# Energy-aware UAV Path Planning with Adaptive Speed

Jonathan Diller
Colorado School of Mines
Golden, USA
jdiller@mines.edu

Qi Han
Colorado School of Mines
Golden, USA
qhan@mines.edu

## ABSTRACT

Unmanned Aerial Vehicles (UAVs) are a versatile platform that can be used for many data collection applications including emergency response, environmental monitoring, surveillance and many others. In this work, we investigate how to plan efficient paths that minimize mission completion time for UAV data collection where the UAV must rendezvous with a moving ground vehicle that cannot stop and wait for the UAV. We also address the limited onboard energy storage issue by adapting UAV speed. We propose a mixed-integer nonlinear program solution to solve the underlying path planning problem to optimality and provide a more tractable alternative approach. We evaluate our two approaches in extensive simulations using real UAV characteristics and prototype our solution on a physical drone testbed. We show that our two approaches can reduce completion time by up to 23.8% and 14.5%, respectively, when compared against other baseline approaches and demonstrate the importance of UAV speed adaptation in route planning for UAVs.

## KEYWORDS

UAV Path Planning; Cooperative Vehicle Routing; Mixed-Integer Nonlinear Program

**ACM Reference Format:**
Jonathan Diller and Qi Han. 2023. Energy-aware UAV Path Planning with Adaptive Speed. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

## 1 INTRODUCTION

Unmanned Aerial Vehicles (UAVs) can easily move over rough terrain and water, are low-cost, commercially available, and can be deployed quickly. This makes them well suited for data collection in various applications such as environmental monitoring, emergency response, search and rescue missions, and surveillance.

In many of these applications, the UAVs often work with ground vehicles which provide battery recharging, battery swap, or data offloading, etc. While a UAV completes its given tasks, the ground vehicle deploying the UAV may need to continue moving rather than wait at the starting location for the UAV to return. Examples include large ships that cannot easily stop, disaster response scenarios where rescue crews must cover large areas quickly, or military applications where it is unsafe to remain stationary. Hence, UAV path planning should account for the movement of a ground vehicle.

Another common requirement in these applications is minimizing mission completion time. Intuitively, this implies that we want to find an optimal path for a UAV to follow while maximizing the UAV's speed. However, UAVs have limited onboard energy storage and there is a trade-off between speed and energy consumption [23, 27]. Any robust path planning approach should consider the speed-energy consumption trade-off and plan for recharging or battery swaps.

Our goal is to minimize UAV mission completion time by planning efficient paths while accounting for a moving ground vehicle that travels along a fixed path, UAV's limited onboard energy, and adaptive UAV speed. In particular, we make the following contributions:

(1) We formally define the problem at hand, termed Minimum-Time while On-The-Move (MT-OTM), and propose a solution framework that reduces the problem to a multiple depot, multiple terminal, Hamiltonian path problem with fixed depots and terminals (fixed-MdMtHPP).
(2) We formulate two solutions for the fixed-MdMtHPP, a Mixed-integer nonlinear program (MINLP) that is tailored to optimize UAV speed and a more tractable heuristics-based K-means clustering algorithm with an Integer Program ($k$-IP).
(3) We conduct extensive simulations and show that our MINLP approach and the $k$-IP solution reduce completion time by 23.8% and 14.5%, respectively, when compared to a baseline approach.
(4) We prototype our solution on a physical UAV testbed to validate our methods and demonstrate how the MT-OTM Problem can be applied in a real world scenario.

## 2 RELATED WORK

In this section we review recent works in literature with a focus on UAV path planning algorithms for mixed UAV and ground vehicle problems. We also review UAV energy models and how they have been applied in UAV path planning.

### 2.1 Mixed UAV & Ground Vehicle Path Planning

Many previous works look at cooperative problems involving UAVs and ground vehicles where a series of waypoints must be visited. In [25] they consider an application where either a UAV or an Unmanned Ground Vehicle (UGV) must be within some distance of a set of waypoints, which is modeled as an orienteering problem, a problem with known algorithms. A related scenario is considered in [26], where a UGV can ferry around the UAV. The authors model the problem as a Generalized Traveling Salesman Problem (TSP) and apply a known solution approaches [18]. In [12] they consider UAV path planning where a ground vehicle can be used to swap-out batteries on the UAV but is constrained to a network of streets. They propose treating all UAV waypoints as a single TSP, then breaking up the route into sub-tours. This work was further expanded in

*Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.* 

[1, 13] where they propose a mixed-integer linear programming (MILP) solution. A similar problem is found in [4], where they first plan ground vehicle routes along a road network then plan UAV routes using Conflict-Based Search. A common theme in all of these works is that the ground vehicle can stop and wait for the UAV to finish flying a sub-tour of waypoints, which often allows the problem to be modeled as a traditional graph theory problem and solved using known techniques. In contrast, our work considers how to plan UAV paths when the ground vehicle cannot stop and wait for the UAV.

A related set of UAV and ground vehicle problems is the UAV parcel delivery problem, where a delivery truck, bound to a network of streets, acts as a launching point for a UAV to deliver a package in a last-mile delivery system [15]. Sometimes multiple UAVs are used, as seen in [17] and [20], and sometimes the UAV can deliver multiple packages [19]. These problems are usually solved by breaking down the larger problem into smaller ones that can be handled using math programming techniques. Most of the literature on this set of problems allows for one of the vehicles to stop and wait for the other. However, in applications over water or rugged terrain the UAV cannot land to wait for the ground vehicle and hovering is energy intensive.

Another set of related problems looks at selecting rendezvous locations for a UAV to be recharged by a ground vehicle. In [14], the authors plan where a ground vehicle should meet up with a UAV on a fixed route by converting the problem into the Generalized TSP and solve it using both integer programming and the Lin-Kernighan-Helsgaun (LKH) heuristic [6]. In [24] both the UAV and UGV have established paths and must select rendezvous points that are ideal for both vehicles, which is done using a Markov Decision Process. Although this work also selects rendezvous locations, our work differs from the current literature because we consider planning rendezvouses with a non-stopping ground vehicle and fit this problem into a larger, combined rendezvous and path planning problem.

For this work, we assume that the drone is capable of landing on the ground vehicle after rendezvousing using techniques such as the ones discussed in [2]. Landing on a moving vehicle falls out of the scope of this research.

## 2.2 UAV Energy Models

UAVs have limited onboard energy. There have been many proposed methods for modeling energy consumption in UAVs including: distance-based method [9, 12], time-based method [8, 22], discretized approaches [25, 26], and velocity-based methods [10]. The velocity-based model maps the UAV's speed to power consumption based on characteristics specific to the UAV. This model was formulated separately in [27] and [10] and validated in [10, 23] through field testing on physical testbeds. In [17] it was found that the other energy modeling approaches are not as accurate.

