International Journal of Engineering Science Invention (IJESI)
ISSN (Online): 2319-6734, ISSN (Print): 2319-6726
www.ijesi.org ||Volume 9 Issue 1 Series. II || Jan.2020 || PP 11-17

Efficient Distributed Flight Path Construction Techniques In Uav

Minakshi Gautam[1], Dr V.Ramesh[2]
[1] *(Dept. of Computer Science and Engg./Presidency university,Bangalore,India)*
[2] *(Dept. of Computer Science and Engg,/Presidency university,Bangalore,India)*
Corresponding Author: Minakshi Gautam

Abstract-The *pioneering area of intelligent unmanned aerial vehicle(UAV) has shown significant developments in recent years particularly drones .Nowadays Unmanned Aerial Vehicles (UAVs) are more reliable, flight autonomous and easier to use with great prospective for commercial use in the near future. The increased availability of distributed spatial data has led to a surge in their popularity. Flight automation of drones is considered in order to maximize their operational value and reduce cost .Different drones suffer from different limitations making it more difficult to plan effective autonomous flights. The paper provides an in depth study of the different flight planning techniques considering different drone types researched so far and their efficiency analysis in distributed environment.***

Keywords- *UAV, autonomous flight plan, conventional drones, Quad copters, DIMPL, DIFPL.*

Date of Submission: 20-01-2020 Date of Acceptance: 05-02-2020

I. Introduction

Drones are increasingly being used to perform crucial and difficult aerial tasks economically and safely. To make sure operating costs are low and drone flight autonomously, their flight plans are required to be pre-built. In the most earliest techniques drone flight paths are not automatically pre-calculated based on drone capabilities and terrain information. Some of the basic techniques used were sensors[1],dynamic shortest paths[2], manually determined paths[3], navigation through camera images[4], and GPS for guidance[5] and vision based algorithms to direct the drone during flight[6], all of which makes flight navigation intricate and hard. Automated flights in several distinct environment has been attempted[7], and automation of flights in environments including those utilizing images in indoor contained environments and the mixture of sonar and image[8].

Pre-planning the paths is another way wherein flights can fly autonomously .Some popular researched algorithms include Genetic algorithms to trace flight paths[9], and the use of ant colony algorithms for 3D[10]. UAV path planning by implementing Particle Swarm Optimization have been discussed in many papers[11].

But in the current scenario the enlarged availability of spatial data distributed approaches has led to a pitch in pre calculation of flight paths which are prefed in the drones to flight autonomously. Flight plans are prebuild using spatial data deliberately stored on cloud , along with accelerated processing with hadoop MapReduce[12]. Spatial Hadoop or Hadoop extension for spatial operations have both been explored.Hadoop based platforms that support distributed queries with Map Reduce have been developed[13].

However here we will be concentrating mainly on the techniques used in distributed environments for flight path construction considering different types of drone and distributed spatial data.The two major approaches under study are DIFPL[14] and DIMPL[15]which are the earliest implemented complete distributed solution for constructing flight plans using spatial data in distributed environment. DIFPL(Distributed flight path builder) was developed with a view to build flight path for a mixed fleet that included conventional drone as well as quadcopter for surveying a large field to detect vegetation overgrowth over electic poles for an aviation company.DIMPL(Distributed in memory drone flight path builder) an advanced version of DIFPL was then developed which optimized the process by querying the data in memory.

II. Distributed System DIFPL

The complexity of covering an area with automated flights increases when there are multiple types of drones available with different capabilities. For this work drones two types of drones were considered shown in Figure 1

Conventional drone with limited take off and landing and Quadcopter with vertical take off and landing. Building a flight plan having different types of drone with maximum optimality was the primary challenge in covering the entire full region. Each drone type has its specific limitations. The cost of operating different types of drones is also varied. Not only should it cover the entire region with numerous flights of the right drone type that can navigate altering terrain but also use the cheaper drone as frequently as possible to minimize cost.
For implementation the terrain data such as networking line length and elevation based climbing angle to optimally divide area into subregions and build the flight plan for each subregion was considered.It construct a set of Flight Plans in order to cover the entire power lines network of aviation organization and reduce the number of drone flights and overall cost. This is achieved by optimizing coverage by each flight in a subregion and assigning to the type of drone needed for that subregion. Network lines and elevation of waypoints in each subregion need to satisfy rules that are represented as autonomy and climbing angle constraints. The constraints decide if the subregion needs to be shrunk or expanded or split between multiple drone types.

