2017 IEEE International Conference on Robotics and Automation (ICRA)
Singapore, May 29 - June 3, 2017

UB-ANC Planner: Energy Efficient Coverage Path Planning with Multiple Drones

Jalil Modares, Farshad Ghanei, Nicholas Mastronarde and Karthik Dantu

Abstract— **Advancements in the design of drones have led to their use in varied environments and applications such as battle field surveillance. In such scenarios, swarms of drones can coordinate to survey a given area. We consider the problem of covering an arbitrary area containing obstacles using multiple drones, i.e., the so-called *coverage path planning* (CPP) problem. The goal of the CPP problem is to find paths for each drone such that the entire area is covered. However, a major limitation in such deployments is drone flight time. To most efficiently use a swarm, we propose to minimize the maximum energy consumption among all drones' flight paths. We perform measurements to understand energy consumption of a drone. Using these measurements, we formulate an Energy Efficient Coverage Path Planning (EECPP) problem. We solve this problem in two steps: a load-balanced allocation of the given area to individual drones, and a minimum energy path planning (MEPP) problem for each drone. We conjecture that MEPP is NP-hard as it is similar to the Traveling Salesman Problem (TSP). We propose an adaptation of the well-known Lin-Kernighan heuristic for the TSP to efficiently solve the problem. We compare our solution to the recently proposed depth-limited search with back tracking algorithm, the optimal solution, and rastering as a baseline. Results show that our algorithm is more computationally efficient and provides more energy-efficient solutions compared to the other heuristics.**

I. Introduction

Networked unmanned aerial vehicles (UAVs) have emerged as an important technology for public safety, commercial, and military applications including search and rescue, disaster relief, precision agriculture, environmental monitoring, and surveillance. Many of these applications require sophisticated mission planning algorithms to coordinate multiple drones to cover an area efficiently. Such scenarios are complicated by the existence of obstacles, such as buildings, requiring detailed planning for effective operation. Although a lot of work has been done on mission planning, optimal mission planning solutions depend heavily on the specific types of vehicles considered (e.g., ground robots, indoor drones, or outdoor drones), their kinematics, and the specific applications. Prior techniques have been optimized for shortest time to completion or control efficiency. However, a major challenge in the realization of such solutions is the limited energy on each drone.

We consider the problem of covering an arbitrary area containing obstacles using multiple UAVs/drones with a max-min fair energy allocation across drones. Through experimental measurements, we have determined that there are two main factors that affect energy consumption in drones: distance traveled and turns. Traditional coverage path planning algorithms, such as those based on the Traveling Salesman Problem (TSP), are not ideal for drones because they only consider the distance traveled. We present a novel Energy Efficient Coverage Path Planning (EECPP) formulation that explicitly considers the energy consumption characteristics of drones in the path planning optimization, i.e., we not only consider the energy consumed traveling between consecutive waypoints (similar to the TSP), but we also consider the energy consumed by the drone when it accelerates into and out of turns. This work makes four contributions:

- From experimental measurements, we develop a linear model for energy consumption during drone flight.
- Using this model, we formulate the EECPP problem and show that it is NP-hard.
- We decompose the EECPP problem into two sub-problems: a load-balancing problem that fairly divides the area among drones and a minimum energy path planning (MEPP) problem for each drone.
- We adapt heuristics proposed for solving the TSP to efficiently solve the MEPP sub-problem on each drone.

The remainder of the paper is organized as follows. In Section II, we discuss related work. In Section III, we describe our experimental energy measurements and the energy model we derive from them. In Section IV, we introduce the EECPP problem formulation. In Section V, we present our simulation results comparing EECPP to rastering (baseline), depth-limited search, and (where possible) an optimal solution. We conclude in Section VI summarizing our main results and future work.

II. Related Work

Recently, there has been a lot of work on coverage path planning for UAVs. Ahmadzadeh et al. [1] introduce a coverage algorithm for surveillance using a set of fixed-wing UAVs. They utilize dynamic programming to maximize the coverage of the area by a camera mounted on the drone.

ACKNOWLEDGEMENT OF SUPPORT AND DISCLAIMER: (a) The University at Buffalo acknowledges the U.S. Government's support in the publication of this paper. This material is based upon work funded by the US Air Force Research Laboratory under Grant No. FA8750-14-1-0073. (b) Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of AFRL.

F. Ghanei and K. Dantu were supported in part by the National Science Foundation under Grant Number IIS-1514395. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

J. Modares and N. Mastronarde are with the Dept. of Electrical Engineering and F. Ghanei and K. Dantu are with the Dept. of Computer Science and Engineering at the University at Buffalo, Buffalo NY, 14260, USA.{jmod, farshadg, nmastron, kdantu}@buffalo.edu

978-1-5090-4633-1/17/$31.00 ©2017 IEEE

Maza et al. [7] propose a full coverage algorithm using a set of heterogeneous UAVs (mostly helicopters). First, they generate a polygonal partition of the area that takes into account the capabilities of each individual UAV, such as flight range. Each polygon in the partition is assigned to a UAV that will cover it in a zig-zag pattern (i.e., a raster scan) using a sweep direction that minimizes the number of turns. Environmental obstacles are not considered in [1], [7].

Barrientos et al. [3] present an approach to cover an area using multiple UAVs based on a depth-limited search with back tracking. First, they present a task scheduler to partition the target area into k non-overlapping areas for the k UAVs. The partitioning procedure is based on a negotiation process in which each UAV claims as much area as possible to cover. Then, the wavefront algorithm is used to cover each subarea. Given that their application objective is similar to ours, we have implemented a version of their algorithm and performed comparisons with respect to computational time as well as solution accuracy in the results.

Di Franco et al. [5] discuss an energy-aware coverage path planning solution for a single multi-rotor. They derive energy models for different operating conditions based on real measurements. However, they only consider distance and do not consider the impact of turns in their formulation.

