Planning Paths for Package Delivery in Heterogeneous Multi-Robot Teams

Neil Mathew, Stephen L. Smith *Senior Member, IEEE,* Steven L. Waslander *Member, IEEE*

***Abstract*—This paper addresses the task scheduling and path planning problem for a team of cooperating vehicles performing autonomous deliveries in urban environments. The cooperating team comprises two vehicles with complementary capabilities, a truck restricted to travel along a street network, and a quadrotor micro-aerial vehicle of capacity one that can be deployed from the truck to perform deliveries. The problem is formulated as an optimal path planning problem on a graph and the goal is to find the shortest cooperative route enabling the quadrotor to deliver items at all requested locations. The problem is shown to be NP-hard. A solution is then proposed using a novel reduction to the Generalized Traveling Salesman Problem, for which well-established heuristic solvers exist. The heterogeneous delivery problem contains as a special case the problem of scheduling deliveries from multiple static warehouses. We propose two additional algorithms, based on enumeration and a reduction to the traveling salesman problem, for this special case. Simulation results compare the performance of the presented algorithms and demonstrate examples of delivery route computations over real urban street maps.**

Note to Practitioners—This work presents a viable approach to optimizing delivery routes for a coordinated team consisting of a small aerial vehicle and a large ground vehicle. The large ground vehicle provides long-range transport and support while the small aerial vehicle completes the final leg of each delivery, thereby reducing overall delivery time and fuel consumption. The approach exploits well known methods from algorithmic graph theory, to efficiently identify route savings that can be achieved with the team over conventional single truck delivery. Results are presented for multiple city environments, with clear improvements in delivery time and total distance travelled observed.

***Index Terms*—Optimal Path Planning, Generalized Traveling Salesman Problem, Urban Delivery, Unmanned Aerial Vehicles.**

I. Introduction

An emerging application for micro-aerial vehicles (MAVs) is in performing autonomous deliveries in urban environments. Indeed, preliminary studies of delivery costs per kilometer seem to indicate that such an application is not only possible, but may represent a significant savings for small packages [1]. As a result, a number of large retailers have recently announced plans to deploy MAVs for expedited small package deliveries. While MAVs have the potential to significantly enhance the speed of deliveries in urban environments as well as the distribution of supplies or aid in inaccessible regions, a number of practical issues such as safety, security and vehicle design still need to be addressed. Current MAV systems are limited by small payload capacities and short operating ranges that severely restrict the extent and efficiency of an autonomous delivery network. Further, current safety regulations usually restrict commercial drone flights to only within line-of-sight of an operator.

In this paper, we propose to overcome these limitations by introducing a heterogeneous delivery team of two cooperating vehicles: a *carrier* truck and a *carried* MAV, such as a quadrotor. The role of the truck is to carry a shipment of packages to be delivered, as well as a docked MAV, and the role of the MAV is to carry individual packages from the truck to specific delivery points in the environment. By requiring the MAV to perform only the last leg of the delivery, both range and line-of-sight limitations can be accommodated. This team architecture requires customized planning algorithms to efficiently exploit the complementary capabilities of the two vehicle types, and these algorithms are the focus of this work.

We assume that the MAV has a payload capacity of one package and hence must return to the truck after each delivery. We also assume that the truck is capable of recharging the MAV after each delivery and that it has an operating range sufficient for the entire delivery mission. We envision that recharging will be done via battery swap to minimize the delay between flights. These swaps could be done either by a human operator, or by using autonomous battery swapping mechanisms [2], [3].

The goal of this paper is to propose a framework to compute a minimum cost cooperative route enabling the MAV to visit each delivery point in the environment as efficiently as possible. To this end, we will abstract the problem on a graph and formulate the *Heterogeneous Delivery Problem* (HDP) as a discrete optimal path planning problem. Solutions consist of routes, computed for the truck and the quadrotor through the graph, that minimize the total cost of deliveries.

A. Contributions

The contributions of this paper are threefold. First, we formulate the HDP as a novel adaptation of a carrier-vehicle system in a discrete environment. Second, we prove NP-hardness of the HDP and present a solution based on an efficient reduction to the Generalized Traveling Salesman Problem (GTSP). The GTSP can then be solved using existing heuristic solvers. Finally, we examine a special case of the HDP consisting of a single vehicle and multiple static warehouses, called the *Multiple Warehouse Delivery Problem*

This research is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

N. Mathew and S. L. Waslander are with the Department of Mechanical and Mechatronics Engineering and S. L. Smith is with the Department of Electrical and Computer Engineering, all at the University of Waterloo, Waterloo ON, N2L 3G1 Canada (nmathew@uwaterloo.ca; stephen.smith@uwaterloo.ca; stevenw@uwaterloo.ca)

(MWDP). We present two algorithms for the MWDP, one, using an alternative transformation to the Traveling Salesman Problem (TSP) and the other, a polynomial time exact algorithm to compute an optimal delivery route.

A preliminary version of this paper appeared as the conference paper [4]. This paper extends the conference version with a more comprehensive review of related work, detailed proofs of all results, a modified reduction to the GTSP to enable computation of minimum-time delivery paths, and new simulation results to investigate the advantages of the proposed delivery team.

B. Related Work

The complementary capabilities of aerial and ground vehicles has led to many works that propose coordinated team operations. Cooperative control in heterogeneous multi-robot teams has been investigated for applications like search and rescue, surveillance, and exploration, target tracking and mapping [5], [6], [7], [8], where robots with complimentary capabilities must coordinate actions to accomplish a common goal. A recent survey [9] presents five separate categories of UAV/UGV coordination problems including docking, mapping, distributed target tracking, formation control and task assignment, of which task assignment [10] is the most relevant to this work.

One of the main challenges with heterogeneous robotic systems is the development of planning algorithms to complete tasks efficiently. Collaborative UAV-UGV teams have been proposed for persistent surveillance, where UAVs can occasionally rendezvous and dock with UGVs to benefit from the larger payload capacity and energy resources of UGVs [11], [12]. Similarly, UAV-UGV teams have been proposed for precision agriculture, where ground vehicle are said to mule the aerial vehicles to convenient access points around the fields from which to most quickly collect high-value plant health data [13].

The HDP can be categorized as a static Vehicle Routing Problem (VRP) with coordination constraints between the two vehicles, and many problems within this class have been studied [14], [15]. We do not consider dynamic or stochastic variants [16], as the team structure requires all delivery packages to be loaded into the ground vehicle in advance. As detailed in the survey [15], these problems are solved using a wide-variety of custom solver approaches, each tailored to the specific problem assumptions. Approaches include mixed integer programming, branch-and-bound, branch-and-price, metaheuristics, and column generation. Unlike the prior work, the HDP solution methods proposed in this work take advantage of the tightly coupled nature of the team motion over the graph to convert the problem to a single-vehicle generalized traveling salesman problem (GTSP), such that more efficient solvers can be employed. In particular, we able to directly apply highly-optimized TSP solvers [17] that can exploit the geometric structure of the problem rather than relying on general purpose IP solvers.

Because of this coupled nature, the HDP can also be considered a part of class of problems referred to as Carrier-Vehicle Traveling Salesman Problems (CV-TSP), extensively studied [18], [19], [20] in the context of a marine carrier and an aircraft visiting a set of locations to conduct a rescue mission in a planar environment. The CV-TSP is defined as a continuous optimization for which solutions can be computed using a sub-optimal heuristic to split the problem into two tractable subproblems: first, a TSP to compute the optimal visit order and second, a convex optimization to compute the specific deployment points for the team in Euclidean space. In [13], a similar class of TSP with neighbourhoods problems are studied for UAV/UGV teams collecting precision agriculture data.

In this work, by selecting a purely discrete formulation for the HDP, a single optimization can be employed that computes cooperative paths for both vehicles over large problem instances. We solve this optimizing through a novel reduction to the GTSP. The GTSP and its special case, the TSP, are both NP-hard [21]. A commonly-used method for solving the GTSP is the Noon-Bean Transformation, which reduces a GTSP to a Traveling Salesman Problem (TSP), for which a wide variety of algorithms can be applied. If the TSP edge lengths satisfy the triangle inequality, then there exist constant-factor approximation algorithms for the TSP. The best known is Christofides' algorithm [22], which gives a $3/2$-factor approximation in $O(n^3)$ time, where n is the number of vertices in the graph. In practice, methods based on Lin-Kernighan heuristic are very effective and have been shown empirically to obtain high-quality solutions in $O(n^{2.2})$ time [23]. The most widely used implementation is the Lin-Kernighan-Helsgaun (LKH) heuristic [17]. Exact solvers also exist for the TSP, such as the Concorde solver [23], which has solved problems on tens of thousands of vertices. The exact solvers, however, are not guaranteed to run in polynomial time. Finally, heuristic solvers have been developed specifically for the GTSP [24]. The GTSP appears in several different path planning problems in which multiple robot configurations can be used to complete a given task, including persistent surveillance [11], and heterogeneous or multi-vehicle TSPs [25].

