

Developing Dijk-Primbert Algorithm for Finding Unpredictable Paths over Time-Varying Networks

P. Sushmitha, R. Kanthavel

Velammal Engineering College, Chennai, India

Email: sushmi.1904@gmail.com

How to cite this paper: Sushmitha, P. and Kanthavel, R. (2016) Developing Dijk-Primbert Algorithm for Finding Unpredictable Paths over Time-Varying Networks. *Circuits and Systems*, 7, 3541-3555.
<http://dx.doi.org/10.4236/cs.2016.711301>

Received: May 12, 2016

Accepted: May 25, 2016

Published: September 8, 2016

Copyright © 2016 by authors and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Cooperation among multiple unmanned vehicles is an intensely challenging topic from a theoretical and practical standpoint, with far reaching indications in scientific and commercial mission scenarios. The difficulty of time coordination for a rapid of multirotor UAVs includes predefined spatial paths according to mission necessities. With the solution proposed, cooperative control is accomplished in the presence of time-varying communication networks, as well as stringent temporal constraints, such as concurrent arrival at the desired final locations. The proposed explanation solves the time-coordination problem under the acceptance that the trajectory-generation and the path-following algorithms meeting convinced cohesion conditions are given. Communication is processed in unpredictable paths by the use of path following and directed communication graph. Dijk-Primbert algorithm for finding the shortest collision free paths is used to avoid and detect collision/congestion in unpredictable paths. Without collision detection, it doesn't seem agreeable to have collision avoidance because there wouldn't be everything to avoid. Dijkloyd algorithm is used for finding shortest paths in a weighted directed graph with positive and negative edges. Primloyd algorithm is used for finding shortest paths in a weighted undirected graph for conquering the complexity in matrix coding. In case of congestion or collision then the whole network is learned about it to all the communicators. Hence, communication is taken place in an unpredictable path in a secured manner.

Keywords

Multirotor, Spacecraft Formation Flying, UAV Control, Collision Avoidance

1. Introduction

With the enormous amount of communication technology development and embedded systems, the space communications have been very much technologically updated. Particularly without the intervention of the human, the aircrafts have been designed with the goal to flyover in the space for commercial and surveillance purposes. The Unmanned Aerial Vehicle (UAV) is a type of aircraft with no pilot on board. Based on the pre-programmable data, the UAV can be remote controlled. In recent times, the field of air vehicle autonomy has many research scopes in the area of sensor fusion, handling communication between multiple agents, path planning, trajectory generation, task allocation and scheduling and cooperative tactics. The ultimate aim is to entirely avoid the pilot in operating the UAV and also to be away from aircraft collision. When the UAV is moved and controlled through the inbuilt programs by the Air Traffic Controller (ATR), there are some toughest tasks to be achieved in order to minimize the distance between the starting point and the ending point. The number of paths available for dynamically may alter the UAV direction during its board time from the usual path is the critical problem.

This paper is focused on controlling the presence of time-varying communication networks, as well as stringent temporal constraints, such as simultaneous arrival at the desired final locations. The coordination task is accomplished by adjusting a coordination variable under trajectory generation algorithm. The proposed solution solves the time-coordination problem under the acceptance that the trajectory-generation and the path-following algorithms meet certain stability conditions [1]. The ability of the approach to avoiding collisions of the formation with static as well as dynamic obstacles during its movement into the desired target region is to use Hawk-eye approach [2], while in [3] new robust and adaptive control scheme is capable of dealing with the system nonlinearities and external disturbances due to close formation. Coordinated path following and vehicle coordination algorithm solves the problem of temporary communication failures under the Lyapunov-based theory [4]. To decentralize Partially Observable Markov Decision Process (POMDPs) as long as maintaining cooperation between multi-robots is to use POMDP policy auctions [5]. In their presented work, optimization algorithms are formulated in the expanded space where subsystems are disjoint. The design action is based on the hierarchical application of convex optimization tools. The proposed solution solves the time-coordination problem under the assumption that a path-following algorithm meets certain stability conditions [6].

A discrete Kalman filter is introduced in [7] for fusing the local estimates of the quadrotors. Control strategy controls the flexible payload to a desired pose while also controlling the deformations to zero; the key idea behind the method is based on common techniques validated by carrying a flexible ring. The merits of this approach are communication is easy in defined path [8]. Model-dependent and model-independent control techniques are presented in [9] [10] for designing the autonomous quadrotors. Relevant work includes Sliding-Mode control theory [11], Model-independent coordination strategy [12], stability properties of a group of mobile agents [13], and the expo-

ponential convergence of the time-coordination error are proven using Lyapunov theory. POMDPs and tractable algorithms are compared with the large teams of interchangeable robots [14]. Iterative algorithm has a local convergence behaviour that employs a semi definite programming solver at each recursive step [15] and linear matrix inequalities [LMI] are combined with logic-based switching shed light on the various control strategies [16].

This limitation motivated us to reformulate the path following problem in a different way. The goal of this paper is to provide a new solution in all defined or undefined path by the use of path following and directed communication graph. Path can be found and communication is processed in unpredictable paths by the use of path following and directed communication graph. Dijk-Primbert algorithm for finding the shortest collision free paths is used to avoid and detect collision/congestion in unpredictable paths. Dijkloyd algorithm is used for finding shortest paths in a weighted directed graph with positive and negative edges. Primloyd algorithm is used for finding shortest paths in a weighted undirected graph for conquering the complexity in matrix coding. In case of congestion or collision then the whole network is learned about it to all the communicators. Hence, communication is taken place in an unpredictable path in an obtained manner. The objectives of this paper are to provide best routing conditions for all type of paths, to find the shortest path in both defined and undefined paths, and to avoid and detect congestion/collision in unpredictable paths.

This paper is created as follows. In Section 2, we introduce the proposed work. In Section 3, we describe the path-following problem by giving a suitable set of algorithms. In Section 4, we formulate the communication flow diagram; simulation results are explained in Section 5. Finally, in Section 6, the main conclusions are presented.

2. Proposed Work

In this section, a proposed work for cooperative path-following control algorithm of UAV is introduced. In order to overcome the existing problems, here it is proposed a Dijk-Primbert algorithm is used for finding collision free shortest paths for source to destination in unpredictable paths.

Overview of a Proposed UAV Environment

In this section, we propose a Dijk-Primbert algorithm for finding collision free shortest paths in a weighted directed graph and weighted undirected graph. **Figure 1** shows the block diagram of the proposed system.