Torres et al. [12] propose a coverage path planning solution for 3D terrain reconstruction with a single UAV. They decompose the area into one or more polygons and have the UAV cover each polygon using a raster scan. They try to minimize the number of turns by calculating the optimal line sweep direction.

Our work is focused on coverage path planning with multiple UAVs. While most previous work attempts to generate paths with minimum distance, our proposed solution instead optimizes energy consumption. Some work has explicitly attempted to minimize turns [3], [7], [12] . However, our empirical results suggest that explicitly optimizing for energy provides more energy-efficient paths than optimizing for turns or distance.

III. Energy Consumption of Drone Flight

Consider a realistic scenario where a team of drones are commanded to survey an area. Such an area could contain buildings, towers, and other man-made obstacles, or trees, hills, and other natural ones. Such obstacles can be convex or non-convex, making path planning fairly complex. In addition, path planning for multiple drones to concurrently cover this area is even more challenging. Further, path planning of a single drone could be optimized for several objectives such as shortest time, least distance traveled, least energy used and others. From empirical flight trials, we have concluded that battery energy is the primary resource that limits flight times of such aerial vehicles. Correspondingly, it would be ideal to optimize paths based on energy consumption to enable the drones to cover the maximum area possible.

In order to better understand the power consumption dynamics of the drones, we equipped one of the drones with a power measurement module as shown in Fig. 1. It has a custom frame, a waypoint-based Pixhawk flight controller, a Raspberry Pi 2 for onboard computing, and a 6-cell 10 Ah battery that gives it over 30 minutes of flight time. Fig. 1 also shows the energy sensing module. This module consists of four current sensing modules with ACS712 IC, which translate the passing currents as analog output voltages. We connected the power supply of the 4 motors to the sensors, and mounted all sensors together with an ADC converter with ADS1115 IC and a logic level converter. The ADC has four channels, each connected to one sensor, and can send the converted read values via I^{2}C. The logic level converter lets the IC communicate with the Raspberry Pi on the drone, which has a different logic voltage level.

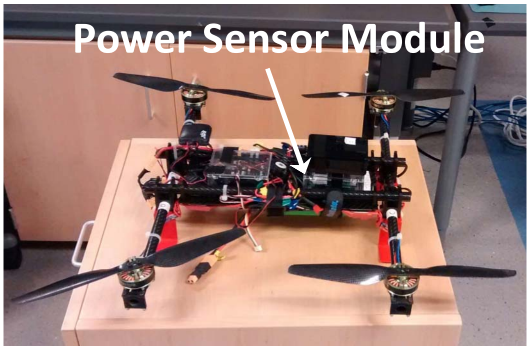
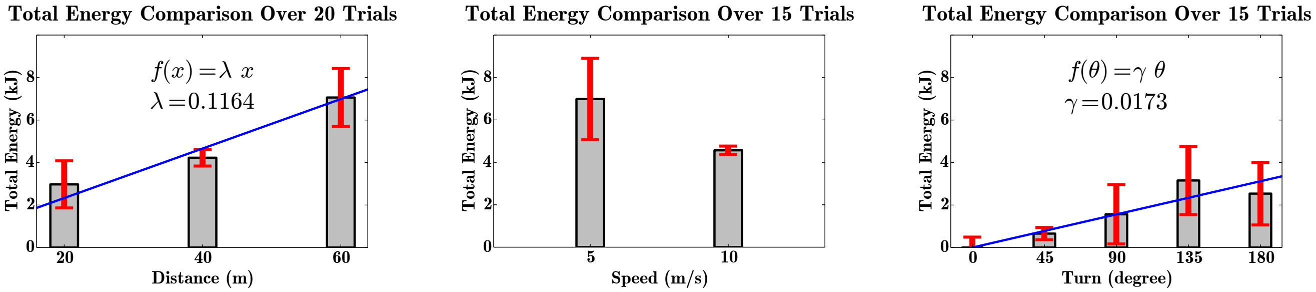


Fig. 1: A UB-ANC Drone [8], [11] with a custom frame, waypoint-based Pixhawk flight controller, Raspberry Pi 2, custom power sensor module, and 10 Ah battery. The drone weighs $3kgs$ and achieves over 30 minutes of flight time.

We used this instrumentation to measure the power consumption of the drone during flight to better understand the relationship between the distance, speed and direction change of the drone and its total power consumption. Each experiment was repeated multiple times and we show averaged results to alleviate anomalies from individual trials.

- **Straight Line Distance:** In this test, we let the drone fly in a straight line with a constant speed of $5m/s$ for 3 different distances: $20m$, $40m$, and $60m$. We want to see how the distance traveled affects the power consumption. During flight, the drone takes time to ramp up to the desired velocity and starts slowing down prior to reaching the destination so it can come to a stop at the destination waypoint.
- **Effect of Velocity:** In this test, we let the drone fly in a straight line for a constant distance of $40m$, but with 2 different target speeds: $5m/s$ and $10m/s$. This is to understand the effect of the target flight speed on the power consumption. As mentioned earlier, the drone must ramp up to its target speed and slow down prior to reaching its destination.
- **Effect of Turning:** In this test, we want to observe how direction changes affect power consumption. For this, we flew the drone $40m$ with a constant speed of $5m/s$ for five turn angles: 0°, 45°, 90°, 135°, and 180°. Specifically, for the 0°turn, we flew a $40m$ straight line path and for the other turn angles we flew a $20m$ straight



(a) Energy vs. Distance

(b) Energy vs. Speed

(c) Energy vs. Turn Angle

Fig. 2: Energy consumed as measured on the UB-ANC Drone for various patterns of flight. In Fig 2c, we omit the 180^o data point for line fitting as our planning does not allow re-visiting a node. It is shown here to demonstrate model validity.

line path, turned, and then flew another $20m$ straight line path. To isolate the energy consumption associated with turning, we subtracted the average straight path energy consumption from the average total energy that we measured for each angle.

