Coverage Path Planning and Convergence for a swarm of UAVs

A report submitted for BTP phase III

by

Sankeerth Reddy Proddduturi

(Roll No. 180108034)

Under the guidance of

Dr. Indrani Kar

DEPARTMENT OF ELECTRONICS & ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

November 2021

Abstract

Recent advances in unmanned aerial vehicles (UAVs) and sensor technology have triggered wide-ranging applications in mobile surveillance, including search and rescue missions, crop management and environmental monitoring. To solve this problem, two principal research components, namely coverage path planning (CPP) and clustering & consensus, are identified. Under the subspace of coverage path planning, partitioning of workspace, waypoint generation, TSP are identified as sub parts. Clustering and Convergence along with those sub parts are the primary hurdles that are needed to be overcome. This report aims to describe solution for partition of the area, linear surveying and a Vanilla-TSP with only distance as a constraint.

Contents

List of Figures

Nomenclature

CPP	Coverage Path Planning
UAV	Unmanned Aerial Vehicle
POI	Point Of Interest
FoV	Field of View
TSP	Travelling Salesman Problem
ROS	Robot Operating System

Chapter 1

Introduction

Multi-agent systems have been accomplish the tasks that would be difficult for the average agent system to complete. Controlling multiple robots has received a lot of attention recently. A multi-agent system is a collection of agents that work together as a single unit through communication. One of the applications where distributed agents can help is Coverage Path Planning (CPP). The coverage period would be minimized, and if one of the agents were to fail, the coverage would not be affected because other agents could take over. Multi agent systems can cover large regions when paired with energy efficient survey patterns and efficient path planning algorithms. This could be quite beneficial in identifying important areas in natural disaster areas that require immediate intervention. The following techniques are used: area partitioning, linear survey, and single agent TSP.

Chapter 2

Literature

Within the robotics and control field, the CPP problem for numerous UAVs has been a major topic. For example, CPP can be used to decrease control effort, time, and single-robot travel distance, among other things [1]. The CPP problem for a single UAV is a subproblem of the CPP problem for multiple UAVs. The Vehicle Routing Problem (VRP) [2] is a well-known approach of tackling the UAV path planning problem. However, as seen below [4], linear surveying and prioritizing straighter lines boost the UAV's range. Our path planner is based on the Traveling Salesman Problem (TSP), which will be updated to include a number of extra constraints. We employ PX4 [5], an autopilot developed to operate a variety of vehicles and provide users greater control at a higher level. The convergence problem for a swarm of robots with energy limits is also addressed in this approach. Traditional optimal motion planning can be conceived of as trajectory generation independent of the coverage problem. The authors of [6] are focused in leader-follower consensus under energy restrictions.

Chapter 3

Problem Statement

Considering a large area with n agents

1. Area partitioning
2. Single agent linear survey
3. TSP on random POI's

Implemented Equal Area Partitioning, Linear Survey using ROS-px4-Gazebo simulation setup and solving for optimal TSP for a undirected $2d$ graph using Stochastic Search Algorithm. This covers the Coverage Path Planning part of the project.