After collecting pictures following the mixed fleet image analysis during post-processing determines if the vegetation has overgrown over poles and needs trimming. Every pole in network was photographed at least 4 times. The drone take 2 passes from each side of the lines, one pass in one direction and another pass in the other direction. Drone is equipped with a advanced 24 Mega pixel camera with 50 mm optical lens. The images are used to perform 3D image reconstruction of each pole to determine if vegetation has overgrown around the pole.

2.1Methodology

The degree of terrain and network lines data for large areas increases rapidly. In order to scale to the wide terrain and networks datasets, DIFPL employ distributed paradigm on Hadoop MapReduce framework. We can discuss the implementation in three broad categories:

1)Preliminaries- In this section we describe the background information to DIFPL including input and output data, constraints on the hardware and subregion and waypoint construction.

a) Input Data-The following data were used as input
 i) (x,y) geocoordinate position of network lines endpoints provided by aviation organization .
 ii) elevation data (x,y,elevation) provided by geographic agency. The elevation points are 25m apart.

b) Output Data-
 i) A set of flight plans, each composed of a set of waypoints (x,y,altitude) .
 ii) One landing point in KML format for the drone. A separate output file is written for each subregion.

A pictorial representation of the sample area is given in fig 2.

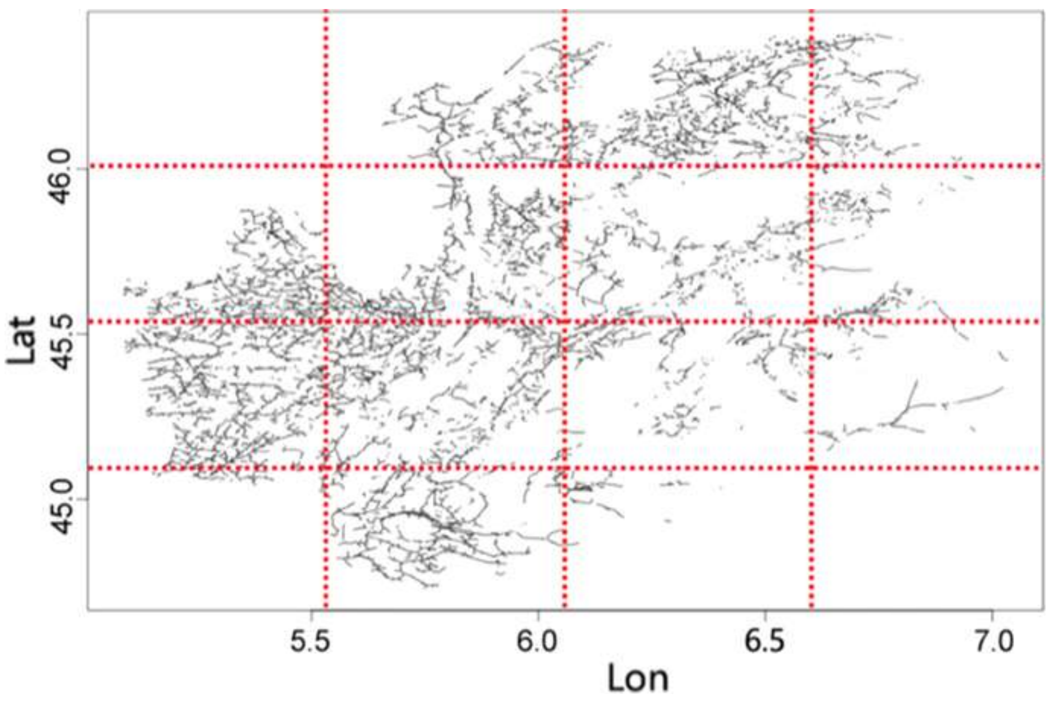


Fig2.