Figs. 2a, 2b, and 2c show the average total energy consumption with standard deviation for the distance, speed, and turn tests, respectively. Fig. 2a shows that increasing the distance traveled increases the energy consumed approximately linearly. Assuming that traveling $0m$ incurs 0 energy cost, we performed a linear fit on the measured data and determined that the drone consumes approximately $\lambda = 0.1164kJ/m$. Fig. 2b shows the relationship between the energy consumption and the target speed. The drone was flown for a distance of $40m$ and commanded to fly at the given speed. Lower speeds result in greater time spent by the drone in the air and correspondingly greater energy consumption. Fig. 2c shows the effect of the turn angle on the energy. It is interesting to observe that increasing the turn angle increases the energy consumed for the same distance traveled in an almost linear manner. It is also noteworthy that the variance in energy consumed grows with greater turn angle. We performed a linear fit on the measured data and determined that the drone consumes approximately $\gamma = 0.0173kJ/deg$. This suggests that intelligently reducing the number and magnitude of turns in a path can potentially reduce energy consumption. Note that we use our measured values of λ and γ when we solve the optimizations proposed in Section IV-B.

Figs. 3a, 3b, and 3c show the power consumption and flight times for the same tests. All three graphs show that the average power consumption is nearly constant for all traveled distances, speeds, and turn angles. Therefore, the difference in energy consumption across tests is primarily due to the duration of the test. Based on the flight controller's design, the drone often slows down considerably when it enters a turn, which results in an increased flight time (Fig. 3c).

We note that for a different choice of drone, the energy consumption patterns might be different, but we believe that the trends are reasonably indicative of the energy consumption in most drones of this size. Our formulation allows for modeling the energy consumed for each of the path primitives (both distances and angles) and using them as parameters of the optimization. We now develop the EECPP formulation based on our empirical power/energy measurements on drone flight.

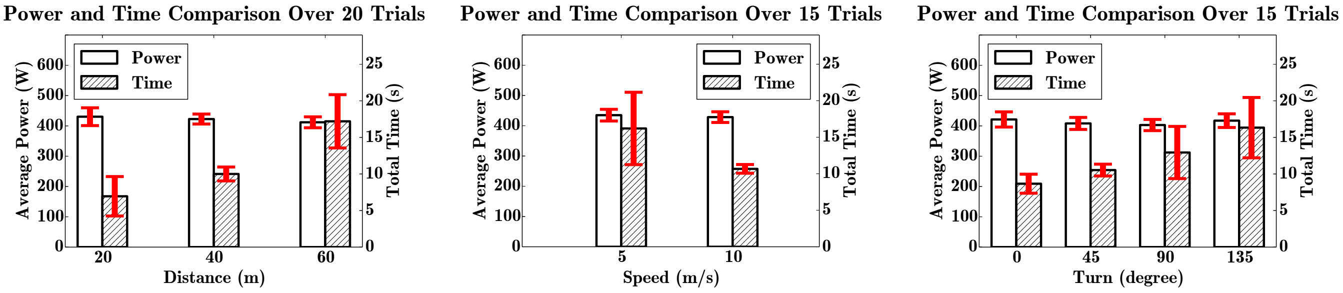
IV. Energy Efficient Coverage Path Planning For Multiple Drones

Following from our energy measurements, we formulate the Energy Efficient Coverage Path Planning (EECPP) problem in this section. The problem is divided into two subproblems: (i) fairly dividing the given area among the drones, and (ii) minimum energy path planning (MEPP) for each drone. Our formulation allows drones to start in different locations and requires each drone to return to its starting point akin to the TSP. These assumptions are drawn from intuition from real applications where surveying is part of a larger operation.

A. Problem Modeling

As our measurements have shown in Section III, a drone's energy consumption depends on the distance it travels and the number and degree of turns in its path. To find the minimum energy path for each drone, we formulate a vehicle routing problem (VRP) [4], which is a more general case of the multiple traveling salesman problem (mTSP). The original VRP problem is a min-sum optimization, which tries to minimize the sum of costs over all vehicles. We adapt that to a min-max formulation where we want to minimize the maximum cost incurred (energy expended) by any drone. In literature, this has been referred to as the Newspaper Routing Problem [2], which considers fairness among all vehicles (agents). However, our problem differs from the Newspaper Routing Problem because the objective function not only depends on the distance traveled, but also on the turns.

Surveillance of a given area requires coverage of all locations. However, given that the drone is flying at a given height, it is able to view a large area from its vantage point. Therefore, we represent the area to be surveyed as a set of grid cells and assume that a grid cell is covered if the drone visits its center. Formally, we represent the grid as a graph $\mathscr{G}(\mathscr{V}, \mathscr{E})$, where $\mathscr{V}$ is the set of nodes and $\mathscr{E}$ is the set of



(a) Power and Time vs. Distance traveled (b) Power and Time vs. Speed of travel (c) Power and Time vs. Turn Angle

Fig. 3: Average power draw and time for different missions.

edges. We let $i,j,k \in \mathscr{V}$ denote a specific node and $e_{ij} \in \mathscr{E}$ denote an edge between nodes i and j.