The velocity-based energy model has been used in various UAV path planning problems. In [17], it was applied to the UAV parcel delivery problem and in [24] it was used to better model the risk of the UAV running out of energy in a rendezvousing problem. In [20], they used this energy model to adapt UAV speed as a post-processing step to further improve mission completion time. However, mission completion time is the product of both the UAV's speed and the distance that the UAV must travel. To directly optimize completion time the energy model should be embedded in the solution formulation. In our work, we add speed adaptation using the velocity-based energy model directly in our mathematical formulation and evaluate speed adaptation against fixed-speed approaches for a multi-waypoint UAV problem.

# 3 PROBLEM FORMULATION

In this section, we describe the system setup and formally define the Minimum-Time while On-the-Move (MT-OTM) problem.

Let $\mathcal{X}$ be a large area with several navigational waypoints that must be visited. A ground vehicle that acts as a moving base station moves through $\mathcal{X}$ on a predetermined, fixed route described by $p_b(t)$, a function that returns the ground vehicle's position at time $t$. Without loss of generality, we assume that the origin of the two-dimensional coordinate system for $\mathcal{X}$ is at the ground vehicle's initial position at the beginning of the considered time window.

Let $P$ be the set of all waypoints that must be visited by a UAV and $p_i$ be the $i^{th}$ waypoint in $P$. Due to energy constraints, the UAV may not be able to visit all of the waypoints in $P$ in a single tour. Let $m$ be the number of sub-tours required to visit every waypoint in $P$, which is initially unknown. We define the $k^{th}$ sub-tour, denoted as $\delta_k$ with path length $d_k$, as an ordered set of waypoints containing a starting waypoint (*depot*), an ending waypoint (*terminal*), and at least one waypoint in $P$. We denote the depot of sub-tour $\delta_k$ as $p_k^d$ and the terminal as $p_k^t$. Let $\mathbf{\Delta}_m$ be a set of $m$ sub-tours such that every waypoint in $P$ is visited exactly once and let the set of speeds for each sub-tour in $\mathbf{\Delta}_m$ be $S_m$. The time it takes a UAV to travel $\delta_k$ while moving at speed $s_k \in S_m$ will be

$$t_\delta(\delta_k, s_k) = \frac{d_k}{s_k}. \tag{1}$$

Suppose that it takes $t_b$ seconds to land the UAV on the ground vehicle and change out the battery between each sub-tour. The total time to complete the set of sub-tours $\mathbf{\Delta}_m$ will be

$$t(\mathbf{\Delta}_m, S_m) = t_b * (m-1) + \sum_{k=1}^{m} t_\delta(\delta_k, s_k). \tag{2}$$

We denote the time lapsed from the start of the mission until the start of sub-tour $k$ as $\tau_k^d$ and the end of sub-tour $k$ as $\tau_k^t$. By definition, for any sub-tour $k$, $\tau_k^t = \tau_k^d + t_\delta(\delta_k, s_k)$. We assume that the UAV is launched on the first sub-tour at $t = 0$ and that the UAV is re-launched on any consecutive sub-tour as soon as the battery has been changed. That is, $\tau_1^d = 0$ and for any $k > 1$, $\tau_k^d = \tau_{k-1}^t + t_b$. We say that sub-tour $\delta_k$ is *consistent* if the depot waypoint of $\delta_k$ is $p_b(\tau_k^d)$ and the terminal waypoint is $p_b(\tau_k^t)$. That is, the UAV must start $\delta_k$ at the ground vehicle's position at time $\tau_k^d$ and end $\delta_k$ at the ground vehicle's position at time $\tau_k^t$ for $\delta_k$ to be considered a consistent sub-tour.

We formally define the MT-OTM problem as:

*Definition 3.1 (MT-OTM Problem).* Given search-space $\mathcal{X}$ with $n$ navigational waypoints and vehicle position function $p_b(t)$, determine the number of sub-tours $m$ and a corresponding sets of

consistent sub-tours $\Delta_m$ and speeds $S_m$ such that $t(\Delta_m, S_m)$ is the minimum time required to visit all waypoints.

We acknowledge but ignore various factors that could also affect flight performance such as wind and the energy consumed by making turns. Although we do not consider these and other minor factors that can impact energy consumption, our proposed solution is versatile and can be reapplied with more comprehensive energy models.

## 4 ADAPTION OF UAV SPEED

Previous work has determined the amount of power consumed by a UAV at varying speeds [10, 27]. Propulsion power consumption of a rotary-wing UAV as a function of speed $v$ can be approximated [27]:

$$\mathcal{P}(v) \approx C_0 \left(1 + \frac{3v^2}{U_{tip}^2}\right) + \frac{C_i v_0}{v} + \frac{1}{2} d_o \rho s_r A v^3, \tag{3}$$

where $C_0$ and $C_i$ are constants representing blade profile power and induced power, respectively, $U_{tip}$ represents the tip speed of the UAV's propellers, $v_0$ is what is known as the mean rotor induced velocity while hovering, $d_0$ is an aircraft-specific drag ratio, $s_r$ is rotor solidity, $\rho$ is the air density and $A$ is the rotor disk area. This function is highly dependent on specific aircraft parameters and will change from aircraft to aircraft, but in general, this function has the shape of an upwards-facing parabola as shown in Figure 1a.

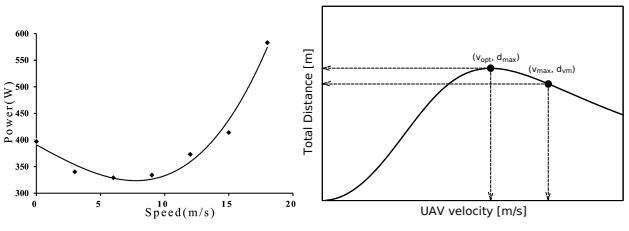
However, many UAV applications depend more on the total distance that a UAV can travel as opposed to just the amount of energy consumed. The relationship between speed and the total distance depends on the voltage that a battery supplies to the UAV and the rate that the battery discharges. We can represent this relationship as

$$\mathcal{D}(v) = \frac{B_{rate} V_{bat} v}{\mathcal{P}(v)}, \tag{4}$$

where $B_{rate}$ is the rate of battery discharge in amp-seconds and $V_{bat}$ is the voltage of the battery. In practice, $V_{bat}$ should be set lower than what the battery is rated for to ensure a safety buffer for uncertainty.

