Multiagent Path Planning for Proximal Coverage of Agricultural Farms

Thesis

Submitted in partial fulfillment of the requirements of
BITS F421T Thesis

By

Suhrudh Sarathy
ID No. 2019A3TS0390G

Under the supervision of:

Dr. Shishir N Y Kolathaya
&
Dr. Kizheppatt Vipin

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, GOA CAMPUS

September 2023

Declaration of Authorship

I, Suhrudh Sarathy, declare that this Thesis titled, 'Multiagent Path Planning for Proximal Coverage of Agricultural Farms' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ________________________________

Date: ________________________________

Certificate

This is to certify that the thesis entitled, "*Multiagent Path Planning for Proximal Coverage of Agricultural Farms*" and submitted by Suhrudh SARATHY ID No. 2019A3TS0390G in partial fulfillment of the requirements of BITS F421T Thesis embodies the work done by him under my supervision.

Supervisor
Dr. Shishir N Y KOLATHAYA
Professor,
Indian Institute of Science, Bangalore
Date:

Co-Supervisor
Dr. Kizheppatt VIPIN
Asst. Professor,
BITS-Pilani Goa Campus
Date:

"All the world's a stage, And all the men and women merely players"

William Shakespeare

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, GOA CAMPUS

Abstract

Bachelor of Engineering

Multiagent Path Planning for Proximal Coverage of Agricultural Farms

by Suhrudh Sarathy

In the field of agriculture, an important application of Robotics is to inspect the biomass. Such inspections can be achieved fruitfully if the robot can reach the area of the biomass carefully and cover all the clusters of biomass in a effective manner. It also becomes logical to use multiple agents for this task as the capacity of a single robot greatly minimises the coverage of the agricultural farm.

The problem of covering a given area with least cost incurred w.r.t path length, control effort, time etc. comes under the class of NP-Hard problems closely resembling the Travelling Salesman Problem (TSP). The solution to the TSP exists and is intractable given the scale of agriculture fields. We observed that the arrangement of biomass clusters in the agricultural fields greatly simplifies this problem.

Firstly, considering certain relaxations and constraints, we formulate the above problem of covering a given area as a Graph Traversal Problem where we construct a graph of nodes proximally positioned to cover the biomass clusters effectively. We show an analytical bound to the constraints and evaluate the costs in simulated environments.

Secondly, we analyse the existing methods for the problem of Multi agent coverage. The problem of Multi agent coverage is similar to the Capacitated Vehicle Routing Problem (CVRP), which is also NP-Hard. We use existing methods to solve the CVRP and our algorithm to perform a two-level decentralised planning to efficiently cover the agricultural fields using multi agents.

Acknowledgements

First and foremost, I thank my thesis mentor, **Mr. Kumar Ankit** for believing in me and providing me with the opportunity to explore, learn and experiment independently throughout my thesis. I thank him for providing the necessary support and help when I was stuck. Discussions with him helped me streamline my thinking and focus on the problem at hand. I can't thank him enough for trusting me and my code for running experiments on costly hardware thus enriching my experience.

I thank my supervisor **Dr. Shishir Kolathaya** and **Dr. Debashish Ghose** for providing timely supervision and correction in the weekly meetings. I would like to thank Dr. Ghose for allowing me to work at the wonderful Mobile Robotics Laboratory (MRL) and use the Lab resources. The environment and peers at MRL helped me learn a lot throughout my thesis. I would specially like to thank **Dr. Vidya Sumathy** for being such a sweet and kind person. I would also like to thank **Hariharan** for being sy collaborator on the project and for being such a genuine and wonderful person.

I express my gratitude to my university **BITS Pilani, KK Birla Goa Campus** for the magic it is. I thank the administration for giving me an opportunity to pursue my thesis. I am thankful to **Prof. Vipin Kizheppatt** for being my On-Campus Supervisor. I am grateful to **Dr. Rakesh Warier** for the guidance during my Undergraduate studies. I thank my peers at the Electronics Robotics Club for the opportunity to build robots and learn along. I appreciate the patience of my friends **Rachit Gupta** and **Ishan Neogi** for being my moral support. I am forever indebted to my mentor, **Harshal Deshpande** for introducing me to the field of Robotics and tolerating and answering all the questions I ask him. I would not have pursued this without him.

I express my deepest gratitude to my Aunt and Uncle, **Mrs. Indumathi** and **Mr. Sampath** for kindly allowing me to stay at their place and provide for me throughout the thesis duration.

Finally, I am indebted, grateful and thankful for the unconditional love and support of my **Parents** and my **Sister**. I am grateful to my parents for allowing me to make my own choices and stand by me during the hard times in life. I thank them for allowing me to question and remain curious. This has helped me numerous time throughout my life. They always taught me to respect people and to respect work. The values I learnt from them keep me humble and hardworking even in the worst of times. I owe them not only this work, but also work to come, as nothing would be possible without their countless sacrifices.

Contents

List of Figures

List of Tables

Abbreviations

TSP	**T**ravelling **S**alesman **P**roblem
CVRP	**C**apacitated **V**ehicle **R**outing **P**roblem
MCC	**M**ultiple **C**luster **C**overage
MA-MCC	**M**ulti **A**gent - **M**ultiple **C**luster **C**overage
MST	**M**inimum **S**panning **T**ree

To my family, friends, teachers

Chapter 1

Introduction

1.1 Motivation

Use of modern technology has been on the rise in the recent years in the field of Agriculture. Robotics plays a huge part in provinding modern solutions to the problems in Agriculture. In recent years, teams of Unmanned Ground Vehicles (UGV) and Unmanned Aerial Vehicles (UAV) have been extensively deployed to perform various tasks in the agriculture fields such as inspection, weeding, spraying etc [11]. It is imperative that these robots remain autonomous for their effective usage. Thus new algorithms are required specifically for deploying UGVs and UAVs in agricultural environments as these environments pose specific challenges due to the scale of the environment. The scale of the environment demands efficient use of power supply resources and hence requires efficient planning. It is also important that the behaviour of the robots remain predictable for safe and repeated uses.

Modern day cameras allow robots to perform high resolution imaging thus helping in tasks like inspection, surveillance etc. A UAV is able to perform efficient 3D reconstruction [8] [14] of the world using a recorded video or multiple recorded images using tools such as Structure from Motion, Visual-SLAM etc. Using these 3D reconstructions enables use to analyse the field in real time. An important requirement for efficient reconstructions is to reach closest to the biomass. This is important as the resolution of the reconstruction depends on the quality of the images taken which depends on the distance the images have been taken from.