Organization: The organization of the paper is as follows. Section II formulates the HDP as an optimal path planning problem in a discrete environment. In Section III, the HDP is proved to be NP-hard. Section IV presents the transformation to the GTSP implemented to solve the HDP. Section V presents two solutions for the MWDP and finally, Section VI compares and benchmarks all proposed algorithms in simulation.

C. Definitions and Nomenclature

A *graph* $G = (V, E, c)$, consists of a set of vertices V, a set of edges E, and a function $c : E \to \mathbb{R}$ that assigns a cost to each edge in E. In a *directed graph*, each edge is an ordered pair of vertices (v_i, v_j) and is assigned a direction from v_i to v_j. A *partitioned graph*, G, is a graph with a partition of its vertex set into ℓ disjoint sets $(V_1, \dots, V_\ell)$ where $\cup_{i=1}^{\ell} V_i = V$.

In this paper, we define a *route* in a graph to be any sequence of vertices $P = (v_1, \dots, v_k)$ linked by edges $(v_i, v_{i+1}), i = 1, \dots, k-1$. Following [26], a *walk* is a route such that no edge is traversed more than once. A *path* is a route where

$v_i \neq v_j$ for all $i, j \in \{1, \dots, k-1\}$. A *closed route* is a route in which $v_1 = v_k$. A *tour* is closed path that visits all vertices in V exactly once.

Given a complete graph $G = (V, E, c)$, the Traveling Salesman Problem (TSP) computes a minimum cost tour of G. Given a partitioned complete graph $G = (V, E, c)$, with a vertex partition $(V_1, \dots, V_\ell)$, the Generalized Traveling Salesman Problem (GTSP) computes a minimum cost closed path, P, that visits exactly one vertex in each vertex set $V_i \subset V$, $i \in \{1, \dots, \ell\}$.

II. The Heterogeneous Delivery Problem (HDP)

The Heterogeneous Delivery Problem (HDP) is defined on a directed graph G that represents the physical locations of the delivery points, the location of a warehouse and a set of drivable routes on a street network. An example HDP graph is shown in Figure 1. The graph G contains the locations of n delivery vertices, denoted by d_i, in set V_d (red vertices in Figure 1), m street vertices, denoted by w_i, in set V_w (blue vertices in Figure 1), and a warehouse vertex, w_0, where the truck and quadrotor are initially located. The m street vertices can be thought of as pre-selected locations at which the truck can safely stop to deploy the quadrotor. The vertices, edges and costs of G are defined as follows:

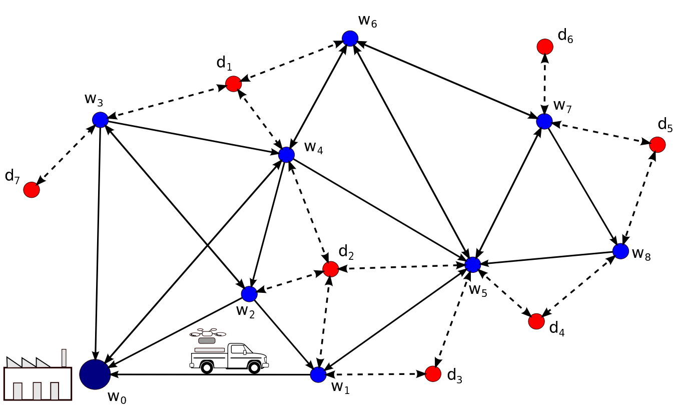


Fig. 1. The Heterogeneous Delivery Problem. The street edges (solid lines) are shown as either single or double arrows, that represent pairs of directed edges. All flight edges (dashed lines) are bidirectional edges between vertices.

Vertices: The vertex set V is defined as a union of three disjoint subsets $V = V_0 \cup V_w \cup V_d$ where $V_0 = \{w_0\}$, $|V_d| = n$, and $|V_w| = m$.

Edges: The edge set, E, is a union of two disjoint subsets, $E = E_w \cup E_d$. The set E_w contains a directed *street edge* of the form (w_i, w_j) for every pair of street vertices $w_i, w_j \in V_w$, and thus the induced subgraph over the vertex set V_w is complete. These edges represent the shortest routes between each pair of street vertices. The set E_d contains pairs of bidirectional *flight edges* of the form (w_i, d_j) and (d_j, w_i), for all $w_i \in V_w$ and $d_j \in V_d$, if w_i is a viable deployment vertex to reach delivery point d_j. These flight edges would have to be computed prior to the first deployment, taking into account the range and line-of-sight constraints. We define the set $W_{d_i} \subset V_w$ to be the set of viable deployment vertices for each delivery point, d_i.

Edge Costs: For full generality, we define three types of edge costs for the truck-quadrotor team. The reason for defining the three different costs will become more apparent in Section IV-C. A flight edge in E_d can be traversed only by a quadrotor between a street vertex, w_i, and a delivery vertex, d_j. A street edge in E_w may be traversed by the truck, either carrying the quadrotor or traveling alone. Thus we define a triple of costs $\mathcal{C} = (c_q, c_t, c_{tq})$ where $c_q : E_d \to \mathbb{R}_{\geq 0}$, assigns a quadrotor travel cost to flight edges in E_d, $c_t : E_w \to \mathbb{R}_{\geq 0}$ assigns a truck travel cost to street edges in E_w, and $c_{tq} : E_w \to \mathbb{R}_{\geq 0}$ assigns a docked truck-quadrotor travel cost on street edges in E_w. Since the street edges E_w represent shortest paths between street vertices, we assume that the edge costs c_t and c_{tq} each satisfy the triangle inequality.

We extend the definition of a graph from Section I-C to a *multi-weighted graph* $G = (V, E, \mathcal{C})$, where $\mathcal{C}$ is a triple of costs. We formulate the HDP on G as the problem of computing a route for the truck and a route for the quadrotor, such that the truck stops at a sequence of delivery points $w_i \in V_w$ at which the quadrotor can take-off, visit a delivery point $d_i \in V_d$ and return to the truck before the next deployment. The goal is for the quadrotor to visit all n delivery points and minimize the total delivery cost of the mission.

To make this definition more precise, let the quadrotor's route be a closed walk P_q along a sequence of unique edges $E_q \subset E$ and let the truck's route be a tour P_t, with a sequence of edges $E_t \subset E$. Routes P_q and P_t share vertices at which the truck and quadrotor meet and share edges during docked travel. The HDP can be formalized as follows.

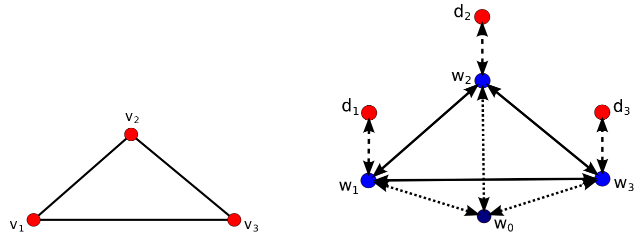
Problem II.1 (Heterogeneous Delivery Problem)**.** Given $G = (V, E, \mathcal{C})$, where $V = V_0 \cup V_w \cup V_d$, $E = E_d \cup E_w$ and $\mathcal{C} = (c_t, c_q, c_{tq})$, compute a closed walk P_q and a closed path P_t that start and end at w_0, such that (i) P_q visits each $d_i \in V_d$ exactly once; (ii) P_t is a sequence of deployment vertices that visits each unique $w_i \in P_q$ exactly once, and in the order defined by the first visit to each w_i in P_q; and (iii) The routes collectively minimize

$$\sum_{e \in E_q \setminus E_t} c_q(e) + \sum_{e \in E_t \setminus E_q} c_t(e) + \sum_{e \in E_q \cap E_t} c_{tq}(e). \qquad (1)$$

Remark II.2 (Objectives in the HDP). Objective (1) can be used to capture metrics such as fuel consumption. In this setting, the costs c_t and c_{tq} capture the fuel consumed by the truck, and would typically be equal. The cost c_q captures the fuel consumed by the quadrotor, and would likely be negligible when compared to the truck fuel consumption. However, the freedom of having the three costs c_t, c_{tq}, and c_t allows one to penalize certain aspects of the solution, or achieve certain desirable properties in the truck and quadrotor routes, and this will be investigated further in Section IV-C. In Section IV-D we discuss the minimum-time objective. □

III. Proof of NP-hardness

To prove NP-hardness of the HDP, we will show that (i) an instance of the TSP can be reduced to an instance of the HDP, and (ii) an optimal HDP solution provides an optimal TSP solution.



