid these regions, known as obstacles, during its path planning. These obstacles in real world environment can either be buildings, trees, no fly zones, etc. In real world environment, the dimensions of obstacles is not perfectly geometric, but for simplicity, we model all the obstacles as cuboid. However, care is taken to prevent collision of UAV with obstacles of uneven dimensions. During environment modelling and testing, all obstacles are encased in a cuboid. The environment, as depicted in Fig2 and Fig3, is designed in a 3-Dimensional Co-ordinate space. Every point can be represented by its co-ordinate point which is of the form P = (x, y, z) where x, y, z are x, y and z axis respectively. The 3D space is represented as a grid and each grid is assigned an integral value beforehand. A grid which contains an obstacle is marked as blocked and the grids free of obstacles are marked as safe. There is a single source node which is denoted by the point S = (Xs, Ys, Zs) while there can be multiple goals with each goal denoted by G = (Xgi, Ygi, Zgi) where i stands for ith goal. The path planning deals with devising an algorithm to find a path from source S to goal G. The map consists of some pre-defined static obstacles. Dynamic obstacles of varying size and random motion are added during run-time and gradually fill up the environment space.

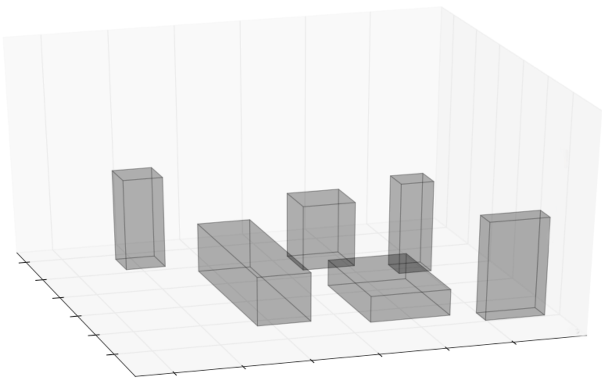

Fig. 2: An Environment with pre-generated Static Obstacles.

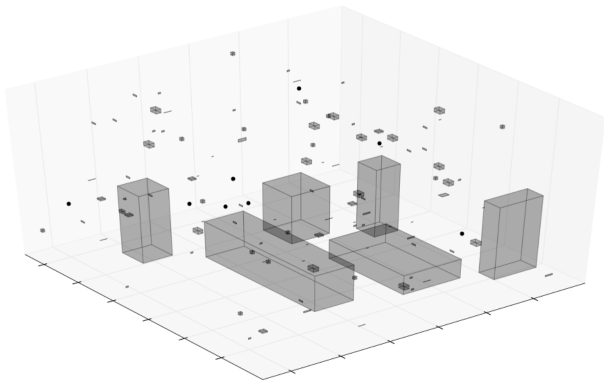

Fig. 3: An Environment with Dynamic Obstacles(Black Dots) and randomly generated static obstacles(Small grey cuboids) and pre-generated obstacles(large grey cuboids).

5.2. EXPERIMENTAL DESIGN

Python 2.7 is used to conduct the simulation. The simulation was performed on computer with 1.7 GHz x 4 of CPU, 4GB of RAM and 2GB(NVIDIA GeForce 820M) Dedicated Graphics memory. The parameters for experimentation taken are defined as follows:

1. Number of agents : 5
2. Initial Luciferin Value : 30
3. Range : 4.0
4. Luciferin decay constant : 0.3
5. Luciferin enhancement factor: 0.7
6. Search radius : 20

5.2.1. *Experiment 1*

The algorithm was tested on path planning for 3 Dimensional Environment with Static Obstacles and multiple static goals. Obstacles were predefined in the map though the particles had no prior knowledge of their location. Fig 4 shows the simulation results.

