Optimal Path Planning in Cooperative Heterogeneous Multi-robot Delivery Systems

Neil Mathew, Stephen L. Smith, and Steven L. Waslander

University of Waterloo, Waterloo, Ontario, N2L3G1, Canada

Abstract. This paper addresses 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 using a reduction from the Travelling Salesman Problem and an algorithmic solution is proposed using a graph transformation to the Generalized Travelling Salesman Problem, which can be solved using existing methods. Simulation results compare the performance of the presented algorithms and demonstrate examples of delivery route computations over real urban street maps.

1 Introduction

An emerging application for micro-aerial vehicles, such as quadrotors, is in performing autonomous deliveries in urban environments. A number of large retailers have recently announced plans to deploy quadrotors for expedited small package deliveries. While quadrotors 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 issues such as safety, security and endurance, still need to be addressed. Current quadrotor 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* quadrotor. The role of the truck is to carry a shipment of packages to be delivered, as well as a docked quadrotor, and the role of the quadrotor is to carry individual packages from the truck to specific delivery points in the environment. By requiring the quadrotor to perform only the last leg of the delivery, both range and line-of-sight limitations are accounted for.

We will assume that the quadrotor 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 quadrotor after each delivery and that it has an operating range sufficient for the entire delivery mission. The goal of this paper is to propose a framework to compute a minimum cost cooperative route enabling the quadrotor to visit all delivery points in the environment. 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 minimise the total cost of deliveries.

Related Work: The HDP belongs to a class of problems referred to as Carrier-Vehicle Travelling Salesman Problems (CV-TSP), extensively studied by Garone et al. [1] in the context of a marine carrier and an aircraft visiting a set of locations to conduct a rescue mission in a planar environment. They formulate a continuous optimization and compute a solution 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 contrast, given the discrete nature of our HDP, we will be able to design a single optimization that computes cooperative paths for both vehicles.

Cooperative control in heterogeneous multi-robot teams has been investigated for applications like search and rescue, surveillance, and exploration, [2–4], where robots with complimentary capabilities must accomplish a common goal. The most relevant are collaborative UAV-UGV teams where UAVs can rendezvous and dock with UGVs to benefit from the larger payload capacity and energy resources of UGVs [5, 6]. One of the main challenges with heterogeneous systems is the development of cooperative planning algorithms to achieve a desired objective. Rathinam et al. explore optimal path planning in heterogeneous teams using variants of the Travelling Salesman Problem (TSP) and the Generalized Travelling Salesman Problem (GTSP) [7, 8] which are well studied problems in operations research literature and can be solved using a number of exact, approximate or heuristic algorithms. In this work we use the Noon-Bean Transformation [9] to cast the GTSP as an Asymmetric Travelling Salesman Problem (ATSP) and solve it using the Lin-Kernighan-Helsgaun (LKH) heuristic solver [10]. Finally, we will draw from existing literature on vehicle routing and pick-up delivery problems [11–13] to inform our work.

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 GTSP. Finally, we examine a special case of the HDP consisting of a single vehicle and multiple static warehouses, called the *Multiple Warehouse Delivery Problem* (MWDP). We present two algorithms for the MWDP, one, using an alternative transformation to the TSP and the other, a polynomial time exact algorithm to compute an optimal delivery route.

The organization of the paper is as follows. Section 2 formulates the HDP as an optimal path planning problem in a discrete environment. In Section 3, the HDP is proved to be NP-hard. Section 4 presents the transformation to the GTSP implemented to solve the HDP. Section 5 presents two algorithmic solutions for the MWDP and finally, Section 6 compares and benchmarks all proposed algorithms through simulation results.

1.1 Definitions and Nomenclature

A *graph* is denoted by $G = (V, E, c)$, where V is the set of vertices, E is the set of edges and $c : E \to \mathbb{R}$ is a function 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 mutually exclusive sets $(V_1, \ldots, V_\ell)$ where $\cup_{i=1}^{\ell} V_i = V$.

A *route* over a graph is a sequence of vertices $P = (v_1, \ldots, v_k)$ linked by edges $(v_i, v_{i+1}), i = 1, \ldots, k-1$. Following [14], 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 $i, j \in \{1, \ldots, k-1\}$. A *closed route* is a route of any type (e.g. route, walk, path) where $v_1 = v_k$. A *tour* is closed path that visits all vertices in V exactly once.