Adding to this, it is also important to cover the entire farm efficiently. The limitations of battery power onboard the robot require that efficient planning be done. In some cases, it also helps to have deterministic global paths for the robot to follow as it helps the ground station determine feasibility of the path before hand to enable safety and repeatability.

In this work, we focus on the problem of travelling a set of Proximal Points around multiple biomass clusters effectively. This problem, in essence is NP-Hard as it is similar to the Travelling Salesman Problem (TSP). According to our problem statement, existing solutions scale extremely poorly in terms of speed and resources required. Because the algorithm will eventually be deployed on resource-constrained hardware, an effective solution must be provided. We develop an algorithm where use exploit some relaxations of the environment and impose certain constraints to solve the traversal problem. We extend our algorithm to use multiple agents and evaluate our algorithm in simulated environments.

1.2 Proximal Points

Proximal points are the locations around a biomass that an agent can reach safely by maintaining certain safe distance to the contour of the canopy. Each biomass cluster has a set of Proximal points that are generated offline and will be used by UAV to traverse.

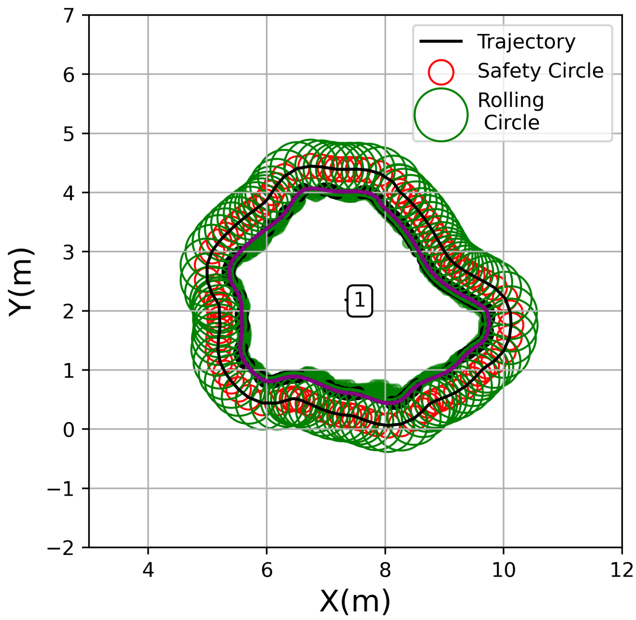


FIGURE 1.1: Proximal points around a given cluster. The safety circle and rolling circle are used to generate Proximal points. A point is considered unsafe when the radius of rolling circle equals the radius of safety circle.

1.3 Our Contribution

The problem of covering biomass clusters is NP-Hard, similar to TSP. However, standard TSP solvers do not account for the grid-based arrangement of agricultural land. Using the principles of Graph Traversal, we attempt to develop our Multiple Cluster Coverage (MCC) algorithm. Our algorithm exploits the geometrical arrangement of the agricultural farm land and produces nearly optimal path with less time and compute required than standard TSP solvers. We then extend our method to use Multiple agents by combining existing solutions to the Vehicle Routing Problem (VRP) and our algorithm.

1.4 Organisation

The document is divided into five chapters. The Multiple Cluster Coverage algorithm is described in Chapter 2, followed by the Multi Agent extension in Chapter 3. Chapter 4 concludes the document followed by Appendices.

Chapter 2

Multiple Cluster Coverage

2.1 Introduction

Given a set of biomass clusters, the problem of Multiple Cluster Coverage is to efficiently find the path connecting all Proximal Points of those biomass clusters. This is analogous to the Travelling Salesman Problem (TSP) where the task is to visit every node such that the total path length is minimised. However, the scale of the problem along with certain restrictions does not allow us to use TSP Solvers directly for this problem as the computation complexity of a TSP problem grows exponentially with the number of nodes.

However, in most agricultural plantations, biomass is planted in grid arrangement. In such cases the distances between successive biomass clusters is relatively uniform. We exploit this constraint along with allowing certain relaxations to build a Undirected Graph using the Proximal points as vertices and connecting the adjacent vertices with weighted edges. We use a lexicographical key with certain costs to select the next vertex to travel to. We show that using our constraints and our problem method reduces the computation complexity and time required to calculate near optimal paths.

2.2 Problem Setting

Given K clusters arranged in a $M \times N$ grid arrangement, each cluster has a set of Proximal points. Let the set of Proximal points belonging to the K clusters be Ω. We define a valid path σ starting from an arbitrary starting point $w_{\text{start}} \in \Omega$ as the path that traverses all the points at least once. Given a cost function $\mathcal{C}$ and a set of feasible paths Σ, the optimal path σ^* w.r.t to the cost function $\mathcal{C}$ can be defined as

$$\sigma^* = \arg\min_{\sigma \in \Sigma} \{\mathcal{C}(\sigma(\omega)) | \forall \omega \in \Omega\} \tag{2.1}$$

2.2.1 Similarity to TSP

The Travelling Salesman Problem is defined broadly as finding the shortest possible routes covering all the given nodes atleast once. This problem belongs to a class of NP-Hard problems and the worst case runtime of any solution to the TSP grows superpolynomially, but not more than exponentially. There are multiple formulations of TSP. But here, we use the Integer Programming Problem formulation, specifically the Dantzig–Fulkerson–Johnson (DFJ) [4] formulation below,

Given the nodes $1 \ldots n$,

$$x_{ij} = \begin{cases} 1 & \text{if edge connects i and j} \\ 0 & else \end{cases} \tag{2.2}$$

Consider the cost of moving from one node to another be $c_{ij} > 0$. Then the TSP can be formulated as

$$\begin{aligned} \min \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} c_{ij} x_{ij} \\ \text{S.T,} \sum_{i=1, i \neq j}^{n} x_{ij} = 1 \\ \sum_{j=1, j \neq i}^{n} x_{ij} = 1 \end{aligned} \tag{2.3}$$

The constraints here ensure that each node is visited only once. We can similarly formulate our problem of Proximal coverage in the lines of TSP where the nodes are the Proximal points of all clusters and the edges are connected such that no collision occurs with the biomass.

2.3 Related Work

The Travelling Salesman Problem has been extensively studied. Over time many exact and heuristic solutions have been developed to solve the problem [12]. This problem is known to be NP-Hard and cannot be solved in Polynomial time. Despite the computational difficulty, TSP can be solved effectively using heuristic methods such as Genetic Algorithm [7], Tabu Search, Simulated Annealing etc [12].