Equation (4) has the following general form (Fig. 1b):

$$\mathcal{D}(v) = \frac{v}{v^3 + v^2 + \frac{1}{v}}. \tag{5}$$




(a) Plot from Ref. [23] (b) Plot of Eqn. 5

**Figure 1: Relationship between UAV power consumption, travel distance, and speed.**

Intuitively, to minimize total mission time, we will want to maximize the UAV's speed. It was found [27] and then verified experimentally [23] that in order to achieve maximum traveling distance, the UAV must travel at a lower speed than its maximum possible speed. Let $d_{max}$ be this maximum achievable distance and $v_{opt}$ be the speed that achieves $d_{max}$. Let $v_{max}$ be the maximum speed that the UAV is capable of traveling and $d_{vm}$ be the distance that the UAV can travel when moving at $v_{max}$. As shown in Figure 1b, if $v_{opt} < v_{max}$, then $d_{max} > d_{vm}$.

Inspired by this finding, we formulate a function $v(d)$ that takes a distance and gives us the maximum speed that a UAV can travel to achieve this distance.

$$v(d) = \begin{cases} v_{max} & \text{if } d \leq d_{vm} \\ \mathcal{D}^{-1}(d) & \text{if } d_{vm} < d \leq d_{max} \\ infeasible & \text{if } d > d_{max} \end{cases} \tag{6}$$

where $\mathcal{D}^{-1}(d)$ is the inverse of Eqn. (4). Note that in the third case (i.e., if $d > d_{max}$) the UAV is not capable of actually traveling distance $d$. This is the adaptive speed we use in our approaches.

If given realistic parameters for a UAV, we could determine $\mathcal{D}(v)$. However, $\mathcal{D}^{-1}(d)$ is not easy to work with. Therefore, we instead propose approximating Eqn. (4) between $d_{vm}$ and $d_{max}$ as a second order polynomial then finding the inverse the polynomial to approximate Eqn. (6). The inverse of such a polynomial will have the form

$$v(d) = \frac{\sqrt{c_1 + c_2 d}}{c_3} + c_4 \tag{7}$$

where $c_1$, $c_2$, $c_3$ and $c_4$ will be constants.

## 5 OUR SOLUTION FRAMEWORK

In this section, we summarize our framework for finding an approximate solution to the MT-OTM problem. Our framework simplifies the problem by first fixing $m$ (the number of sub-tours), estimating the time required to complete each sub-tour, then treating the problem as a multi-Hamiltonian paths problem.

**Algorithm 1:** MT-OTM Solver

**Input:** $P$ : set of waypoints to visit, $p_b(t)$ : ground vehicle position function
**Output:** $\Delta^o$ : optimal set of routes, $S^o$ : optimal set of UAV speeds

**function** *MT-OTM-Solver*$(P, p_b(t))$
  $\Delta^o \leftarrow \emptyset,\ S^o \leftarrow \emptyset,\ m \leftarrow 0$
  **do**
    $m \leftarrow m + 1$
    $\Delta_m, S_m \leftarrow$ *path-planning*$(P, p_b(t), m)$
    **if** $t(\Delta_m, S_m) < t(\Delta^o, S^o)$ **then**
      $\Delta^o \leftarrow \Delta_m, S^o \leftarrow S_m$
    **end**
  **while** $t(\Delta_m, S_m) \not\leq t(\Delta_{m-1}, S_{m-1})$ ***and*** $m < |P|$;
  **return** $\Delta^o, S^o$
**end**

Algorithm 1 depicts our approach for determining a value for $m$. We do this by setting $m = 1$, solving for $\Delta_m$ and $S_m$ using function *path-planning()*, then incrementing $m$ until we do not see an improvement in mission completion time, $t(\Delta_m, S_m)$. The problem becomes writing a function *path-planning()* that can find a $\Delta_m$ and $S_m$ that can minimize total mission time for any $m$.

Algorithm 2 describes our proposed *path-planning* function. We start by guessing at the total time, $t$, that it will take to visit the waypoints in $P$ based on $p_b(t)$ and $m$. Vector $A$ represents the time required for each of the $m$ sub-tours. In the while loop, we set our guess for the total time and sub-tour times to $t'$ and $A'$, respectively. We then form graph $G'_m$ using $P$, $p_b(t)$, $m$, and $A$ in function *form-graph*(), which determines the locations of all $p^d_k$ and $p^t_k$ (the depots and terminals of each sub-tour) using $A$. This graph can be described mathematically as $G'_m = (V_m, E_m)$, where $V_m = P \cup \{p^d_1, \cdots, p^d_m\} \cup \{p^t_1, \cdots, p^t_m\}$ and

$$\begin{aligned} E_m = &\{(p_i, p_j) \mid 1 \le i, j \le n \text{ and } i \ne j\} \\ &\cup \{(p^d_k, p_i) \mid 1 \le k \le m \text{ and } 1 \le i \le n\} \\ &\cup \{(p_i, p^t_k) \mid 1 \le i \le n \text{ and } 1 \le k \le m\}. \end{aligned}$$

Once we find $G'_m$ the problem becomes solving an underlying Hamiltonian Path problem (function *solve-HP()*). We set $\Delta$ and $S$ as the found set of sub-tours and set of assigned speeds, respectively, then update the value of $t$ based on the actual total mission time to run all sub-tours in $\Delta$ given $S$. We update $A$ using function *sub-tour-times*(), which updates each sub-tour $k$'s entry in $A$ based on the time required to travel sub-tour $\delta_k$ while moving at speed $s_k$.

The loop continues until one of the following conditions is met:

(1) The predicted mission time, $t$, is within some epsilon of the actual time, $t'$, and every entry of our predicted vector $A'$ is within some epsilon of the actual vector $A$, or
(2) Some iteration limit has been met.

We then return the last found graph $G'$ and the corresponding set of sub-tours $\Delta$. If an iteration time-out condition occurs, we propose updating $G'$ based on $t$ and $A$ without changing the found set of tours $\Delta$.

To make an initial guess for $t$, we find the minimum spanning forest of $m$ trees on $P \cup \{p^d_1\}$, determine the minimum time that a UAV needs to fly the total distance of the forest while making $m-1$ stops to swap batteries. We can use this time to predict where the base station will be located for each of the $m-1$ stops. Using this intermediate guess, we can create a $G'_m$, find a new minimum forest of $m$ trees in $G'_m$ then use the distance of this new forest to get a guess on the time required to complete the search.