Communication network is used to exchange the information among the vehicles. Once the communication takes place, UAV will select Robert algorithm to avoid collision/congestion in unpredictable paths. After that Dijkloyd and Primloyd algorithm is referred by UAV which way it needs to travel. If the path is clear the UAV will select the Dijkloyd algorithm and go through the path already assigned by us *i.e.* predefined path. If the path is not clear the UAV will select the Primloyd algorithm and go through

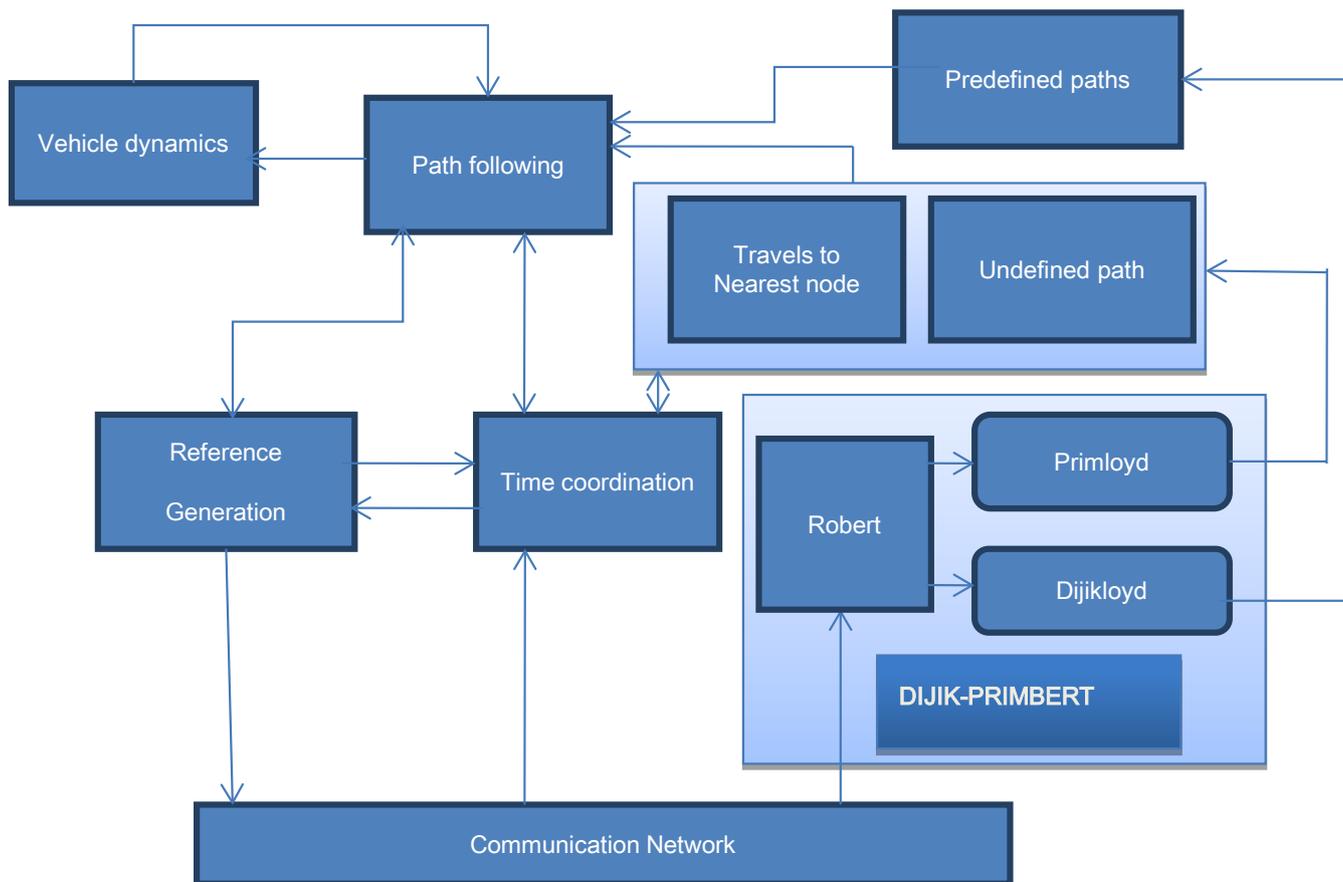


Figure 1. Block diagram of proposed UAV environment and interactions between them.

the undefined path. The undefined path is identified by nearest path/node from the original/predefined path. Path following allows each vehicle to follow its assigned path with the desired speed profile. If the path is identified it will check the time coordination if it's clear it will sustain to the travel path. If the time coordination is not clear it will wait until to get the clear paths. Time coordination regulates the vehicle according to time where one vehicle cannot collide with each other. Time coordination will check the reference generation whether any vehicles is travelling on the path or not. If all are clear the vehicle will travel on the path following.

3. Path Following: Problem Formulation

We now address the path-following problem of a fleet of multirotor UAVs. As already mentioned earlier, path can be found and communication is processed in all direction and informed to overall network increase the network topology. Dijik-Primbert algorithm is used for finding collision free shortest paths for source to destination in unpredictable paths.

3.1. Dijik-Primbert Algorithm for Unpredictable Paths

Dijik-Primbert algorithm is the combination of Dijikloyd algorithm, Primloyd algo-

rithm and Robert algorithm. Dijk-Primbert algorithm for finding the collision free shortest paths, which is used to avoids/detects collision/congestion in unpredictable paths. Collision detection is simply the act of surveying the known vicinity and detecting the presence of a possible collision. Without collision detection, it doesn't seem acceptable to have collision avoidance because there wouldn't be anything to avoid.

Collision avoidance is the plan for action the robot algorithm takes to evade the on-coming collision. As previously declared, there is no need for collision avoidance algorithm if there are no collisions to avoid. The node diagram of finding unpredictable path using Dijk-Primbert algorithm is shown in **Figure 2**. Collision avoidance, in comparison to collision disclosure, is what is done after a possible collision is detected. When making an effort to avoid an obstacle, there are many verdicts to face. The verdicts available depend on the available paths the robot can take to prevent the obstruction. Finding the correct path to use can be annoying depending on which algorithm is used and the granularity correlated with it.