2)Constraints- The constraints on the flight path of drones were designed as inequalities. The inequalities were applied for each subregion for the type of drones. The inequalities were modelled as:

→For climbing angle:
Max(Clp) ≤ Ctype

Which described that elevation angle drone has to climb to fly from one waypoint to next along network line should be less than Ctype which is the climbing angle of the drone

→For autonomy:
$\sum L\ (2 * dL + iL) + 3 * l * 2 * \pi * r + t + n \leq Atype$
d is the distance of each network line
t is the takeoff distance to get to required elevation over first network pole with the climbing angle of each drone type
n is the landing distance with the descent angle for drone type
i distance between two network lines
r is the turn distance for the drone type for L lines.

Turning radius of conventional drone is 150m while that of quadcopter is 0m. The distance i is calculated by ordering network lines in the subregion by their start and calculating the distance between one line to next. Since there are 3 turns for a drone to cover a line segment twice and proceed to the next line segment 3 turning circumferences have to be added to the equation. The requirement of photographing each pole 4 times is satisfied by setting camera to take an image a second. The number of waypoints in output is achieved by collecting waypoints along network lines every 200m for conventional drone and every 100m for quadcopter and increasing it if the number of waypoints exceed the maximum.

3) Number of Waypoints. The hardware of conventional drone can be programmed with up to 200 waypoints and the quadcopter can be programmed with 50 waypoints

2.2Architecture

The architecture of DIFPL is based on distributed paradigm. The distribution approach in DIFPL was implemented using Apache Hadoop MapReduce framework. Hadoop is an open source framework which facilitates distributed computations on large clusters. A master node supervises data storage and computation distribution over multiple slave nodes. Files are uploaded into distributed file storage called HDFS, split into 64MB blocks and then processed.All the nodes are tracked by master node by keeping storage information of the blocks. MapReduce allows master node to break down huge computation tasks into mappers and reducers distributed over slave nodes. Mappers read input data as key value pairs and show intermediate key value pairs . Reducer receive the intermediate key value pairs grouped by k value and processes them to generate the final set of key value pairs . The distribution of flight path builder is important due to memory limitations of indexing large elevation and network datasets on a single node. Several opportunities for distribution of the flight plan builder process are available. The identification of quadcopter subregions and the shrinking of quadcopter and conventional drone subregions can be performed in parallel. The distributed application runs on a cluster on Amazon Web Services (AWS). MapReduce jobs are run on AWS as Elastic MapReduce (EMR) and data is read from and written to S3 buckets similar to HDFS. Hadoop 2.5.1 and MapReduce2 were considered for this execuiton. The experiments were performed on a 5 node Hadoop cluster with 1 master and 4 slave nodes. An overview of the distributed system architecture is shown in Figure 3.

