Informative path planning for a collaborative robot system in precision agriculture applications

Vishnu Sashank Dorbala, Vishnuu AD, Rodrigo Perez-Vincente

***Abstract*—We perform a detailed analysis of a paper that does informative path planning for precision agriculture systems. The authors propose a symbiotic UAV + UGV setup to identify and take nitrogen soil samples from a field. In order to achieve this, the authors describe a procedure that involves finding Potentially Mislabelled Points (PMLs) from a set of soil samples, and use this to plan paths for navigation. In particular, two path planning approaches are described, both of which are variants of the Travelling Salesman Problem (TSPN). We present results in the form of plots showing the optimized path.**

I. Introduction

The purpose of this project is to implement an informative path planning algorithm in a robot system used for precision agriculture. The path planning problem of this robot system will be solved by a symbiotic collaboration between a UAV and a UGV that will oversee the data collection of soil nitrogen level measurements across a farm.

The reason for using this collaborative system is to combine the sensing capability strengths of aerial and ground robots. The ground robot (UGV) can travel long distances, carrying heavy loads and measuring soil data. However, the aerial robot (UAV) has the ability to take aerial images from a low altitude despite not being able to take solid measurements and having a limited battery life.

The informative path planning of the system will consist of two different planning problems. The first path planning problem will be to reduce the time of the UGV to obtain soil measurements by combining close measurement locations with overlapping neighbors. This will be done by applying a novel modification of the Euclidian traveling salesperson (TSP) which is known as sampling traveling salesperson problem with neighborhoods (SAMPLINGTSPN).

The other path planning algorithm will be centered on the UAV+UGV. The objective is to make the UAV visit the maximum number of measurement locations of interest concerning the constraint of the maximum battery life of the aerial robot. The UAV will have the possibility to land on the UGV and use the UGV to travel to the next take-off location. This type of path planning problem of visiting the maximum number of locations with a budged constraint is known as Orienteering.

The planning problems will also have in common the input, prior key locations to obtain the measurements from. These locations will have enough descriptive information to understand the whole environment. Also, these key locations are the goals that the robot system will have to reach and are referred to as Potentially Mislabeled (PML) points.

A. Definitions

In the following paragraphs it will be briefly explained key definitions to understand better the content of the project:

- **Precision agriculture**: precision agriculture is a technique to forecast the conditions of harvests in a farm by gathering sensor data. This information is used to improve productivity and increase environmental quality. As for this project the final objective is to measure the levels of nitrogen in the soil to be able to reduce fertilizer usage by applying the right level of nitrogen at the right location and time.
- **Informative path planning**: informative path planning is a branch of path planning that involves the development of paths for robots that act as mobile sensors to gather data from a specific environment. As it will be studied during the development of this project many locations of the environment are correlated which simplifies and makes more effective the use of informative path planning. For this project the symbiotic system of UAV+UGV will act as the mobile sensor, were the UGV will gather the data from the soil measurements while the UAV will take information from aerial images.
- **PML - Potentially Mislabeled Points**: Given a map of Nitrogen levels (N-Levels) constructed using prior information, some points have a high probability of uncertainty. These points are called Potentially Mislabeled Points or PML.
- **TSP - Travelling Salesman Problem**: This is a classic NP-Hard optimization problem. Given a set of points having finite distances between them, we need to find the least cost path to traverse through all the points and return to the point of origin.
- **Orienteering**: Given a set of vertices on a graph, the objective of the *orienteering* problem is to find a tour of a subset of vertices collecting the maximum reward, such that the sum of weights of edges on the tour is less than a given budget.
- **UAV and UGV** - These refer to Unmanned Aerial Vehicle (UAV) and Unmanned Ground Vehicle (UGV).

B. Background

UAV's and UGV are increasingly being used in Agricultural Robotics. One of the most sought after tasks to solve in this area is one where sensor capabilities are improved for autonomous navigation within farms. Nitrogen, being a key element for plant's chlorophyll pigment, helps increase yield in plants. Therefore, it is crucial to have this data to carry out essential steps for harvesting crops. It is also important for this data to be reliable. This paper focuses on collecting the Nitrogen data from a given map of points where we're uncertain of the nitrogen content(Potentially Mislabeled Points). Identifying these PML's, and constructing an optimal UAV+UGV path through them form the crux of the paper.