3.2. Dijkloyd Algorithm for Defined Path

Dijkloyd algorithm is the combination of dijkstra's algorithm and Floyd-warshall

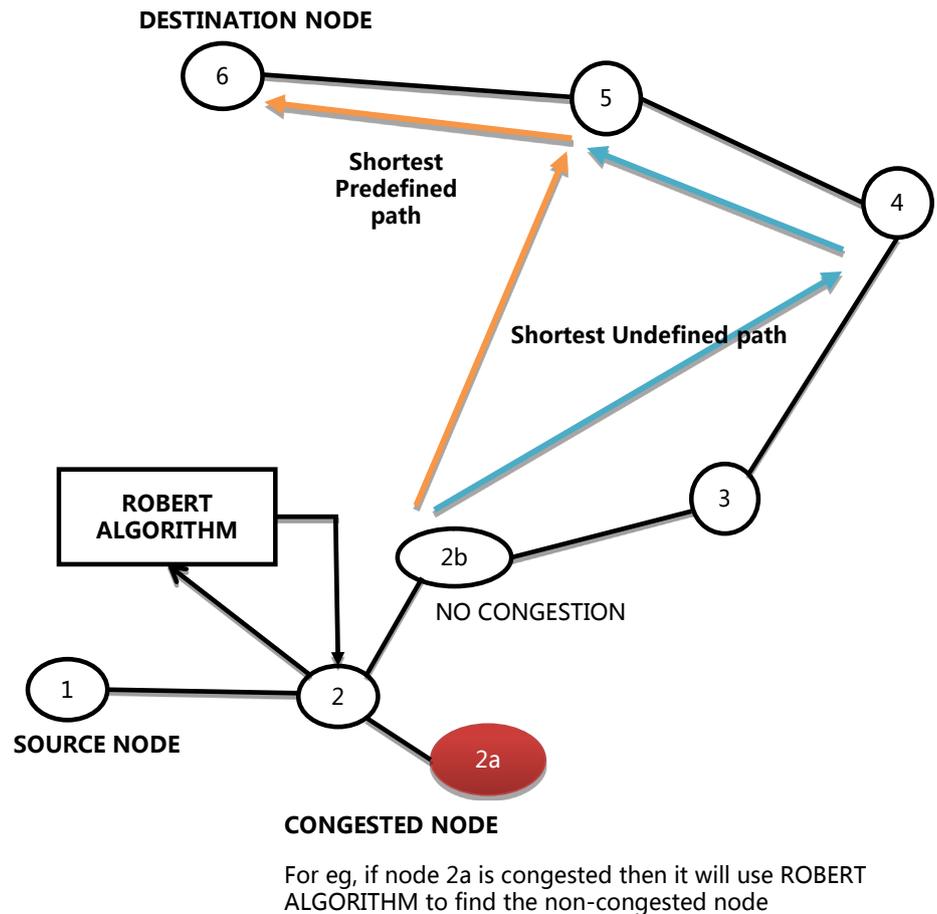


Figure 2. Node diagram of the Dijk-Primbert algorithm for unpredictable paths.

algorithm. Dijkstra's algorithm is a graph search method that finds the path with reduced cost between the vertex and it solves the single-source shortest path dispute for a graph with non-negative edge path costs, producing a shortest path tree to the destination vertex. This algorithm is often used in routing and subroutine in other graph algorithms for superficial directed graphs with non-negative weights. Floyd-Warshall algorithm is a graph analysis method for finding shortest paths in a weighted graph with either positive or negative edge weights. The Floyd algorithm is an example of dynamic programming. Dijkloyd algorithm is a graph search algorithm for finding shortest paths in a weighted directed graph with positive and negative edge weights. The merits of this algorithm are easier to code and fast computation of path finder networks.

3.3. Primloyd Algorithm for Undefined Path

Primloyd algorithm is the combination of Prims algorithm and Floyd-warshall algorithm. Prims algorithm is a rapacious algorithm that finds a minimum spanning tree and ordering the edges by weight for a weighted undirected graph. This means it finds a subset of the corner that forms a tree that contains every vertex using a priority queue, where the total weight of all the corners in the tree is minimized. The Floyd-Warshall method analyses all possible paths through the graph for finding the simplest path between each pair of vertices. If there are negative periods, the Floyd-Warshall algorithm can be used to identify them. The Floyd-Warshall algorithm typically only supports the lengths of the paths between all pairs of vertices. When a new shortest path is found between two vertices, the matrix including the paths with the maximum flow between the vertices is updated. Primloyd algorithm is used for overcoming the complexity in matrix coding. The merits of this algorithm are simple and easier to code.

4. Algorithm and Communication Flow Diagram of Dijk-Primbert Algorithm

Let the node at which the UAV is starting called the **initial node**. The **distance** from initial node to destination node is taken as **Y**.

4.1. Algorithm for Finding Unpredictable Paths Using Dijk-Primbert Algorithm

Let the Dijk-Primbert algorithm is a dignified version of simple one liner and it works well for an over simplified paths. Dijk-Primbert algorithm will also assign initial node values and will try to find out collision free shortest path in a step by step manner.

- 1) Declare the inputs in the communication networks.
- 2) Check whether the congestion/collision condition $y \leq n$.
- 3) If the above condition satisfied then it will detect and correct the congestion/collision using Robert algorithm.
- 4) After that the primloyd algorithm is used to trace the possible undefined paths and it finds the shortest distance between the traced paths.
- 5) If the condition is not satisfied then it will go through with the Dijkloyd algo-

rithm.

- 6) The Dijkloyd algorithm is used to find the shortest predefined path.

4.2. Algorithm for Finding Defined Path Using Dijkloyd Algorithm

Let the node at which we are starting UAV be called the **initial node**. Let the **distance of node Y** be the interval from the **initial node** to Y . Dijkloyd algorithm will assign some initial length values and will try to improve them step by step.

- 1) Assign a conditional distance value to each and every node: set zero to the initial node and to infinity for all other nodes.
- 2) Mark all the nodes interval from the initial node and from a queue.
- 3) Take the essential distance value from the queue.
- 4) If any queue is empty or distance is maximum then get the acquaintance of the current node.
- 5) If there is no neighbour compute new distance for a neighbour distance which is **neighbour + current node distance**.
- 6) If there is neighbour calculate the negative characteristic of node *i.e.* the threshold value is more than the negative value.
- 7) The new distance value should be less than the old one in both the cases of step 5 and step 6.
- 8) If step 7 satisfies proceed to the next node, if not discard.

4.3. Algorithm for Finding Defined Path Using Dijkloyd Algorithm

Let the node at which we are selecting be called the **initial node**. Primloyd algorithm will also selects the initial distance values and will try to find out undefined path in a step by step manner.

- 1) Initialize a tree with single vertex, allow a tentative distance value to each and every node.
- 2) Insist the variables $n, X, Y, \text{num} = 1$. Set the input number of rows then initialize $X = 1$ and $Y = 1$.
- 3) Trace the possible paths and find the shortest distance between the traced paths.
- 4) If X is less than or equal to n and Y is less than or equal to n then print num value and increment num, Y .
- 5) If they recognize the candidate model then get a neighbour of the current node.
- 6) If X is less than or equal to n and y is not equal to n then print n and increment X value.
- 7) If X is not less than n then stop.