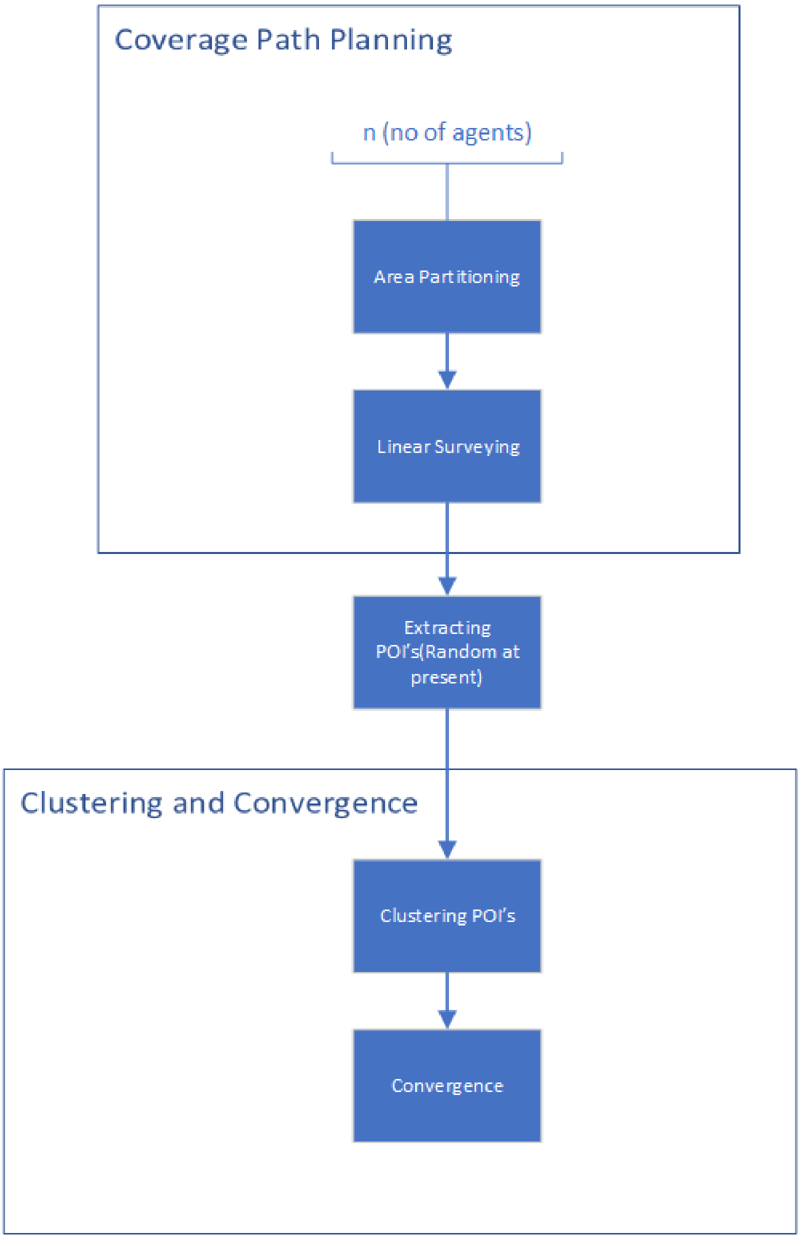


Figure 3.1: Problem Statement

Chapter 4

Simulation and Results

4.1 Simulation Setup

Simulation has been done using ROS-PX4-Gazebo setup. MAVROS is a ROS package which communicates with PX4 autopilot firmware using MAVLINK protocol. Gazebo is used as the 3d simulator to visualise and test the algorithms.

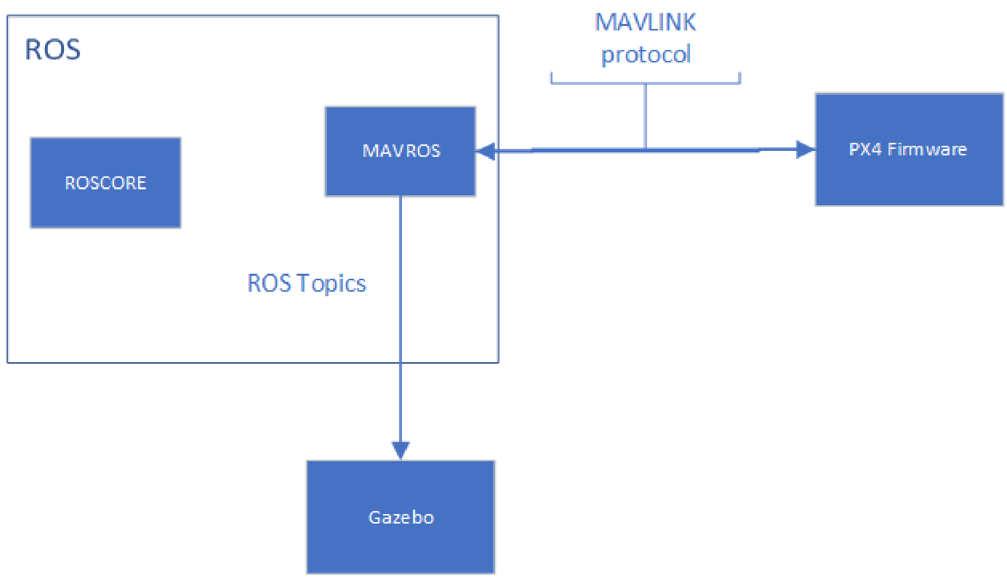
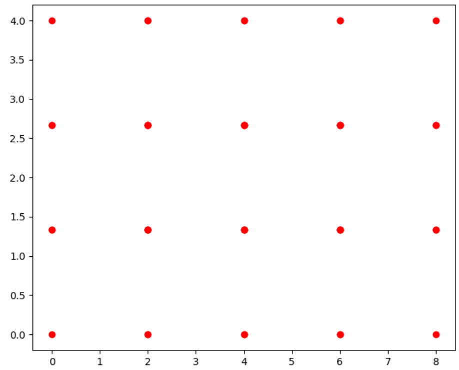


