Optimizing for Transfers in a Multi-Vehicle Collection and Delivery Problem

Brian Coltin and Manuela Veloso

Abstract We address the Collection and Delivery Problem (CDP) with multiple vehicles, such that each collects a set of items at different locations and delivers them to a dropoff point. The goal is to minimize either delivery time or the total distance traveled. We introduce an extension to the CDP: what if a vehicle can transfer items to another vehicle before making the final delivery? By dividing the labor among multiple vehicles, the delivery time and cost may be reduced. However, introducing transfers increases the number of feasible schedules exponentially. In this paper, we investigate this Collection and Delivery Problem with Transfers (CDP-T), discuss its theoretical underpinnings, and introduce a two-approximate polynomial time algorithm to minimize total distance travelled. Furthermore, we show that allowing transfers to take place at any location for the CDP-T results in at most a factor of two improvement. We demonstrate our approximation algorithms on large simulated problem instances. Finally, we deploy our algorithms on robots that transfer and deliver items autonomously in an office building.

1 Introduction

Previously, we deployed a team of robots in an office building. The robots navigate autonomously on multiple floors to complete tasks requested by users at a centralized website. These tasks include retrieving and delivering objects. For example, users can ask the robots to bring them printouts from the printer or coffee from the kitchen, or to transfer written messages, USB sticks, or other items between offices.

Forming a schedule for the robots to fulfill all requests is an instance of the Vehicle Routing Problem (VRP). In this work, we propose and examine an extension of the VRP: what if the robots can *transfer* items between each other? By having one robot pick up an object and transfer it to a different robot (or a series of robots) for delivery, we can conserve both time and the battery life of the robots. Transferring items makes the problem significantly harder because this creates exponentially more possible schedules. Scheduling with transfers has potential energy-saving and productivity-increasing applications in both transportation domains (e.g., taxis that exchange occupants heading to nearby locations) and in warehouse automation.

Brian Coltin
The Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, e-mail: bcoltin@cs.cmu.edu

Manuela Veloso
Computer Science Department, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, e-mail: veloso@cs.cmu.edu

In particular, we examine the role of transfers in the *collection and delivery problem* (CDP), a subproblem of the VRP in which the robots retrieve a set of items and deliver them to a single central location. In our building, popular requests for the robots include retrieving mail from the secretaries' offices and delivering it to the central office, as well as delivering business cards, fliers or candy to many offices and returning the leftovers. Both problems are instances of the CDP. We call the CDP with transfers the CDP-T. The CDP-T is NP-hard, so we introduce algorithms both to solve it optimally and approximately.

In this paper, we first discuss selected related work, introduce the CDP-T, and formulate the problem as a delivery tree. We consider CDP with transfers anywhere (CDP-TA), where transfers are not limited to pickup locations, and show that the optimal solution to the CDP-T costs at most twice that of the CDP-TA in a metric space. We propose a two-approximate polynomial time algorithm for the CDP-T, and a metaheuristic to improve on this solution. We show the effectiveness of these algorithms in simulation and on physical robots, and demonstrate an approach for two robots to autonomously transfer objects. To our knowledge, this is the first time multiple robots have created and executed a schedule with transfers.

2 Related Work

Extensive research has been done on the Vehicle Routing Problem (VRP) in which a set of vehicles visit a set of locations to service customer requests. Variants of the VRP include the Pickup and Delivery Problem (PDP), in which a set of goods must be picked up from one set of locations and delivered to another set; and the Capacitated PDP (CPDP). The VRP and its variants are generally NP-hard, and are often solved optimally with branch and bound algorithms such as mixed integer programming. See [19] for an extensive discussion of the VRP.

A plethora of approaches have been proposed to solve the PDP, including branch and bound methods, heuristics, and metaheuristics [3, 16]. The Collection and Delivery Problem (CDP) is a subproblem of the PDP in which there is only a single delivery point. Approximation algorithms with guarantees have been developed for the VRP with release times at which jobs can first be performed but without deadlines [4], for the VRP with time windows [2], and for the CPDP [6, 11].