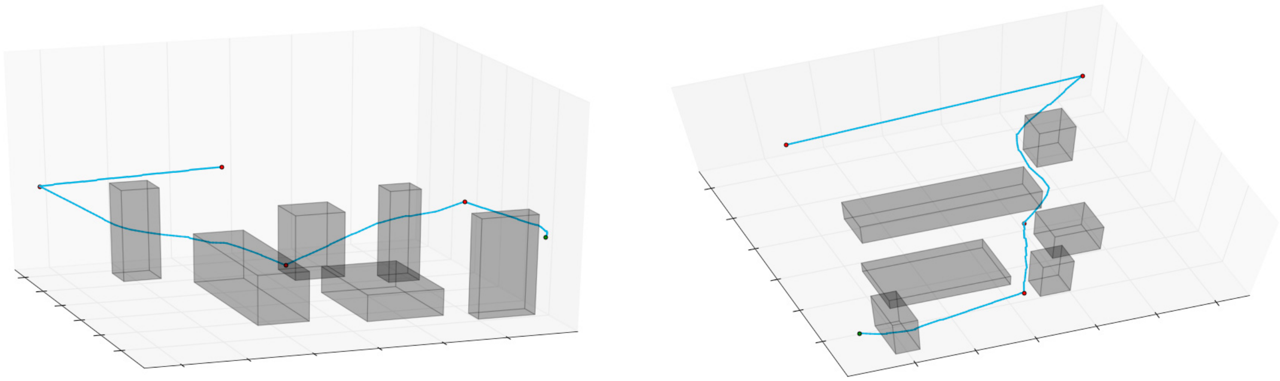

Fig. 4: Path planning with static obstacles and multiple static goals. Grey cuboids are static obstacles. Green dot is source and red dots are goals.

5.2.2. *Experiment 2*

The algorithm was tested on path planning for 3 Dimensional Environment with randomly generated static obstacles of varying sizes during run-time and multiple static goals. Obstacles are randomly generated during run-time in addition to the pre-defined static obstacles. Fig 5 shows the simulation results.

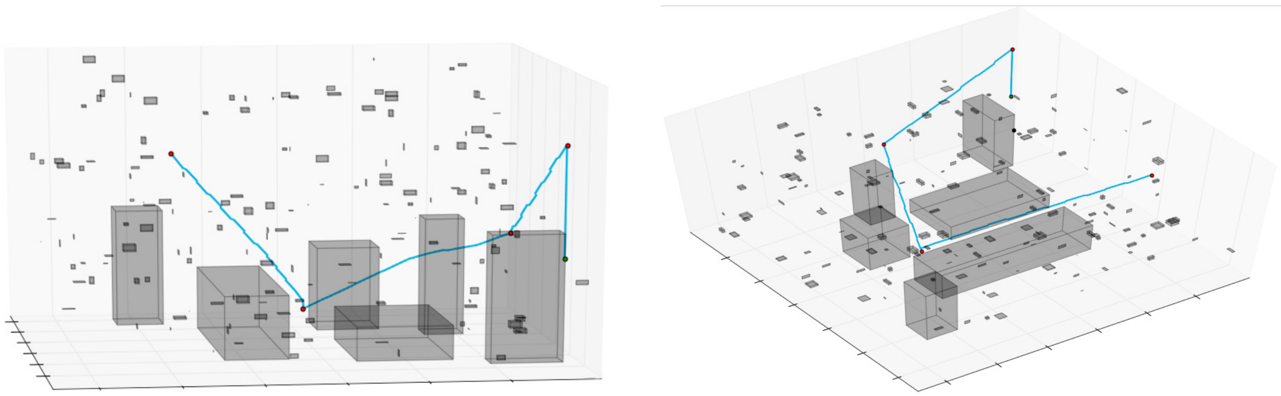

Fig. 5: Path planning with randomly generated static obstacles during runtime and multiple static goals. Grey cuboids are static obstacles. Green dot is source and red dots are goals.

5.2.3. *Experiment 3*

The algorithm was tested for path planning for 3 Dimensional Environment with Dynamic Obstacles and multiple moving goals. Obstacles are randomly generated during runtime in addition to the pre-defined static obstacles. Goals move randomly in the environment. Fig 6 shows the simulation results.