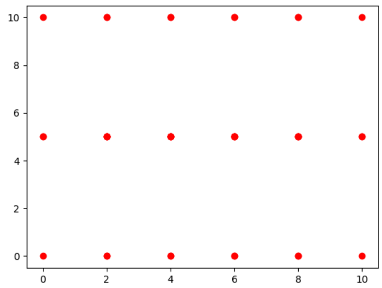
Figure 4.1: Simulation setup

4.2 Area partitioning

Given any rectangular area and the number of agents n the algorithm divides the area into n equal regions. The algorithm calculates the factors pairs of n which have minimum difference uses them to slice the area horizontally and vertically. This way the area tends to be nearer to original and as a square.



(a) 12 regions in 8x4 area



(b) 10 regions in 10x2 area

Figure 4.2: Area partition in different cases

4.3 Linear Surveying

The area partitioning algorithm sends out coordinates of the area which they need to cover for each of the agent. The agents then plan a linear survey path given the altitude and the field of view of the on-board camera/sensor which maps/extracts POI's from the area. The agents take velocity messages as input and move accordingly. A simple proportional controller over the PX4 autopilot is used to achieve this. Linear surveying has the advantage of straighter paths and less abrupt turns which optimizes the battery life as discussed in this paper [3]. Linear surveying can be further improved upon using Zamboni pattern. Linear surveying decreases the computational overhead which can be useful in detecting POI's.