4.4. Communication Flow of Dijk-Primbert Algorithm

The flow diagram of finding defined path using Dijkloyd algorithm is shown below in **Figure 3**. The below **Figure 4** shows the flow diagram of finding undefined path using Primloyd algorithm. The flow diagram of finding unpredictable path using Dijk-Primbert algorithm is shown below in **Figure 5**. The Dijk-Primbert algorithm is a

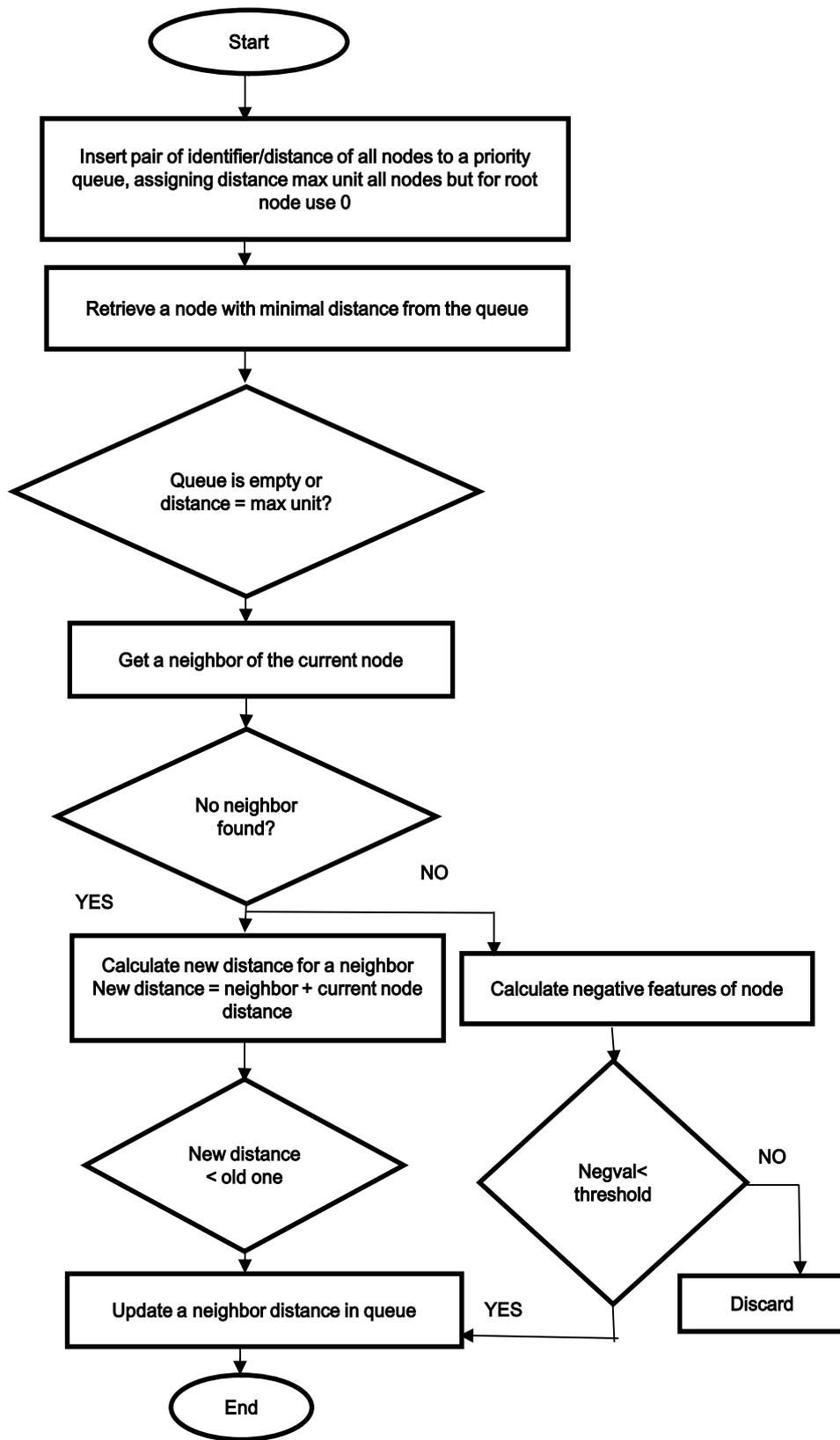


Figure 3. Finding defined path using Dijkstra algorithm.

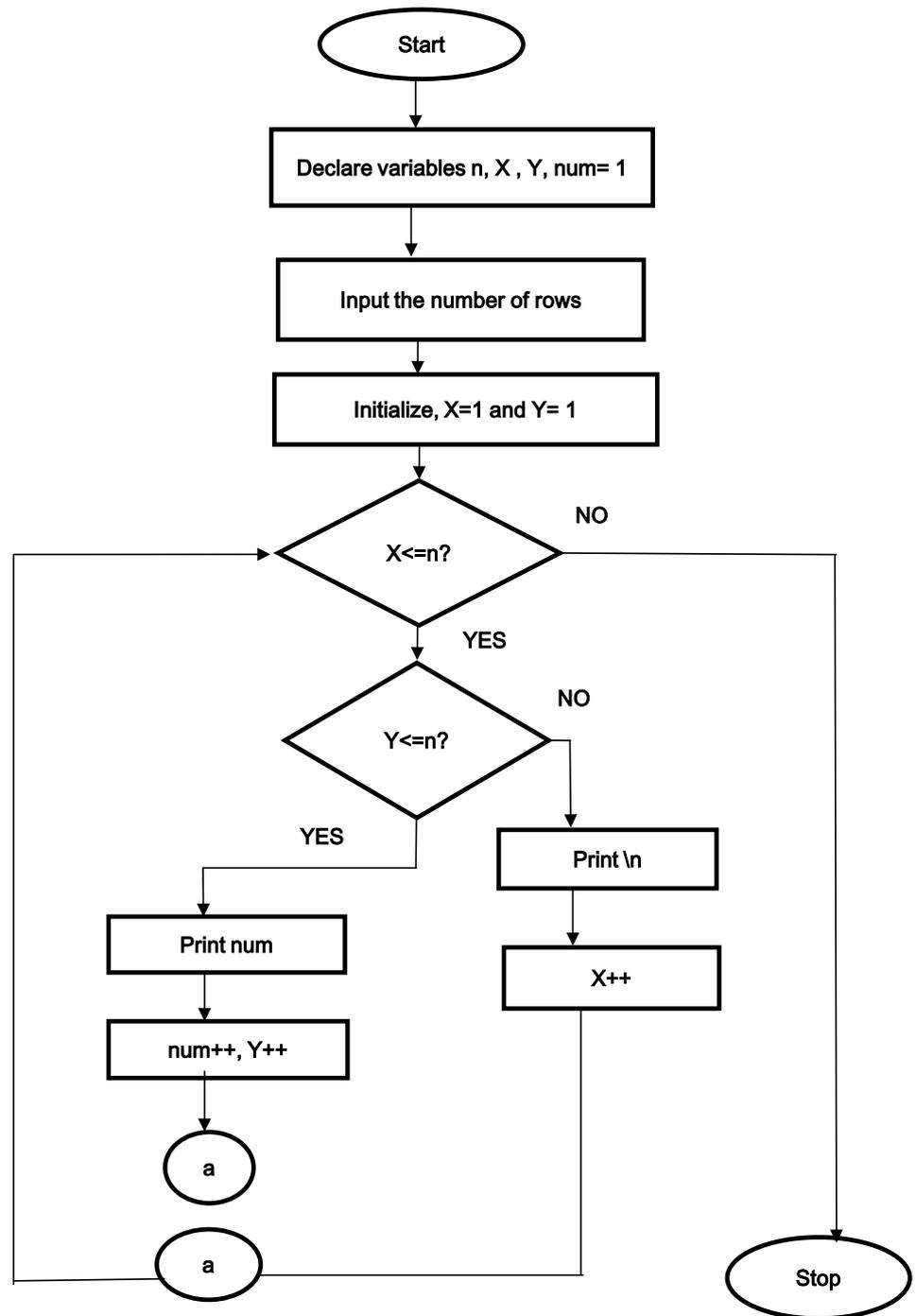


Figure 4. Finding undefined path using Primloyd algorithm.