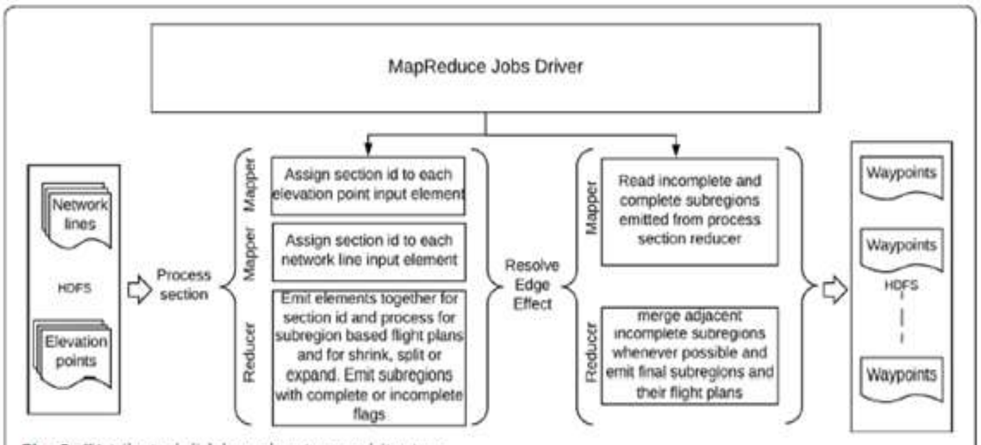
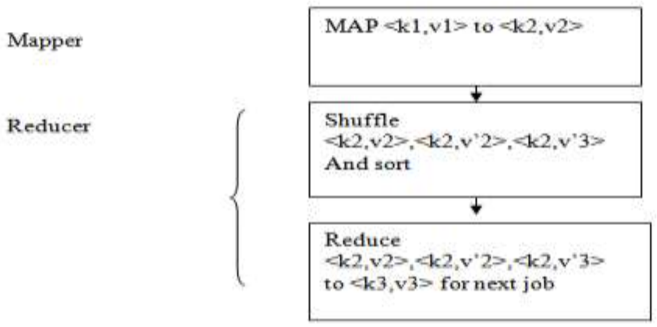

MapReduce Jobs Driver
Network lines
HDFS
Elevation points
Process section
Mapper
Assign section id to each elevation point input element
Mapper
Assign section id to each network line input element
Reducer
Emit elements together for section id and process for subregion based flight plans and for shrink, split or expand. Emit subregions with complete or incomplete flags
Resolve Edge Effect
Mapper
Read incomplete and complete subregions emitted from process section reducer
Reducer
merge adjacent incomplete subregions whenever possible and emit final subregions and their flight plans
Waypoints
Waypoints
HDFS
Waypoints

Fig. 3 Distributed disk based system architecture

The flight plan building is implemented in 3 steps. The input data is stored in a distributed file system HDFS. The input data in converted into key value pair <k,v> where key is an index and v are geometric coordinates values of the network line and elevation data .The idea is to combine similar data under one key removing redundant appearance of the data and distribute them over n nodes to execute parallallly.The three phases that transform the data are as follows:


Mapper
MAP <k1,v1> to <k2,v2>
Reducer
Shuffle <k2,v2>,<k2,v'2>,<k2,v'3> And sort
Reduce <k2,v2>,<k2,v'2>,<k2,v'3> to <k3,v3> for next job

1. ***Map*** reads the input as key-value pairs < k1, v1 > and transforms them
to key-value pairs < k2, v2 >
2. ***Shuffle*** The key-value pairs < k2, v2 > are distributed across all machines.
This stage guarantees to sort the keys and key value are combined together for reduce stage.
3**. *Reduce*** It group the values together as < k2,< v2, v′2, v′′2 , . . . >> and emitanother set of key-value pairs < k3, v3 > to be processed in the next job.

2.3 Construction of flight plan

A simple pseudocode how a flight plan is constructed

Step1 :Take input network and elevation data from the disk.

```
Step 2:Checks for spatial index

Step3: if present(spatial index(network data))
Load index from the memory
else create spatial index(netwok data) & write to disk

Step4: Construct waypoints with elevation network points
using KNN algorithm.

Step5: if CC and AC are satisfied for conventional drone
execute query in CD
else  shrink or expand the region &
again Construct the waypoint
if CC and AC are satisfied for QC
execute query in QC
else  take new points go to step4.

Step6: Output  the flight plan.
```

Explanation:

The software reads index file from disk if present otherwise reads the network line and elevation data {network i} and {elevation i} and assign them spatial index Si. The algorithm then starts searching index with conventional drone Dc sized subregions .Obtained result is set of the query that include network lines and elevation points inside the subregion. It calculates waypoints along network lines in query and then elevation of waypoints using kNN query and taking average of the elevation of nearest neighbors. If the network lines suit the conventional drone autonomy constraint Ac and does not satisfy the conventional drone climbing angle constraint Cc, then it searches spatial index for the subregion using default size of quadcopter Dq . Each consecutive quadcopter size subregion that simply satisfies the climbing angle constraint is merged with previous one. Regions that do not satisfy Cc require quadcopter. If the total length of network lines are extremely large or less than β% threshold of the autonomy of quadcopter or conventional drone, then the subregion is shrunk or expanded recursively till it completely satisfies the constraint of autonomy for the respective drone type.The output for each subregion as waypoints and landing point is written.

III. Distributed Sytem DIMPL

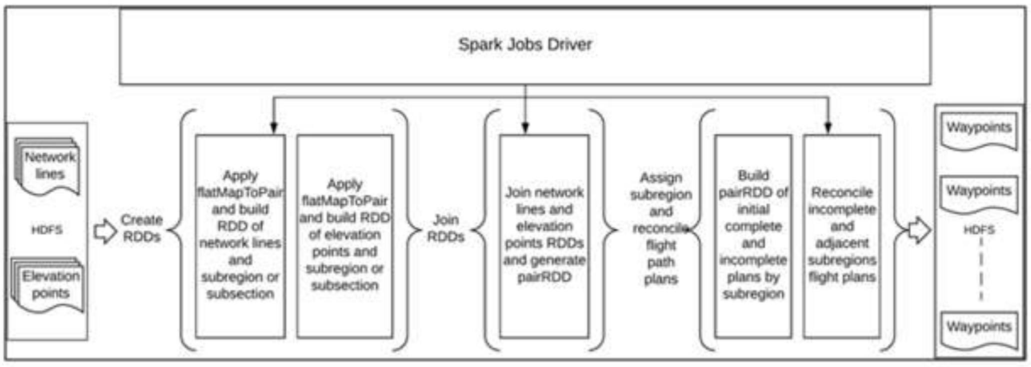
Since the amount of data is huge writing immediate output to the disk became a costly affair in DIFPL.This led to the formulation of DIMPL that constructed flight path using an in-memory technique . The in-memory distribution implementation, unlike the DIFPL implementation, does not need to write the intermediate output to disk, it performs all operations in-memory. For that purpose Apache Spark an in-memory based framework was considered that allows computations to be distributed in-memory over a large number of nodes in a cluster.

3.1 Methodology

Since the algorithm is a successor of DIFPL the data consideration and climbing and autonomy constraint remain same for DIMPL as well. We will directly focus on the in-memory approach that optimized DIFPL.

3.2Architecture

The programming construct in Spark retransform the data on a disk into RDDs (resilient distributed datasets) and then deploy further actions to the RDDs on subsets of data in processes called *executors* on cluster nodes to produce values that can be returned to next level processing. RDDs are highly fault tolerant in case one or more nodes of the cluster fail. The algorithms provided by Spark are ML and statistical functions that are highly iterative in nature. Performing complex distributed operation on a disk based distribution framework such as MapReduce is quiet expensive computationally due to the need to write data to the disk during each iteration. The in-memorapplication runs as a cluster on AWS. Spark jobs are run on AWS and data is read from and written to S3 buckets which is similar to HDFS[16]. As the figure shows, the various in-memory RDD transforms that are performed allow the construction of flight plans by subregion-wise distributions. A Spark jobs driver controls the sequence of RDD transformations and actions.For each data RDD is prepared which is then converted into a pair RDD which is then processed parallely.