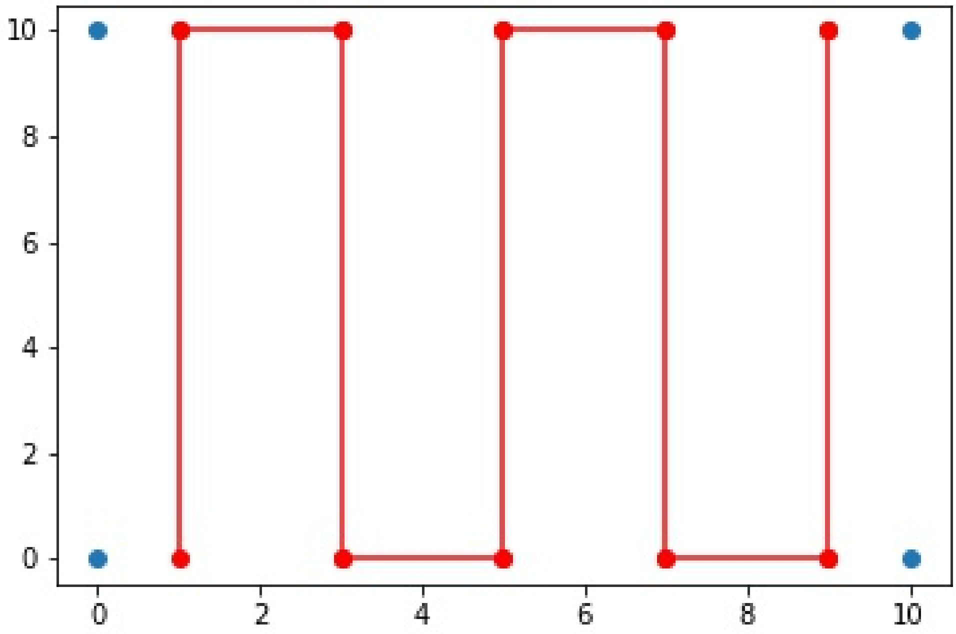


Figure 4.3: Linear survey in a 10x10 area with FoV of 2x1

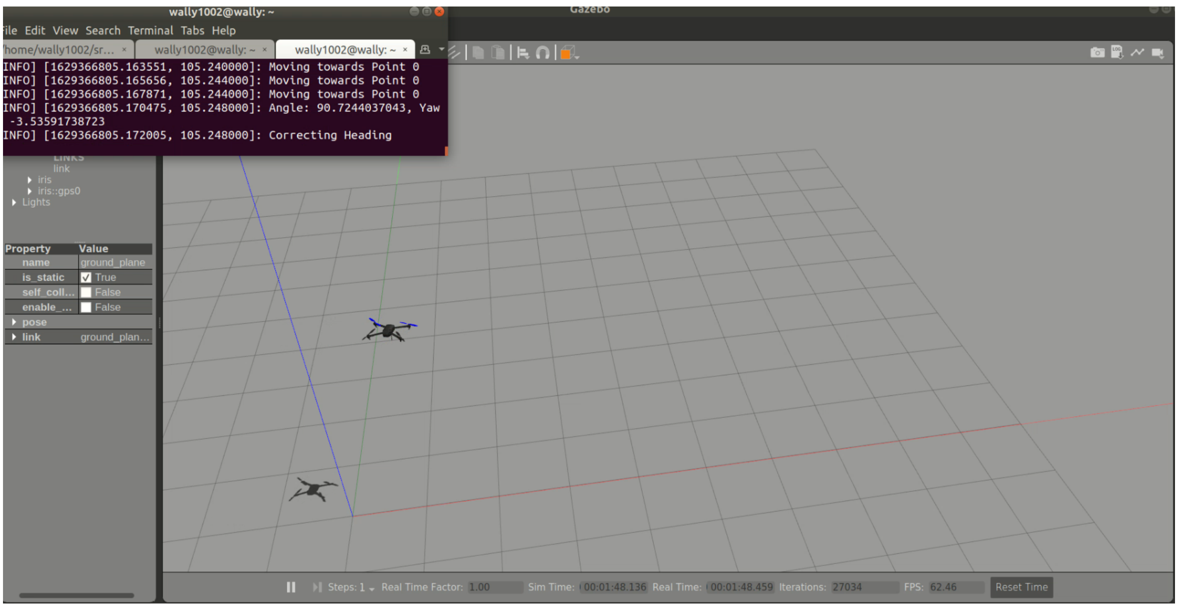


Figure 4.4: Linear survey in Gazebo

4.4 Modelling TSP

After the linear survey we extract POI's using mapping or detection algorithms. For now we randomly select n points in the given area. We then generate the distance matrix for each point in those points. We then pass the distance matrix to the Stochastic Search algorithm which returns the optimal path to follow.

```
def solve_tsp():
        S, y = self.init_tsp()
        y_opt=y[0] #check for optimum solution at each and every step
        path_opt=S[0]
        for i in range(numIter):
            p=generateFitnessScores(i)
            S_new=[] #same as S_t+1
            #generating children by fitness score and storing in S_new
            #we generate K children for S[i] if p[i]=K
            for j in range(len(S)):
                for k in range(p[j]):
                    S_new.append(list(generateChild(S[j])))
            y_new=generateCostArray(S_new)
            S,y=chooseBestchild(S,S_new,y,y_new,self.m)
            for i in range(self.m):#checking for optimum solution
                if y[i]>y_opt:
                    y_opt=y[i]#update optimum solution
                    path_opt=S[i] #save optimum path
        return path_opt
```

Stochastic search generates a population of children using random swaps and selects the best of them using fitness scores which in turn is calculated using the sum of distances in the round trip considering that child. This method doesn't always guarantee the best solution but works very efficiently and fast for smaller number of points. Here are some paths generated by this algorithm.