dignified version of simple one liner and it works well for an over simplified paths. Dijkstra-Primbert algorithm will also assign initial node values and will try to find out collision free shortest pathway. In dijkloyd algorithm, start with a starting vertex with distance zero. Trace the possible paths from starting vertex and find the shortest distance. Mark it as a next starting terminal add the present distance with previous. Repeat

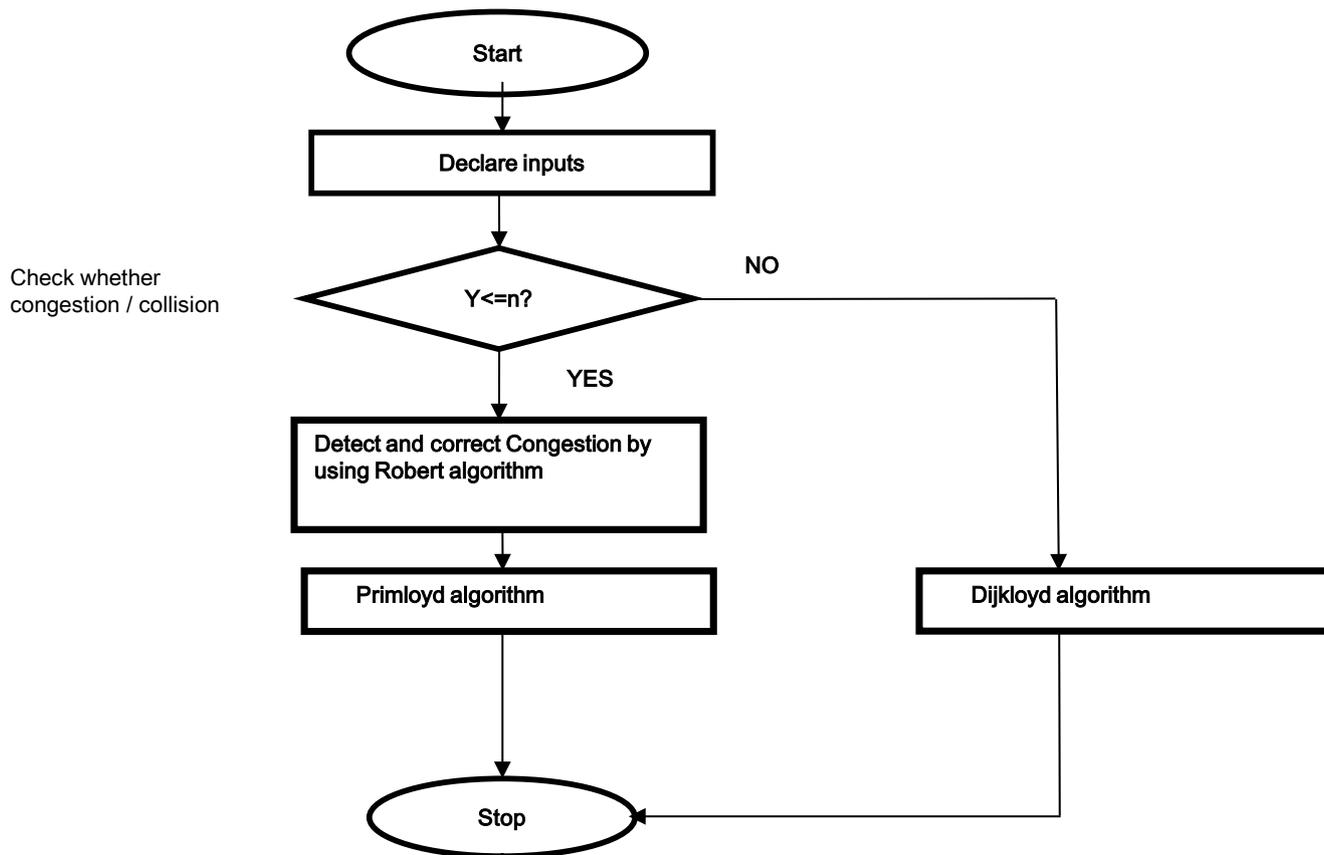
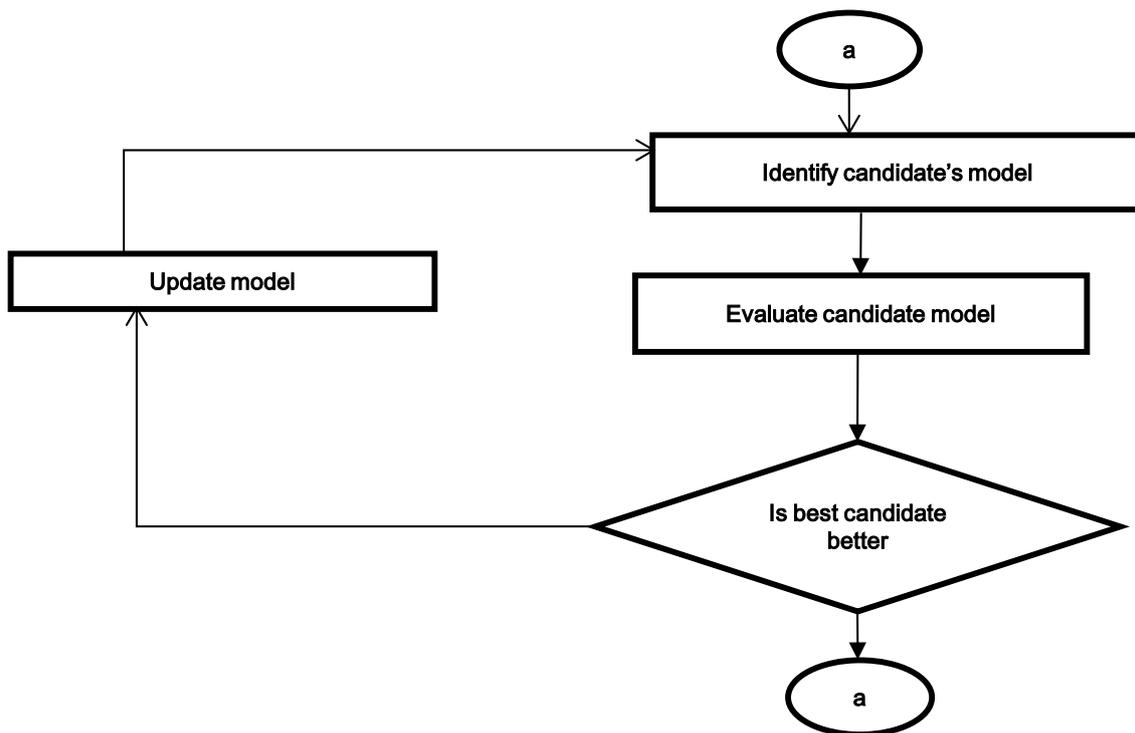