In the following section we further describe the underlying Hamiltonian path problem and show how to find an optimal solution for it.

## 6 FINDING FIXED HAMILTONIAN PATHS

Our solution framework reduces the MT-OTM problem into an underlying fixed multi-depot, multi-terminal Hamiltonian path problem (fixed-MdMtHPP). This is a special case of the more general MdMtHPP, which is formally defined as: "Given $m$ salesmen that start from distinct depots, $m$ terminals and $n$ destinations, the problem is to choose paths for each of the salesmen so that (1) each salesman starts at his respective depot, visits at least one destination and reaches any one of the terminals not visited by other salesmen, (2) each destination is visited exactly once, and (3) the cost of the paths is minimum among all possible paths for the salesmen" [3]. fixed-MdMtHPP differs from the traditional MdMtHPP in that the depots and terminals are fixed. Each tour that starts at some depot ($p^d_k$) must end at a specific terminal (the corresponding $p^t_k$).

MdMtHPP is NP-Hard ([3]). Fixing the matching between depots and terminals does not make the problem easier to solve.

**Algorithm 2:** Path Planning

**Input:** $P$ : set of waypoints to visit, $p_b(t)$ : ground vehicle position function, $m$ : number of sub-tours
**Output:** $\Delta$ : set of routes, $S$ : set of UAV speeds

```
function path-planning(P, p_b(t), m)
    t ← guess-time(P, p_b(t), m)
    A ← {t/m, ··· , t/m}
    do
        t' ← t, A' ← A
        G'_m ← form-graph(P, p_b(t), m, A')
        Δ, S ← solve-HP(G'_m)
        t ← t(Δ, S)
        A ← sub-tour-times(Δ, S)
    while (|t − t'| ≥ ε_t or |max_i(A − A')| ≥ ε_A) and iterations < iteration-limit;
    return Δ, S
end
```

Theorem 6.1. *The fixed multiple depot, multiple terminal, Hamiltonian path problem (fixed-MdMtHPP) is NP-Hard.*

Before beginning the proof for Theorem 6.1, we would like to remind the reader of the fixed destination multi-salesmen, multi-depot Traveling Salesman Problem (MmTSP). In MmTSP, there is a given set of depots with one or more salesmen and a set of destination vertices that must be visited by exactly one salesman. In fixed-destination MmTSP, each salesman must end their tour at the vertex they started at, which is known to be NP-Hard.

Proof. We form a reduction from fixed-destination MmTSP to fixed-MdMtHPP as follows. Let $G$ be the graph for the fixed-destination MmTSP problem. Form a new graph $G'$ by taking the set of destination vertices and depots from $G$. If any depot has more than one salesman that starts at it, then form a new depot at this same location and assign the additional salesman to this depot. For every salesman, create a terminal vertex that lies on top of the salesman's starting vertex. Solve the fixed-MdMtHPP that we have just formed. The routes from our solution to fixed-MdMtHPP will also be a solution to the fixed-destination MmTSP.

This reduction can be performed in linear time in terms of the number of vertices in $G$. As we have found a polynomial time reduction from fixed-destination MmTSP to fixed-MdMtHPP, and we know that fixed-destination MmTSP is NP-Hard, then fixed-MdMtHPP must also be NP-Hard. □

### 6.1 MINLP Formulation for fixed-MdMtHPP

In this sub-section we formulate a MINLP to solve the underlying fixed-MdMtHPP for our UAV path planning problem. We also show how a simplified version of the formulation can be used to solve the general case of the fixed-MdMtHPP.

Our MINLP formulation minimizes mission completion time by jointly minimizing the distance of each sub-tour and maximizing the speed that the UAV travels on each sub-tour. Due to the relationship between travel distance and speed discussed in Section 4, we cannot simply use a fixed UAV speed but must adapt the speed based on distance, which is why we optimize these two jointly. We use a variation of the Miller-Tucker-Zemlin formulation [7, 16] for the capacitated vehicle routing problem because it provides us the flexibility to force fixed depots and terminals. Although we cannot guarantee an optimal solution to the MT-OTM problem, our MINLP can find an optimal solution to the underlying fixed-MdMtHPP.

For the set of waypoints $i, j \in P$ and sub-tour $k$ in the set of tours $K$, let $X_{ijk}$ be a binary decision variable that determines if edge $(i,\ j)$ is included in tour $k$. We denote the Euclidean distance between $i$ and $j$ as $d_{ij}$. We use binary variables $Y_{ki}$ and $Z_{jk}$ to connect each depot $k$ to some waypoint $i$, and some waypoint $j$ to some terminal $k$, respectively. Let $L_k$ and $S_k$ be continuous variables for the total distance that the UAV must fly on sub-tour $k$ and the constant speed of the UAV on $k$, respectively. To prevent cycles within each sub-tour, we use the integer variable $U_i$ to give ordering assignments to each waypoint.

Our MINLP formulation of the problem is as follows.

$$\min \sum_{k \in K} \frac{L_k}{S_k} \tag{8}$$

subject to:

$$L_k = \sum_{i \in P} \sum_{j \in P} d_{ij} X_{ijk} + \sum_{i \in P} d_{ki} Y_{ki} + \sum_{j \in P} d_{jk} Z_{jk}, \qquad \forall_{k \in K} \tag{9}$$

$$S_k \leq \frac{\sqrt{c_1 - c_2 L_k}}{c_3} + c_4, \qquad \forall_{k \in K} \tag{10}$$

$$1 - \mathcal{M}(1 - X_{ijk}) \leq U_j - U_i \leq 1 + \mathcal{M}(1 - X_{ijk}), \qquad \forall_{i,j \in P}, \forall_{k \in K} \tag{11}$$

$$1 \leq U_i - U_k^d Y_{ki} \leq 1 + \mathcal{M}(1 - Y_{ki}), \qquad \forall_{i \in P} \forall_{k \in K} \tag{12}$$

$$\sum_{j \in P} X_{j,i,k} + Y_{k,i} = \sum_{j \in P} X_{i,j,k} + Z_{i,k}, \qquad \forall_{i \in P} \forall_{k \in K} \tag{13}$$

$$\sum_{j \in P} \sum_{k \in K} X_{j,i,k} + \sum_{k \in K} Y_{k,i} + \sum_{j \in P} \sum_{k \in K} X_{i,j,k} + \sum_{k \in K} Z_{i,k} = 2, \qquad \forall_{i \in P} \tag{14}$$

$$\sum_{i \in P} Y_{k,i} = 1, \qquad \forall_{k \in K} \tag{15}$$

$$\sum_{i \in P} Z_{i,k} = 1, \qquad \forall_{k \in K} \tag{16}$$

$$S_k \leq v_{max}, \qquad \forall_{k \in K} \tag{17}$$

$$L_k \leq d_{max}, \qquad \forall_{k \in K} \tag{18}$$

Our objective function (8) is minimizing the total time required for the UAV to travel all sub-tours. Note that we can remove possible domain violations in the objective using an additional auxiliary variable, $A_k$, to separate $L_k$ and $S_k$ as $L_k = A_k S_k$ then minimizing $A_k$. We forego adding additional variables here for brevity. Constraint (9) forces variable $L_k$ to equal the distance of sub-tour $k$. Constraint (10) adapts a sub-tour speed based on the distance of sub-tour $k$ and is derived from Eqn. (7).