We define the pair (α_i, β_i) as the Cartesian coordinate of node $i \in \mathscr{V}$ and let c_{ij} denote the cost of traversing the edge e_{ij} between node $i \in \mathscr{V}$ and node $j \in \mathscr{V}$. In our path planning optimization, we assume that drones traverse adjacent cells. In other words, $e_{ij} \in \mathscr{E}$ if nodes $i, j \in \mathscr{V}$ are adjacent and $e_{ij} \notin \mathscr{E}$ otherwise. Based on our measurements in Section III, we assume that the cost (energy) is proportional to the distance traveled: i.e.,

$$c_{ij} = \begin{cases} \lambda\sqrt{(\alpha_i-\alpha_j)^2+(\beta_i-\beta_j)^2}, & \text{if } e_{ij} \in \mathscr{E} \\ \infty, & \text{otherwise.} \end{cases} \quad (1)$$

In other words, if two nodes are adjacent, then the energy cost to traverse the edge between them is proportional to the Euclidean distance between the centers of their corresponding cells (where the parameter λ kJ/m is specific to the drone as described in Section III); however, if two nodes are not adjacent, then the cost to traverse them is infinite (i.e., it is not possible to directly traverse the two nodes because they are not connected by any edge).

Let θ_{ijk} denote the exterior angle between nodes $i,j,k \in \mathscr{V}$ (Fig. 4). The squared length of the edges of the triangle made by nodes i,j,k can be determined as follows:

$$r = (\alpha_i-\alpha_j)^2+(\beta_i-\beta_j)^2, \quad (2)$$

$$s = (\alpha_j-\alpha_k)^2+(\beta_j-\beta_k)^2, \text{and} \quad (3)$$

$$t = (\alpha_k-\alpha_i)^2+(\beta_k-\beta_i)^2 \quad (4)$$

We know that given the lengths of three sides of a triangle, we can calculate an internal angle using the Law of Cosines. It follows that the exterior angle between nodes $i,j,k \in \mathscr{V}$ can be written as:

$$\theta_{ijk} = \pi - cos^{-1}\left[\frac{(r+s-t)}{\sqrt{4rs}}\right] \text{ radians.} \quad (5)$$

From our empirical energy measurements, we model the cost associated with a feasible turn, denoted by q_{ijk}, to be proportional to the angle of the turn (where the parameter γ kJ/deg is specific to the drone as described in Section III):

$$q_{ijk} = \begin{cases} \gamma\frac{180}{\pi}\theta_{ijk}, & \text{if } e_{ij}, e_{jk} \in \mathscr{E} \\ \infty, & \text{otherwise.} \end{cases} \quad (6)$$

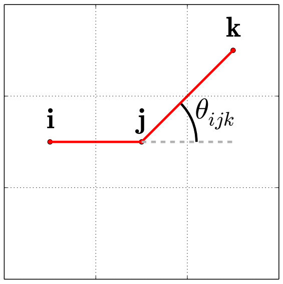


Fig. 4: A cell and its neighbors, and the exterior angle for three nodes on the path.

B. Problem Formulation

Let $\mathscr{A}$ denote the set of agents that will cover the area and let $v_a \in \mathscr{V}$ denote the starting node for agent $a \in \mathscr{A}$. We indicate which edges each agent traverses using the binary decision variable $x_{ij}^a \in \{0,1\}$, where

$$x_{ij}^a = \begin{cases} 1, & \text{if agent } a \in \mathscr{A} \text{ traverses edge } e_{ij} \in \mathscr{E} \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

Given a feasible path assignment (i.e., a sequence of edges), agent $a \in \mathscr{A}$ will incur a total cost of

$$\sum_{i\in\mathscr{V}}\sum_{j\in\mathscr{V}\setminus\{i\}} c_{ij}x_{ij}^a + \sum_{i\in\mathscr{V}}\sum_{j\in\mathscr{V}\setminus\{i,v_a\}}\sum_{k\in\mathscr{V}\setminus\{j\}} q_{ijk}x_{ij}^a x_{jk}^a, \quad (8)$$

where the first term in the cost function is proportional to the distance traveled (as in the TSP) and the second term is proportional to the sum of turn angles (unique to the EECPP).

Formally, the EECPP problem can be stated in Equations 9. The objective of the EECPP (Equation (9a)) is to determine the paths for each drone that minimize the maximum cost incurred by any individual drone, where the cost function is defined in Equation (8). There are several constraints governing this optimization. First, each node should be visited exactly once (Equation (9b)). Next, we need flow conservation constraints which ensure that, once a drone visits a node, it also departs from the same node (Equation (9c)). Third, we incorporate extensions of MTZ-based SECs [4] (subtour elimination constraints) to a threeindex model (Equation (9d)).[1] Finally, u_i is a dummy variable associated with node $i \in \mathscr{V}$ which is assigned by the solver.

$$\min \max_{a\in\mathscr{A}} \sum_{i\in\mathscr{V}} \sum_{j\in\mathscr{V}\setminus\{i\}} c_{ij}x^a_{ij} + \sum_{i\in\mathscr{V}} \sum_{j\in\mathscr{V}\setminus\{i,v_a\}} \sum_{k\in\mathscr{V}\setminus\{j\}} q_{ijk}x^a_{ij}x^a_{jk} \tag{9a}$$

$$\text{s.t.} \quad \sum_{a\in\mathscr{A}} \sum_{i\in\mathscr{V}\setminus\{j\}} x^a_{ij} = 1,\ \forall j \in \mathscr{V} \tag{9b}$$

$$\sum_{i\in\mathscr{V}\setminus\{j\}} x^a_{ij} - \sum_{k\in\mathscr{V}\setminus\{j\}} x^a_{jk} = 0,\ \forall a \in \mathscr{A}, \forall j \in \mathscr{V} \tag{9c}$$

$$u_i - u_j + |\mathscr{V}|x^a_{ij} \leq |\mathscr{V}| - 1, \quad \forall a \in \mathscr{A}, \forall i,j \in \mathscr{V}\setminus\{v_a\} \text{ and } i \neq j \tag{9d}$$

$$u_i \in \mathscr{Z},\ \forall i \in \mathscr{V} \tag{9e}$$

$$x^a_{ij} \in \{0,1\},\ \forall i,j \in \mathscr{V} \text{ and } \forall a \in \mathscr{A} \tag{9f}$$

The problem shown above is an NP-hard mixed integer quadratic constrained program (MIQCP); therefore, it does not scale well beyond a few drones and dozens of cells. To overcome this limitation, we decompose the problem into two sub-problems: the first sub-problem assigns a set of cells to each drone and the second determines the minimum energy path that each drone will follow to cover these cells.