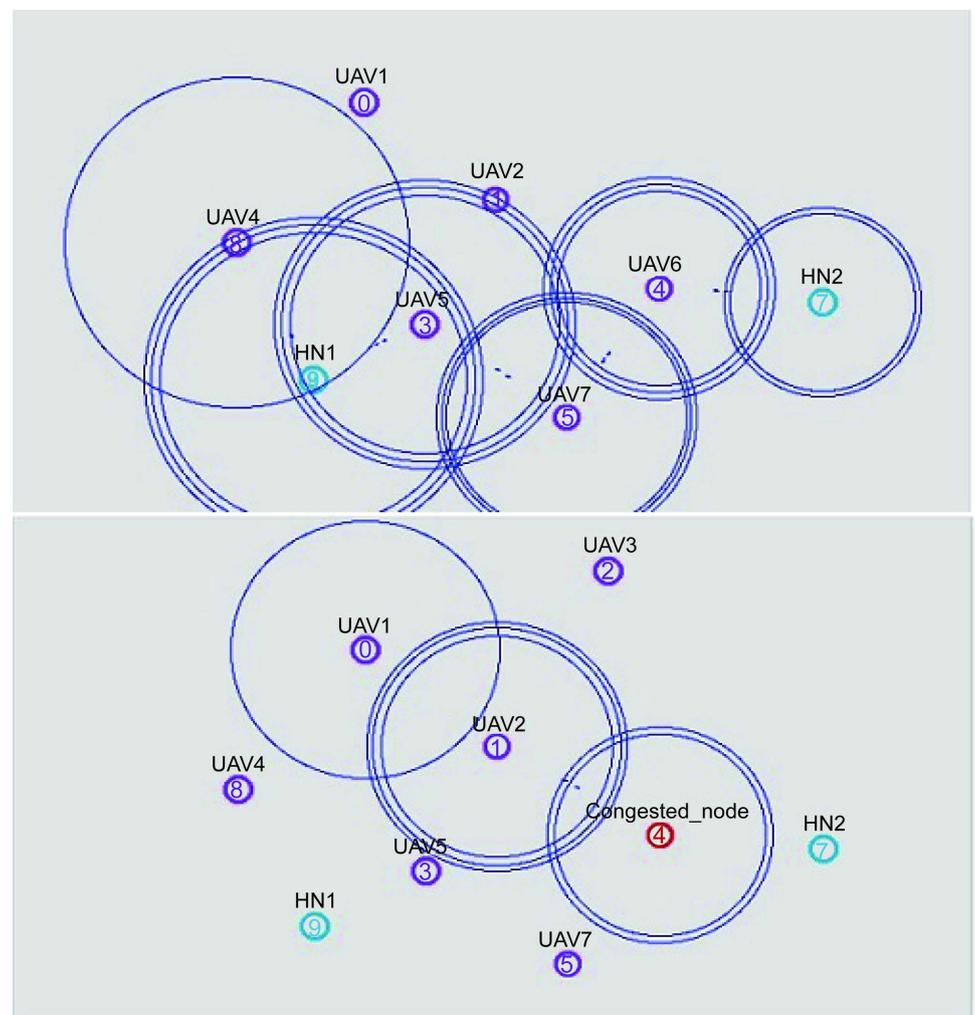
Figure 5. Finding unpredictable paths using Dijk-Primbert algorithm.

the above steps. In primloyd algorithm, start with an initial vertex and trace the possible paths. Find the shortest distance between the traced paths and fix the shortest path. From the new vertex again repeat the above steps and finally check all the vertices have traced.

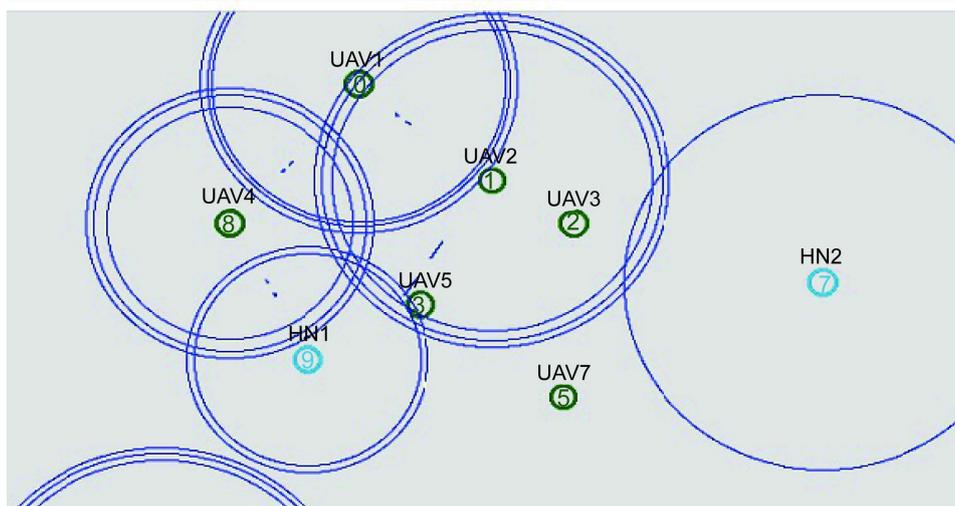
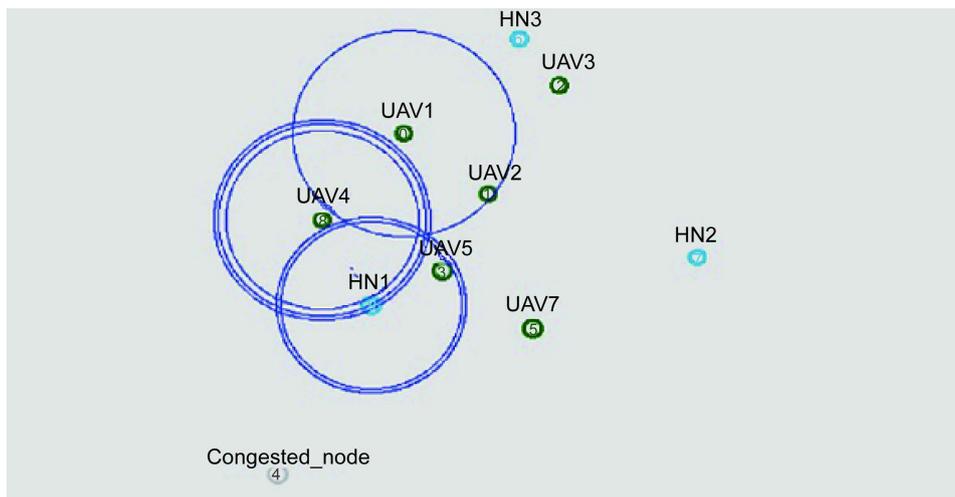
5. Experimental Results and Outputs

In this section, we present simulation results for Dijk-Primbert algorithm run in MATLAB and NS2. Here nine nodes are generated, three acts as header nodes and seven as UAVs nodes. Dijk-Primbert algorithm is used for finding collision free shortest paths for source to destination in unpredictable paths. Initially explains the command window of showing the defined and undefined paths with less computational cost and communication overhead.