Constraint (11) enforces a tight numbering scheme for consecutive waypoints in each sub-tour where $\mathcal{M}$ is some sufficiently large number which will be further discussed shortly. Constraint (12) forces the waypoint after depot $k$ to be assigned a sequence number of $U_k^d + 1$, where $U_k^d$ is an implied, fixed sequence number assigned to depot $k$ that remains constant. For each waypoint $i$ on sub-tour $k$, we want to assign a sequence number to $U_i$ that is within a designated range to force waypoints that are on the same sub-tour to be numbered together. We define the bounds of this numbering range for sub-tour $k$ using implied depot and terminal sequence numbers $U_k^d$ and $U_k^t$, respectively. If $l_m$ is the maximum number of waypoints that can be assigned to a sub-tour then we want to have $l_m$ sequence numbers available between $U_k^d$ and $U_k^t$. When given $n$ waypoints to visit on $m$ sub-tours with at least one waypoint on each sub-tour, by the pigeon hole principle, $l_m = n - m + 1$. For the first sub-tour, if we set $U_1^d = 0$ then $U_1^t$ must be $l_m + 1$. This makes $U_2^d = l_m + 2$. Following this trend, we find that for any sub-tour $k$, $U_k^d = (l_m + 2)(k-1)$ and $U_k^t = k(l_m + 1) + (k-1)$. In constraints (11) and (12) we want a value for $\mathcal{M}$ that is large enough to allow for all feasible sequence number assignments for each $U_i$. To keep a tight bound on our constraints, we set $\mathcal{M} = U_m^t = ml_m + 2m - 1$.

Constraint (13) ensures that the in-degree is equal to the out-degree of each waypoint while constraint (14) forces each waypoint to have a degree of two. Constraint (15) and (16) force every depot and terminal to be used, respectively. Finally, constraints (17) and (18) bound the maximum allowable speed of the UAV and maximum allowable distance of any sub-tour, respectively.

The numbering scheme described above is what allows us to fix the depots with their corresponding terminals. We are also ensuring that each sub-tour contains at least one waypoint by not defining an edge from depot $k$ to terminal $k$ in our formulation.

We can modify our MINLP formulation to solve the general case of fixed-MdMtHPP by making the objective function to be

$$\min \sum_{k \in K} L_k \tag{19}$$

and removing constraints (10), (17) and (18). The general fixed-MdMtHPP formulation avoids the additional complexity of variable multiplication seen in our MINLP formulation but does not adapt UAV speed and is not directly optimizing mission completion time.

### 6.2 $k$-IP Approach for fixed-MdMtHPP

Mixed-Integer Non-Linear Programs are often very hard to solve, even for commercial solvers on high-performance computers, so in

this sub-section we propose a more tractable approach that combines a heuristics-based k-means clustering algorithm and an Integer Program ($k$-IP). The general concept for $k$-IP is to partition the waypoints into $m$ groups then form sub-tours by solving a traveling salesman problem (TSP).

To partition the waypoints, we use Lloyd's algorithm to form $k$-means clusters [11]. We use the centroid of each $p_k^d$ and $p_k^t$ pair as the initial cluster centroids. We then limit the number of iterations that the algorithm runs which prevents the cluster centroids from migrating too far away from their corresponding $p_k^d$ and $p_k^t$ pairs.

Once we have put each waypoint into a cluster, we combine the clusters with the corresponding $p_k^d$ and $p_k^t$. We then solve a TSP on the resulting graph and force the edge connecting $p_k^d$ and $p_k^t$ to be part of the solution. To do this we use the IP formulation found in [21] that uses sub-cycle cuts to enforce closed tours. To avoid an exponential number of sub-cycle cuts, we treat these as lazy constraints where the constraint is only added to the solver when a found solution would break the constraint, a feature available in many commercially available solvers such as Gurobi.

To adapt sub-tour speeds for the $k$-IP approach, we first solve the problem as described above then assign a speed to each sub-tour using the distance of the found sub-tour and Eqn. (6).

## 7 SIMULATION EVALUATION

In this section we discuss our evaluation of our framework for solving the MT-OTM Problem in simulation using parameters from previous field testing and commercially available hardware. Our simulations were conducted on a machine with an Intel 3.4 GHz 16-Core CPU and 64 GiB of RAM. We use Gurobi Optimizer version 9.5.1 for our optimization solver. Our solution framework is implemented in C++ and provided as open-source[1].

### 7.1 Realistic Energy Model from Field Tests

In Section 4 we presented $\mathcal{P}(v)$, a theoretical equation that represents the power consumed by a UAV based on speed $v$. $\mathcal{P}(v)$ was verified through field testing in [23]. For their specific UAV, $\mathcal{P}(v)$ can be approximated as

$$\mathcal{P}(v) = 0.07v^3 + 0.0391v^2 - 13.196v + 390.95 \tag{20}$$

If we equip the UAV with a commercially available LiPo battery rated at 2,200 mAh, 12.6 volt, then we can plug Eqn. (20) into Eqn. (4) and get

$$d(v) = \frac{99,792v}{0.07v^3 + 0.0391v^2 - 13.196v + 390.95} \tag{21}$$

We can use Eqn. (21) to determine $d_{max}$, the maximum distance that the UAV can travel. By approximating the curve of Eqn. (21) between $d_{vm}$ and $d_{max}$ as a polynomial, we can find an approximation for $\mathcal{D}^{-1}$ between $v_{opt}$ and $v_{max}$ in the form of Eqn. (4).