C. Literature Review

Low *et al.* [1] have developed a method to reduce the probability of mis-classification with Gaussian Process. Zhang et. al [2] present an adaptive sensing algorithm to estimate a scalar field. The algorithm accepts the measurements made by static nodes as inputs and computes a path for the mobile robot which minimizes the integrated mean square error of the reconstructed field subject to the constraint that the robot has limited energy. In the above work, it is assumed that measurements take no time and measurements are done at every time step. However, our paper of interest considers the measurement time as well and thereby minimizes measurement time and travel distance together. Singh *et al.* [3] presented an approximation algorithm to find a path for a mobile robot for using information from visited and un-visited nodes subject to budget constraints in the robot.

Solving the problem of maximizing exploration among the vertex set V with a budget constraint on the path length is called the orienteering problem. Blum *et al.* [4] have shown a 4 approximation algorithm for this problem on a metric graph. Our paper of interest shows how to model the problem of visiting most points with a symbiotic UAV and UGV as an orienteering problem. The classical problem of finding shortest length path to a set of cities is known as the Travelling Sales Man problem. This paper deals with a slight modification of the original TSP where only the Neighbourhood is considered (TSPN) where the optimal path should visit a point in the neighbourhood of interest sites.

Dumitrescu and Mitchell [5] have presented an approximation algorithm for cases where neighboring disks overlap with each other. However, in our paper of interest, the authors consider the cost to be the total time taken for travelling and obtaining soil measurements. Bhadauria *et al.* [6] studied the problem of finding minimum time tour where k robots robots visit sensors and query each sensor's data. However, in this case the authors claim that it is possible to visit a single point if there are multiple points in a neighborhood. Alt *et al.* [7] studied the problem of given set of points with fixed radio antenna and circular ranges with centers and radius r_i. The cost function is a weighted sum of length of tour and the sum of r_i^{α} for each disk. The paper of our interest's authors do not have a fixed constraint number of interest points and also penalise higher number of points.
There has been a surge in interest of work in the area of cooperative robots to solve tasks such as in [8] [9] [10] and this paper's authors solve a novel problem where the UAV lands on the UGV at times.

II. Potentially Mislabelled points

The quest to solve the problem of labelling points efficiently starts with a prior information of nitrogen soil map. For each point in $X = \{x_1, x_2, x_3, ..., x_n\}$ we have a most its likely estimate as $N(x_i)$, with variance of the estimate given by $\sigma^2(x_i)$. In general, there are a set of labels and each Label L_i is defined by a range $\{L_i^- \ and \ L_i^+\}$.

Since our labels and estimates are deterministic, we find the probabilities of a single point belonging to several labels. For each point, we choose the label with highest probability and call that label as the most likely label for that point. We finally assign the most likely label as the label to a point if the probability is greater than a user chosen threshold $P_d \in (0, 1)$.

In a nutshell, a point is labelled as *potentially mislabeled* if either of either of its probabilities of belonging to a Label is less than a user set threshold P_d.

A. Our implementation of the algorithm in absence of soil data

The paper uses soil data from a source which is currently inaccessible to us. Thereby we go ahead to find the PML points through random initialization. Our pipeline for PML points is mentioned below.

Algorithm 1 Find PML points parameters: $N(x_i),\ \sigma(x_i),\ r_i$

Initialize "N" random points(Nx2) in a grid of size 500x500

Initialize "N" random radius values in a fixed range(Nx1)
Create PML point as (X, Y, R) (Nx3)

A sample output is the algorithm is shown below.