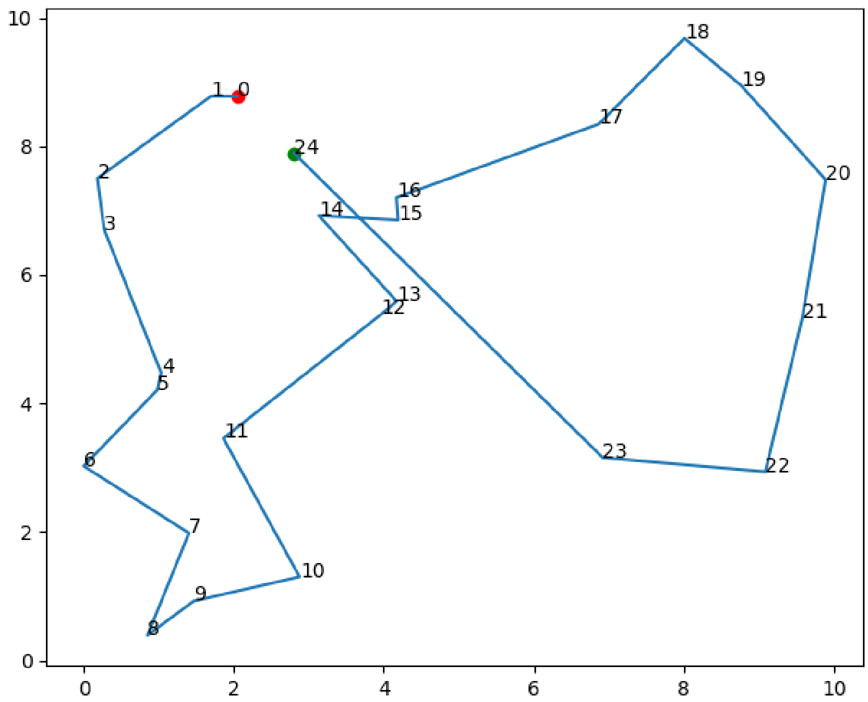


Figure 4.5: Solution for TSP for 25 random POI's

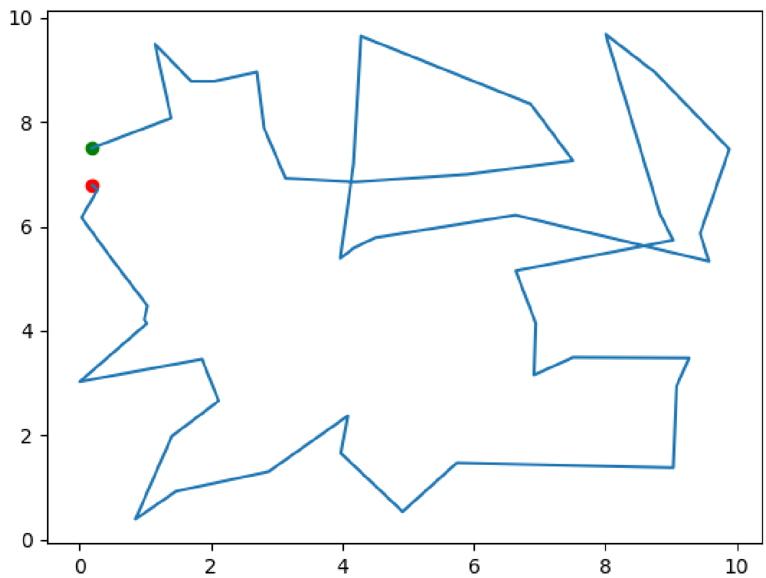


Figure 4.6: Solution for TSP for 50 random POI's

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Most of the work in coverage path planning and modelled a vanilla-TSP for clustering and convergence. TSP now has distance as the only metric to find the best route and is only for single agent. Linear survey is very energy efficient and requires less computational resources. TSP is not dependent on the number of agents so it can scale easily and help in convergence problem for multiple agents.

5.2 Future Work

Future work for this project mainly lies in extracting the point of interests. POI's can be extracted using 3d mapping or detection algorithms. These POI's should clustered optimally to ensure that the battery capacity of each drone is enough to converge on each of the cluster formed. TSP is to be improved using other constraints such as battery remaining, coverage time etc. Implementing a obstacle avoidance algorithm can be useful. Implementing a single agent system on hardware.

Bibliography

[1] S. Chung, A. A. Paranjape, P. Dames, S. Shen and V. Kumar, ”A Survey on Aerial Swarm Robotics,” in IEEE Transactions on Robotics, vol. 34, no. 4, pp. 837-855, Aug. 2018, doi: 10.1109/TRO.2018.2857475.

[2] J. Yu and S. M. LaValle. Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics. IEEE Transactions on Robotics, 32(5):1163–1177, 2016

[3] J. Modares, F. Ghanei, N. Mastronarde and K. Dantu, ”UB-ANC planner: Energy efficient coverage path planning with multiple drones,” 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 6182-6189, doi: 10.1109/ICRA.2017.7989732.

[4] J. Xie, L. R. Garcia Carrillo and L. Jin, ”Path Planning for UAV to Cover Multiple Separated Convex Polygonal Regions,” in IEEE Access, vol. 8, pp. 51770-51785, 2020, doi: 10.1109/ACCESS.2020.2980203.

[5] L. Meier, D. Honegger and M. Pollefeys, ”PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms,” 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 6235-6240, doi: 10.1109/ICRA.2015.7140074.

[6] Dan Zhang, Zhenhua Xu, Dipti Srinivasan, and Li Yu. Leader–follower consensus of multiagent systems with energy constraints: A Markovian system approach. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(7):1727–1736, 2017.