(a) $G' = (V', E', c')$ (b) $G = (V, E, \mathcal{C})$

Fig. 2. A reduction from the TSP on graph G' to the HDP on graph G.

Theorem III.1. *The Heterogeneous Delivery problem is NP-hard.*

Proof. Let $G' = (V', E', c')$, with $|V'| = n$, be an input to the TSP. To prove (i), we give a polynomial-time transformation of G' into an input $G = (V, E, \mathcal{C})$ of the HDP as shown in Figure 2.

The HDP is constructed such that each delivery vertex, $d_i \in V_d$ corresponds to a vertex $v_i \in V'$, and has only one unique viable street deployment vertex $w_j \in W_{d_i} \subset V_w$. Thus, construct the vertex set $V = V_0 \cup V_d \cup V_w$, where $|V_d| = n$, $|V_w| = n$ and V_0 contains an additional start vertex w_0.

Now for each edge (v_i, v_j) in E' with a cost $c'(v_i, v_j)$, add a sequence of directed edges to E, from d_i to d_j, given by $(d_i, w_i), (w_i, w_j), (w_j, d_j)$, denoting the feasible flight and street edges, and resulting in a total cost of $c_q(d_i, w_i) + c_{tq}(w_i, w_j) + c_q(w_j, d_j)$. Let $c_q(e) = 0$ for all flight edges $e = (d_i, w_i)$ or $e = (w_i, d_i)$. For all street edges, e, we set $c_t(e) = c_{tq}(e) = c'(e)$. Finally, add bidirectional edges from all $w_i \in V_w$ to w_0 and set $c(w_0, w_i) = 0$ and $c(w_i, w_0) = 0$. This transformation defines G, the required input to the HDP.

We can now demonstrate (ii) by showing that an optimal HDP solution, comprised of P_q and P_t, corresponds to the optimal TSP solution, P'. From Figure 2, note that an HDP solution of the form, $P_q = (w_0, w_1, d_1, w_1 \ldots, w_n, d_n, w_n, w_0)$ and $P_t = (w_0, w_1, \ldots, w_n, w_0)$, can be used to generate a TSP tour of the form $P' = (v_1, \ldots, v_n, v_1)$ by simply extracting the order of street vertices $(w_1, \ldots, w_n)$ in P_t, since the truck must visit every $w_i \in V_w$ to service each $d_i \in V_d$. If E'_P contains the sequence of edges in P', then $E_t = E'_P$. Now, since $c_q(e) = 0$ and $c_t(e) = c_{tq}(e) = c'(e)$, we can see that

$$\sum_{e \in E_q \setminus E_t} c_q(e) + \sum_{e \in E_t \setminus E_q} c_t(e) + \sum_{e \in E_q \cap E_t} c_{tq}(e) = \sum_{e \in E'_P} c'(e),$$

completing the proof. $\square$

IV. Solution Approach for HDP

Given that the HDP is NP-hard, our method for solving the problem will be to look for reductions to an NP-hard problem for which good solvers exist. In what follows we present an efficient reduction to the Generalized Traveling Salesman Problem (GTSP). We proceed by giving the details of the reduction, and then showing that an optimal GTSP solution provides an optimal HDP solution of equal cost.

Referring to Figures 3 and 4, note that the approach will be presented in two transformations. The first, T_1 is a procedure

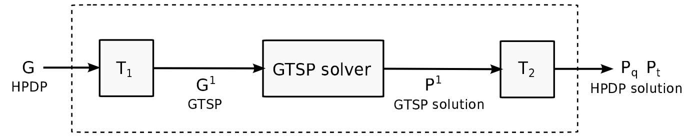
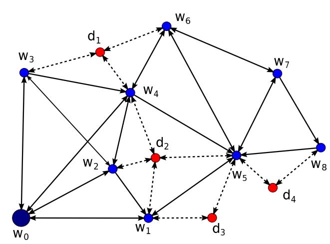


Fig. 3. A high-level schematic of the solution approach for the HDP.

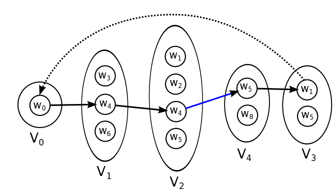
to cast an HDP on graph G as a GTSP on a partitioned graph $G^1 = (V^1, E^1, c^1)$, where each vertex set $V_i^1 \in V^1$ corresponds to a delivery point $d_i \in V_d$ and the vertices in V_i^1 correspond to the set of viable street deployment points, $w_j \in W_{d_i} \subset V_w$, for each d_i. Edges correspond to feasible routes between deliveries. The second transformation, T_2, is a method to extract the HDP solution, P_q, P_t, from a GTSP solution, P^1. Lemmas IV.1 and IV.2 prove the correctness of the transformations.

A. Reduction to the GTSP

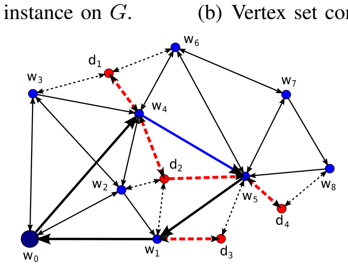
Figure 4 illustrates the graph transformations on a sample HDP instance to aid in the description. The problem in Figure 4(a) is a simplified version of the example problem in Figure 1 and contains an environment $G = (V, E, \mathcal{C})$, where $|V_d| = 4$ and $|V_w| = 8$. Figure 4(b) shows the transformed GTSP graph G^1, as well as an optimal solution, P^1, through it. Finally, Figure 4(c) shows how the GTSP solution can be translated to an HDP solution on G. We will refer to these figures throughout the descriptions below.



(a) An HDP instance on G.



(b) Vertex set construction in G^1.

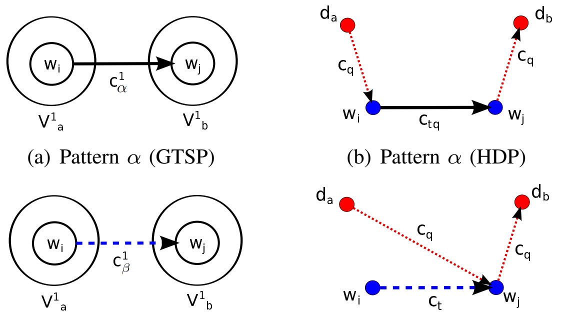


(c) An optimal HDP Solution.

Fig. 4. Transformation of HDP to GTSP. Figures 4(b) and 4(c) highlight red edges (quadrotor travel), blue edges (truck travel) and black edges (docked truck-quadrotor travel).

1) Transformation T_1: HDP to GTSP : Let the input to transformation T_1 be an instance of the HDP defined on the directed graph $G = (V, E, \mathcal{C})$. The output of T_1 is a partitioned directed graph $G^1 = (V^1, E^1, c^1)$ with V^1 partitioned into $n + 1$ disjoint subsets $V^1 = \{V_0^1, \ldots, V_n^1\}$, such that $V^1 = \cup_{i=0}^{n} V_i^1$, corresponding to the initial location w_0 and each of n delivery vertices.

Algorithm 1 describes the transformation of the *input* $G = (V, E, \mathcal{C})$ into the *output* $G^1 = (V^1, E^1, c^1)$. In the graph G^1,



(a) Pattern α (GTSP) (b) Pattern α (HDP)

(c) Pattern β (GTSP) (d) Pattern β (HDP)

Fig. 5. Deployment patterns α and β used in creating the edge weights in the GTSP graph G^1.

the vertex set V_0^1 contains w_0, and each vertex set $V_i^1, i = \{1, \ldots, n\}$, contains a copy of all street vertices $w_j \in V_w$ for which the flight edges (w_j, d_i) and (d_i, w_j) exist in E.