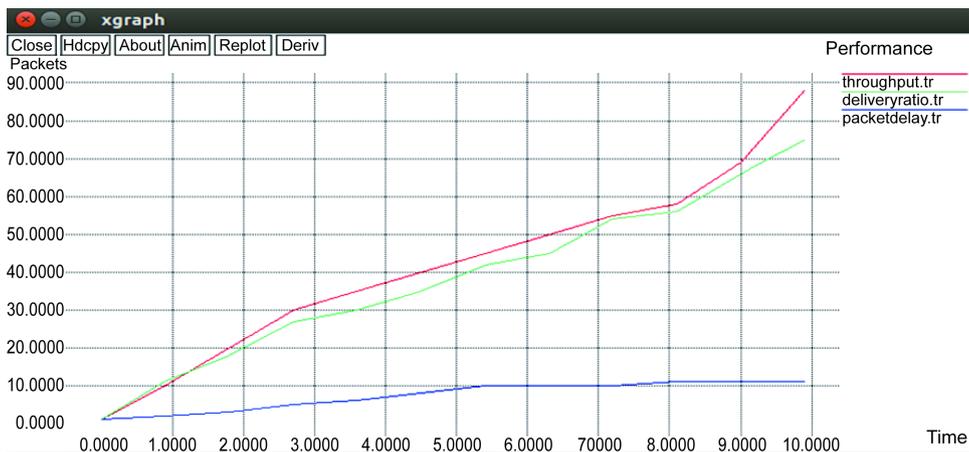
Figure 6(a) shows that UAV6 is congested due to packet loss and it is removed from the path. **Figure 6(b)** shows the UAV communicating in predefined and undefined path. **Figure 6(c)** shows the X graph for the performance of packets with required time.



(a)



(b)



(c)

Figure 6. Finding unpredictable paths using Dijk-Primbert algorithm. (a) UAVs communicating in a predefined Path UAV6 is CONGESTED due to packet loss; (b) UAV6 is CONGESTED and it is Removed UAVs communicating in a undefined path; (c) X GRAPH for the performance of packets.

Comparative Analysis of the Proposed System and Existing System

In the existing system, Multivehicle cooperative supremacy pursues the trajectory-generation algorithm, path-following algorithm and the time coordination control algorithm have been used. A Path-following controller allows each vehicle to converge to and follow its assigned path but it detects the congestion in the predefined path only. So there is a need of implementing an algorithm to find out the unpredictable path. So, it is proposed to implement a Dijk-Primbert algorithm to find out the shortest collision free path in unpredictable paths.

Figure 7 shows the comparative analysis chart for the existing path communication in defined path with the proposed path communication in unpredictable paths. The **Table 1** shows the comparative table for the existing path communication in defined path with the proposed path communication in unpredictable paths for different number of nodes with required time.

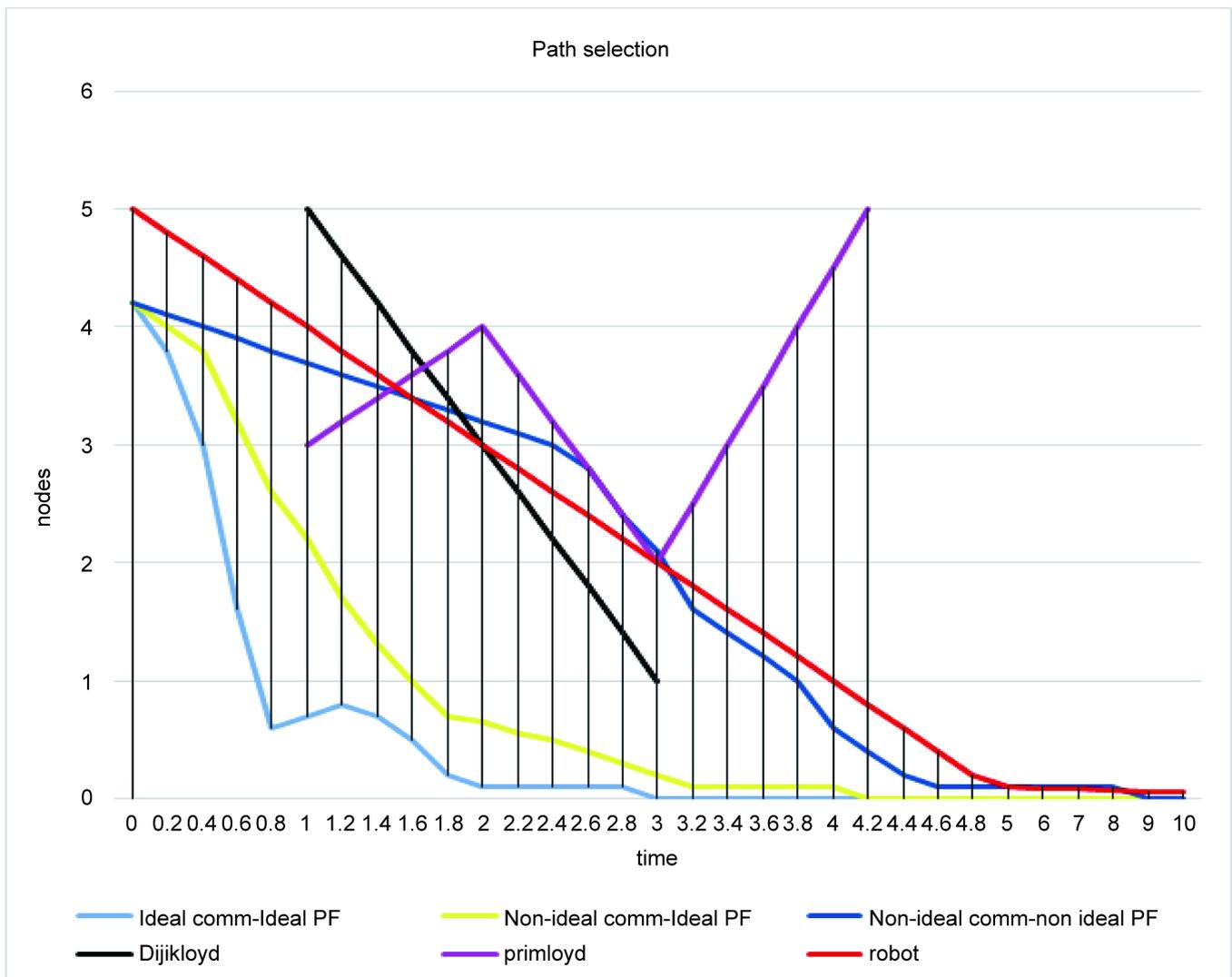


Figure 7. Comparison of existing and proposed method.