C. Sub-Problem 1: Load Balancing (LB)

The first sub-problem is the so-called load balancing (LB) problem. We use mixed integer linear programming (MILP) to divide the nodes among the users with linear complexity. Let c_{ai} denote the distance between agent $a \in \mathscr{A}$ and node $i \in \mathscr{V}$. Let x_{ai} denote a decision variable that is set to 1 if node $i \in \mathscr{V}$ is assigned to agent $a \in \mathscr{A}$ and is set to 0 otherwise: i.e.,

$$x_{ai} = \begin{cases} 1, & \text{if node } i \in \mathscr{V} \text{ is assigned to agent } a \in \mathscr{A} \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Based on the starting positions of the drones, we would like to assign grid cells to them to minimize the maximum energy incurred across them. This can be formulated as:

$$\min \max_{a\in\mathscr{A}} \sum_{i\in\mathscr{V}} c_{ai}x_{ai} \tag{11a}$$

$$\text{s.t.} \quad \sum_{a\in\mathscr{A}} x_{ai} = 1,\ \forall i \in \mathscr{V} \tag{11b}$$

$$x_{ai} \in \{0,1\},\ \forall a \in \mathscr{A} \text{ and } \forall i \in \mathscr{V} \tag{11c}$$

This can be solved by a linear program with complexity that is linear in the product of the number of drones and the number of grid cells.

D. Sub-Problem 2: Minimum Energy Path Planning (MEPP)

The second sub-problem is the minimum energy path planning (MEPP) problem. After dividing the area in sub-problem 1, we use mixed integer quadratic programming (MIQP) to formulate the MEPP problem. The problem is very similar to the EECPP (9a) except that there are no indices for individual drones. It is shown below.

$$\begin{aligned} \min \quad & \sum_{i\in\mathscr{V}} \sum_{j\in\mathscr{V}\setminus\{i\}} c_{ij}x_{ij} + \sum_{i\in\mathscr{V}} \sum_{j\in\mathscr{V}\setminus\{i,v_a\}} \sum_{k\in\mathscr{V}\setminus\{j\}} q_{ijk}x_{ij}x_{jk} \\ \text{s.t.} \quad & \sum_{i\in\mathscr{V}\setminus\{j\}} x_{ij} = 1,\ \forall j \in \mathscr{V} \\ & \sum_{j\in\mathscr{V}\setminus\{i\}} x_{ij} = 1,\ \forall i \in \mathscr{V} \\ & u_i - u_j + |\mathscr{V}|x_{ij} \leq |\mathscr{V}| - 1, \\ & \qquad \forall i,j \in \mathscr{V}\setminus\{v_a\} \text{ and } i \neq j \\ & u_i \in \mathscr{Z},\ \forall i \in \mathscr{V} \\ & x_{ij} \in \{0,1\},\ \forall i,j \in \mathscr{V} \end{aligned}$$

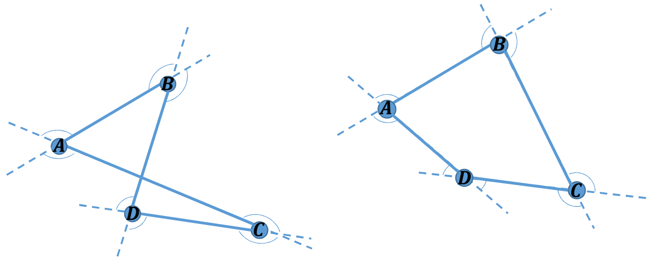
We conjecture that the above minimum energy path planning problem is NP-hard since it is similar to the TSP, but with additional quadratic terms to account for the turning costs. To solve this problem efficiently when a large number of nodes are assigned to a drone, we propose a modification of the well-known Lin-Kernighan Heuristic (LKH [6]). While the conventional LKH only considers the distance traveled, we modify it to also account for the drone's turning costs.

Lin-Kernighan Heuristic for Drones (LKH-D): We now briefly describe the LKH used to solve the TSP. We then show how we have modified the LKH to account for the cost of turns in the MEPP problem. We call the new algorithm LKH for Drones (LKH-D).

LKH begins by determining a feasible tour T that visits each node exactly once and returns to the origin node. The tour T is associated with a cost $f(T)$, which is equal to the length of the tour. LKH works by iteratively improving the initial tour using a specific transformation (see, e.g., [6]). After applying the transformation on the tour T, a new feasible tour T' is obtained, which has a cost $f(T')$. If the gain $g(T,T') = f(T) - f(T')$ is positive (i.e., the tour T' has a lower cost than the tour T), then the new tour is adopted; otherwise, it is thrown away. This procedure is repeated iteratively until a specific stopping condition is met (see, e.g., [6]). Fig. 5a and Fig. 5b show a four node tour T and a feasible tour T' obtained by an appropriate transformation on T: in this example, T' is obtained from T by replacing edges $\overline{BD}$ and $\overline{CA}$ with edges $\overline{BC}$ and $\overline{DA}$, respectively.

Our proposed LKH-D algorithm follows the same iterative approach as the conventional LKH, but uses a different cost function to account for the drone's turning costs. Specifically, $f(T)$ is calculated as a weighted sum of the length of the tour and the sum of the turn angles within the tour (these are the four exterior angles illustrated in Figs. 5a and 5b.) In other words, $f(T)$ is equal to the energy cost defined in (8) associated with the tour T.

[1] A subtour is a closed path that starts from one node and returns to that node. The subtour elimination constraints prevent the optimization solver from returning undesirable subtours as solutions.



(a) Initial feasible tour T.