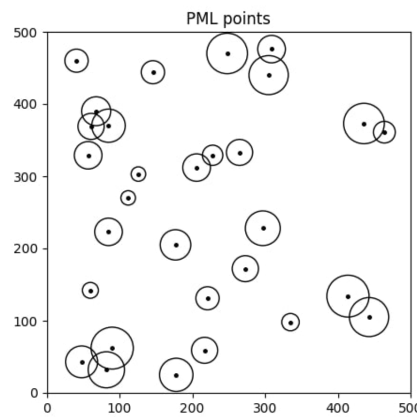

Figure 1: PML points

III. Planning Algorithms

The paper describes two path planning algorithms, specific to UAV and UGV navigation. Both these algorithms are based on variants of the Travelling Salesperson Problem (***TSPN***) described below.

A. *TSPN*

Consider a situation where you are a salesman, and want to distribute your merchandise across a number of cities (say, n). The cost of travelling to every other city from a particular city is known. Given an initial city as a starting point, the Travelling Salesman Problem (TSPN) tries to find the least cost path that traverses through all n cities and return to the starting point.

A brute force method to solve this problem gives us a time complexity of $\mathcal{O}(n!)$. As the number of cities increase, the number of paths to check increases greatly. Thus, solving this in a time and cost effective manner is a non-trivial task. However, there exist several heuristic approaches to solve this problem.

In this paper, the PML points are considered as cities, and the authors try to solve a TSPN on both a UGV and a UAV. For the UGV, a grid sampling based method is used. Due to the limited battery life of the UAV, a Prize Collecting version of the TSPN (PC-TSPN) is used. This is discussed further in detail below.

B. UGV Planning: SamplingTSPN

The SamplingTSPN problem is similar to the TSPN problem except for two differences:

- The total cost also takes into consideration the sampling time at each node. This sampling time is the amount of time taken to obtain a measurement of the soil at a particular PML.
- The nodes (cities) behave as disks having different radii. Each of these disks describes an area around the PML in which measurement samples can be taken.

Each disk around a PML point i can be defined on the map as having center x_i and radius r_i. The SamplingTSPN algorithm then takes a set of these disks $X = (x_1, r_1), \ldots, (x_n, r_n)$ as input. The maximum and minimum radii are defined as r_max and r_min respectively. The objective of the SamplingTSPN problem can then be modelled as follows: *Given X, find a tour τ of N distinct PML sample locations to maximize the cost $len(\tau) + C_g.N$ such that each disk contains a sample location.*
where C_g is the cost to obtain a sample.

As mentioned earlier, solving the TSPN problem is a non trivial task, as it is an NP-Hard problem. There exist several methods to solve the problem however, that make use of certain heuristics to obtain the correct solution. The authors of the paper use a $(1+\epsilon)$ approximation method described by Sanjeev Arora in [11] to solve the problem. This reduces the complexity of the TSPN from the brute force $\mathcal{O}(n!)$ to $\mathcal{O}\left(n(\log n)^{\mathcal{O}\left(\epsilon^{-1}\right)}\right)$. In our case, this $\mathcal{O}(\frac{r^2_{max}}{r^2_{min}})$ at most. This is further reduced to $\mathcal{O}(\frac{r_{max}}{r_{min}})$ by computing hitting sets, wherein the number of points selected are reduced. This is further explained below.

Algorithm 2 Sampling TSPN: $\mathcal{O}(\frac{r^2_{max}}{r^2_{min}})$

1) Stage 0: $\bar{X}$ is the set of all disks X with radius r_{max}.
2) Stage 1: Compute I = MIS($\bar{X}$).
3) Stage 2: For each disk $d \in I$, place a grid co centered with d and with dimensions $6r_{max} \times 6r_{max}$ with a resolution of $\frac{r_{min}}{\sqrt{2}}$.
4) Stage 3: Output a TSPN tour of all grid points withing $2r_{min}$ of a sampling location in Stage 2.