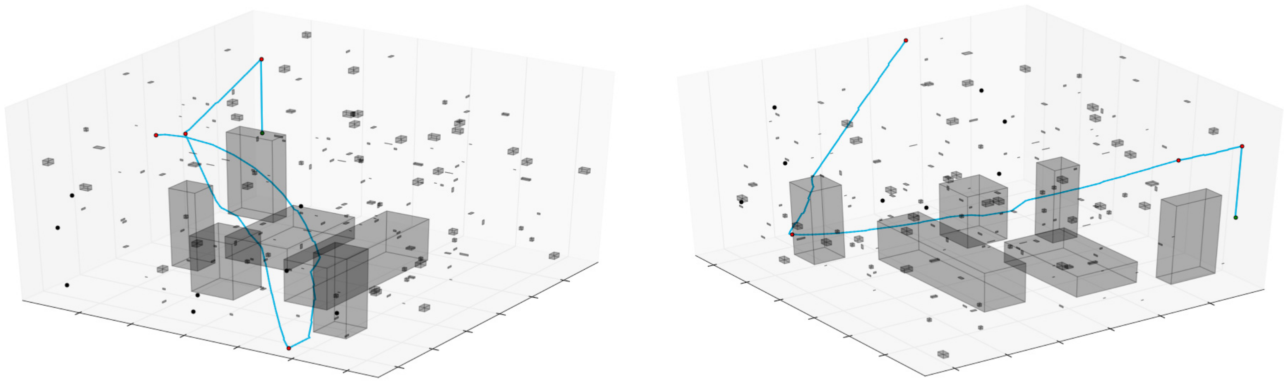

Fig. 6: Path planning in completely dynamic environment with dynamic obstacles and multiple goals in motion. Grey cuboids are static obstacles. Black dots are dynamic obstacles, green dot is source and red dots are goals.

5.3. RESULTS

Table 1 shows the results obtained when the algorithm is applied for generating path for the mobile robots. Figure 7 shows the Graphical Analysis of results by comparing the results obtained in the three experiments. As the complexity of environment increases, more nodes have to be searched for finding optimal route which is evident from the number of nodes expanded in corresponding simulation. Also, increased environment complexity suggests an increased cost and time which is evident too from the results.

Table 1: Analysis of results

Experiment Number	Time (sec)	Cost(units)	Number of Expanded Nodes
Experiment 1	34.84	311.34	13908
Experiment 2	43.58	337.24	21435
Experiment 3	48.37	406.44	28005

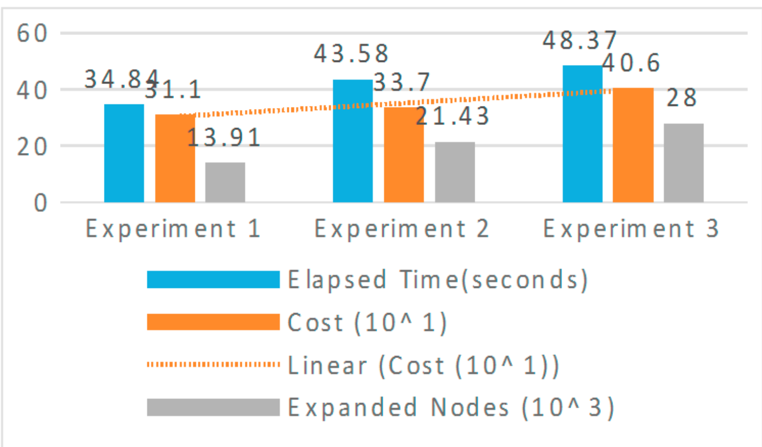

Fig. 7: Graphical Analysis of results.

6. CONCLUSION AND FUTURE SCOPE

This paper is aimed at establishing Glow-worm Swarm Optimization (GSO) algorithm as a potential solution for solving the 3D Path Planning in Dynamic Environment. The use of Swarm Intelligence helps in faster convergence and speeds up the task of path planning. With the results obtained after performing the simulation, it can be inferred about the viability of using Glow-worm Swarm Optimization for path planning of UAV in 3D Dynamic Environment. The fitness of the algorithm is judged on the basis of number of nodes expanded, cost and time taken to reach the goal. The cost, number of expanded nodes and path finding time increases with the increase in the complexity of the environment. Path Planning Problem for UAV has a great deal of applications in present scenario. It offers an excellent solution for finding paths in uncertain terrains. It can be used to track people in a flood-hit area, or can be used in warfare/other military service. It can also be used for surveillance operations on other planets, water-bodies, and no-mans land. It can be seen that various natural factors like wind speed, atmospheric pressure, temperature, etc. have to be taken into account for real life application. The algorithm can be clubbed with Deep Learning (Image Processing) which will help the UAVs to generate a map of the environment in real time. Genetic and chaotic operators can be introduced in GSO to improve the convergence rate and hence efficiency. Evolutionary algorithms can also be used along with GSO to increase the accuracy and speed of the path-planning problem.