There also exists approximate solutions for solving the TSP. Approximate algorithms are efficient algorithms that provide approximate solutions to optimisation problems with theoretical bounds [13].

Previous works also shows a use of Minimum Spanning Trees (MST) on Graphs to reduce the TSP problem into multiple smaller problems [9]. However, the usage of MST starkly differs to that of done in this work.

2.4 Method

Consider a grid arrangement of K biomass clusters in $M \times N$ rows and columns. The Proximal points belonging to a cluster $k \in K$ can be denoted as P_i^k. The total cost of circumnavigating a cluster k can be given as $C^k = \sum_{i=0}^{n-1} ||P_{i+1}^k - P_i^k||$.The cost of switching from an arbitrary node i in cluster k and another node j in cluster l can be written similarly as $C_{ij}^{kl} = ||P_i^k - P_j^l||$. The total cost of the path to cover multiple clusters can be formulated as below

$$\begin{aligned} \min \sum_{i=0}^{K} C^k + \sum_i \sum_j C_{ij}^{kl} x_{ij} \\ \text{S.T,} \sum_{i=1, i \neq j}^{n} x_{ij} = 1 \\ \sum_{j=1, j \neq i}^{n} x_{ij} = 1 \end{aligned} \tag{2.4}$$

Where,

$$x_{ij} = \begin{cases} 1 & \text{if edge connects i and j} \\ 0 & else \end{cases}$$

The formulation above is similar to that of the TSP [2.3]. We propose certain relaxations and add additional constraints to modify the formulation into a Graph traversal problem.

2.4.1 Proposed Approach

The cost of any valid path $\sigma \in \Sigma$ can be represented as the sum of cost of covering all clusters and the cost of switching from once cluster to another. Similar to the TSP, the variable x_{ij} affects the optimality of the solution. In our approach, we convert the problem into a Graph Traversal problem. We construct a cyclic graph around each cluster with Proximal points of the cluster as the vertices and connect alternate clusters with edges building a Weighted Undirected graph. We later use lexicographical keys to obtain next best vertex to traverse to and completely traverse the graph. The proposed algorithm is divided into two steps, Graph Construction and

Graph Traversal. Both these algorithms are independent of each other in a sense that once a Graph is constructed, it can be traversed multiple times.

Additionally, we impose the following relaxation and constraints to obtain near optimal solutions.

1. Any vertex in the Graph can be traversed more than once.
2. Any vertex can only be connected to two neighbours of the same cluster and one vertex of other cluster.
3. Two clusters can be connected only by a predefined switching edge.
4. The agent is constrained to always switch at the switching edge

2.4.2 Graph Construction

Given the Proximal points P^n for n^{th} cluster, we construct a graph $G_\phi(V_\phi, E_\phi)$. For vertices $v_i^n \in P^n$:

$$\begin{aligned} v_{i-}^n &= P^n_{(i-1) \mod S} \\ v_{i+}^n &= P^n_{(i+1) \mod S} \\ \text{where}, S &= |P^n| \\ \phi^n_{i,i-} &= (v_i^n, v_{i-}^n) \\ \phi^n_{i,i+} &= (v_i^n, v_{i+}^n) \end{aligned} \tag{2.5}$$

For all the Proximal points P^n of a n^{th} cluster we can define the set of all edges that completely cover the cluster forming a cyclic graph as

$$\Phi^n = \bigcup_{i=1}^{S} \{\phi^n_{i,i-}, \phi^n_{i,i+}\} \tag{2.6}$$

Now the graph defined by V_ϕ and E_ϕ covers all the clusters with each cluster fully connected.

$$\begin{aligned} V_\phi &= \bigcup_{n=1}^{N} P^n \\ E_\phi &= \bigcup_{n=1}^{N} \Phi^n \end{aligned} \tag{2.7}$$

To connect alternate clusters, we define a special edge called *switching edge.* A switching edge carries a connection from a vertex in once cluster to a vertex in a different cluster. A switching edge can be defined as

$$\psi^{n,m} = (P_i^n, P_j^m) \tag{2.8}$$

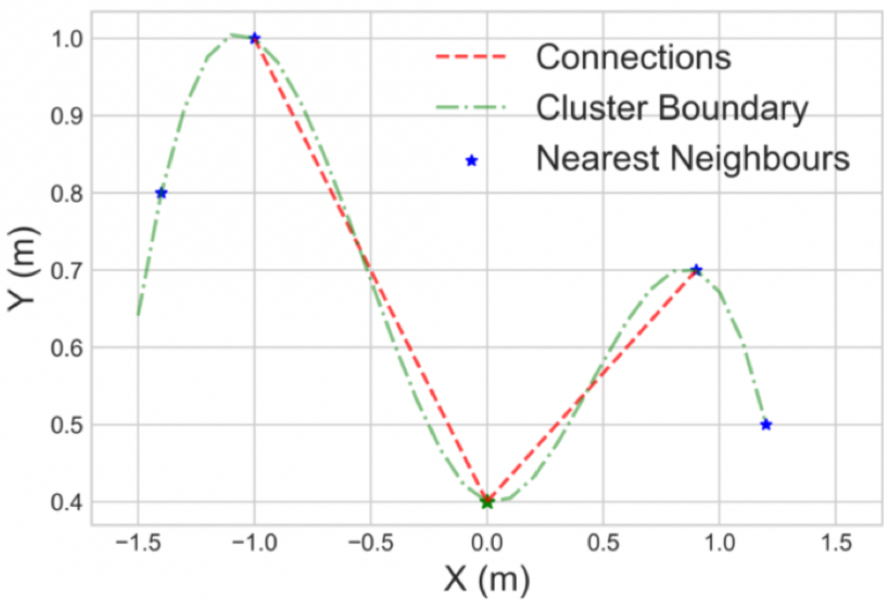


FIGURE 2.1: Every point is connected to two of its neighbours each in opposite sides to each other with respect to the normal at a point. This ensures that the Graph constructed is complete.