Table 1. Comparison table of existing and proposed method.

Path selection mode	Number of nodes	Required time
Ideal communication-ideal PF	4	0
	2	0.6
	0	2.8
Non-ideal communication-ideal PF	4	0
	2	1.4
	0	4.2
Non-ideal communication-non ideal PF	4	0
	2	3.2
	1	4.0
Dijkloyd	5	1.0
	3	2.2
	1	3.2
Primloyd	5	4.2
	3	1.2
	2	3.0
Robert	5	0
	3	2.2
	1	4.0

6. Conclusion

In this work, an efficient path is determined by proposing the Dijk-Primbert algorithm to direct UAVs and to consider out the collision free shortest paths in an unpredictable path. Dijk-Primbert algorithm has the features of Dijkloyd, Primloyd and Robert algorithm. Dijkloyd has the features of finding shortest path in positive and negative edges. Primloyd algorithm is used for overcoming the complexity in matrix coding. Here the performance of calculating the shortest collision free path is improved. It is also used efficiently for UAVS. Future works will implement time delayed communication, *i.e.*, Markovian rule for finding the collision free paths. Markov Decision Process (MDP) produces the multiple threats resolution logic for the collision avoidance system and the unique feature is to formulate the collision avoidance problem in real-time.

References

- [1] Cichella, V., Kaminer, I. and Dobrokhodov, V. (2015) Cooperative Path Following of Multiple Multirotors over Time-Varying Networks. *IEEE Transactions on Automation Science and Engineering*, **12**, 945-957. <http://dx.doi.org/10.1109/TASE.2015.2406758>
- [2] Saska, M., Vonásek, V., Krajník, T. and Přeučil, L. (2014) Coordination and Navigation of Heterogeneous MAV-UGV Formations Localized by a Hawk-Eye-Like Approach under a Model Predictive Control Scheme. *International Journal of Robotics Research*.

- <http://dx.doi.org/10.1177/0278364914530482>
- [3] Song, Y.D., Li, Y. and Liao, X.H. (2005) Orthogonal Transformation Based Robust Adaptive Close Formation Control of Multi-UAVs. *Proceedings of American Control Conference*, **5**, 2983-2988.
- [4] Ghabcheloo, R., Pascoal, A.M., Silvestre, C. and Kaminer, I. (2006) Coordinated Path Following Control of Multiple Wheeled Robots Using Linearization Techniques. *International Journal of Systems Science*, **37**, 399-414. <http://dx.doi.org/10.1080/00207720500438324>
- [5] Turpin, M., Michael, N. and Kumar, V. (2013) Concurrent Assignment and Planning of Trajectories for Large Teams of Interchangeable Robots. *IEEE International Conference on Robotics and Automation (ICRA)*, 6-10 May 2013, 842-848. <http://dx.doi.org/10.1109/icra.2013.6630671>
- [6] Stipanović, M., Inalhan, G., Teo, R. and Tomlin, C.J. (2004) Decentralized Overlapping Control of a Formation of Unmanned Aerial Vehicles. *Automatica*, **40**, 1285-1296. <http://dx.doi.org/10.1016/j.automatica.2004.02.017>
- [7] Gurcuoglu, U., Puerto-Souza, G.A., Morbidi, F. and Mariottini, G.L. (2013) Hierarchical Control of a Team of Quad Rotors for Cooperative Active Target Tracking. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5730-5735.
- [8] Ritz, R. and D'Andrea, R. (2013) Carrying a Flexible Payload with Multiple Flying Vehicles. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3-7 November 2013, 3465-3471. <http://dx.doi.org/10.1109/iros.2013.6696850>
- [9] Gupte, S., Mohandas, P.I.T. and Conrad, J.M. (2012) A Survey of Quadrotor Unmanned Aerial Vehicles. *Proceedings of IEEE Southeastcon*, 15-18 March 2012, 1-6. <http://dx.doi.org/10.1109/secon.2012.6196930>
- [10] Cichella, V., Kaminer, I., Xargay, E., Dobrokhodov, V., Hovakimyan, N., Aguiar, A.P. and Pascoal, A.M. (2012) A Lyapunov-Based Approach for Time-Coordinated 3D Path-Following of Multiple Quad Rotors. *IEEE Conference on Decision and Control*, Maui, December 2012, 1776-1781.
- [11] Gazi, V. (2005) Swarm Aggregations Using Artificial Potentials and Sliding Mode Control. *IEEE Transactions on Robotics*, **21**, 1208-1214. <http://dx.doi.org/10.1109/TRO.2005.853487>
- [12] Egerstedt, M. and Hu, X. (2001) Formation Constrained Multi-Agent Control. *IEEE Transactions on Robotics and Automation*, **17**, 947-251. <http://dx.doi.org/10.1109/70.976029>
- [13] Tanner, H., Jadbabaie, A. and Pappas, G. (2007) Flocking in Fixed and Switching Networks. *IEEE Transactions on Automatic Control*, **52**, 863-868. <http://dx.doi.org/10.1109/TAC.2007.895948>
- [14] Capitan, J., Spaan, M.T., Merino, L. and Ollero, A. (2013) Decentralized Multi-Robot Cooperation with Auctioned POMDPs. *International Journal of Robotics Research*, **32**, 650-671. <http://dx.doi.org/10.1177/0278364913483345>
- [15] Kim, Y. and Mesbahi, M. (2006) On Maximizing the Second Smallest Eigenvalue of State-Dependent Graph Laplacian. *IEEE Transactions on Automatic Control*, **51**, 116-120. <http://dx.doi.org/10.1109/TAC.2005.861710>
- [16] Mesbahi, M. and Hadaegh, F.Y. (2001) Formation Flying Control of Multiple Spacecraft via Graphs, Matrix Inequalities, and Switching. *Journal of Guidance, Control, and Dynamics*, **24**, 369-377. <http://dx.doi.org/10.2514/2.4721>



Submit or recommend next manuscript to SCIRP and we will provide best service for you:

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.

A wide selection of journals (inclusive of 9 subjects, more than 200 journals)

Providing 24-hour high-quality service

User-friendly online submission system

Fair and swift peer-review system

Efficient typesetting and proofreading procedure

Display of the result of downloads and visits, as well as the number of cited articles

Maximum dissemination of your research work

Submit your manuscript at: <http://papersubmission.scirp.org/>