The authors of the paper also present a novel GRIDSAMPLE algorithm whose compute complexity is $\mathcal{O}(\frac{r_{max}}{r_{min}})$

Algorithm 3 GRIDSAMPLE

Stage 1: Create a set *(P, R)* from arrangement of all disks in X. Compute the hitting set solution C = HITTINGSET*(P, R)*
Stage 2: Output a TSP tour of C

A hitting set problem is one where for a set *(P, R)*, P is set of points and R is a collection of subset of P. The solution to the hitting set point is the smallest subset of P such that C intersects with all R_i where $R_i \in R$. For example, given a set $\mathcal{U} = 1, 2, 3, 4, 5$, and the collection of sets $\mathcal{S} = 1, 2, 3, 4, 1, 3, 5, 2, 4$. Here, the union of $\mathcal{S}$ is $\mathcal{U}$. The hitting set in this case however is $1, 3, 5, 2, 4$, since all the elements in $\mathcal{U}$ can be computed using this set. This is clearly illustrated in table 1. Figure 1 of Table 1 shows a set of intersecting circles. Figure 2 shows disjoint sets of patches on the grid. Figure 3 shows a random point assigned in each of these patches. Figure 4 shows the hitting set solution which happens to lie in the brown region (Intersecting region common to all the points)

C. UAV Planning: Orienteering (PC - TSPN)

As previously mentioned, the planning for the UAV will also be computed by a variation of the Travelling Salesman Problem (TSPN). The limitation of the UAV is its battery life. For this reason, the path planning algorithm of the UAV will have to select from all the PML points the most rewarding to visit,taking into consideration a travelling budget. This type of problem is called Orienteering which belongs to the family of the Prize Collecting TSPN problems (PC-TSPN).
The PC-TSPN problems can be compared to having a set of cities with positive reward values and a set of pairwise distances, a salesman must visit a subset of the cities in order to minimize the total distance traveled while also maximizing the sum of rewards gathered.As it can be noticed in the PC-TSPN problem there is a trade of between the cost distance of a tour and the total reward maximization. On the other hand, for the UAV path planning a variation of PC-TSPN will be used, called Orienteering. The Orienteering problem is a budget-driven version of the PC-TSPN, in which the distance traveled has a limit threshold and is used as a budget to maximize the reward collected. For these reason the Orienteering problem can also be called maximum-prize path.

The Orienteering algorithm to be applied appears in [4]. This algorithm computes a four-approximation and for its input it is required a undirected weighted graph G. The undirected graph is represented as $G(V, E, \pi, w)$ and it is obtained as follows:

1) Over the plane compute a grid of resolution $C/\sqrt{2}$. C is the cameras diameter footprint that the UAV carries. The grid locations that have one or more points from $\mathcal{X}_{pml}$(space that includes all the PML points obtained in Section 2) will be represented as grid vertices, $v \epsilon V$.
2) The amount of PML points of each grid vertex $v \epsilon V$ is stored in $\pi(v)$.
3) E will represent all the edges between $(u, v) \epsilon V$.
4) For each edge calculate the weight $w(u, v) = min \{d(u, v), 2C_a\}$. This means that there are two types of edges with different costs: $d(u, v)$ is the Euclidean distance between two points and represents the cost of the UAV to travel from u to v; while $2C_a$ is the cost of the UAV for the action of takeoff and landing.

Because the edge weights are Euclidean distances, this graph is a considered an undirected metric graph and can be implemented in the algorithm [4]. In the original paper this Orienteering tour is define by using the implementation of the SFOToolbox in MATLAB[**?**]. However, it was decided to implement an original Orienteerig algorithm in Python. This algorithm works imitating a Brute Force technique as follows
The elements in Table II shows the expected results from the Orienteering path planning:

IV. Symbiotic UGV + UAV Planning

Having solutions to the two path planning problems, we use the combination of UAV and UGV for a better path. We first plan the Path for UAV, where the algorithm chooses the grid points which give the maximum reward to the UAV. Given the grid points of interest of the UAV, we take all pml points belonging to these grid points. The UGV then plans its path using the GRIDSAMPLE solution mentioned above. The exact algorithm is mentioned here.

V. Results

We conduct experiments with a fixed number of generated PML points. During each trial, we vary the battery life cost and the radius of visibility of the camera of the drone. The results of this are shown in table II. The four trials conducted are on a fixed set of PML points (30). We vary the battery life of the UAV and the sparsity of the points to optimize for the path. These parameters are shown in table IV.