We construct E^1 as follows. In line 5 of Algorithm 1, we add an edge to E^1 between every pair of street vertices $w_i \in V_a^1$ and $w_j \in V_b^1$, where $a \neq b$. The edge will have a finite cost if either one of two subsets of edges,

$$\alpha = \{(d_a, w_i), (w_i, w_j), (w_j, d_b)\}, \quad \text{or}$$
$$\beta = \{(d_a, w_j), (w_i, w_j), (w_j, d_b)\}$$

exist in E: i.e., if the quadrotor can deliver to d_a from w_i, followed by d_b from w_j.

Figure 5 illustrates this mapping between the edges of E^1 and E. The edge $e \in E^1$ maps to either α or β in E between delivery vertices d_a and d_b as follows. In pattern α, shown in Figures 5(a) and 5(b), the quadrotor, having delivered an item at d_a from w_i, returns to the truck at w_i and travels in a docked state to w_j, to be redeployed towards d_b. In pattern β, shown in Figures 5(c) and 5(d), the quadrotor, having delivered an item at d_a from w_i travels directly from d_a to w_j to rendezvous with the truck and pickup the item to be delivered at d_b.

In Section IV-B, Lemma IV.1 states that the edge subsets α and β encode all potential truck-quadrotor deployment patterns between any two delivery vertices, d_a and d_b, for a chosen pair of respective street deployment vertices w_i and w_j. Thus, deployment patterns α and β present the only two potential edge costs for edges in E^1, and can be computed as follows:

$$\begin{aligned} c_\alpha^1(w_i, w_j) &= c_q(d_a, w_i) + c_{tq}(w_i, w_j) + c_q(w_j, d_b) \\ c_\beta^1(w_i, w_j) &= c_q(d_a, w_j) + c_t(w_i, w_j) + c_q(w_j, d_b) \end{aligned} \tag{2}$$

Given these two costs, the minimum cost deployment pattern between $w_i \in V_a^1$ and $w_j \in V_b^1$ is chosen and a cost,

$$c^1(e) = \min\{c_\alpha^1(e), c_\beta^1(e)\}$$

is assigned to the edge $(w_i, w_j) \in E^1$. This calculation is performed in lines 6-13 of Algorithm 1. Figure 4(b) illustrates the vertex sets of the constructed GTSP graph G^1 as a result of Algorithm 1.

Run-time: For an HDP with $|V_d| = n$ delivery vertices and $|V_w| = m$ street vertices, the the corresponding GTSP problem contains at most $1 + nm$ vertices, and thus $O(n^2 m^2)$ edges. The time to compute the GTSP graph is linear in the size of the graph.

Algorithm 1: Graph Transformation: G to G^1.

Input: $G = (V, E, \mathcal{C})$
Output: $G^1 = (V^1, E^1, c^1)$
1 $V_0^1 = V_0$
2 **foreach** $d_i \in V_d$ **do**
3 $\quad V_i^1 = \{w_j | \ w_j \in V_w, \ (w_j, d_i) \in E, \ (d_i, w_j) \in E\}$
4 $V^1 = \{V_0^1, V_1^1, \ldots, V_n^1\}$
5 $E^1 = \{(w_i, w_j) | \ w_i \in V_a^1, w_j \in V_b^1, a \neq b\}$
6 **foreach** $e = (w_i, w_j) \in E^1$ *where* $w_i \in V_a^1, w_j \in V_b^1$ **do**
7 $\quad$ **if** $a = 0$ **then**
8 $\quad\quad c_q(d_a, w_i) = 0$
9 $\quad$ **if** $b = 0$ **then**
10 $\quad\quad c_q(w_j, d_b) = 0$
11 $\quad c_\alpha^1(e) = c_q(d_a, w_i) + c_{tq}(w_i, w_j) + c_q(w_j, d_b)$
12 $\quad c_\beta^1(e) = c_q(d_a, w_j) + c_t(w_i, w_j) + c_q(w_j, d_b)$
13 $\quad c^1(e) = \min\{c_\alpha^1, c_\beta^1\}$

Note, however that this worst-case bound is attained only for a quadrotor having sufficiently large operating range such that each delivery vertex can be reached from every street vertex, i.e., $|W_{d_i}| = m$ for all $d_i \in V_d$. In many applications, where the quadrotor is limited to line-of-site deployments, we expect $|W_{d_i}|$ to be $O(1)$ and the size of the GTSP $1 + \sum_{i=1}^n |W_{d_i}|$ to be $O(n)$, giving a transformation that runs in $O(n^2)$ time. The simulation results in Section VI, Figure 11 further explore the effect of quadrotor range on the size and the runtime of the GTSP transformation.

The GTSP can now be solved using a variety of solvers in existing literature and as seen in Figure 4(a), the solution to the GTSP is a closed path of the form $P^1 = (w_0, w_1, \ldots, w_n, w_0)$, where w_0 is the starting vertex and $(w_1, \ldots, w_n)$ is a sequence containing one vertex from each set $V_i^1 \subset V^1$.

2) Transformation T_2: GTSP Solution to HDP Solution: Given the optimal GTSP solution P^1, the optimal HDP solution composed of a closed walk P_q and a closed path P_t can be obtained using Algorithm 2 as briefly described below. The algorithm computes the quadrotor path P_q (lines 2-11) and then computes the truck path P_t from the ordering of the street vertices in P_q (lines 12-15).

Let the computed GTSP solution be defined by the sequence of vertices $P^1 = (w_0, w_1, \ldots, w_n, w_0)$ where each vertex $w_i, i \in \{1, \ldots, n\}$, belongs to a unique vertex set V_j^1 in G^1. Since the optimal deployment pattern for every pair of deployment points $w_i \in V_a^1$ and $w_j \in V_b^1$ was predetermined during the construction of G^1, we can construct P_q by inserting the vertices of the complete quadrotor path between every consecutive vertex in P^1. The truck path P_t can be constructed by copying all unique street network vertices $w_i \in V_w$ from P_q in the order in which they occur in P_q.

In the HDP solution to the example problem, as shown in Figure 4(c), $P_q = (w_0, w_4, d_1, w_4, d_2, w_5, d_3, w_5, w_1, d_4, w_1, w_0)$ and $P_t = (w_0, w_4, w_5, w_1, w_0)$. Transformation T_2 is a linear in time, $O(n)$, algorithm since the deployment patterns between

Algorithm 2: Reconstructing P_q and P_t from P^1.

Input: $P^1 = (w_0, w_1, \ldots, w_n, w_0)$
Output: P_q, P_t
Initialize P_q and P_t as empty lists
$P_q.\mathrm{append}(w_0, w_1, d_a)$, where $w_1 \in V_a^1$
foreach $i \in \{1, \ldots, n-1\}$ **do**
    if $c^1(w_i, w_{i+1}) = c_\alpha^1(w_i, w_{i+1})$ **then**
        $P_q.\mathrm{append}(w_i, w_{i+1}, d_b)$, where $w_{i+1} \in V_b^1$
    else
        $P_q.\mathrm{append}(w_{i+1}, d_b)$, where $w_{i+1} \in V_b^1$
if $c^1(w_n, w_0) = c_\alpha^1(w_n, w_0)$ **then**
    $P_q.\mathrm{append}(w_n, w_0)$
else
    $P_q.\mathrm{append}(w_0)$
foreach $w_i \in P_q$ **do**
    if $w_i \notin P_t$ **then**
        $P_t.\mathrm{append}(w_i)$
$P_t.\mathrm{append}(w_0)$

each consecutive pair of vertices in P^1 were computed in T_1.

B. Correctness of the Transformation

This section proves that the GTSP transformation encodes all possible HDP solutions and that the optimal solution to the GTSP can be used to generate the optimal solution to the HDP.

Lemma IV.1 follows immediately from the discussion in Transformation T_1, that describes patterns α and β and the fact that there is a street edge between all pairs of street vertices. Thus, if the GTSP solution, P^1, contains the edge (w_i, w_j) and pattern α is chosen, then in the HDP solution, P_q will contain a subsequence of edges $\{(d_a, w_i), (w_i, w_j), (w_j, d_b)\}$. If pattern β is chosen, P_q will contain a subsequence $\{(d_a, w_j), (w_j, d_b)\}$. P_t will contain edge (w_i, w_j) in both cases. In the case where d_a and d_b share deployment points (i.e. $w_i = w_j$), the truck does not move and hence $\alpha = \beta$.