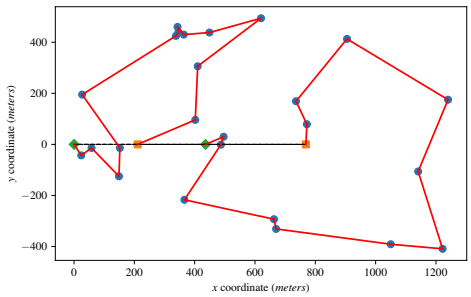
### 7.2 Baseline Approach

For a baseline, we adapt the heuristics-based approach found in [13]. This approach finds a TSP tour on the entire set of waypoints using the LKH heuristic [6] then has the UAV follow this tour until it runs out of energy. We refer to this as *tour-splitting* (TS). We chose this approach because it avoids repeatedly solve for Hamiltonian paths, allows us to adapt UAV speeds to sub-tour distances, and has been proposed for similar problems in recent literature [1, 13].

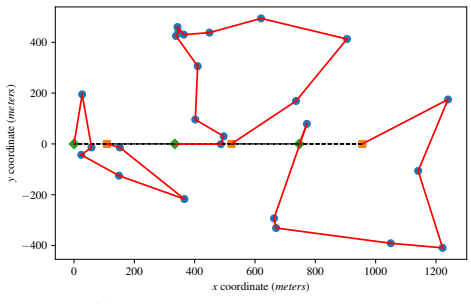
This approach as proposed in [13] solves for a cycle while we wish for the UAV to generally follow the path of the ground vehicle. This requires us to slightly adapt the original approach instead of using it directly. To adapt this approach, we find a minimum distance Hamiltonian path using the LKH heuristic that starts from the waypoint closest to vehicle's starting point and ends where we predict the ground vehicle to be at the end of the entire data-collection mission, as described in Section 5. We then divide this total path into $m$ roughly equal segments. For each segment, we form a sub-tour by determining where the ground vehicle will be at for the beginning of the sub-tour based on the previous sub-tour. We then iteratively approximate the maximum possible UAV speed allowed for the sub-tour using Eqn. (6) and determine a corresponding sub-tour terminating location until we settle on a consistent solution. The approach is plugged in to algorithm 1 as the *path-planning()* function.

### 7.3 Comparison of Approaches

We ran all three approaches on randomized graphs with $n$ ranging from 5 up to 80 at increments of 5. We generated 50 graphs at each increment. To keep the results comparable across each input graph,




**(a) UAV paths generated using MINLP**




**(b) UAV paths generated using $k$-IP**

**Figure 2: Sample UAV paths generated using our approaches. The blue circles are the navigational waypoints, the dashed, black line is the ground vehicle's trajectory, the green diamonds are UAV launch points and the orange squares are UAV receiving points along the ground vehicle's path.**

[1] github.com/pervasive-computing-systems-group/MT-OTM-Solver

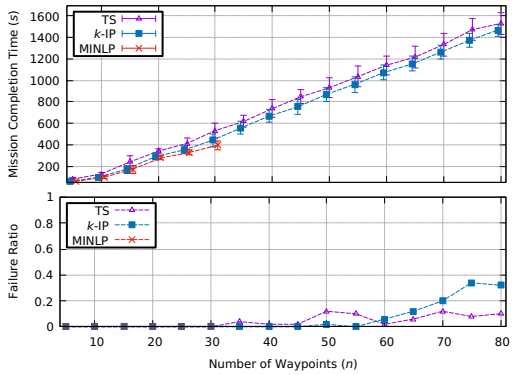



**Figure 3: Simulation results for the three considered approaches on randomized graphs with $n$ up to 80. Top plot shows impact of number of waypoints on mission completion time. Bottom plot shows the ratio of graphs where each approach failed to find a solution.**

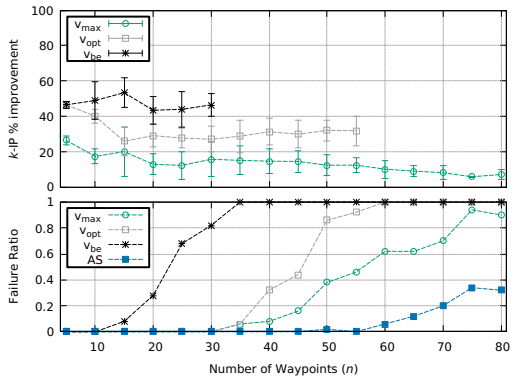



**Figure 4: Simulation results for adaptive speed compared to fixed speeds on randomized graphs with $n$ up to 80. Top plot shows impact of number of waypoints on mission completion time. Bottom plot shows the ratio of graphs where each approach failed to find a solution.**

we had the ground vehicle move at a fixed speed of 2.5 $m/s$ along the $x$-axis.

Figure 2a shows the output of the MINLP approach and Figure 2b shows the output of the $k$-IP approach on one of the randomly generated graphs with 25 waypoints. The MINLP approach found a superior solution with a mission completion time of 307.5 $s$ while the $k$-IP approach found a solution with a completion time of 382.4 $s$. However, the $k$-IP only took 0.038 $s$ to compute this solution while using the MINLP formulation took 130.9 $s$ to compute a solution. In the following sections we further analyze this trade-off between performance (mission completion time) and computation time.

## 7.4 Mission Completion time Evaluation

Figure 3 (*top*) shows how the number of waypoints affects average mission completion time. The error bars show the standard deviation for each approach. We stopped the MINLP at $n = 30$ due to long computation times, which are further discussed below. The TS and $k$-IP approaches were not always able to find a feasible solution and their failure rates are also documented below.

On average, the MINLP and $k$-IP approaches provide a 23.8% and 14.5% improvement over the TS approach, respectively. $k$-IP's performance is not as good as the MINLP solution but only averages a 3.8% increase in mission completion time over the MINLP approach and provides a nice alternative to the MINLP in larger sized problems.

We acknowledge this as a limitation in this work and observe how well each approach can find a solution. Figure 3(*bottom*) shows the ratio of graphs where each approach failed to find a solution. The $k$-IP approach failed to find a valid solution for 6.6% of the graphs while the TS approach failed at 4.3% of the graphs. This suggests that although $k$-IP outperforms TS in mission completion time, TS may be able to find more solutions than $k$-IP. The MINLP approach was able to find a solution of all graphs up to the cut-off point ($n = 30$).

It can be particularly hard to find solutions for some combinations of waypoints and ground vehicle paths. In our evaluation, we observed that the $k$-IP approach struggles on inputs with subregions with a high concentration of waypoints while the rest of the graph is sparse. On these graphs, the clustering algorithm forms a single cluster with too many waypoints while the rest of the clusters are fairly small creating many short sub-tours with one very long tour that cannot be completed in a single flight.