As Table II shows the paths determined by the Orienteering algorithm are very dependable of the initial parameters. In the case of the first row the 30 PML were originated concentrated close to each other and a high battery life for the UAV was established. This allowed for the UAV to make just one long flying path gathering as

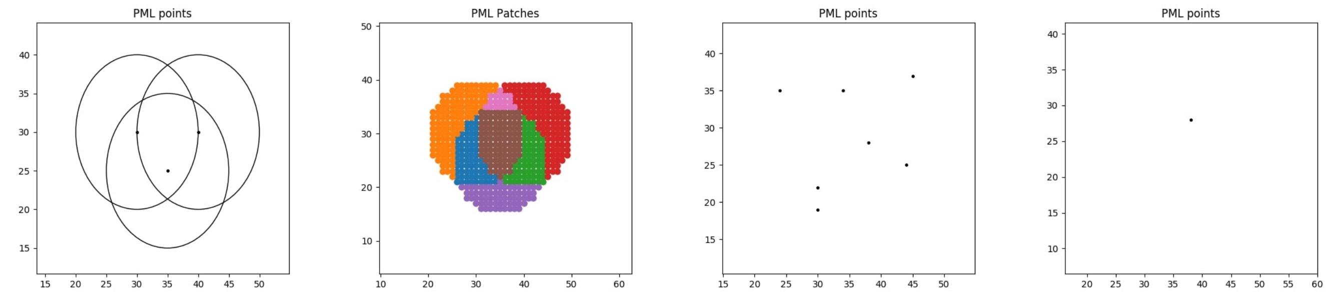

Table I: HITTINGSET: Solution (1) Set of 3 PML circles, (2) Segmeneted Patches, (3) 1 point per Segmented Patch (4) Hitting set solution