Lemma IV.1. *Deployment patterns α and β are the only two HDP routes between any two delivery vertices, d_a and d_b, given their respective street deployment points w_i and w_j.*

Lemma IV.2 validates transformation T_2 by showing that any feasible or optimal GTSP solution P^1 directly corresponds to an HDP solution P_q, P_t.

Lemma IV.2. *Any feasible GTSP tour on G^1 corresponds to a pair of feasible HDP routes on G of equal cost. Moreover, an optimal GTSP solution corresponds to the optimal HDP solution.*

Proof. Consider a feasible GTSP tour on G^1. By definition, this tour visits all vertex sets $V_a^1 \subset V^1$, and uses edges with finite cost to travel between vertex sets. Since each vertex set corresponds to a delivery vertex $d_a \in V_d$, the corresponding quadrotor route visits all delivery points. By Lemma IV.1, each edge $(w_i, w_j) \in E^1$ in the GTSP tour, where $w_i \in V_a$ and $w_j \in V_b$, has cost equal to that of the lowest cost HDP route from d_a to d_b for the respective w_i and w_j (as given by the minimum of the two deployment patterns α and β). Thus, the feasible GTSP solution on G^1 corresponds to a unique HDP solution on G of equal cost.

We prove that an optimal GTSP solution provides the optimal HDP solution, by contradiction, as follows. Consider an optimal GTSP solution of the form $P^1 = (w_0, w_1, \ldots, w_n, w_0)$. We know that each edge $(w_i, w_{i+1}) \in P^1$, where $w_i \in V_a$ and $w_{i+1} \in V_b$ represents an optimal subsequence of edges in P_q and P_t, based on the choice of α or β. Thus, a sub-optimal HDP solution can only be obtained if P^1 contains (i) a sub-optimal ordering of vertex sets, or (ii) a sub-optimal selection of vertices in any vertex set. This violates the definition of an optimal GTSP solution and hence optimality is preserved in the transformation from P^1 to P. □

C. Altering the HDP Solution Through Edge Weights

In a typical HDP solution, the truck-quadrotor team conducts deliveries in a clustered manner, with the truck stopping at a sequence of deployment points given by P_t, such that $|P_t| \leq m$, while the quadrotor visits a subset of delivery vertices $D_{w_i} \subset V_d$, from each $w_i \in P_t$, such that $\cup_{i=1}^{|P_t|} D_{w_i} = V_d$.

Given an HDP instance, the structure and total cost of P_t, P_q, and the choice of deployment patterns between each truck stop depend entirely on the relative values of the edge weights c_q, c_t and c_{tq} in G. Figure 6 qualitatively illustrates the effect of varying edge cost parameters on the nature of the HDP solution.

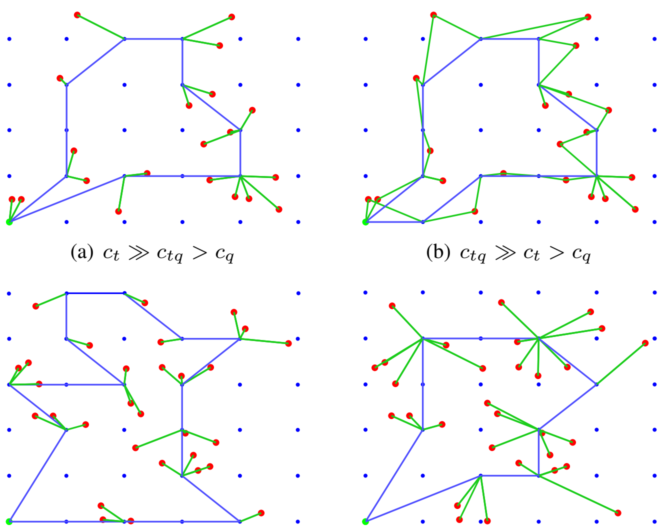
Figures 6(a) and 6(b) show two special cases of the HDP solution that arise when the costs, c_t and c_{tq} are greater than c_q as follows. When $c_t \gg c_{tq}$, the cost of the truck traveling alone is heavily penalized and all deployments occur using pattern α as seen in Figure 6(a). Conversely, when $c_{tq} \gg c_t$, docked truck-quadrotor travel is penalized, making deployment pattern β consistently preferable to α as shown in Figure 6(b).

Finally, Figures 6(c) and 6(d) illustrate the effect of the relative truck and quadrotor costs on the HDP solution. Low values of c_t and c_{tq} relative to c_q encourage greater truck effort in the HDP solution, as in Figure 6(c), while higher values of c_t and c_{tq} relative to c_q result in a greater quadrotor effort, limited by its operating range, as in Figure 6(d).

D. Minimum-Time HDP Solutions

In the case where c_t, c_{tq}, and c_q correspond to travel times rather than fuel costs, the more relevant objective is the time to perform the n deliveries. The key difference when computing the time of a HDP route is that when the truck and quadrotor travel simultaneously (as in deployment pattern β of Figure 5), we should take the maximum of these travel times, rather than the sum.

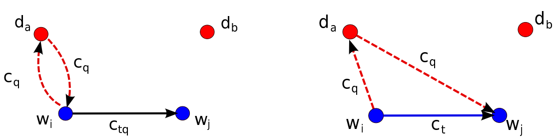
We can solve this problem through a small modification of the edge weights in the GTSP graph G^1. Let us denote the min-time graph as $\overline{G}^1 = (\overline{V}^1, \overline{E}^1, \overline{c}^1)$. Its vertex set and partition are identical to that of G^1. The edge (w_i, w_j) is added to $\overline{E}^1$



(a) $c_t \gg c_{tq} > c_q$ (b) $c_{tq} \gg c_t > c_q$

(c) Low truck cost $c_t, c_{tq} \ll c_q$. (d) High truck cost: $c_t, c_{tq} \gg c_q$

Fig. 6. HDP solution characterization based on c_q, c_t and c_{tq}. All figures show $w_0 = (0, 0)$, delivery points (red vertices), a gridded street network (blue vertices), the truck path (blue paths) and quadrotor flight paths (green paths).



(a) Pattern $\overline{\alpha}$ for min-time. (b) Pattern $\overline{\beta}$ for min-time.

Fig. 7. Deployment patterns $\overline{\alpha}$ and $\overline{\beta}$ used for the min-time formulation.

if either one of two subsets of edges,

$$\overline{\alpha} = \{(w_i, d_a), (d_a, w_i), (w_i, w_j)\}, \quad \text{or}$$
$$\overline{\beta} = \{(w_i, d_a), (d_a, w_j), (w_i, w_j)\}$$

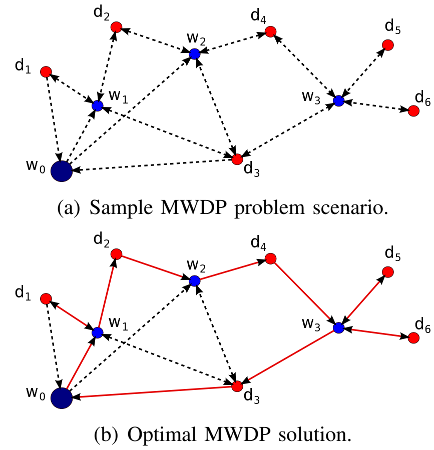
exist in E. The cost of the edge is defined to be

$$\begin{aligned} c^1_{\overline{\alpha}}(w_i, w_j) &= c_q(w_i, d_a) + c_q(d_a, w_i) + c_{tq}(w_i, w_j) \\ c^1_{\overline{\beta}}(w_i, w_j) &= \max\{c_q(w_i, d_a) + c_q(d_a, w_j), \\ &\qquad c_t(w_i, w_j)\} \end{aligned} \tag{3}$$

The cost of the edge in $\overline{G}^1$ is then, $c^1(e) = \min\{c^1_{\overline{\alpha}}(e), c^1_{\overline{\beta}}(e)\}$. The new deployment patterns and associated edge weights are illustrated in Figure 7.