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

2 The Heterogeneous Delivery Problem (HDP)

The HDP is abstracted 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

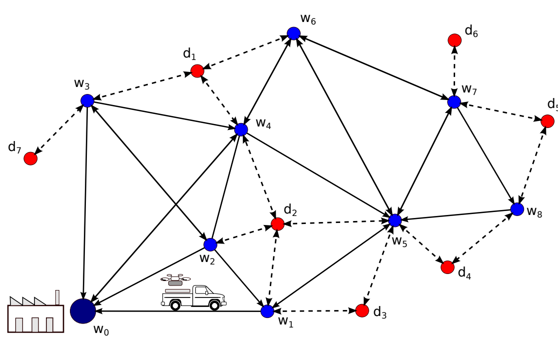


Fig. 1. The Heterogeneous Delivery Problem. The street edges (solid lines) are shown as either single or double arrows, that represent pairs of directed edges. All flight edges (dashed lines) are bidirectional edges between vertices.

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 vertices, edges and costs of G are defined as follows:

Vertices: The vertex set V is defined as a union of three mutually exclusive 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 mutually exclusive subsets, $E = E_w \cup E_d$. The set E_w contains directed *street edges* of the form (w_i, w_j), that represent shortest routes between street vertices for all $w_i, w_j \in V_w$. 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. 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 travelling 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.

We extend the definition of a graph from Section 1.1 to $G = (V, E, \mathcal{C})$, where $\mathcal{C}$ is a triple of costs, and formulate the HDP, on G, as the problem of computing two routes, for the truck and quadrotor, both starting and ending at vertex w_0, such that the truck, travelling on street edges, stops at a sequence of deployment 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.

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 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

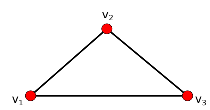
$$\sum_{e \in E_q \setminus E_t} c_q(e) + \sum_{e \in E_t \setminus E_q} c_t(e) + \sum_{e \in E_q \cap E_t} c_{tq}(e). \tag{1}$$

3 Proof of NP-hardness

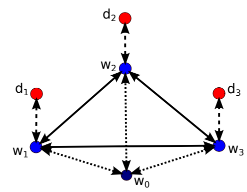
To prove NP-hardness of the HDP, we will show that (i) an instance of the TSP can be reduced to an instance of the HDP, and (ii) an optimal HDP solution can be used to generate an optimal TSP solution.

Theorem 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.



(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.

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

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

Thus, the optimal solution to the HDP can indeed be transformed into the optimal solution of the TSP, completing the proof. $\square$

4 Solution Approach

Given the NP-hardness of the HDP and the fact that it contains the TSP as a special case, our solution approach will be to polynomially transform an instance of the HDP into a GTSP, such that the optimal GTSP solution provides an optimal HDP solution of equal cost.

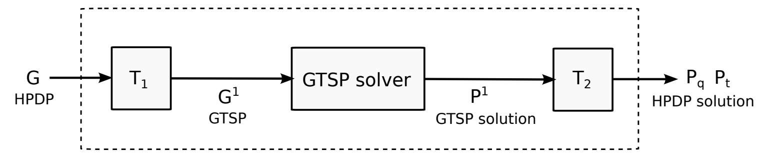


Fig. 3. Graph transformation based solution approach to the HDP.

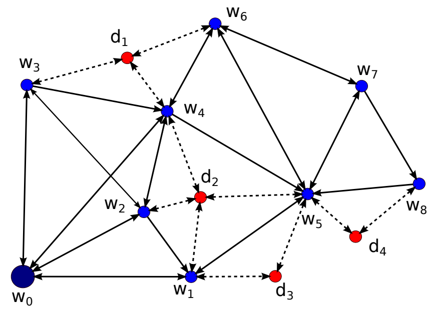
Referring to Figures 3 and 4, note that the approach will be presented in two transformations. The first, T_1 is a procedure to cast an HDP on graph G as a GTSP on a partitioned graph $G^1 = (V^1, E^1, c^1)$, where each vertex set $V_i^1 \in V^1$ corresponds to a delivery point $d_i \in V_d$ and the 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 1 and 2 prove the correctness of the transformations.

4.1 Transformation Algorithms

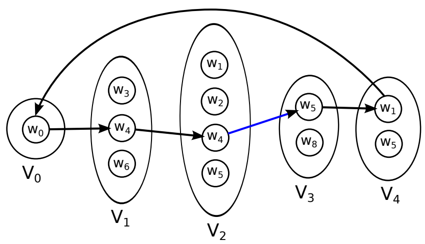
Figure 4 illustrates the graph transformations on a sample HDP instance to aid in the description. The problem in Figure 4(a) is a simplified version of the example problem in Figure 1 and contains an environment $G = (V, E, \mathcal{C})$, where $|V_d| = 4$ and $|V_w| = 8$. Figure 4(b) shows the transformed 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.