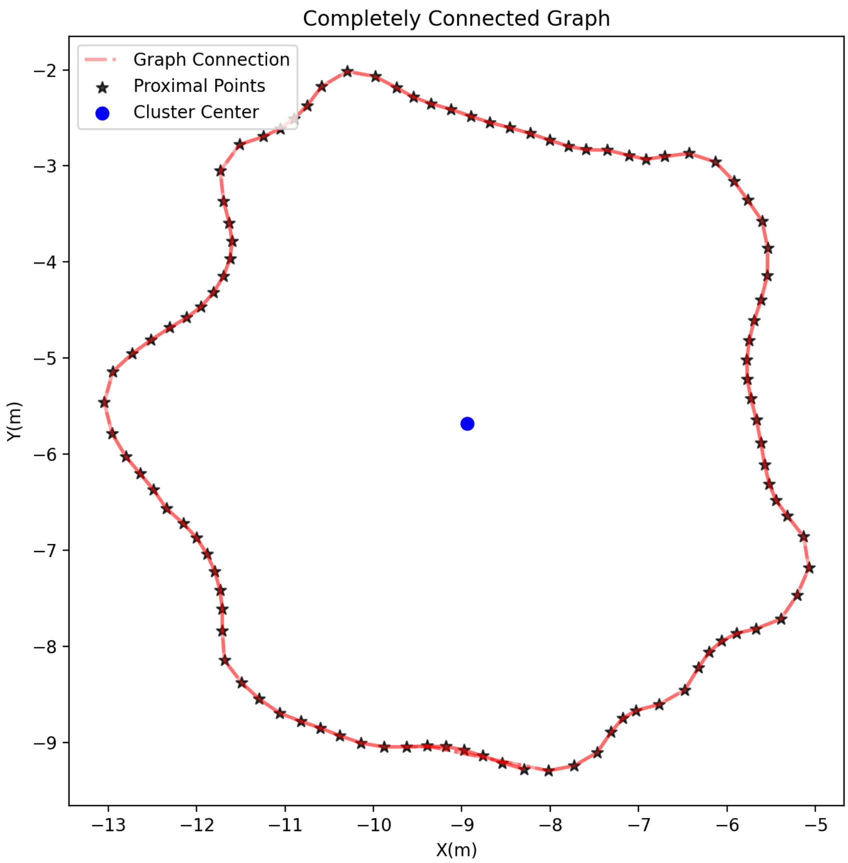


FIGURE 2.2: Fully connected graph generated using the Proximal points of a cluster

where the points P_i^n, P_j^m are the points belonging to n^{th}, m^{th} cluster respectively.
Using Proximal points P^n, the center of a cluster μ^n and radius of the cluster r^n, can be defined as

$$\begin{gathered}\mu^n = \frac{\Sigma P_i^n}{|P^n|} \\ r^n = \max_{p \in P^n} ||\mu^n - p||\end{gathered} \tag{2.9}$$

We construct a graph $G_\phi(V_\phi, E_\phi)$ given the location of the depot D and the cluster centers μ^n as

$$\begin{gathered}V_\psi = \{\bigcup_{n=0}^{K} \mu^n\} \cup \{D\} \\ E_\psi = \{(v^i, v^j) \mid v^i, v^j \in V_\psi \text{ and } v^i \neq v^j\}\end{gathered} \tag{2.10}$$

We can define the total path cost to be

$$C = \sum_{i=0}^{K} C^k + \sum_i \sum_j C_{ij}^{kl} x_{ij} \tag{2.11}$$

Owing to our constraint of the switching edges, we can reformulate the cost function as follows

$$C = \sum_{i=0}^{K} C^k + 2 \sum_{(i,j) \in E_\psi} C^{ij} \tag{2.12}$$

Since we fix the switching edge and constrain the agent to move from once cluster to another via the switching edge, the total cost now becomes the sum of cost to cover all the clusters and the cost of going and coming through clusters using the switching edge.

To minimise the cost function now is to minimise

$$\min C \implies \min \sum_{(i,j) \in E_\psi} C^{ij} \tag{2.13}$$

We obtain the Minimum Spanning Tree (MST), $T(V_\phi, E_\phi^*)$ of the graph G_ϕ where $T \subset G_\phi$. The MST T consists of the edges with the least sum of costs and hence minimises our cost function.

$$E_\psi^* = \min \sum_{(i,j) \in E_\psi} C^{ij} \tag{2.14}$$

Using the edges E_ψ^* we find the nearest vertex of a the cluster that is closest to the point where the line joining the centers intersects with the boundary. For an edge $e \in E_\psi^*$ we find a new edge $\hat{e}$ that uses the Proximal points as vertices instead of the cluster centers to connect two clusters.

$$e = (v^n, v^m)$$

$$
\begin{aligned}
\mu^n, r^n &= v^n \\
\mu^m, r^m &= v^m \\
\hat{\mathbf{x}}_{mn} &= (v^m - v^n)/||v^m - v^n|| \\
x^n &= v^n + r^n \cdot \hat{\mathbf{x}}_m n \\
x^m &= v^m - r^m \cdot \hat{\mathbf{x}}_m n \\
\hat{v}^n &= \underset{v \in V\phi}{||x^n - v||} \\
\hat{v}^m &= \underset{v \in V\phi}{||x^m - v||} \\
\hat{e} &= (\hat{v}^n, \hat{v}^m)
\end{aligned}
$$

Given the new edges $\hat{e}$ $\forall e$, the set of all the edges that connect alternate clusters can be written as

$$\Psi^{ij} = \{\hat{e} \mid \forall e \in E^*_\psi\} \tag{2.15}$$

Finally, the complete Graph connecting all the Proximal points along with switching edges can be represented as $G(V, E)$

$$
\begin{aligned}
V &= V_\phi \\
E &= E_\phi \ \cup \Psi^{ij}
\end{aligned} \tag{2.16}
$$

2.4.3 Graph Traversal

Given a fully connected graph $G(V, E)$, we traverse the graph completely using the edges $e \in E$, starting from the depot D until all the vertices $v \in V$ are not traversed atleast once. To traverse the graph, we evaluate all the edges emanating from a vertex v and choose the edge with least cost. We continue this until, we reach the depot D thus completely covering all the Proximal waypoints.