Notice that we now "cut" the HDP solution into pieces at street vertices instead of delivery vertices. An edge in $\overline{G}^1$ represents the following: the quadrotor and truck begin at w_i, the quadrotor delivers to d_a, and the team finishes at w_j (moving there together or separately), where the quadrotor picks up the package for delivery at d_b. The advantage to this approach for the minimium-time formulation is that the simultaneous travel of the quadrotor and truck between street vertices is contained on a single edge, creating edge weights that are independent of one another. In the original G^1 definition, simultaneous travel would be split over two edges, creating coupling between edge weights of different edges. Our original cut formulation proves useful, however, when considering the multiple warehouse delivery problem variant of the HDP, as discussed in Section V.

After this modification, an optimal GTSP solution on $\overline{G}^1$ will correspond to minimum-time HDP routes for the truck and quadrotor. The truck path P_t is given by the sequence of warehouses in the GTSP solution. The quadrotor path P_q is obtained by inserting the appropriate deployment pattern between each warehouse.



(a) Sample MWDP problem scenario.

(b) Optimal MWDP solution.

Fig. 8. The Multiple Warehouse Delivery Problem (MWDP).

V. The Multiple Warehouse Delivery Problem (MWDP)

Consider the case in which each street vertex is a static warehouse. The truck is no longer a part of the problem, and the goal is to find a quadrotor route that alternately visits street vertices (i.e., warehouses) and delivery vertices. We call this problem the Multiple Warehouse Delivery Problem (MWDP), where a set of delivery requests, $V_d = \{d_1, \ldots, d_n\}$ must be fulfilled by a single vehicle from a set of warehouses $V_w = \{w_1, \ldots, w_m\}$.

It is possible to note that the MWDP is a special case of the HDP, in which we set $c_{tq}(e) = \infty$, and $c_t(e) = 0$ for each edge $e \in E$. In doing this, we prevent docked travel and assume that the truck travels alone with zero cost (i.e., infinite speed.) From Figure 5 we can see that, $c_q(d_a, w_i) + c_{tq}(w_i, w_j) \geq c_q(d_a, w_j) + c_t(w_i, w_j)$, is always true in this case, and hence every edge of P_q in the HDP solution will be a flight edge of the form $e = (w_i, d_j)$ or $e = (d_j, w_i)$, with a cost $c_q(e)$. The total cost of P_t is $\sum_{e \in E_t} c_t(e) = 0$.

Figure 8(a) illustrates an MWDP graph, $G = (V, E, c)$, where $V = V_0 \cup V_w \cup V_d$, E contains directed edges (d_i, w_j) for all $d_i \in V_d, w_j \in V_w$ and edges (w_j, d_i) if $w_j \in W_{d_i}$. Cost function, $c : E \to \mathbb{R}_{\geq 0}$, represents the non-negative travel cost, that satisfies the triangle inequality. The MWDP is stated in Problem V.1.

Problem V.1 (Multiple Warehouse Delivery Problem)**.** Given $G = (V, E, c)$, where $V = V_0 \cup V_d \cup V_w$, compute a closed walk P, that starts and ends at w_0, such that each delivery vertex in V_d is visited exactly once.

Since the MWDP is a special case of the HDP, it can be solved using the methods in Section IV. However, the downside of this approach is that it results in an increase in the size of the problem instance as described in Section IV-A. Exploiting the simplifications in the MWDP relative to the HDP, we present two improved solution approaches. First, we present a transformation of the MWDP into a TSP and, second, we present an exact algorithm to solve instances with a small, fixed number of warehouses. This problem is similar to the TSP with refueling constraints [27], with the key difference being that the quadrotor is required to return to a warehouse (refueling depot) after each delivery. As such, we will be able to exploit this additional structure to develop more efficient algorithms.

A. Transformation Algorithm: MWDP to TSP

Since, in the MWDP, the quadrotor uses pattern β for each delivery, there is only one shortest path between any pair of delivery vertices d_a and d_b, and it passes through the warehouse vertex $w_i \in W_{d_b}$, such that $c(d_a, w_i) + c(w_i, d_b)$ is minimized. Therefore, we can cast the MWDP as a TSP, by transforming an MWDP instance $G = (V, E, c)$, into a TSP instance, $G^1 = (V^1, E^1, c^1)$, where $V^1 = V_0 \cup V_d$ and E^1 contains edges $e = (v_a, v_b)$, for all $v_a, v_b \in V^1$. Now for each edge, (v_a, v_b), we identify the warehouse $w_i \in W_{d_b}$ that minimizes $c(d_a, w_i) + c(w_i, d_b)$, and set the cost $c^1(v_a, v_b) = c(d_a, w_i) + c(w_i, d_b)$

Graph G' is a TSP instance of size $|V_d| = n$, which is significant smaller than the GTSP and can be solved using a number of exact or heuristic algorithms in existing literature such as the Lin-Kernighan [28] or LKH [17] heuristics. The TSP solution is a sequence of vertices of the form $P^1 = (v_0, v_1, \ldots, v_n, v_0)$, from which an MWDP solution may be obtained by inserting the stored warehouse vertex w_i, between each consecutive pair of vertices $\{v_a, v_b\}$ in P^1. An optimal MWDP solution is illustrated in Figure 8(b).

B. Kernel Sequence Enumeration (KSE) Algorithm

Figure 8(b) shows that an optimal MWDP solution will always be of the form

$$P = (w_0, w_{k_1}, d_1, w_{k_2}, d_2 \ldots, w_{k_n}, d_n, w_0),$$

where we have numbered the delivery points so that they are visited in the order $d_1, d_2, \ldots, d_n$ and each k_i is a warehouse index in $\{1, \ldots, m\}$. All delivery vertices are visited in sub-sequences, $(w_{k_i}, d_i, w_{k_{i+1}})$ where w_{k_i} is the warehouse assigned to service d_i. Given this property, we identify two classes of delivery vertices in P,

1) a *localized delivery vertex*, d_i, for which $k_i = k_{i+1}$, and
2) a *transitional delivery vertex*, d_i, for which $k_i \neq k_{i+1}$.

We also say that d_n is a transitional delivery vertex since it returns to w_0. Two additional properties of P, that can be easily proven by the triangle inequality are:

1) For every localized delivery vertex d_i in P, where $(w_{k_i}, d_i, w_{k_{i+1}})$ and $k_i = k_{i+1}$, we must have that $w_{k_i} = \arg\min_{w \in V_w} c(w, d_i)$. Thus $w_{k_i} = w_{k_{i+1}}$ is the closest warehouse to d_i.

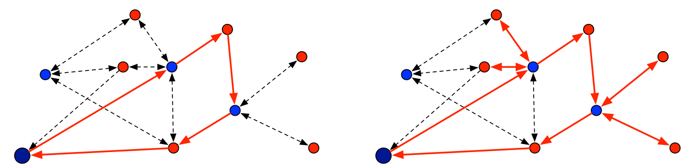


Fig. 9. The left figure shows a kernel sequence consisting of transitional deliveries. The right figure shows the completion of this sequence with localized deliveries.

2) If the path P visits $m_P < m$ unique warehouses in V_w, then the number of transitional delivery vertices $|D_t| = m_P$. This implies that the quadrotor never revisits a warehouse w_{k_i} once it has transitioned to warehouse $w_{k_{i+1}}$ with $k_{i+1} \neq k_i$.

From the second property, we can define a *kernel sequence* to be the sequence transitional deliveries (each consisting the start warehouse, delivery vertex, and end warehouse) in the quadrotor path. Based on this, the following procedure gives us an exact algorithm for solving the MWDP:

1) Enumerate all *kernel sequences* consisting of an ordered subset of warehouses and a transitional delivery point between each pair of warehouses. In total there are $O(n^m m^m)$ possible kernel sequences.
2) For each kernel sequence, create a complete path by assigning all remaining delivery points as localized deliveries, using their closest warehouse in the kernel sequence.
3) Output the shortest path among all completed kernel sequences.

An example kernel sequence and its completion is shown in Figure 9.

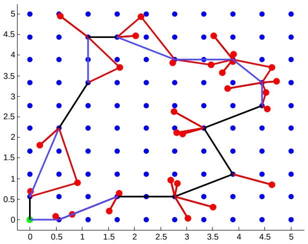
To complete each kernel sequence we must compute the closest warehouse for each remaining delivery point. Since there are at most m warehouses in the kernel sequence and n delivery points that are not in the kernel sequence, the complexity of each kernel completion step is $O(nm)$. Therefore, the total runtime of this brute force algorithm is $O\big((nm)^{m+1}\big)$.