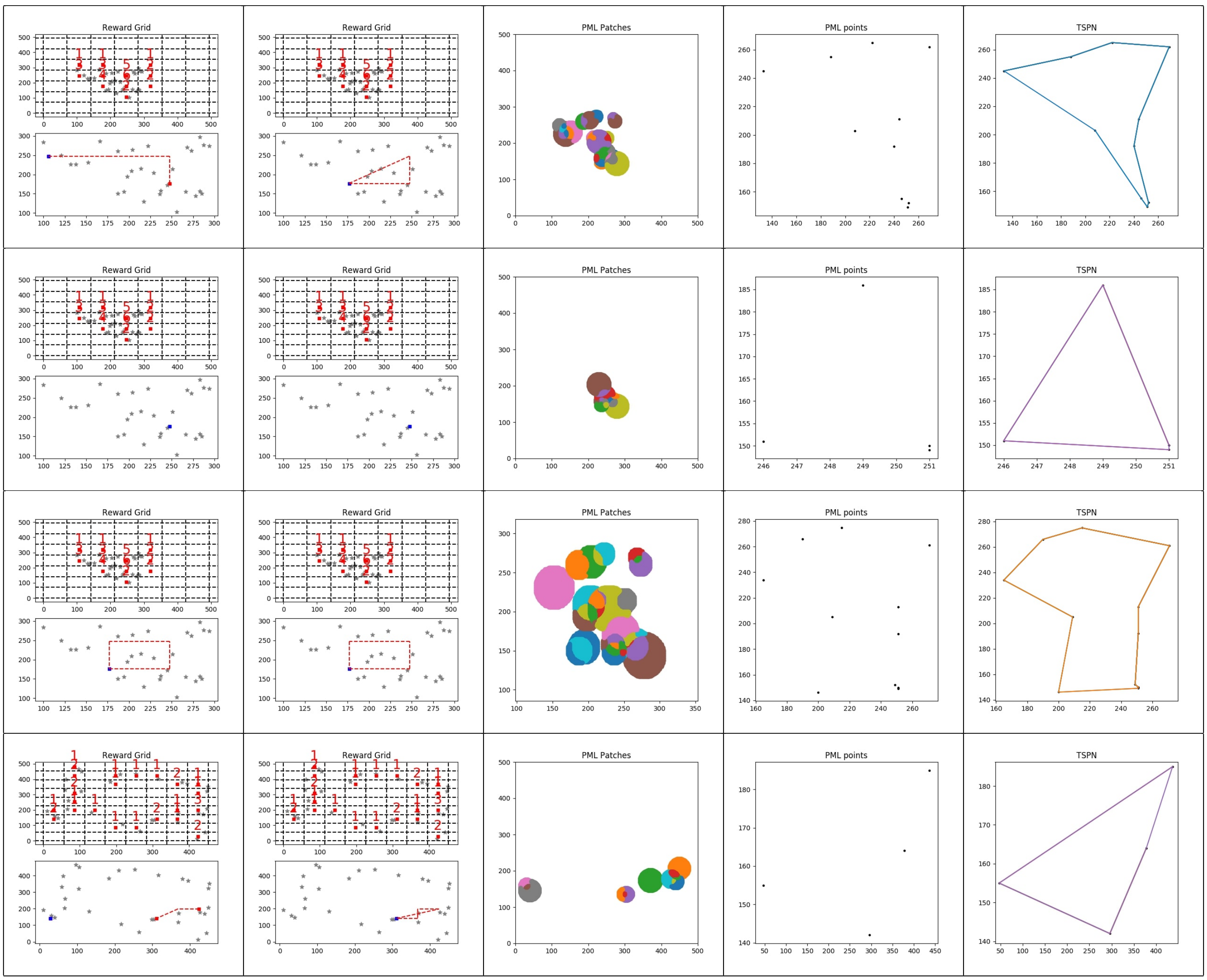

Table II: These are our results. The first column represents the path of the UAV, when a **UAV + UGV** system is used. The second column represents the path, when only the UAV is used. These two columns are used to perform planning on a UAV. The third column shows the regions of intersecting PML circles, and how we segment them out. The fourth column shows the PML points obtained after finding the *Hitting Set* of regions in column 3. The final column is the **SamplingTSPN** tour obtained for the UGV. The outputs of the final column and the 2nd column are combined to give us a tour of UGV+UAV.

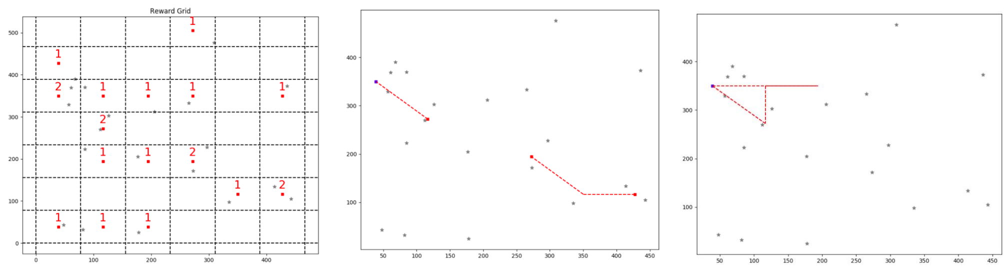

Table III: From left to right (a) Grid map with PML rewards (b) UAV paths using UGV, the square points represent takeoff and landing points, while the dashed line represents the drone flying. The UAV uses the UGV to travel from lading to takeoff points, (c) Closed flying tour for the UAV only.

S_no	Num PML points	Radius of Visibility	Battery Life	Takeoff/Landing cost
1	30	100	400	150
2	30	100	180	150
3	30	70	600	300
4	30	80	400	150

Table IV: Table of parameters for the experiments shown in Table II

Algorithm 4 Orienteering for symbiotic UAV and UGV

Initialize metric graph "G"
Initialize iteration path queue
Initialize complete paths queue
for Each point v_i in V **do**
  Initialize new path with starting node v_i.
  while iteration path queue not empty **do**
    $path = pop.(iterationpathqueue)$
    for Each point v_i in V **do**
      if $cost < budget$ **then**
        $cost+ = cost$
        $reward+ = reward$
        $iterationpathqueue \leftarrow path.append(v_i)$
      end if
      if $startingnode = v_i$ **then**
        $cost+ = cost$
        $completepathqueue \leftarrow path.append(v_i)$
      end if
    end for
  end while
  for path in complete path queue **do**
    if $path.reward > finalpath.reward$ **then**
      $finalpath = path$
    end if
  end for
end for