## 7.5 Impact of Speed Evaluation

We also evaluated how our approach to adaptive speed affected mission completion time. Using the $k$-IP approach, we compared how well fixing the UAVs speed at $v_{max}$, $v_{opt}$, and the speed that minimizes energy consumption (termed best endurance, or $v_{be}$) performed against our adaptive speed approach (AS). We chose these different speed settings because they have all been proposed for UAV path planning problems in recent literature [17, 19, 24].

Figure 4 shows the results of comparing our adaptive speed approach against using a fixed-speed. On average, the adaptive speed approach improved the mission completion by 11.9%, 31.9%, and 47.1% compared against fixing the velocity at $v_{max}$, $v_{opt}$, and $v_{be}$, respectively. With speed fixed at $v_{max}$, $v_{opt}$, and $v_{be}$ the solver only found solutions for 69.3%, 52.5%, and 25.9% of the graphs, respectively, while using an adaptive speed approach found a solution for 93.4% of the inputs. In fact, when fixing speed at $v_{be}$ no graphs were solved with $n \geq 35$.

These results show that our proposed UAV speed adaptation greatly outperforms approaches where the UAV's speed is kept fixed. These results also show the impact that UAV speed has on UAV path planning problems and the necessity for considering speed adaptation for UAV path planning.

## 7.6 Computational Efficiency

Table 1 shows the minimum, average, and maximum computation times required to solve the randomized graph set for the three approaches. This data only includes the results from graphs that each approach was able to find a feasible solution on.

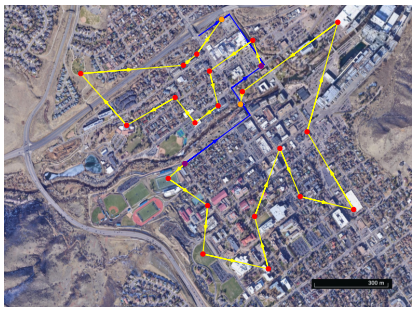


**Figure 5: UAV path generated using MINLP approach for a case study in an urban environment.**

The table shows that the computational time of the MINLP increases exponentially with the number of waypoints and is not an ideal approach for larger graphs. Both of the TS and $k$-IP approaches solve the MT-OTM problem very quickly, with the TS approach generally outperforming $k$-IP. However, we showed above that the $k$-IP approach greatly outperforms the TS approach in mission completion time.

**Table 1: Computation Times (in *seconds*)**

| | $n$ = | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 |
|---|---|---|---|---|---|---|---|---|---|
| TS | Avg. | 0 | 0 | 0 | 0 | 0.001 | 0.002 | 0.004 | **0.007** |
| | 95% CI | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.001 |
| $k$-IP | Avg. | 0.01 | 0.03 | 0.06 | 0.11 | 0.20 | 0.32 | 0.46 | **0.74** |
| | 95% CI | 0.001 | 0.005 | 0.008 | 0.011 | 0.024 | 0.031 | 0.066 | 0.090 |
| MP | Avg. | 0.73 | 135 | **5139** | - | - | - | - | - |
| | 95% CI | 0.151 | 48.34 | 3215 | - | - | - | - | - |

Average computation time and 95% confidence interval (95% CI) for the *tour-splitting* (TS), $k$-IP, and MINLP (MP) approaches with varying number of waypoints ($n$).

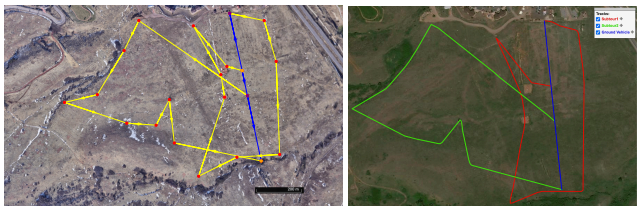
## 7.7 Urban Environment Simulation

To demonstrate how our MT-OTM solution framework can be used in real-world scenarios, we apply it to an example in an urban environment where the UAV must visit a set of waypoints in a city while the ground vehicle follows a set route on city streets, i.e., not a straight line as in the previous simulation settings. We selected 20 waypoints for the UAV to visit in the urban environment. We selected these point using Google Earth, then converted the GPS coordinates into relative distances and generated a graph for our MT-OTM solution framework to solve. The ground vehicle follows a series of city streets, moving at a constant 3 $\frac{m}{s}$.

Figure 5 shows the UAV path generated using the MINLP approach. It took the MINLP approach 43.1 seconds to find this solution. We argue that this demonstrates that for most common scenarios the number of waypoints will be small enough that the MINLP approach can be used to find superior solutions. The solution found using the MINLP has a mission completion time of 355.8 seconds while the $k$-IP finds a solution with 373.1 seconds.

## 8 FIELD TEST

To further validate our solution framework we created a field prototype of the MT-OTM Problem using our own physical UAV testbed



**(a) UAV paths generated using $k$-IP** **(b) GPS trace of UAV path**

**Figure 6: Field prototype results. (a) UAV path generated using $k$-IP approach. (b) UAV's GPS trace while following the path found offline.**

[5]. We selected 18 waypoints in an empty field using Google Earth with the ground vehicle moving in a straight line across the field at 2.5 $\frac{m}{s}$. For simplicity, we landed the UAV manually at the end of each sub-tour and substituted the ground vehicle by walking the path of the vehicle on foot. Because the UAVs in [5] can travel up to 8 $km$, we shortened the max flying distance to 1.7 $km$ and set a max velocity of 11 $\frac{m}{s}$ to scale down the physical prototype.

Figure 6a shows the UAV path found using the $k$-IP approach. Figure 6b shows the GPS trace of the UAV while following the found paths. The result on the physical prototype demonstrates that our approach works well when the MT-OTM problem is applied in real world scenarios.

## 9 CONCLUSIONS

In this paper, we formulated the Minimum-Time while On-The-Move (MT-OTM) problem and presented an algorithm that solves the MT-OTM problem by boiling the problem down into an underlying fixed Multi-Depot, Multi-Terminal Hamiltonian Path Problem (fixed-MdMtHPP). We developed two approaches for solving fixed-MdMtHPP, a Mixed-integer nonlinear program (MINLP) that optimizes UAV speeds and a $k$-means clustering algorithm paired with an Integer Program ($k$-IP approach).

Our field-test-based simulation results show that on graphs that have 30 waypoints or less our MINLP finds superior mission completion time results while maintaining reasonable computation times. For larger graphs we recommend our $k$-IP approach, which gives good mission completion time results while reducing computation time. Our simulation results also show that UAV speed adaptation can reduce mission completion time and should be considered when planning paths for UAVs. We also demonstrated how our solution framework can still be used when the MT-OTM problem is applied to new scenarios such as urban environments. We further validated our solution in a case study of the MT-OTM problem on a physical UAV testbed. Our case study demonstrates how the MT-OTM problem can be used in real-world scenarios.