Less work has focused on robots that transfer items. Mail couriers and the transportation industries use fixed "transshipment" points, hubs such as post offices and airports, where objects are deposited for another agent to retrieve. Algorithms have been developed to find the optimal solution [8] and heuristics [14] for the PDP with fixed transshipment points, and for the CPDP with time windows and a single transshipment point [15]. Solving the PDP with online requests to minimize delivery delay has been considered in [20] with a single relay between the pickup and delivery vehicles. In [18], transfers at any pickup or delivery location are considered in the construction of heuristics for the PDP with time windows. In [10], an approximation is given for the preemptive CPDP, where objects are dropped off at pickup points and retrieved by other vehicles later.

Robotics researchers have extensively studied the task allocation problem, in which robots are assigned to (independent) tasks to maximize utility. In the VRP and CDP, tasks are not independent [9]. Other researchers have devised algorithms to find the optimal rendezvous locations given a schedule of meetings, which could be used in conjunction with the schedules we create [1].

In this work, we consider a version of the PDP with transfers and only a single dropoff point, and provide a constant factor approximation algorithm. We prove that this is also a constant factor approximation to the CDP-TA, where transfers take place at any location (see Section 3.4). We develop further heuristics to improve these solutions and demonstrate their effectiveness in real-world scenarios. To our knowledge, we are the first to deploy robots which plan and execute a schedule with autonomous transfers.

3 The Collection and Delivery Problem with Transfers

We are given a set of robots, located at positions $R \subseteq M$, where M is a metric space with distance metric d. The robots must retrieve items from pickup locations $L \subseteq M$, then deliver all the objects to a final drop-off location $f \in M$. Robots may meet up to transfer objects, but only at the pickup locations. We assume that the robots have infinite capacity.

We construct the complete graph $G = (V, E)$ where $V = R \cup L \cup \{f\}$ and the edge weights are defined by the metric d. The goal is to find a path P_i on G for each robot such that all the objects are delivered. In a valid solution, the endpoint of every path P_i is either f or, in the case of a transfer, an intermediate node in another robot's path. Every vertex $l \in L$ is part of at least one path P_i since every item is delivered.

Many such sets of paths exist, and we consider two objectives to distinguish between them: a) minimize the total movement energy of the robots, corresponding to the sum of edge weights, and b) minimize the completion time, corresponding to the length of the longest path from any robot to the destination.

Now, consider the directed graph T which is the union of all the paths of used robots P_i, where the paths are chosen to either minimize the total distance travelled or the maximum distance travelled to deliver any object.

Claim. There exists an optimal directed graph T that is a tree.

Proof. T is a tree iff it is connected and has $|E| = |V| - 1$ edges. T is connected, since every pickup location and used robot is part of a path to f. Every robot route P_i either ends at f or at a transfer point. If not, travelling to the final location on the path is either needless or retrieves an item that is not delivered to f. Furthermore, every route P_i contains f or a transfer point where items are given to a different robot nowhere else in the path. If these points do exist elsewhere, a solution of less than or equal cost can be constructed by removing the points due to the triangle inequality. Any items transferred earlier can still be delivered by transferring them at the route's final point instead. Hence, there is an optimal set of paths where each robot transfers or delivers items exactly once at the end of its path. Furthermore, there is an optimal solution where each node $l \in L$ is not the endpoint of exactly one path P_i, as otherwise a point could be omitted from one of the paths to construct

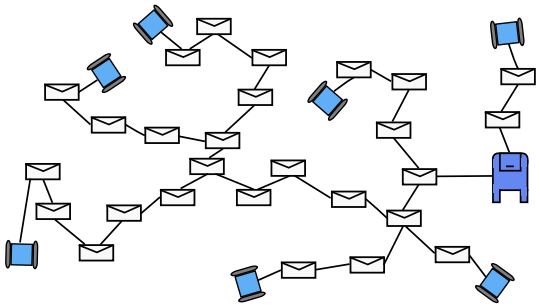


Fig. 1 The optimal solution to the CDP-T can be formulated as a delivery tree. Edges represent the motion of a single robot. Where the tree branches, all the robots at that branch point transfer their entire load to a single robot which continues alone.

a solution with no worse cost. So there exists a set of paths T which has $|E| = |L| + |R \cap T|$ edges, as each vertex aside from f is not the endpoint of exactly one path P_i. T has $|V| = |L| + |R \cap T| + 1$ vertices, and is hence a tree.