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 $n+1$ mutually exclusive 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.

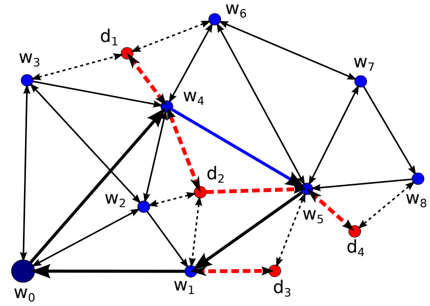
Algorithm 1 describes the transformation of the *input* $G = (V, E, \mathcal{C})$ into the *output* $G^1 = (V^1, E^1, c^1)$. In the graph G^1, the vertex set V_0^1 contains w_0, and each vertex set $V_i^1, i = \{1, \ldots, n\}$, contains a copy of all street vertices $w_j \in V_w$ for which the flight edges (w_j, d_i) and (d_i, w_j) exist in E. We construct E^1 as follows. Consider two street vertices in G^1 defined by $w_i \in V_a^1$ and $w_j \in V_b^1$, where $a \neq b$. The edge (w_i, w_j) is added to E^1 if either one of two subsets of



(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).

edges, $\alpha = \{(d_a, w_i), (w_i, w_j), (w_j, d_b)\}$, 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 4.2, Lemma 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:

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

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 $(w_i, w_j) \in E^1$. Figure 4(b) illustrates the vertex sets of the constructed GTSP graph G^1 as a result of Algorithm 1.

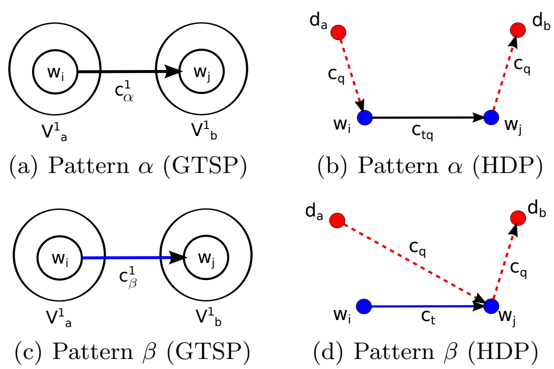


Fig. 5. Mapping between edges in GTSP and HDP.

Transformation T_1 has a runtime complexity of $O(n^2)$ and for an HDP with $|V_d| = n$ and $|V_w| = m$, it generates a GTSP of size $1 + nm$. While this is a significant increase in problem size, it represents the worst case with the quadrotor having an infinite operating range such that for each $d_i \in V_d$, $|W_{d_i}| = m$. In practice, $|W_{d_i}| < m$ and the size of the GTSP is $1 + \sum_{i=1}^{n} |W_{d_i}|$. The simulation results in Section 6, Figure 8(d) show how the quadrotor range affects size and runtime complexity 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 $V_i^1 \subset V^1$.

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.

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. 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 $P_t = (w_0, w_4, w_5, w_1, w_0)$.

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

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

Transformation T_2 is a linear in time, $O(n)$, algorithm since the deployment patterns between each consecutive pair of vertices in P^1 were computed in T_1.

4.2 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 1 follows immediately from the discussion in Transformation T_1, that describes patterns α and β. 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 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 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 2. *Any feasible GTSP tour on G^1 corresponds to a pair of feasible HDP routes on G. Moreover, an optimal GTSP solution corresponds to the optimal HDP solution of identical cost.*

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

P_q.append(w_0, w_1, d_a), where $w_1 \in V_a^1$
foreach $i \in \{1, \ldots, n-1\}$ **do**
  if $c^1(w_i, w_{i+1}) = c^1_\alpha(w_i, w_{i+1})$ **then**
    P_q.append(w_i, w_{i+1}, d_b), where $w_{i+1} \in V_b^1$
  else
    P_q.append(w_{i+1}, d_b), where $w_{i+1} \in V_b^1$
if $c^1(w_n, w_0) = c^1_\alpha(w_n, w_0)$ **then**
  P_q.append(w_n, w_0)
else
  P_q.append(w_0)
foreach $w_i \in P_q$ **do**
  if $w_i \notin P_t$ **then**
    P_t.append(w_i)
P_t.append(w_0)

Proof. Each vertex set, $V_a^1 \subset V^1$, corresponds to a delivery vertex $d_a \in V_d$. Lemma 1 proves that an edge $(w_i, w_j) \in E^1$, where $w_i \in V_a$ and $w_j \in V_b$, represents the lowest cost HDP route from d_a to d_b for a respective w_i and w_j. Thus the set of edges between all $w_i \in V_a$ and $w_j \in V_b$ will encode any optimal route between d_a and d_b, and this implies that any feasible GTSP solution on G^1 will correspond to a feasible HDP solution on graph G.