For a vertex v_i, the edge to be taken e_i^* can be chosen by,

$$
\begin{aligned}
e_i = \{(v_i, v_j) \mid (v_i, v_j) \in E\} \\
e_i^* = \underset{e \in e_i}{\operatorname{argmin}}\ k(e)
\end{aligned} \tag{2.17}
$$

Where $k(e)$ is defined as a priority key with the components: *traversal cost, switching cost, distance cost* in order of decreasing priority. Comparison of keys happens in lexicographical ordering.

$$k(e) = [\ c_{traversed}, c_{switching}, ||e||\ ] \tag{2.18}$$

Hence, the optimal path can be described as

$$\sigma^* = (e_1^*, e_2^*, \dots e_n^*)\ \forall e_i^* \in E \tag{2.19}$$

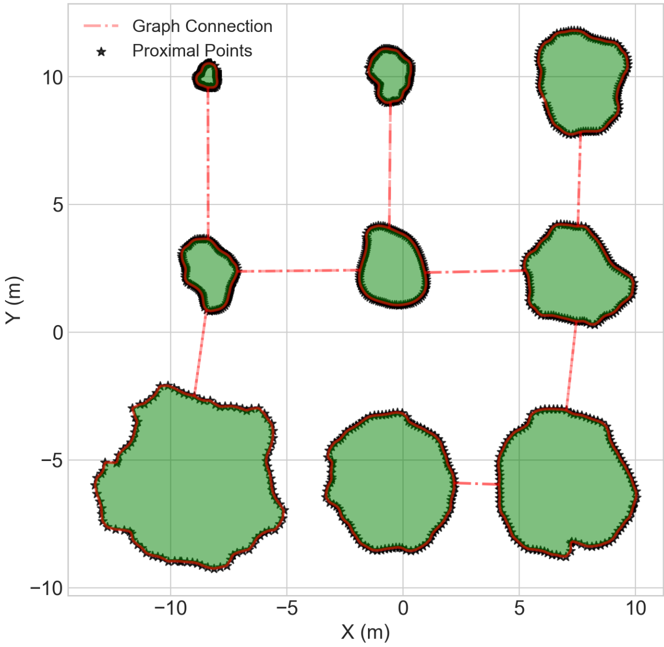


FIGURE 2.3: Fully connected Graph given multiple clusters. The edges connecting alternate clusters are the switching edges.

Appendix A contains a detailed example using the Graph traversal algorithm.

2.5 Experiments

2.5.1 Setup

We compare the performance of the Multiple Cluster Coverage algorithm in a set of simulated environments that mimic an agricultural farm. Specifically, we simulate environments where the biomass clusters are approximated as circles with a radius of 2m. The centers of the clusters are placed in a grid configuration with $M \times N$ rows and columns. We evaluate the performance of the proposed algorithm using length of path generated and the time taken for computation as metrics. We compare our algorithm with traditional TSP solver from Google OR-tools. We consider every vertex of our graph as a node and connect edges that are collision free with the biomass to construct the TSP problem.

2.5.2 Results

The results of comparisons can be found in Table 2.1. The proposed algorithm is capable of producing near-optimal solutions compared to the TSP solver while reducing the computation time by 3.7× on an average. The simulations were performed on MacBook Air with 8GB RAM and an M1 Processor running MacOS 12.4

TABLE 2.1: Comparison of the length of the path L and computation time t

Environment	L_{ours}	L_{TSP}
Grid-(2, 1)	$40.78 \pm 0.11m$	$39.14 \pm 0.01m$
Grid-(4, 1)	$83.10 \pm 1.12m$	$76.81 \pm 0.02m$
Grid-(5, 3)	$357.57 \pm 9.25m$	$290.73 \pm 0.42m$
Grid-(8, 8)	$1321.01 \pm 6.14m$	$1186 \pm 1.83m$
Environment	t_{ours}	t_{TSP}
Grid-(2, 1)	$0.01 \pm 3 \times 10^{-5}s$	$0.03 \pm 12 \times 10^{-5}s$
Grid-(4, 1)	$0.04 \pm 17 \times 10^{-5}s$	$0.13 \pm 1.6^{-3}s$
Grid-(5, 3)	$0.57 \pm 50 \times 10^{-5}s$	$3.25 \pm 0.06s$
Grid-(8, 8)	$26.70 \pm 3.40s$	$98.64 \pm 4.58s$

2.6 Limitations

The algorithm produces near-optimal results when the environment is similar to a Grid arrangement with $M \times N$ rows and columns. Otherwise, there is no guarantee that the above algorithm shall produce comparable results as reported above. This is due to the fact that the constraints and relaxations hold well in a Grid setting and might not generalise. In other cases, the problem remains NP-Hard and cannot be solved trivially. However in Appendix B we provide an analytical condition for near-optimality of the algorithm.

Chapter 3

Multi Agent Multiple Cluster Coverage

3.1 Introduction

In the previous section, we provide an effective solution to the problem of Multiple cluster Coverage. However, in real life, the Agricultural land to be covered is huge and a single agent might not be able to cover due to energy constraints. In such cases, multiple agents are deployed to share the work and to cover the agriculture field effectively. In literature, this is a well studied problem called the Vehicle Routing Problem (VRP) where the task is to distribute the total demand of the job among multiple agents with their respective demands.

3.2 Problem definition

Given K clusters arranged in $M \times N$ grid arrangement. Each cluster has a demand D^k. Given a set of P heterogeneous agents with respective capacities as C^p, the optimal solution partitions the clusters such that the demand of each cluster is met. However, it is always assumed that the sum of capacities of the agents is always greater than the sum of demands. This problem is analogous to the Capacitated Vehicle Routing Problem (CVRP). CVRP is NP-Hard and solutions to the problem have been studied extensively studied in literature. Below is a formulation of CVRP.

3.2.1 Similarity to CVRP

Given a topological Graph $G(V, E)$ where $V = \{1, 2 \dots n\}$ and $E = \{(i, j) \forall i, j \in V\}$. Let the cost of travelling from a node i to j be denoted by c_{ij}. Let Q^k denote the maximum capacity of the vehicle. A decision variable Y_{ij}^k decides if an agent k travels from node i to j. Where,

$$Y_{ij}^k = \begin{cases} 1 & \text{if } k \text{ travels from } i \text{ to } j \\ 0 & \text{if else} \end{cases} \tag{3.1}$$

Then the Capacitated Vehicle Routing Problem can be posed as an optimisation problem.

$$\begin{aligned} & \min \sum_{i}^{N} \sum_{j}^{N} \sum_{k=1}^{K} c_{ij} Y_{ij}^k \\ \text{S.T } & \sum_{k=1}^{N} \sum_{i=0}^{N} Y_{ij}^k = 1 \; \{j = 1 \ldots n\} \\ & \sum_{k=1}^{N} \sum_{j=0}^{N} Y_{ij}^k = 1 \; \{i = 1 \ldots n\} \\ & \sum_{i=0}^{N} \sum_{j=0}^{N} Y_{ij}^k \leq Q^k \; \forall k \in \{1, \ldots n\} \\ & \sum_{j=1}^{N} Y_{ij}^k = \sum_{i=1}^{N} Y_{ij}^k \leq 1 \end{aligned} \tag{3.2}$$

The constraints here can be explained as

1. Customers can only be serviced once and only by one vehicle.
2. The total capacity along the route cannot the exceed the capacity of any vehicle.
3. All routes must start and end at the depot.