An equivalent formulation for the CDP-T is to construct a *delivery tree* D with the following properties (see Fig. 1):

1. The interior nodes are the pickup locations $l \in L$ and the final delivery point f.
2. The leaf nodes are a subset of R, the starting locations of the used robots.
3. Branch points represent transfers of one robot's load to another. All but one robot at a transfer point remain behind and are not used again.

Our goal is to either find the delivery tree of minimum weight $w(D)$ (minimum movement cost), of minimum weighted depth $\texttt{depth}(d)$ (minimum delivery time), or to minimize a linear combination $\alpha w(D) + (1-\alpha)\texttt{depth}(D)$ where $0 \leq \alpha \leq 1$.

We show that the CDP-T problem is NP-hard by reducing the TSP to the CDP-T. In the TSP, a salesman aims to find the minimum distance Hamiltonian tour that visits every city in a set V exactly once, and return to the initial city. The TSP can be reduced to the CDP-T problem by setting the cities V as the pickup locations L. We have one robot r, which begins at the same location as the travelling salesman. We set the dropoff location f to be the same location, r. So if the CDP-T can be solved in polynomial time, so can the TSP. Hence the CDP-T is NP-hard.

3.1 Optimal Approach

The CDP-T can be formulated as a mixed integer program. We solve for binary variables $x_{a,b}$, which indicate whether the directed edge from node a to node b is in the solution. The final destination f has zero outgoing edges ($\sum_{v \in V} x_{f,v} = 0$), each location $l \in L$ has exactly one outgoing edge ($\forall l \in L \ \sum_{v \in V} x_{l,v} = 1$), and each robot $r \in R$ has zero incoming edges and at most one outgoing edge, but may have zero ($\forall r \in R \ \sum_{v \in V} x_{r,v} \leq 1, \sum_{v \in V} x_{v,r} = 0$). The vertex f and each point $l \in L$ have at least one incoming edge ($\forall n \in (L \cup \{f\}) \ \sum_{v \in V} x_{v,n} \geq 1$).

The formulation as it stands still allows subtrees, in which the delivery tree is not connected. To address this, we introduce constraints similar to the subtour elimination constraints that are used to formulate the TSP as an MIP: $\forall U \subset V, \ U \neq \emptyset \ \sum_{e \in \delta(U)} x_e \geq 1$, where $\delta(U)$ is the set of edges which link U and $V \setminus U$.

This MIP formulation gives paths which solve the CDP-T. The graph is connected (aside from unused robots), so every item is part of a robot's path to f. The

Algorithm 1 `cdp_t`(L,f,R): Construct a delivery tree given the set of pickup points L, the dropoff point f, and robots R. $N_T(v)$ gives the set of neighbors of v in T.

$G \leftarrow$ `complete_graph`$(L \cup \{f\}, d)$
$T \leftarrow$ `mst`(G)
$s, v \leftarrow \text{argmin}_{s \in R, v \in V(G)}\, d(s,v)$
$T' \leftarrow$ `deliv_tree`$(R, L, f, T \cup$ `edge`$(s,v))$
for $r \in R, r \notin T'$ **do**
  $v \leftarrow \text{argmin}_{v \in V(G), N_{T'}(v) \cap R = \emptyset}\, d(r,v)$
  $T'' \leftarrow$ `deliv_tree`$(R, L,$ `edge`$(r,v))$
  if `cost`$(T'') <$ `cost`(T') **then**
    $T' \leftarrow T''$
  end if
end for
return T'

Algorithm 2 `deliv_tree`(R,L,f,T): Construct a delivery tree given the set of pickup points L, the dropoff point f, robots R, and an intermediate delivery tree T.