For future work, we will expand our energy model to include other factors such as making turns and the influence of wind. We will also consider the multi-UAV version of the MT-OTM problem and look at how to determine where along the ground vehicle's path the UAVs should start being launched.

## REFERENCES


[1] Divansh Arora, Parikshit Maini, Pedro Pinacho-Davidson, and Christian Blum. 2019. Route planning for cooperative air-ground robots with fuel constraints: an approach based on CMSA. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 207–214.

[2] Tomas Baca, Petr Stepan, Vojtech Spurny, Daniel Hert, Robert Penicka, Martin Saska, Justin Thomas, Giuseppe Loianno, and Vijay Kumar. 2019. Autonomous landing on a moving vehicle with an unmanned aerial vehicle. *Journal of Field Robotics* 36, 5 (2019), 874–891.

[3] Jungyun Bae and Sivakumar Rathinam. 2012. Approximation algorithms for multiple terminal, hamiltonian path problems. *Optimization Letters* 6, 1 (2012), 69–85.

[4] Shushman Choudhury, Kiril Solovey, Mykel Kochenderfer, and Marco Pavone. 2021. Coordinated Multi-Agent Pathfinding for Drones and Trucks over Road Networks. *arXiv preprint arXiv:2110.08802* (2021).

[5] Jonathan Diller, Peter Hall, Corey Schanker, Kristen Ung, Philip Belous, Peter Russell, and Qi Han. 2022. ICCSwarm: A Framework for Integrated Communication and Control in UAV Swarms. In *Proceedings of the Eighth Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*. ACM, 1–6.

[6] Keld Helsgaun. 2017. An extension of the Lin-Kernighan-Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems. *Roskilde: Roskilde University* (2017), 24–50.

[7] Imdat Kara, Gilbert Laporte, and Tolga Bektas. 2004. A note on the lifted Miller–Tucker–Zemlin subtour elimination constraints for the capacitated vehicle routing problem. *European Journal of Operational Research* 158, 3 (2004), 793–795.

[8] Songhua Li, Minming Li, Lingjie Duan, and Victor Lee. 2021. Online ride-hitching in UAV travelling. In *International Computing and Combinatorics Conference*. Springer, 565–576.

[9] Chi Harold Liu, Chengzhe Piao, and Jian Tang. 2020. Energy-efficient UAV crowdsensing with multiple charging stations by deep learning. In *INFOCOM - Conference on Computer Communications*. IEEE, 199–208.

[10] Zhilong Liu, Raja Sengupta, and Alex Kurzhanskiy. 2017. A power consumption model for multi-rotor small unmanned aircraft systems. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 310–315.

[11] Stuart Lloyd. 1982. Least squares quantization in PCM. *IEEE transactions on information theory* 28, 2 (1982), 129–137.

[12] Parikshit Maini and PB Sujit. 2015. On cooperation between a fuel constrained UAV and a refueling UGV for large scale mapping applications. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 1370–1377.

[13] Parikshit Maini, Kaarthik Sundar, Mandeep Singh, Sivakumar Rathinam, and PB Sujit. 2019. Cooperative aerial–ground vehicle route planning with fuel constraints for coverage applications. *Transactions on Aerospace and Electronic Systems* 55, 6 (2019), 3016–3028.

[14] Neil Mathew, Stephen L Smith, and Steven L Waslander. 2015. Multirobot rendezvous planning for recharging in persistent tasks. *Transactions on Robotics* 31, 1 (2015), 128–142.

[15] Neil Mathew, Stephen L Smith, and Steven L Waslander. 2015. Planning paths for package delivery in heterogeneous multirobot teams. *Transactions on Automation Science and Engineering* 12, 4 (2015), 1298–1308.

[16] Clair E Miller, Albert W Tucker, and Richard A Zemlin. 1960. Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)* 7, 4 (1960), 326–329.

[17] Chase C Murray and Ritwik Raj. 2020. The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. *Transportation Research Part C: Emerging Technologies* 110 (2020), 368–398.

[18] Charles E Noon and James C Bean. 1993. An efficient transformation of the generalized traveling salesman problem. *INFOR: Information Systems and Operational Research* 31, 1 (1993), 39–44.

[19] Stefan Poikonen and Bruce Golden. 2020. Multi-visit drone routing problem. *Computers & Operations Research* 113 (2020), 104802.

[20] Ritwik Raj and Chase Murray. 2020. The multiple flying sidekicks traveling salesman problem with variable drone speeds. *Transportation Research Part C: Emerging Technologies* 120 (2020), 102813.

[21] Ronald L Rardin. 1998. *Optimization in operations research*. Vol. 166. Prentice Hall Upper Saddle River, NJ.

[22] Jürgen Scherer and Bernhard Rinner. 2017. Short and full horizon motion planning for persistent multi-UAV surveillance with energy and communication constraints. In *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 230–235.

[23] Feng Shan, Junzhou Luo, Runqun Xiong, Wenjia Wu, and Jiashuo Li. 2020. Looking before Crossing: An Optimal Algorithm to Minimize UAV Energy by Speed Scheduling with a Practical Flight Energy Model. In *INFOCOM - Conference on Computer Communications*. IEEE, 1758–1767. https://doi.org/10.1109/INFOCOM41043.2020.9155376

[24] Guangyao Shi, Nare Karapetyan, Ahmad Bilal Asghar, Jean-Paul Reddinger, James Dotterweich, James Humann, and Pratap Tokekar. 2022. Risk-aware UAV-UGV Rendezvous with Chance-Constrained Markov Decision Process. In *Conference on Decision and Control (CDC)*. IEEE.

[25] Pratap Tokekar, Joshua Vander Hook, David Mulla, and Volkan Isler. 2016. Sensor planning for a symbiotic UAV and UGV system for precision agriculture. *Transactions on Robotics* 32, 6 (2016), 1498–1511.

[26] Kevin Yu, Ashish Kumar Budhiraja, Spencer Buebel, and Pratap Tokekar. 2019. Algorithms and experiments on routing of unmanned aerial vehicles with mobile recharging stations. *Journal of Field Robotics* 36, 3 (2019), 602–616.

[27] Yong Zeng, Jie Xu, and Rui Zhang. 2019. Energy minimization for wireless communication with rotary-wing UAV. *IEEE Transactions on Wireless Communications* 18, 4 (2019), 2329–2345.