The problem of Multi agent-Multiple Cluster Coverage (MA-MCC) is similar to the CVRP after constructing a topological graph using the centers of all clusters. The edges can be constructed after considering collisions or using Nearest Neighbour based connections.

3.3 Related Work

Vehicle Routing Problem is NP-Hard and is an optimisation problem. As with optimisation problems, there are Exact methods and Heuristic and Meta-Heuristic methods. In general, the CVRP is solved in two steps. First assigning customers under different vehicles and then finding an optimal path [1]. There exists exact solutions to the CVRP. In [2] talk about Branch-and-cut algorithms for solving the CVRP. They show that the success of these recent exact algorithms is effective combination of set partitioning formulations with families of cuts into column generation based algorithms.

Since the CVRP is a Combinatorial Optimisation problem, several heuristic methods have been proposed. Genetic Algorithms have been extensively studied and used to solve the CVRP. Recently, in [3] performs a study over recombination operators for the CVRP and propose a new recombination operator called *Best Route Better Adjustment recombination* (BRBAX). In [5] a Graph theoretic based approach is studied for deploying multi-agent systems over various nodes over a graph. This uses a Voronoi partitioning based method to split the graph.

In recent times, Deep Reinforcement Learning based solutions have been proposed to solve VRP. In [10] a single model is trained to find near-optimal solutions to the VRP sampled from a given distribution. They show that their method out performs classical methods and Google OR-Tools in medium scale instances. They also talk about extensions to other VRP based problems and Combinatorial optimisation problems in general. In [6] Multi Agent Reinforcement Learning is used to solve CVRP. They show that their method performs equally to Google OR-Tools but is more adaptive.

3.4 Method

Given K clusters arranged in $M \times N$ grid arrangement, we construct a topological graph $G_\tau(V_\tau, E_\tau)$ where the vertices of the Graph are the centers of the clusters and the edges are connected after considering collisions. We define the demand of each cluster D^k. Given P heterogeneous agents with capacities C^p, we deploy a CVRP solver to partition the topological graph thus obtaining the set of clusters each agent has to traverse. We then deploy our MCC algorithm for each agent to traverse the clusters.

Algorithm 1 Multi agent Multiple Cluster Coverage

```
V_τ ← {μ^k | ∀k ∈ K}
E_τ ← {(μ^i, μ^j) | ∀i, j ∈ V_τ}
G_τ ← G(V_τ, E_τ)                          ▷ Construct the Topological Graph
D ← {D^k | ∀k ∈ K}                         ▷ Demands of clusters
C ← {C^p | ∀pinP}                          ▷ Capacities of agents
Ξ ← CVRP(G_τ, D, C)                        ▷ Graph partitioning
for ξ ∈ Ξ do
    MCC(ξ)                                 ▷ Multiple Cluster Coverage
end for
```

3.4.1 Demand of a cluster

Demand of a cluster is important to define. Given the radius R and crown height H, two types of demands can be constructed

1. **Volumetric Demand**: $\pi R^2 H$
2. **Time Demand**: $2\pi RH$

3.5 Experiments

3.5.1 Setup

We compare the performance of Combinatorial Optimisation based CVRP solver from Google OR Tools, Voronoi Graph Partitioning and Genetic Algorithm in a simulated environments that mimic an agricultural field. We simulate a 5×5 Grid arrangements of the centers to compare the algorithms on Time taken to report a solution and whether the solution satisfies the demand constraints.

We use the Genetic Algorithm formulation used in [3] along with the new BRBAX crossover operator. The Voronoi Graph Partitioning used is shown in Algorithm

Algorithm 2 Voronoi Partitioning of Graph

$G = (V, E)$
$P = \{(p_i, L_i)\ \forall i \in R\}$ ▷ All robot positions and capacities
$g_i \leftarrow \emptyset\ \forall r_i \in R$
for $v \in V$ **do**
    if $g_i = \emptyset$ **then**
        $c_i, r_i = \text{Dijkstra}(p_i, v)$
    else
        $c_i, r_i = \text{Dijkstra}(g_i[-1], v)$ ▷ Compute path from latest location
    end if
    while $\{c_i, r_i\} \neq \emptyset$ **do**
        $c_i^* = \arg\min(\{c_i, r_i\})$
        if $c_i^* < L_i$ **then**
            $g_i \cup \{v\}$
            $L_i \leftarrow L_i - c_i$
            break
        else
            $\text{Pop}(c_i, r_i)$ ▷ Remove the cost and robot tuple
        end if
    end while
end for

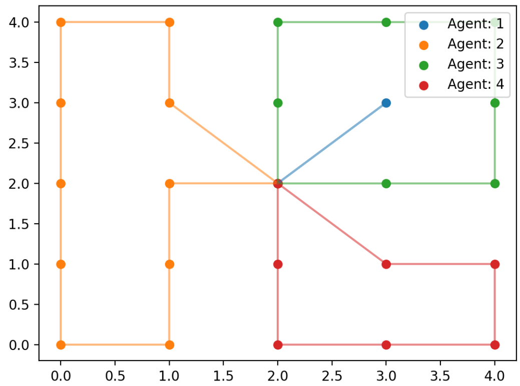


FIGURE 3.1: Solution to a 5x5 Grid arrangement of clusters split amongst 4 agents.

3.5.2 Results

TABLE 3.1: Comparision of Time taken to solve the CVRP problem

Algorithm	$T(s)$	Success rate (%)
CO - ORTools	$0.234 \pm 0.011s$	99.98 ± 0.02
CO - Voronoi	$0.531 \pm 0.12s$	76.81 ± 5.12
GA	$4.573 \pm 1.25s$	40.73 ± 9.87

The results of the comparison can be found in the Table 3.1. We find that the CVRP solver from Google OR Tools out performs other methods and is the most reliable. This is corroborated across literature too. It is to note that the performance of the Genetic Algorithm heavily depends on the hyper parameters and the choice of Crossover and Mutation functions.

Chapter 4

Conclusion

The problem of traversing Proximal points around a biomass in an Agricultural farm is analogous to the TSP. We exploit the Grid arrangement commonly found in most Agricultural farms to impose the following constraints and relaxations.

1. Any vertex in the Graph can be traversed more than once.
2. Any vertex can only be connected to two neighbours of the same cluster and one vertex of other cluster.
3. Two clusters can be connected only by a predefined switching edge.
4. The agent is constrained to always switch at the switching edge

Using the following constraints, we convert the problem into a Graph traversal problem where we construct a Graph using the Proximal points as vertices. We connect alternate vertices using switching edges. We use a lexicographical key to select the next best vertex and traverse the graph completely.