Algorithm 5 Symbiotic Path Planning

Input: A map of soil N level Labels and desired Label uncertainity.
Ouput: A symbiotic UAV+UGV tour of Measurement Locations
Identify the set of PML points $\mathcal{X}_pml$, from a prior and calculate the required disk with radii
Find the larget subset of PML points, $\mathcal{X}_s \subseteq \mathcal{X}_{pml}$, that can be visited by the UAV using the symbiotic system of UAV+UGV system, subject to battery life.
Compute the UGV tour to obtain ground measurements for each PML point in $\mathcal{X}_s$

much PML's as possible and then using the UGV to come back to the initial node. However, if the UAV battery life was decreased as it can be observed in the second row image, the UAV will only have enough battery to make a single takeoff and landing in the same spot as where it starts. In the following cases of the last two rows of the table, the take off and landing cost was increased. This forced the UAV to consider doing more than one takeoff and landing preferring to fly. As it can be seen in the third row of the table, the UAV completes a closed flying tour instead of using the UGV to travel. In the last case, the 30 PML points were distributed along the plane further apart from each other and the same parameters as in the last case were used. In this case, the UAV had to travel longer distances to get more reward so it decided to use the UGV to travel to the highest reward area of the plane and do a short flying path to then use the UGV again to go back to the initial point (this initial point was also visited by the UAV by doing a takeoff/landing in the same location).

Finally, it was concluded by trying different parameter variations, that in every occasion the UAV and UGV symbiotic system was capable of gathering more PML rewards than just using the UAV to complete a closed flying tour. With the help of the UGV to travel long distances, the UAV was capable of creating a more optimal path and saving battery to reach the most rewarding locations along the plane.

A code repository of our work can be found here. [1]

[1] https://github.com/vdorbala/ENPM661-Final_Project

REFERENCES

[1] K. H. Low, J. Chen, J. M. Dolan, S. Chien, and D. R. Thompson, "Decentralized active robotic exploration and mapping for probabilistic field classification in environmental sensing," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-12)*, June 2012, pp. 105–112.

[2] B. Zhang and G. S. Sukhatme, "Adaptive sampling for estimating a scalar field using a robotic boat and a sensor network," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 3673–3680.

[3] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *J. Artif. Int. Res.*, vol. 34, no. 1, p. 707–755, Apr. 2009.

[4] A. Blum, S. Chawla, D. R. Karger, T. Lane, A. Meyerson, and M. Minkoff, "Approximation algorithms for orienteering and discounted-reward tsp," in *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, 2003, pp. 46–55.

[5] A. Dumitrescu and J. S. Mitchell, "Approximation algorithms for tsp with neighborhoods in the plane," *Journal of Algorithms*, vol. 48, no. 1, p. 135–159, 2003.

[6] D. Bhadauria, O. Tekdas, and V. Isler, "Robotic data mules for collecting data over sparse sensor fields," *Journal of Field Robotics*, vol. 28, no. 3, p. 388–404, 2011.

[7] H. Alt, E. M. Arkin, H. Brönnimann, J. Erickson, S. P. Fekete, C. Knauer, J. Lenchner, J. S. B. Mitchell, and K. Whittlesey, "Minimum-cost coverage of point sets by disks," *Proceedings of the twenty-second annual symposium on Computational geometry - SCG 06*, 2006.

[8] H. G. Tanner, "Switched uav-ugv cooperation scheme for target detection," *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007.

[9] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, p. 16–25, 2006.

[10] Gerkey, Mataric, and M. Dias, "An empirical evaluation of coordination strategies for an auv and uav," Jan 1970. [Online]. Available: https://link.springer.com/article/10.1007/s10846-012-9728-z

[11] S. Arora, "Nearly linear time approximation schemes for euclidean tsp and other geometric problems," in *Proceedings 38th Annual Symposium on Foundations of Computer Science*, 1997, pp. 554–563.