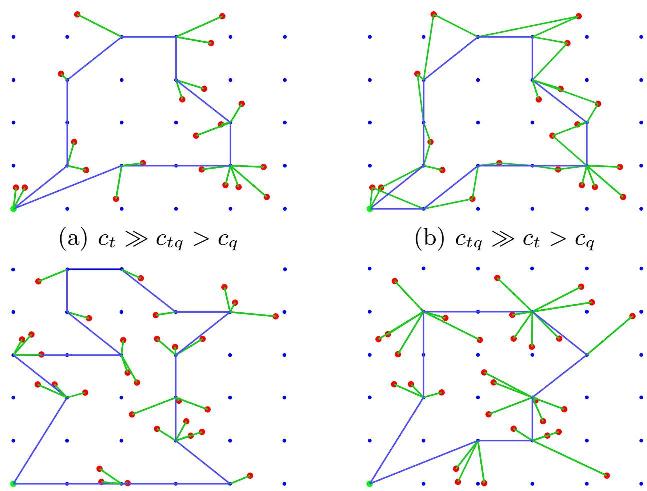
We prove that an optimal GTSP solution provides the optimal HDP solution, by contradiction, as follows. Consider an optimal GTSP solution of the form $P^1 = (w_0, w_1, \ldots, w_n, w_0)$. We know that each edge $(w_i, w_{i+1}) \in P^1$, where $w_i \in V_a$ and $w_{i+1} \in V_b$ represents an optimal 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. □

4.3 Characterizing the HDP Solution

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 cost functions 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 travelling 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).



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

(c) Low truck cost. (d) High truck cost.

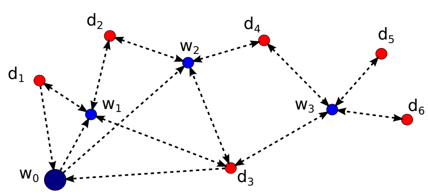
Fig. 6. HDP solution characterization based on c_q, c_t and c_{tq}. All figures show $w_0 = (0, 0)$, delivery points (red vertices), a gridded street network (blue vertices), the truck path (blue paths) and quadrotor flight paths (green paths).

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).

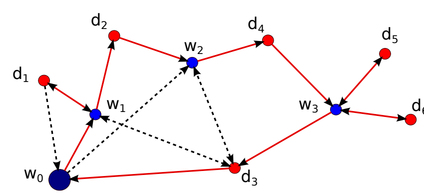
5 The Multiple Warehouse Delivery Problem (MWDP)

In this section, we further examine the special case of the HDP where all quadrotor deployments occur using pattern β, similar to Figure 6(b). A limiting case of this problem arises when $c_{tq}(e) = \infty$, and $c_t(e) = 0$ for all $e \in E$, thereby completely preventing docked travel and assuming that the truck travelling alone has a zero cost and 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 $\sum_{e \in E_t} c_t(e) = 0$. Note that a zero cost, infinite speed truck can be interpreted as a static warehouse at each street vertex and we will define a special case of the HDP: the Multiple Warehouse Delivery Problem (MWDP), where a set of delivery requests, $V_d = \{d_1, \dots, d_n\}$ must be fulfilled by a single vehicle from a set of warehouses $V_w = \{w_1, \dots, w_m\}$. Figure 7(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 2.



(a) Sample MWDP problem scenario.



(b) Optimal MWDP solution.

Fig. 7. The Multiple Warehouse Delivery Problem (MWDP).

Problem 2 (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.

The MWDP can be solved as an HDP using the methods in Section 4. However, the downside of this approach is that it results in an increase in the size of the problem instance as described in Section 4.1. Exploiting the simplifications in the MWDP, relative to the HDP, we present two improved solution approaches, first, a graph transformation of the MWDP into a TSP and, second, an exact algorithm to solve instances with a small, fixed number of warehouses.