$A = \{l \in L : \exists r \in R \text{ s.t. } l \in$ `path(r, f)`$\}$
$T' =$ Null Graph
for $r \in R$ **do**
  $n = r$, $P = []$
  while $n \notin T'$ **do**
    Append n to P
    If $n = f$, break
    Choose $n' = v \in N_T(n)$ s.t. $v \notin P$ and $v \notin A$
    If $\nexists n',\ n' = v \in N_T(n)$ s.t. $v \notin P$ and $v \in A$
    $n = n'$
  end while
  $P' = P$ with duplicate vertices removed
  $T' = T' \cup P$
end for
return T'

graph has $|R| + |L| = |V| - 1$ edges since each used robot and item has exactly one outgoing edge, and is hence a tree. The robots are the only leaf nodes in the tree, since they have one edge while the items have at least two. We minimize the total distance $D = \sum_{e \in E} d(e)$ travelled by all the robots.

Alterntiavely, to minimize the time taken to deliver all items (the length of the longest path from a robot to the goal) we define binary variables $p_{e,r}$ which indicate whether the directed edge e is part of the path from r to f. Edges from robots are part of that robot's path if the edge exists. No edges exist *to* robots, and f has no outgoing edges. We add the constraints $\forall r \in R,\ l \in L \quad p_{e_{r,l},r} = x_{r,l},\ p_{e_{l,r},r} = 0,\ p_{e_{f,l},r} = 0$. Edges with robots as nodes are not part of another robot's extended path: $\forall r_1, r_2 \in R, l \in L \cup \{f\}\ p_{e_{r_2,l},r_1} = 0$. Each edge between retrieval and delivery points that is in the solution is part of the path for at least one robot (or more, with transfers). $\forall l_1, l_2 \in L \cup \{f\}, l_1 \neq l_2\ \sum_{r \in R} p_{e_{l_1,l_2},r} \geq x_{l_1,l_2}$. Finally, edges between two retrieval or delivery locations are only on a robot's path to f if an incoming node was also on the extended path. $\forall l_1 \in L, l_2 \in L \cup \{f\}, l_1 \neq l_2, r \in R,\ p_{e_{l_1,l_2},r} \geq x_{l_1,l_2} - 1 + \sum_{l_3 \in L \setminus l_1, l_2 \cup \{f\}} p_{e_{l_3,l_1},r}$.

We define a variable G to be the length of the longest extended path. Then we add constraints so that G is greater than the length of every robot's extended path: $\forall r \in R\ G \geq \sum_{e \in E} p_{e,r} d(e)$. Our objective function is then $\min G$, or a weighted sum of the two objectives, $\min \alpha D + (1-\alpha)G$.

3.2 Minimum Length Approximation

Computing the exact solution to the CDP-T is difficult since the problem is NP-hard. Hence we are interested in approximation algorithms, which find a solution in polynomial time that is not optimal, but provably close to optimal.

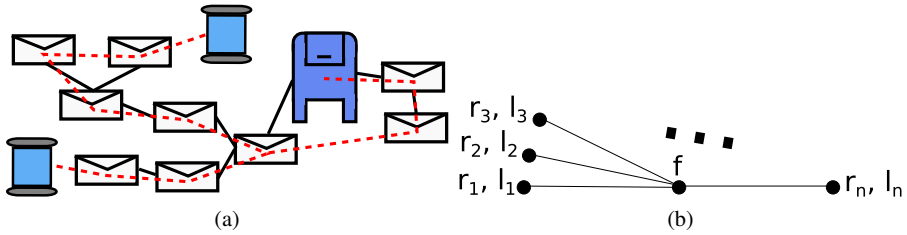


Fig. 2 (a) An example of the approximation algorithm. Solid black lines indicate edges on the minimum spanning tree, and dashed lines show the generated delivery tree. (b) Robots and items are situated at the end of n hallways of length one emanating from the delivery point f. $O_n = n$, where every robots travels to f. $O_1 = 2n-1$, where one robot retrieves every item.