Spark Jobs Driver
Network lines
HDFS
Elevation points
Create RDDs
Apply flatMapToPair and build RDD of network lines and subregion or subsection
Apply flatMapToPair and build RDD of elevation points and subregion or subsection
Join RDDs
Join network lines and elevation points RDDs and generate pairRDD
Assign subregion and reconcile flight path plans
Build pairRDD of initial complete and incomplete plans by subregion
Reconcile incomplete and adjacent subregions flight plans
Waypoints
Waypoints
HDFS
Waypoints

Fig 4- Distrubuted in memory architecture

IV. Basic Performance Analysis

The performance diagram shown below clearly states that the query execution time in DIMPL is less than in DIFPL .Also for sequential execution time consumption is more than when the query was executed on different nodes.However the performance remain constant even if try to increase the amount of nodes in the cluster.

4.1 Performance Graph

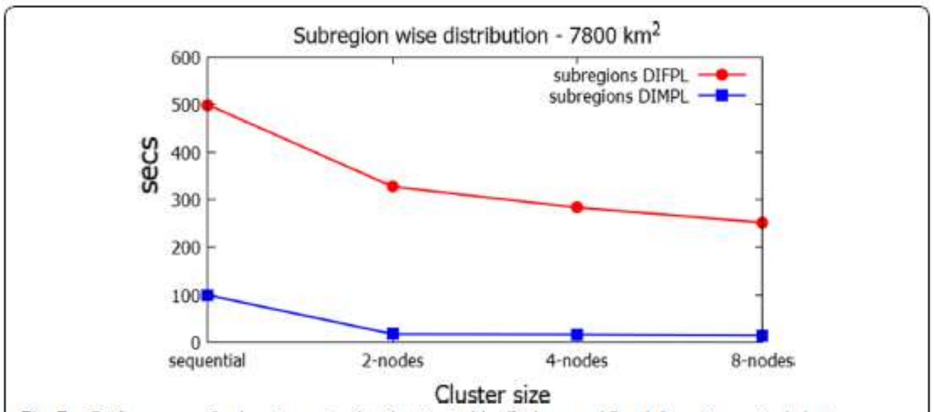


Fig. 5 Performance of subregion-wise distribution in MapReduce and Spark for various sized clusters

4.2 Comparision table DIFPL VS DIMPL

DIFPL	DIMPL
Runs on AWS and uses hadoop map reduce framework for preparing flight plans.	Runs on Amazon spark utilizing RDDs framework .
Disk based implementation	In-memory Query execution
Considers different drone capabilities	Considers different drone capabilities
Query execution is slow compared to DIMPL	Query execution is fast comparitively
Increased number of nodes does not cause more optimization	Increased number of nodes does not cause more optimization
Hadoop 2.5.1,MapReduce2 is used	Spark 1.5.1 is used
Less fault tolerant	more fault tolerant

V. Conclusions

Both the algorithms provided a decent approach to flexibly divide a large area into subregions and dynamically readjust them to optimally cover each region with a single drone flight. The algorithms combine spatial data and drone limitations or constraints, which are modeled as linear inequalities, to automate the flight paths of the drones. The distributed implementation provides an outstanding way to handle large datasets that cannot be processed on a single node. Utilizing in-memory distribution significantly speeds up the flight paths building process compared to disk based distribution . The flight plans produced by the new distributed model are similar in numbers to those obtained by single node implementations but aregenerated more efficiently. The technique applied here is not only useful for the assigned task of surveying power lines but can easily be extended to a host of other drone applications such as surveying coastlines for hurricane damage, forest surveys for logging, farm surveys for fertilization, insecticide spraying and watering[17], and many others.

VI. Future Scope

The presented scenario is easy to implement ,reliable and cost effective.Considering the huge environment it has been designed for ,we can build a complete medical solution for high altitude areas with varying terrain where human reachability is limited.The drone can be planned to effectively deliver urgent medicine in medical centres in these areas at times of natural calamity like earthquakes and flood.However an effective solution needs to be in process to handle unavoidable obstruction and bad weather conditions.

References

[1]. Visse A, Dijkshoorn N, van der Veen M, Jurriaans R. Closing the gap between simulation and reality in the sensor and motion models of an autonomous AR drone. In: Proceedings of the international micro air vehicles conference. 2011.