5.1 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_i and d_j, and it passes through the warehouse vertex $w_a \in W_{d_j}$, such that $c(d_i, w_a) + c(w_a, d_j)$ 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 $e = (v_i, v_j)$, for all $v_i, v_j \in V^1$. Now for each edge, (v_i, v_j), we identify the warehouse $w_a \in W_{d_j}$ that minimizes $c(d_i, w_a) + c(w_a, d_j)$, and set the cost $c^1(v_i, v_j) = c(d_i, w_a) + c(w_a, d_j)$

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 [15] or LKH [10] 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_a, between each consecutive pair of vertices $\{v_i, v_j\}$ in P^1. An optimal MWDP solution is illustrated in Figure 7(b).

5.2 Kernel Sequence Enumeration (KSE) Algorithm

Figure 7(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 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, (i) a *localized delivery vertex*, d_i, for which $k_i = k_{i+1}$ and (ii) 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 are 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.
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 $k_{i+1} \neq k_i$.

Given these properties the following procedure gives us an exact algorithm for solving the problem:

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.

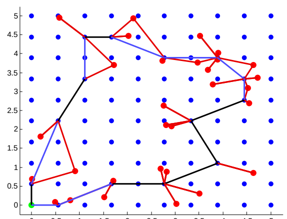
To complete each kernel sequence we must compute the closest warehouse for each remaining delivery point. Since there are at most m warehouses in the kernel sequence and n delivery points that are not in the kernel sequence, the complexity of each kernel completion step is $O(nm)$. Therefore, the total 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 a larger number of 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.

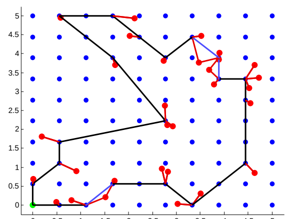
6 Simulation Results

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.

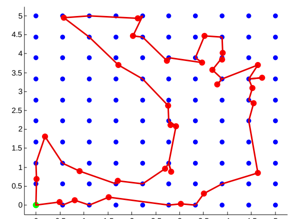
The first set of results in Figure 8, presents HDP solutions on a sample problem instance with 30 delivery points and a gridded terrain with 100 street vertices in an environment of arbitrary size r_{env}. The key simulation parameters are c_t,c_q,c_{tq} and r_q, the operating range of the quadrotor, 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.



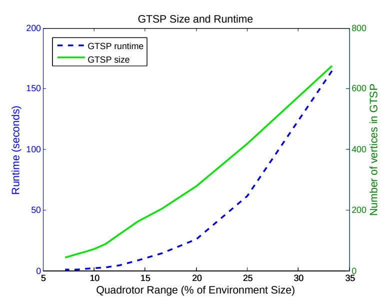
(a) HDP ($r_q = 0.3\, r_{env}$).



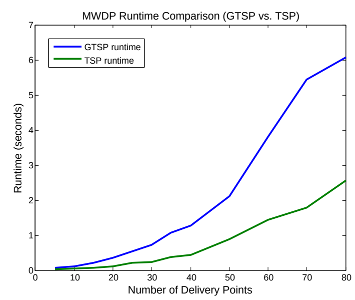
(b) HDP ($r_q = 0.1\, r_{env}$).



(c) MWDP TSP solution.



(d) GTSP performance.



(e) GTSP vs. TSP for MWDP.

Fig. 8. HDP simulation results and GTSP performance

In Figure 8(a), $r_q = 0.3\, r_{env}$, which resulted in a GTSP with 170 vertices and took 5.7 seconds to compute a solution. When r_q was reduced to $r_q = 0.1\, r_{env}$, the resulting GTSP contained 82 vertices and took 2.3 seconds to compute the solution, shown in Figure 8(b). From the Figures 8(a) and 8(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 4.3 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 8(c), for which the TSP method computes a solution in 0.45 seconds. To assess this further,

Delivery Points	Runtime			Solution Quality		
	GTSP	TSP	KSE	GTSP	TSP	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 1. MWDP algorithm comparison. $|V_w| = 3$.

Figure 8(d) shows the effect of the quadrotor range on the size (right y-axis) and runtime complexity (left y-axis) of the GTSP solution. Figure 8(e) 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.

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 with greater complexity and the TSP method stands out as the appropriate approach, as evident in Table 1, which shows runtime and solution quality results for an MWDP problem with $|V_w| = 3$ and an increasing number of delivery points.