We compare our algorithm to standard TSP solvers and show that our algorithm produces near-optimal solutions and uses less time to compute them.

We extend our Multiple Cluster Algorithm to using Multiple heterogeneous agents using existing VRP solutions. We compare various available methods and find that the Optimisation based method produces better results.

Appendix A

Graph Traversal Example

According to 2.18, for a given edge (i, j), the priority key is a combination of three parameters, *traversal cost*, *switching cost* and *distance cost*. *Traversal cost* is used to discriminate between vertices that are already traversed. *Switching cost* is used to prioritise switching vertices over other vertices. *Distance cost* is used to select the vertex that is the closest.

In our implementation, every time a vertex is traversed, we add a *traversal cost* of 100 to itself. If an edge is switching, we subtract 100 from *switching cost*. A Python implementation of calculating keys is shown below

```python
class Vertex:
    def __init__(self, state):
        self.state = state

        # Costs
        self.traversal_cost = 0
        self.switching_cost = 0

        self.isSwitching = False # or True
```

LISTING A.1: Vertex Class definition

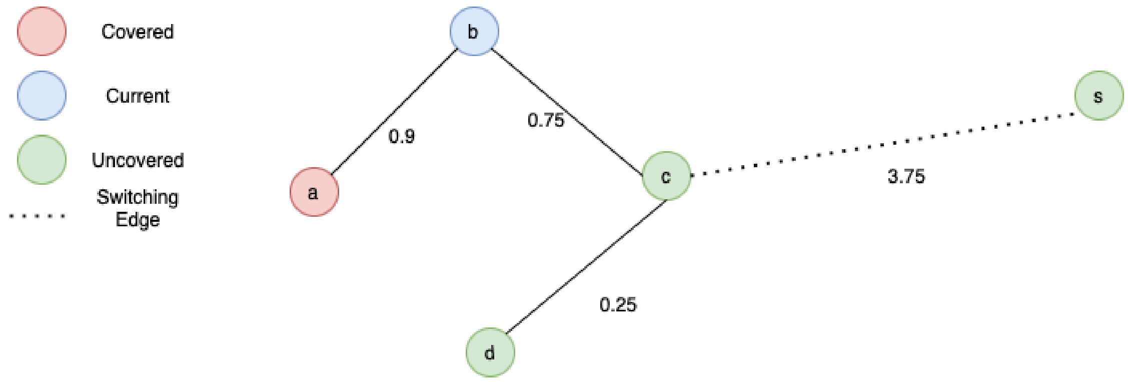
```
class Key:
    def __init__(self, k1, k2, k3):
        self.k1 = k1
        self.k2 = k2
        self.k3 = k3

    def __lt__(self, __ot):
        if self.k1 < __ot.k1:
            return True
        else:
            if self.k2 < __ot.k2:
                return True
            else:
                if self.k3 < __ot.k3:
                    return True
                else:
                    return False
```

Listing A.2: Key Definition and calculation

```
def get_next_vertex(vertex):
    keys = []
    for neighbour in vertex.neighbours:

        if neighbour.isSwitching:
            switchingCost = -100
        else:
            switchingCost = 0

        traversalCost = neighbour.traversal_cost

        distanceCost = distance(vertex, neighbour)

        keys.append([neighbour, Key(traversalCost, switchingCost, distanceCost])

    return sorted(keys, lambda x: x[1])[0]
```

Listing A.3: Next Vertex Selection

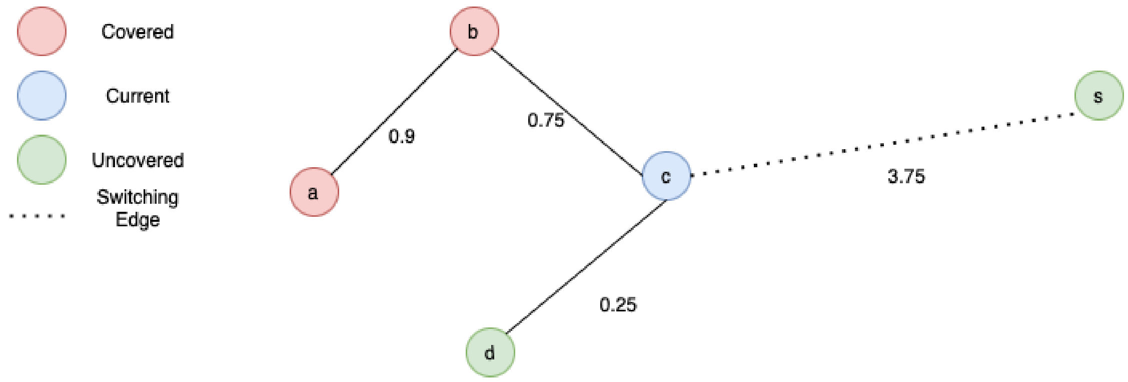
Let us assume a scenario where we have 4 vertices of a cluster, namely a, b, c, d and another vertex s from a different cluster. The vertex c is connected to the vertex s via a switching edge. Now given that we are currently at the vertex b, the next vertex to traverse can be chosen by constructing the corresponding Priority keys for each choice and using lexicographical ordering to select the next vertex.



The keys at vertex *b*

1. **Edge (b, a)** : [100, 0, 0.9]
2. **Edge (b, c)** : [0, 0, 0.75]

The **Edge (b, c)** is selected over the other edges.



Now, for the vertex c, the keys can be written as

1. **Edge (c, b)** : [100, 0, 0.75]
2. **Edge (c, d)** : [0, 0, 0.25]
3. **Edge (c, s)** : [0, -100, 3.75]

Here, since the **Edge (c, s)** is a switching edge, it is prioritised.

Appendix B

Analytical bound for Near Optimality

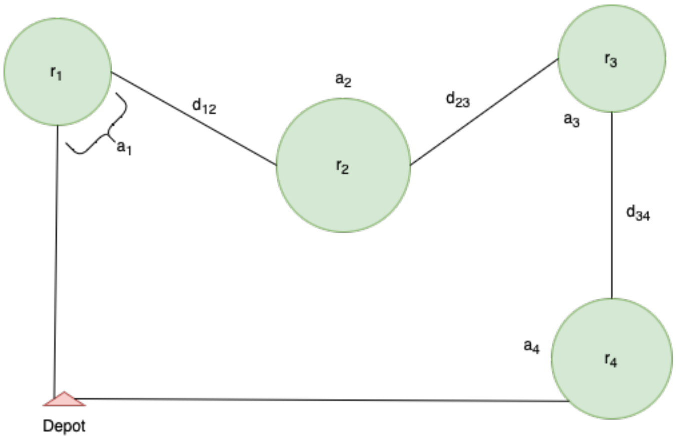


FIGURE B.1: Sample scenario consisting of four clusters approximated as circles.