References

[1] Aljarah, Ibrahim et al.: 2013, A new clustering approach based on glowworm swarm optimization, pp. 2642–2649.

[2] Bircher, Kemel et al.: 2015, Three-dimensional coverage path planning via viewpoint re-sampling and tour optimization for aerial robots., *International Conference on Computer and Communication Technologies* pp. 1 – 3.

[3] Chestnutt, Takaoka et al.: 2009, Biped Navigation in Rough Environments using On-board Sensing., *IEEE-Intelligent Robots and System.* pp. 1 – 2.

[4] Kitampura, Tanaka et al.: 1995, 3-D path planning in a dynamic environment using an octree and an artificial potential field., *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems, Human Robot Interaction and Cooperative Robots* pp. 1 – 7.

[5] Krishnanand, K. and Ghose, D.: 2009, Glow-worm Swarm Optimization for simultaneous capture of multiple local optima of multimodal functions(Swarm intelligence), *Springer* pp. 87 – 124.

[6] Mishra, A. and Shukla, A.: 2016a, Mathematical analysis of the cumulative effect of novel ternary crossover operator and mutation on probability of survival of a schema., *Theoretical Computer Science* pp. 1 – 11.

[7] Mishra, A. and Shukla, A.: 2016b, Analysis of the Effect of Defining Length and Order of Schemata on Probability of Survival of a Group of Schemata., *Annual International Conference on Intelligent Computing, Computer Science and Information Systems* .

[8] Mishra, A. and Shukla, A.: 2017, Mathematical analysis of schema survival for genetic algorithms having dual mutation., *Soft Computing-A Fusion of Foundations, Methodologies and Applications* pp. 1 – 9.

[9] Mishra, A. and Shukla, A.: 2018, A new insight into the schema survival after crossover and mutation for genetic algorithms having distributed population set., *International Journal of Information Technology* .

[10] Pandey, P., Shukla, A. and Tiwari, R.: 2017, Three-dimensional path planning for Unmanned Aerial Vehicles using glowworm swarm optimization algorithm., *International Journal of System Assurance Engineering and Management* .

[11] Sanci, S. and Isler, V.: 2011, A Parallel Algorithm for UAV Flight Route Planning on GPU., *International Journal of Parallel Programming* pp. 1 – 3.

[12] Tang, Z., Zhou, Y. et al.: 2013, An Improved Glow-worm Swarm Optimization Algorithm Based on Parallel Hybrid Mutation., *International Conference on Intelligent Computing* pp. 1 – 4.

[13] Uc, F. and Altilar, D. T.: 2012, Using Genetic Algorithms for Navigation Planning in Dynamic Environments, *Applied Computational Intelligence and Soft Computing Volume* pp. 1 – 3.

[14] Wang, G., Guo, L. et al.: 2012, A Bat Algorithm with Mutation for UCAV Path Planning, *International Journal of Hybrid Information Technology* pp. 123 – 144.

[15] Wei-guo, Z., Guang-wen, L. et al.: 2012, UAV Path Planning in Dynamic Environment., *32nd chinese control conference* pp. 1 – 2.

[16] Xie, C. and Zheng, H.: 2016, Application of improved Cuckoo search algorithm to path planning unmanned aerial vehicle., *International Conference on Intelligent Computing* pp. 722–729.

[17] Yang, L., Qi, J. et al.: 2014, A Literature Review of UAV 3D Path Planning., *IEEE Intelligent Control and Automation (WCICA)* pp. 1 – 5.