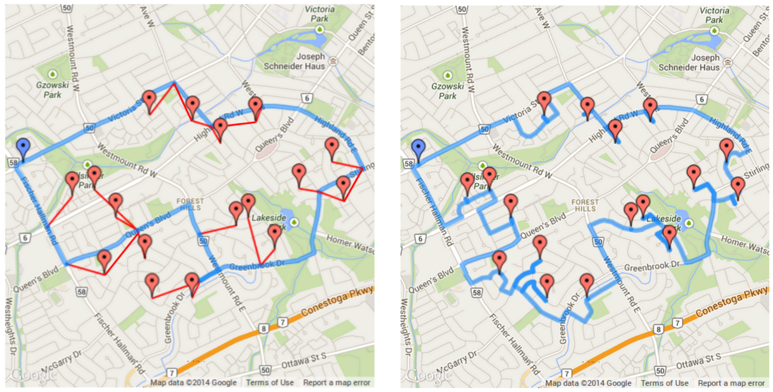
Finally, Figure 9 presents a realistic delivery scenario on a Google street map of a 15 km^2 area in a residential neighbourhood in Waterloo, Ontario, Canada. Figure 9(a) shows an HDP solution for 17 delivery points in contrast to a single delivery truck conducting deliveries in Figure 9(b). Given a maximum range of $150m$ for the quadrotor to ensure line of sight, the HDP solution in this problem instance results in a $\approx 50\%$ reduction in travel distance for the truck.

7 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, scheduled delivery requests, and dynamic scenarios where new requests arrive during execution.

References

1. Garone, E., Naldi, R., Casavola, A., Frazzoli, E.: Cooperative path planning for a class of carrier-vehicle systems. In: IEEE CDC. (Dec 2008) 2456–2462
2. Parker, L.: Current state of the art in distributed autonomous mobile robotics. In Parker, L.E., Bekey, G., Barhen, J., eds.: Distributed Autonomous Robotic Systems 4. Springer Japan (2000) 3–12



(a) HDP truck-quadrotor solution. (b) Truck delivery route.

Fig. 9. HDP Solution on a map of Waterloo, Ontario.

3. Pimenta, L., Kumar, V., Mesquita, R., Pereira, G.: Sensing and coverage for a network of heterogeneous robots. In: IEEE CDC. (2008) 1–8
4. Chand, P., Carnegie, D.A.: Mapping and exploration in a hierarchical heterogeneous multi-robot system using limited capability robots. Robotics and Autonomous Systems. **61**(6) (2013) 565 – 579
5. Mathew, N., Smith, S.L., Waslander, S.L.: A graph-based approach to multi-robot rendezvous for recharging in persistent tasks. In: ICRA. (May 2013) 3497–3502
6. Phan, C., Liu, H.: A cooperative UAV/UGV platform for wildfire detection and fighting. In: Int. Conf. on System Sim. and Scientific Computing. (2008) 494–498
7. Bae, J., Rathinam, S.: An approximation algorithm for a heterogeneous traveling salesman problem. In: ASME 2011 Dynamic Systems and Control Conf. and Bath/ASME Symposium on Fluid Power and Motion Control. (2011) 637–644
8. Oberlin, P., Rathinam, S., Darbha, S.: A transformation for a heterogeneous, multiple depot, multiple traveling salesman problem. In: ACC. (2009) 1292–1297
9. Noon, C.E., Bean, J.C.: An efficient transformation of the generalized traveling salesman problem. INFOR **31**(1) (1993) 39 – 44
10. Helsgaun, K.: General k-opt submoves for the Linkernighan TSP heuristic. Mathematical Programming Computation **1** (2009) 119–163
11. Nagy, G., Salhi, S.: Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. European Journal of Operational Research **162**(1) (2005) 126 – 141
12. Bullo, F., Frazzoli, E., Pavone, M., Savla, K., Smith, S.: Dynamic vehicle routing for robotic systems. Proceedings of the IEEE **99**(9) (Sept 2011) 1482–1504
13. Qu, Y., Bard, J.F.: The heterogeneous pickup and delivery problem with configurable vehicle capacity. Transportation Research Part C **32** (2013) 1–20
14. Korte, B., Vygen, J.: Combinatorial Optimization: Theory and Algorithms. 4 edn. Volume 21 of Algorithmics and Combinatorics. Springer (2007)
15. Lin, S., Kernighan, B.W.: An effective heuristic algorithm for the traveling-salesman problem. Operations Research **21**(2) (1973) pp. 498–516