[2]. Wang X, Poikonen S, Golden B. The vehicle routing problem with drones: several worst-case results. Optim Lett. 2017;11:679.

[3]. Babel L. Flight path planning for unmanned aerial vehicles with landmark-based visual navigation. Robot Autonom Syst. 2014;62(2):142–50.

[4]. Bills C, Chen J, Saxena A. Autonomous MAV flight in indoor environments using single image perspective cues. In: IEEE international conference on robotics and automation (ICRA). 2011. p. 5776–83.

[5]. Yu J, Wu J, Sarwat M. GeoSpark: a cluster computing framework for processing large-scale spatial data. In: Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems, SIGSPATIAL '15. Seattle, Washington: ACM; 2015. p. 1–70.Zefang he,long jaho, The Comparison of Four UAV Path Planning Algorithms Based on Geometry Search Algorithm 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC),2017

[6]. Saska M, Krajnik T, Faigl J, Vonasek V, Preucil L. Low cost MAV platform AR-drone in experimental verifications of methods for vision based autonomous navigation. In: 2012 IEEE/RSJ international conference on intelligent robots and systems (IROS). 2012. p. 4808–9.

[7]. Hall J, Anderson D. Reactive route selection from pre-calculated trajectories—application to micro-UAV path planning. Aeronaut J. 2011;115(1172):635–40.

[8]. Do T, Carrillo-Arce LC, Roumeliotis SI. Autonomous flights through image-defined paths. In: Bicchi A, Burgard W, editors. Robotics research. Springer proceedings in advanced robotics, vol. 2. Cham: Springer; 2018.

[9]. De Paula Santos, G., Garcia Marques, L., Miranda Neto,M., Cardoso, A., Lamounier, E., and Yamanaka, K.(2013). Development of a genetic algorithm to improvea uav route tracer applied to a man-in-the-loopflight simulator. In Virtual and Augmented Reality(SVR), 2013 XV Symposium on, pages 284–287.

[10]. Stanisław Konatowski, Piotr Pawłowski. Ant colony optimization algorithm for UAV path planning.Published in 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET). Slavske, Ukraine.2018

[11]. Holub JS. Improving particle swarm optimization path planning through inclusion of flight mechanics. Graduate theses and dissertations. Paper 11741. 2010.

[12]. Eldawy A, Li Y, Mokbel MF, Janardan R. CG_Hadoop: computational geometry in MapReduce. In: ACM SIGSPATIAL.2013. p. 294–303.

[13]. Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters. Commun ACM. 2008;51(1):107–13.

[14]. Shukla M, Chen Z, Lu CT. DIFPL—distributed drone flight path builder system. In: Proceedings of the 1st internationalconference on geographical information systems theory, applications and management, GISTAM, Barcelona,Spain. 2015, p. 17–26.

[15]. Shukla M,chen Z,Lu CT. "DIMPL: a distributed in-memory drone flight path builder system." *Journal of Big Data*, vol. 5, no. 1, 2018. *Gale Academic Onefile*, Accessed 17 Dec. 2019.

[16]. Zaharia M, Chowdhury M, Franklin MJ, Shenker S, Stoica I. Spark: cluster computing with working sets. In: Proceedings of the 2Nd USENIX conference on hot topics in cloud computing, HotCloud'10. Berkeley: USENIX Association; 2010. p. 10.

[17]. Tripicchio P, Satler M, Dabisias G, Ruffaldi E, Avizzano CA. Towards smart farming and sustainable agriculture with drones. IN: 2015 international conference on intelligent environments, Prague. 2015. p. 140–[18]Floreano D, Wood RJ. Science, technology and the future of small autonomous drones. Nature. 2015;521:460–6.

Minakshi Gautam"Efficient Distributed Flight Path Construction Techniques In Uav " *International Journal of Engineering Science Invention (IJESI)*, Vol. 09(01), 2020, PP0 11-17