Thus, the key point is that the algorithm is polynomial for a fixed number of warehouses m. For example, if there are three warehouses and n delivery points, this exact algorithm runs in $O(n^4)$ time, which may be acceptable, and does not require a transformation to an NP-hard problem. However, for a larger number of warehouses, this algorithm is less practical.

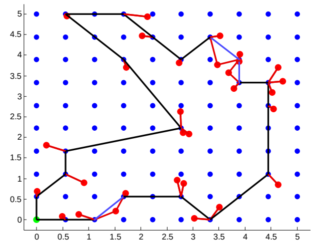
VI. Simulation Results

The optimization framework for this paper was implemented in MATLAB. The solutions were computed on a laptop computer running a 32 bit Ubuntu 12.04 operating system with a 2.53 GHz Intel Core2 Duo processor and 4GB of RAM. To solve TSP instances, we used the LKH solver [17], which is a heuristic that quickly produces high-quality, but in general suboptimal solutions. To solve GTSP instances, we reduced the problem to a TSP using the Noon-Bean transformation, and then called LKH on the TSP instance.

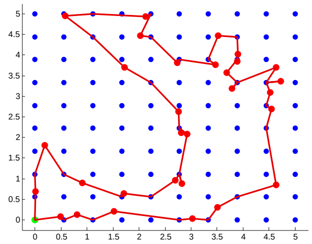
Figure 10 presents HDP solutions on a sample problem instance with 30 delivery points and a gridded environment



(a) HDP ($r_q = 0.3\,r_{\text{env}}$).



(b) HDP ($r_q = 0.1\,r_{\text{env}}$).



(c) MWDP TSP solution.

Fig. 10. Subfigures (a) and (b) show solutions for the HDP as a function of the quadrotor range. Subfigure (c) shows the MWDP solution in which each street vertex is a static warehouse.

with 100 street vertices in an environment of arbitrary size r_{env}. The key simulation parameters are c_t, c_q, c_{tq} and r_q, the operating range of the quadrotor, defined as a percentage of r_{env}, which dictates the size of W_{d_i} for each delivery point d_i and consequently, the size of the GTSP. For these results, we set c_q to be the Euclidean distance between vertices and $c_t(e) = c_{tq}(e) = 3c_q(e)$ for all edges e.

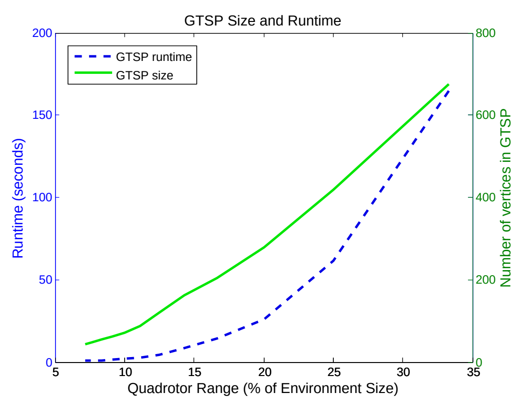


Fig. 11. The size of the GTSP graph and computation time for the HDP shown in Figure 10.

In Figure 10(a), $r_q = 0.3\,r_{\text{env}}$, which resulted in a GTSP with 170 vertices and took 5.7 seconds to compute a solution. When r_q was reduced to $r_q = 0.1\,r_{\text{env}}$, the resulting GTSP contained 82 vertices and took 2.3 seconds to compute the solution, shown in Figure 10(b). From the Figures 10(a) and 10(b), we can see that reducing the quadrotor range resulted in a smaller problem size, and an increasing truck effort, similar to the properties observed in Section IV-C where a lower truck cost resulted in longer truck path in the HDP solution. In the limiting case, the HDP approaches the MWDP special case in Figure 10(c), for which the TSP method computes a solution in 0.45 seconds. To assess this further, Figure 11 shows the effect of the quadrotor range on the size (right y-axis) and runtime complexity (left y-axis) of the GTSP solution for the environment shown in Figure 10. The runtime on each instance is dominated by that of LKH, which internally performs 10 runs for a given instance. Figure 12 shows that for the MWDP case, the TSP of size n presents a faster and more scalable solution than the GTSP approach as shown by the average growth of runtime complexity as $|V_d|$ in increased, keeping other parameters and $|V_w|$ constant.

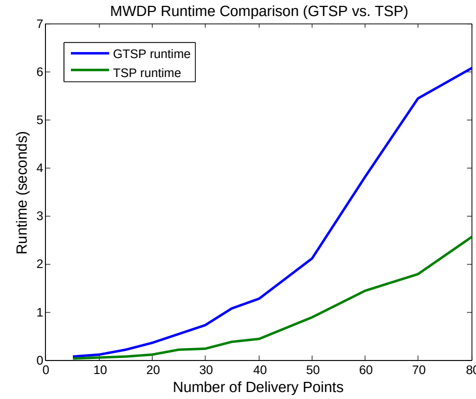


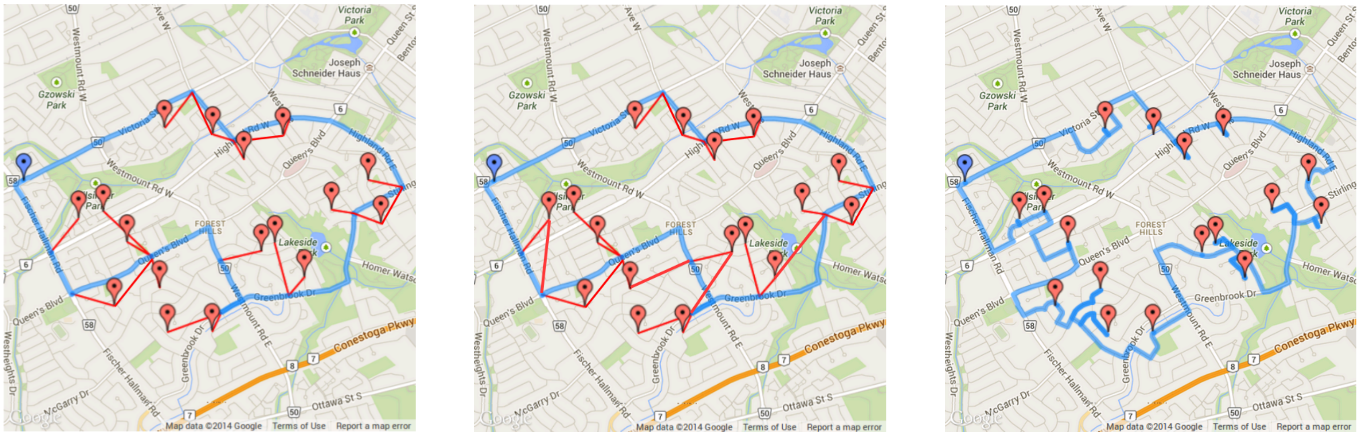
Fig. 12. A comparison of the GTSP and TSP approaches to the MWDP for the MWDP shown in Figure 10.

Delivery Points	Runtime GTSP	Runtime TSP	Runtime KSE	Solution Quality GTSP	Solution Quality TSP	Solution Quality KSE
3	0.05	0.04	0.06	10.56	10.56	9.95
6	0.20	0.06	1.11	16.21	16.61	16.21
9	0.26	0.14	5.55	30.47	30.20	29.21
12	0.44	0.26	21.46	35.12	34.27	33.52

TABLE I
ALGORITHM COMPARISON FOR THE MULTIPLE WAREHOUSE DELIVERY PROBLEM WITH THREE WAREHOUSE (I.E., $|V_w| = 3$).

In the case of the MWDP, all three solution methods can be employed with comparable results in terms of solution quality. While the KSE algorithm is useful to obtain the optimal MWDP solution for smaller problem sizes, it quickly becomes impractical as the number of warehouses grows and the TSP method stands out as the appropriate approach. This is evident in Table I, which shows runtime and solution quality results for an MWDP problem with $|V_w| = 3$ and an increasing number of delivery points.