Without loss of generality, let us assume that the clusters are circular in shape with radius r_i. Each cluster is connected to the other. Let us denote the shortest arc between the connections from one cluster to another as a_i where i denotes the cluster. The distance between one cluster to another can be denoted as $d_{i,j}$. Let us assume that the agent starts from Depot and reaches the first cluster. The distance between Depot to a cluster i can be denoted by D_i. We also assume that $D_1 < D_4$.

Now, according to Eq. 2.12, the cost of the path given the configuration above can be denoted by

$$C_{\text{our}} = \sum_i C^i + 2\sum_{i,j} C^{ij} \implies (2\pi r_1 + 2\pi r_2 + 2\pi r_3 + 2\pi r_4) + 2 \cdot (d_{12} + d_{23} + d_{34}) + 2 \cdot D_1 \quad \text{(B.1)}$$

Given the above configuration, the only other valid path is to cover a cluster completely and traverse the shortest arc again to switch to a new cluster. The cost of the path can be given as

$$\begin{aligned} C_{\text{other}} = (2\pi r_1 + 2\pi r_2 + 2\pi r_3 + 2\pi r_4) + \\ (d_{12} + d_{23} + d_{34}) + \\ (a_1 + a_2 + a_3 + a_4) + D_1 + D_4 \end{aligned} \tag{B.2}$$

Our algorithm produces near optimal solutions when $C_{\text{our}} \leq C_{\text{other}}$.

$$\begin{aligned} \sum_i 2\pi r_i + 2\sum_i d_{i,i+1} + 2 \cdot D_1 &\leq \sum_i 2\pi r_i + \sum_i d_{i,i+1} + \sum_i a_i + D_1 + D_4 \\ \implies \sum_i d_{i,i+1} &\leq \sum_i a_i + (D_4 - D_1) \\ \implies \sum_i d_{i,i+1} &\leq \sum_i a_i \end{aligned} \tag{B.3}$$

Hence, if the summation of distance between the clusters are less than the summation of extra distance to cover, our MCC algorithm provides near-optimal solutions. However, this result is not complete as the complexity of choices increases as the number and arrangement of clusters varies.

Bibliography

[1] M. A. H. Akhand et al. "Solving Capacitated Vehicle Routing Problem with route optimization using Swarm Intelligence". In: *2015 2nd International Conference on Electrical Information and Communication Technologies (EICT)*. 2015, pp. 112–117. DOI: 10.1109/EICT.2015.7391932.

[2] Roberto Baldacci, Aristide Mingozzi, and Roberto Roberti. "Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints". In: *European Journal of Operational Research* 218.1 (2012), pp. 1–6. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2011.07.037. URL: https://www.sciencedirect.com/science/article/pii/S0377221711006692.

[3] Carlos Bermúdez et al. "A Comparison of Recombination Operators for Capacitate Vehicle Routing Problem". In: *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial* 14 (Mar. 2010), pp. 34–44. DOI: 10.4114/ia.v14i46.1520.

[4] A. Claus. "A New Formulation for the Travelling Salesman Problem". In: *SIAM Journal on Algebraic Discrete Methods* 5.1 (1984), pp. 21–25. DOI: 10.1137/0605004. eprint: https://doi.org/10.1137/0605004. URL: https://doi.org/10.1137/0605004.

[5] Mohammadreza Davoodi, Saba Faryadi, and Javad Mohammadpour Velni. "A Graph Theoretic-Based Approach for Deploying Heterogeneous Multi-agent Systems with Application in Precision Agriculture". In: *Journal of Intelligent and Robotic Systems* 101.1 (2020). DOI: 10.1007/s10846-020-01263-4.

[6] Yamen Habib and Andrey Filchenkov. "Multi-Agent Reinforcement Learning For Multi Vehicles One-commodity Vehicle Routing Problem". In: *Procedia Computer Science* 212 (2022). 11th International Young Scientist Conference on Computational Science, pp. 418–428. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2022.11.026. URL: https://www.sciencedirect.com/science/article/pii/S1877050922017173.

[7] Abid Hussain et al. "Genetic Algorithm for Traveling Salesman Problem with Modified Cycle Crossover Operator". en. In: *Computational Intelligence and Neuroscience* 2017 (Oct. 2017), e7430125. ISSN: 1687-5265. DOI: 10.1155/2017/7430125. URL: https://www.hindawi.com/journals/cin/2017/7430125/ (visited on 12/21/2022).

[8] Qi Kuang et al. "Real-Time UAV Path Planning for Autonomous Urban Scene Reconstruction". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 1156–1162. DOI: 10.1109/ICRA40945.2020.9196558.

[9] Huichuan Liao. "Research on agricultural machinery operation path optimization based on traveling salesman problem (TSP)". In: *Journal of Physics: Conference Series* 1976 (July 2021), p. 012051. DOI: 10.1088/1742-6596/1976/1/012051.

[10] Mohammadreza Nazari et al. "Reinforcement learning for solving the vehicle routing problem". In: *Advances in neural information processing systems* 31 (2018).

[11] Catur Aries Rokhmana. "The Potential of UAV-based Remote Sensing for Supporting Precision Agriculture in Indonesia". In: *Procedia Environmental Sciences* 24 (2015). The 1st International Symposium on LAPAN-IPB Satellite (LISAT) for Food Security and Environmental Monitoring, pp. 245–253. ISSN: 1878-0296. DOI: https://doi.org/10.1016/j.proenv.2015.03.032. URL: https://www.sciencedirect.com/science/article/pii/S1878029615001000.

[12] Shabnam Sangwan. "Literature Review on Travelling Salesman Problem". In: *International Journal of Research* 5 (June 2018), p. 1152.

[13] S. I. Sergeev. *Approximate algorithms for the traveling salesman problem. II.* en. 2015. DOI: 10.1134/s0005117915030108. URL: http://dx.doi.org/10.1134/S0005117915030108.

[14] Xiaocui Zheng, Fei Wang, and Zhanghua Li. "A multi-UAV cooperative route planning methodology for 3D fine-resolution building model reconstruction". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 146 (2018), pp. 483–494. ISSN: 0924-2716. DOI: https://doi.org/10.1016/j.isprsjprs.2018.11.004. URL: https://www.sciencedirect.com/science/article/pii/S0924271618303009.