(b) New feasible tour T' obtained from tour T.

Fig. 5: Illustration of the Lin-Kernighan Heuristic (LKH).

V. Simulation Results

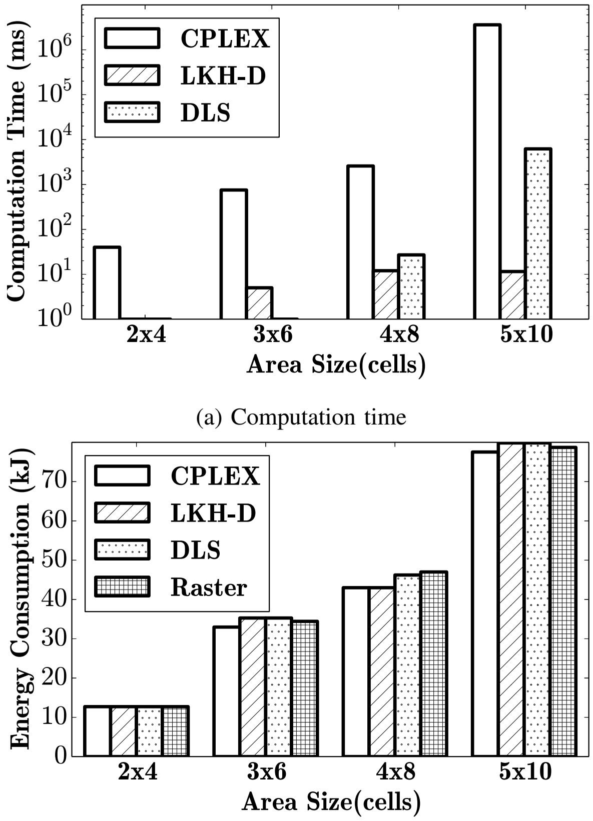
We perform several sets of simulations to demonstrate the benefits of our proposed heuristics for the EECPP problem with multiple drones and the MEPP problem (i.e., subproblem 2) for a single drone. We compare our solution with (a) simple rastering, which has no planning cost, (b) a previously proposed depth-limited search (DLS) algorithm with backtracking for multi-robot search [3],[2] and (c) an brute-force optimal solution wherever possible. (We imposed a 1 hour execution time limit for all algorithms, and present the best result obtained in that time). We have compared these algorithms in several scenarios to demonstrate the utility of our solution.

For all our simulations, we use the UB-ANC Emulator [9], [10],[3] which is an emulation framework that we developed to simplify the transition from simulation to experimentation with multi-agent drone systems. In particular, our simulations leverage a software-in-the-loop (SITL) version of the flight controller's firmware, which allows our simulations to be immediately ported to actual hardware.

We perform comparisons with the benchmark algorithms on several dimensions. We vary the area of coverage by a single drone to compare the scalability of the algorithms with increased area (number of grid cells). We compare the efficiency of each approach by comparing the total energy consumed by the drone for that mission. We also perform these comparisons in four scenarios with one or more obstacles as shown in Fig. 7. Finally, we demonstrate a full run of our multi-robot path planning problem by simulating a set of drones covering a large area. Our results are from simulations on a standard laptop. We impose a 1-hr run limit on all our algorithms.

We note that Sections V-A and V-B focus primarily on the MEPP problem for a single drone. In Section V-C, we consider the full EECPP problem with multiple drones covering a large area.

A. Algorithm Scalability



(a) Computation time

(b) Performance

Fig. 6: Run-time (max: 1hr) and performance of MEPP on rectangular grids of different sizes without obstacles.

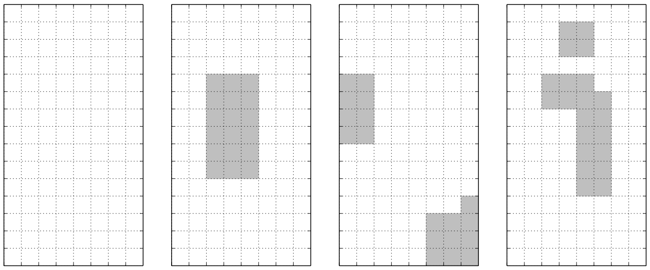
First, we show how the compared algorithms perform in terms of computation time and energy efficiency for simple rectangular maps with dimensions 2×4, 3×6, 4×8, and 5×10. The cells in each of these maps are $10m \times 10m$ squares and there are no obstacles. Fig. 6a shows how the computation time scales with the number of cells for each algorithm and Fig. 6b shows the energy consumption under each algorithm. Please note that the time axis in Fig. 6a is in log-scale. As would be expected, the optimal solution for the MEPP problem, which is similar to the TSP, is computationally expensive. Rastering does not require any planning and is not represented in the figure. In comparison to DLS [3], the proposed LKH-D is three orders of magnitude faster for the large map. Fig. 6b shows that for small areas without obstacles all algorithms achieve comparable performance to the exhaustive CPLEX solution (which is optimal for grid sizes up to 4×8).