Figure 13 presents a realistic delivery scenario on a Google street map of a residential neighbourhood in Waterloo, Ontario, Canada. Figure 13(a) shows an HDP solution for 17 delivery points in contrast to a single delivery truck conducting deliveries in Figure 13(c). Given a maximum range of 150m for the quadrotor to ensure line of sight, the HDP solution in this problem instance results in approximately a 50% reduction in travel distance for the truck and thus the fuel consumption (assuming that the fuel consumption of the quadrotor is negligible when compared to the truck). Finally, Figure 13(b) shows the minimum-time version of the optimal solution. Given, a quadrotor speed of 30 km/hr, a landing time of 30 seconds per landing (deliveries and return to truck), and a truck speed of 40 km/hr, the optimal solution completes all deliveries



(a) HDP truck-quadrotor solution with the minimum fuel objective.

(b) HDP truck-quadrotor delivery route with the minimum-time objective.

(c) Truck delivery route.

Fig. 13. HDP solutions for deliveries in a neighborhood of Waterloo, Ontario.

in one hour and twenty three minutes. For this example, the HDP solution results in approximately a 35% reduction in travel time compared to the single truck solution shown in Figure 13(c). To make this comparison we assume that the delivery time for each parcel from truck to door is 30 seconds.

VII. Conclusions

This paper presents a novel adaptation of a heterogeneous carrier-vehicle system for cooperative deliveries in urban environments. The HDP represents a class of cooperative carrier-vehicle path planning problems in discrete environments, applicable to a number of multi-robot systems in scenarios like search and rescue, surveillance and exploration. In future work, we are interested in generalizing the HDP to allow multiple simultaneous quadrotor deliveries, quadrotors with capacity larger than one, and dynamic scenarios where new requests arrive during execution.

References

[1] R. D'Andrea, "Guest editorial: Can drones deliver?" *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 3, pp. 647–648, 2014.

[2] K. Swieringa, C. Hanson, J. Richardson, J. White, Z. Hasan, E. Qian, and A. Girard, "Autonomous battery swapping system for small-scale helicopters," in *IEEE International Conference on Robotics and Automation*, May 2010, pp. 3335 –3340.

[3] K. Suzuki, P. Kemper Filho, and J. Morrison, "Automatic battery replacement system for UAVs: Analysis and design," *Journal of Intelligent and Robotic Systems*, vol. 65, pp. 563–586, 2012.

[4] N. Mathew, S. L. Waslander, and S. L. Smith, "Optimal path planning in cooperative heterogeneous multi-robot delivery systems," in *Workshop on the Algorithmic Foundations of Robotics*, Istanbul, Turkey, Aug. 2014.

[5] L. E. Parker, "Current state of the art in distributed autonomous mobile robotics," in *Distributed Autonomous Robotic Systems 4*, L. E. Parker, G. Bekey, and J. Barhen, Eds. Springer Japan, 2000, pp. 3–12.

[6] L. Pimenta, V. Kumar, R. Mesquita, and G. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *IEEE Conference on Decision and Control*, 2008, pp. 1–8.

[7] P. Chand and D. A. Carnegie, "Mapping and exploration in a hierarchical heterogeneous multi-robot system using limited capability robots," *Robotics and Autonomous Systems.*, vol. 61, no. 6, pp. 565 – 579, 2013.

[8] B. Grocholsky and N. Michael, "Exploiting mobility heterogeneity in micro-aerial vehicle deployments for environment exploration," in *IEEE Globecom Workshops*, 2013, pp. 1421–1425.

[9] S. L. Waslander, "Unmanned aerial and ground vehicle teams: Recent work and open problems," in *Autonomous Control Systems and Vehicles*, ser. Intelligent Systems, Control and Automation: Science and Engineering, K. Nonami, M. Kartidjo, K.-J. Yoon, and A. Budiyono, Eds. Springer Japan, 2013, vol. 65, pp. 21–36.

[10] J. Chen and D. Sun, "Coalition-based approach to task allocation of multiple robots with resource constraints," *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 3, pp. 516–528, July 2012.

[11] N. Mathew, S. L. Smith, and S. L. Waslander, "Multi-robot rendezvous planning for recharging in persistent tasks," *IEEE Transactions on Robotics*, vol. 31, no. 1, pp. 128–142, 2014.

[12] C. Phan and H. Liu, "A cooperative UAV/UGV platform for wildfire detection and fighting," in *International Conference on System Simulation and Scientific Computing*, 2008, pp. 494–498.

[13] P. Tokekar, J. V. Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic UAV and UGV system for precision agriculture," in *IEEE/RSJ Conference on Intelligent Robots and Systems*, 2013, pp. 5321–5326.

[14] G. Nagy and S. Salhi, "Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries," *European Journal of Operational Research*, vol. 162, no. 1, pp. 126 – 141, 2005.

[15] M. Drexl, "Synchronization in vehicle routing-a survey of VRPs with multiple synchronization constraints," *Transportation Science*, vol. 46, no. 3, pp. 297–316, 2012.

[16] F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. Smith, "Dynamic vehicle routing for robotic systems," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1482–1504, Sept 2011.

[17] K. Helsgaun, "General k-opt submoves for the Linkernighan TSP heuristic," *Mathematical Programming Computation*, vol. 1, pp. 119–163, 2009.

[18] E. Garone, R. Naldi, A. Casavola, and E. Frazzoli, "Cooperative path planning for a class of carrier-vehicle systems." in *IEEE Conference on Decision and Control*, 2008, pp. 2456–2462.

[19] ——, "Cooperative mission planning for a class of carrier-vehicle systems," in *IEEE Conference on Decision and Control*, Dec 2010, pp. 1354–1359.

[20] E. Garone, J.-F. Determe, and R. Naldi, "Generalized traveling salesman problem for carrier-vehicle systems," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 3, pp. 766–774, 2014.

[21] B. Korte and J. Vygen, *Combinatorial optimization*. Springer, 2012.

[22] N. Christofides, "Worst-case analysis of a new heuristic for the traveling salesman problem," Graduate School of Industrial Administration, Carnegie Mellon University, Tech. Rep. 388, 1976.

[23] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook, *The traveling salesman problem: a computational study*. Princeton University Press, 2011.

[24] D. Karapetyan and G. Gutin, "Lin–kernighan heuristic adaptations for the generalized traveling salesman problem," *European Journal of Operational Research*, vol. 208, no. 3, pp. 221–232, 2011.

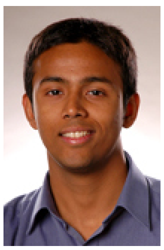
[25] P. Oberlin, S. Rathinam, and S. Darbha, "A transformation for a

heterogeneous, multiple depot, multiple traveling salesman problem," in *American Control Conference*, 2009, pp. 1292–1297.

[26] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, 4th ed., ser. Algorithmics and Combinatorics. Springer, 2007, vol. 21.

[27] K. Sundar and S. Rathinam, "Algorithms for routing an unmanned aerial vehicle in the presence of refueling depots," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 1, pp. 287–294, Jan 2014.

[28] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," *Operations Research*, vol. 21, no. 2, pp. 498–516, 1973.



Neil Mathew received the B.A.Sc. degree from the University of Waterloo, Waterloo, ON, Canada in 2010 and the M.A.Sc. degree in Mechanical Engineering with the Waterloo Autonomous Vehicles Laboratory at the University of Waterloo in 2014. His research interests include robotic motion planning in unknown environments, multi-robot coordination and energy aware task scheduling with an emphasis on long-term autonomy.

Stephen L. Smith received the B.Sc. degree in engineering physics from Queen's University, Kingston, ON, Canada, in 2003, the M.A.Sc. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, in 2005, and the Ph.D. degree in mechanical engineering from the University of California, Santa Barbara, in 2009.

He is an Assistant Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON. From 2009 to 2011, he was a Postdoctoral Associate with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge. His main research interests are in control and optimization for autonomous systems, with a particular emphasis on robotic motion planning and distributed coordination.

Steven L. Waslander received the B.Sc.E degree in Applied Mathematics and Mechanical Engineering from Queen's University, Kingston, ON, Canada, in 1998, the M.S. in Aeronautics and Astronautics from Stanford University in 2002 and the Ph.D. in Aeronautics and Astronautics from Stanford University in 2007.

He is currently an Assistant Professor with the Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, ON. He is also the Principal Investigator at the Waterloo Autonomous Vehicles Laboratory. His main research interests include the planning and control of autonomous aerial rotorcrafts and ground rovers with a focus on perception and navigation in unknown environments.