B. Energy Efficiency and Algorithm Adaptivity

As discussed in the introduction, a major challenge in path planning is adapting to real-world constraints such as obstacles and areas shaped in a non-standard manner. To understand the effects of obstacles on the computation time and performance of the compared algorithms, we generated four 8×15 rectangular maps as illustrated in Fig. 7. Given the size of the area (120 cells), we were unable to run the CPLEX to solve the MEPP problem to completion in all cases. Instead, we run the optimization for one hour and

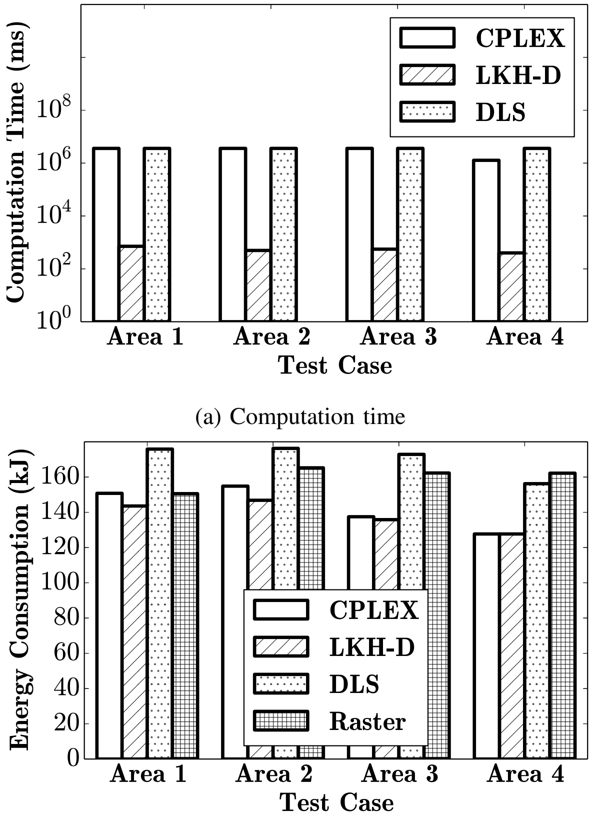
[2] Since no source code was available for this algorithm, we have faithfully implemented our version of the proposed algorithm and verified its functionality with results from various papers on this algorithm.

[3] https://github.com/jmodares/UB-ANC-Emulator



(a) Area 1 (b) Area 2 (c) Area 3 (d) Area 4

Fig. 7: Rectangular grid maps used to evaluate the algorithms. Grey cells represents obstacles.



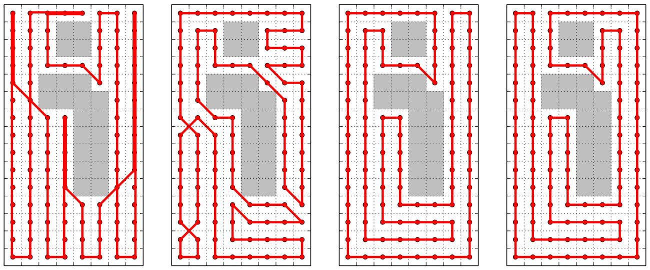
(a) Computation time

(b) Performance

Fig. 8: Comparison of algorithms over areas in Fig. 7.

report results returned by the CPLEX optimization solver.

Fig. 8a shows the computation time for the compared algorithms when they are applied to the four areas defined in Fig. 7, and Fig. 8b shows the corresponding energy consumption. In all scenarios, our proposed heuristic (LKH-D) performs at least as well as the time-limited CPLEX solution and does approximately 15-20% better than the DLS and rastering solutions. Moreover, LKH-D achieves this performance in 3-orders of magnitude less time than we allowed for the CPLEX and DLS solutions. These results highlight the benefits of both our modeling and the proposed heuristic for energy-efficient path planning.



(a) Rastering (b) DLS (c) CPLEX (d) LKH-D

Fig. 9: Visualization of the planned paths for different algorithms in area 4 of Fig. 7. The drone's tour starts and ends in the upper-right corner of the grid.

TABLE I: Simulation statistics for area 4 in Fig. 7.

Algorithm	Raster	DLS (1 hr)	CPLEX	LKH-D
Energy (kJ)	162.2	156.2	127.7	127.7
Comp Time (s)	0	3600	1276	0.4
Distance (m)	1244.8	1043.8	994.1	994.1
Turns (degree)	1620	2970	1620	1620

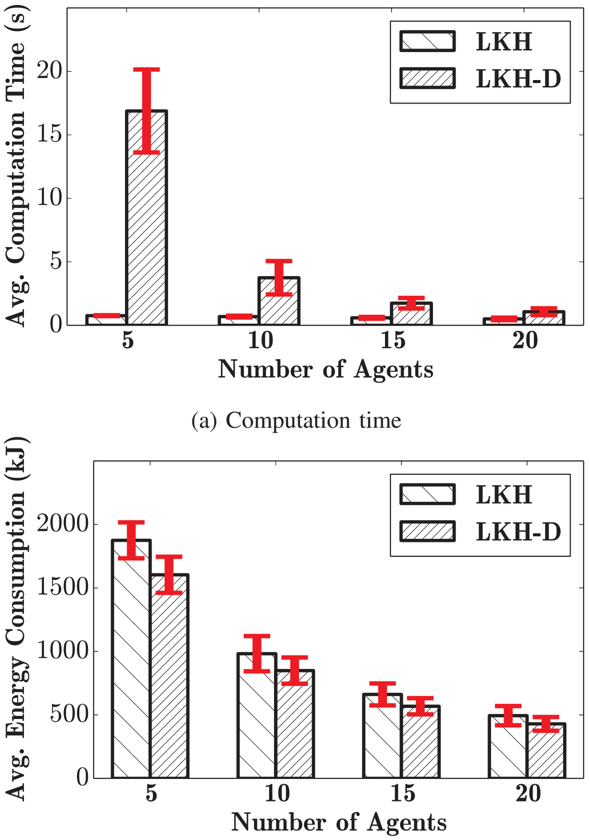
Fig. 9 illustrates the planned paths under each algorithm for the area shown in Fig. 7d, and Table I provides some statistics about these paths (energy, computation time, total distance, and total degree of turns). The proposed heuristic (LKH-D) achieves the optimal energy consumption, the minimum distance traveled, and the minimum total turns, and is executed in less than half a second.

C. Multi-Drone Path Planning

To show the application of the proposed method for solving the EECPP problem (load balancing followed by LKH-D to solve the MEPP) in a large-scale scenario, we executed it with a varying number of drones (5, 10, 15, and 20) on the University at Buffalo's North Campus in simulation. A top view of the area can be seen in Fig. 11. The area is decomposed into over 3000 cells measuring $20m \times 20m$. Running the multi-agent coverage path planning on the area with different numbers of drones, with both LKH and LKH-D algorithms, we obtain the results shown in Fig. 10a and 10b. Fig. 10a compares them with respect to average computation time required for the path planning per drone across all drones in each mission. Fig. 10b compares these algorithms with respect to average energy expended per drone. As expected, increasing the number of available drones allows each agent to cover less area, and correspondingly decreases the computation time and energy expended for the mission. Finally, LKH-D achieves lower energy consumption than LKH, at the expense of increased computation time, because it considers the turning costs.

VI. Conclusion

We formulated the Energy Efficient Coverage Path Planning (EECPP) problem for covering an arbitrary area containing obstacles using multiple drones. The goal of the EECPP problem is to minimize the maximum energy required for any individual drone to traverse its assigned



(a) Computation time

(b) Performance

Fig. 10: Average path planning computation time (per drone), and energy consumption (per drone) for a set of drones covering UB North campus. Comparison between LKH and LKH-D algorithms.

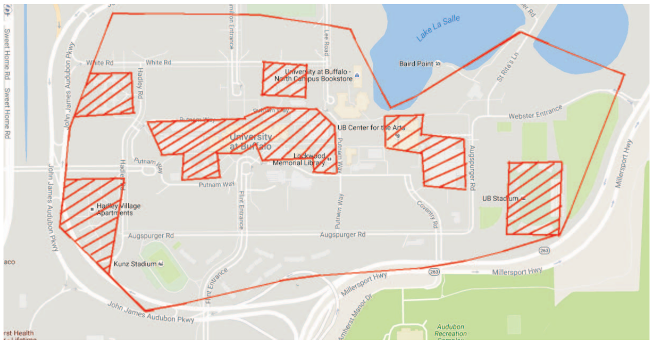


Fig. 11: UB North Campus. Areas dense with buildings are assumed to obstacles (shaded with diagonal lines).

path. Unlike the conventional multiple traveling salesman problem (mTSP), the vehicle routing problem (VRP), and the newspaper routing problem, which only consider the distance traveled by each agent, we accounted for the energy consumption characteristics of drones in our optimization. In particular, we included a term to account for the additional energy consumed by drones when they accelerate into and out of turns. This decision was justified by experimental energy measurements on an actual drone. However, we conjectured that this problem is NP-hard, and we decomposed it into two sub-problems. The first sub-problem divides the area among drones and has linear complexity. The second sub-problem then determines the path for each drone to cover its assigned area. Although the second sub-problem is similar to the TSP (which is NP-hard), we believe that this is acceptable as long as there are enough drones to divide the areas into reasonably sized regions (with under 50 cells) for the solver to process in a timely manner. More complex grids can be solved using sub-optimal heuristic algorithms. To this end, we adapted a heuristic for the TSP problem (LKH) and proposed the LKH-D algorithm incorporating energy consumption into the solution. We showed in simulation that our proposed heuristic is computationally faster than previously proposed algorithms and comes up with more energy-efficient paths.

In future work, we plan to extend the proposed framework to integrate network performance metrics into the path planning optimization, e.g., to optimize connectivity, throughput, or delay. We implemented UB-ANC Planner based on the algorithms mentioned in the paper. UB-ANC Planner is available as open-source at `https://github.com/jmodares/UB-ANC-Planner`.

References

[1] Ahmadzadeh, A., Keller, J., Pappas, G., Jadbabaie, A., Kumar, V.: An optimization-based approach to time-critical cooperative surveillance and coverage with uavs. In: Experimental Robotics, pp. 491–500. Springer (2008)

[2] Applegate, D., Cook, W., Dash, S., Rohe, A.: Solution of a min-max vehicle routing problem. INFORMS Journal on Computing **14**(2), 132–143 (2002)

[3] Barrientos, A., Colorado, J., Cerro, J.d., Martinez, A., Rossi, C., Sanz, D., Valente, J.: Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. Journal of Field Robotics **28**(5), 667–689 (2011)

[4] Bektas, T.: The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega **34**(3), 209–219 (2006)

[5] Di Franco, C., Buttazzo, G.: Energy-aware coverage path planning of uavs. In: Autonomous Robot Systems and Competitions (ICARSC), 2015 IEEE International Conference on, pp. 111–117. IEEE (2015)

[6] Lin, S., Kernighan, B.W.: An effective heuristic algorithm for the traveling-salesman problem. Operations research **21**(2), 498–516 (1973)

[7] Maza, I., Ollero, A.: Multiple uav cooperative searching operation using polygon area decomposition and efficient coverage algorithms. In: Distributed Autonomous Robotic Systems 6, pp. 221–230. Springer (2007)

[8] Modares, J., Mastronarde, N.: Ub-anc: A flexible airborne networking and communications testbed: Poster. In: Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization, WiNTECH '16, pp. 95–96. ACM, New York, NY, USA (2016)

[9] Modares, J., Mastronarde, N., Dantu, K.: Ub-anc emulator: An emulation framework for multi-agent drone networks. In: IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR) (2016)

[10] Modares, J., Mastronarde, N., Dantu, K.: Ub-anc emulator: An emulation framework for multi-agent drone networks: Demo. In: Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization, WiNTECH '16, pp. 93–94. ACM, New York, NY, USA (2016)

[11] Modares, J., Mastronarde, N., Medley, M.J., Matyjas, J.D.: Ub-anc: An open platform testbed for software-defined airborne networking and communications. arXiv preprint arXiv:1509.08346 (2015)

[12] Torres, M., Pelta, D.A., Verdegay, J.L., Torres, J.C.: Coverage path planning with unmanned aerial vehicles for 3d terrain reconstruction. Expert Systems with Applications **55**, 441–451 (2016)