The approximation algorithm we introduce generates two-approximate solutions in terms of total distance to the CDP-T using only a single robot. In fact, a two-approximate solution to the CDP-T is the best guarantee we can possibly make with only a *single* robot. If O_n is the optimal solution with n robots, and O_1 the optimal solution with any one of those robots, Fig. 2b shows a case where $w(O_1)$ approaches arbitrarily close to $2w(O_n)$. With multiple robots, the heuristic further reduces both the distance travelled and the delivery time.

Our approximation is based on the minimum spanning tree two-approximate heuristic for the TSP, but is extended for multiple robots that transfer items. The algorithm is shown in Algorithm 1. First, we construct the complete graph G with pickup locations L and drop-off location f with edge lengths determined by the distance metric d, and its minimum spanning tree T. Next, choose the edge e of lowest weight from a node in T to a node in $r \in R$, and add e and r to T to construct T'.

If O is the delivery tree for the optimal solution, then $w(T) \leq w(O \setminus R)$, since T is the minimum spanning tree over the same nodes. Since a valid delivery tree must have at least one edge connected to a robot, and we chose the minimum one, $w(T') \leq w(O)$. We call T' an *intermediate delivery tree*, since it can be used to construct a delivery tree but not all leaf nodes are robots.

We then construct a tour P, starting at r and ending at f, which visits each vertex in T' at least once with the procedure `deliv_tree` (see Alg. 2). When T' contains only a single robot, this algorithm is equivalent to the two-approximate TSP approximation, which traverses each edge at most twice. The `deliv_tree`(T) algorithm extends this heuristic to multiple robots which can transfer items, and creates a valid delivery tree with weight at most $2w(T)$. Thus, $w(P) \leq 2w(T) + w(e) \leq 2w(O \setminus R) + w(e) \leq 2w(O \setminus R)$. So our single robot algorithm is two-approximate to the optimal multi-robot solution in terms of total distance.

We can lower the cost further with multiple robots, although no better guarantees on the approximation bound are obtained. We begin with the constructed intermediate delivery tree for a single robot, T', and for each robot r, greedily add the shortest edge from r to a node in T to the tree T'. We construct a new delivery tree from T' with `deliv_tree`, and keep the edge and robot in the intermediate delivery tree T' if and only if the delivery tree's cost decreases according to our objective function. We then attempt to add the next robot to the updated T', iterating through

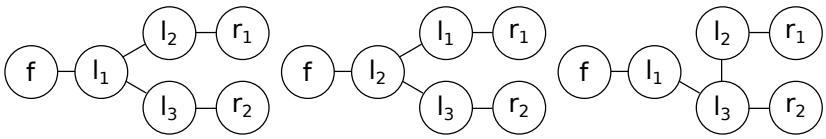


Fig. 3 (a) The initial state. (b) A neighbor found by swapping l_1 and l_2 (the edges linking them to other nodes are different). (c) A neighbor with l_2 grafted from l_1 to l_3.

every robot. This procedure still gives a two-approximation to the CDP, and additional robots will sometimes decrease both the total distance travelled and the time to completion, depending on the problem instance. See Fig. 2a for an example of the algorithm's results.

We have constructed a two-approximate delivery tree in polynomial time that uses multiple robots, since the cost of the multi-robot heuristic is at most the cost of the single-robot heuristic which is two-approximate.

3.3 Improvement with Local Search

Next, we introduce a metaheuristic to improve upon the two-approximate solution with local search techniques. Specifically, we make use of simulated annealing [13]. Simulated annealing is a metaheuristic that begins at some state, and chooses a random "neighbor" of that state. With probability $accept(e,e',t)$ the new state is accepted as the current state, where e is the "energy" (in our case, the cost) of the current state, e' is the energy of the new state, and t is the temperature, or the fraction of iterations of the algorithm currently completed. If the new state is rejected we remain at the current state and repeat with a new neighbor. The algorithm continues either for a fixed number of iterations or until the energy crosses some threshold, when the best solution that has been encountered thus far is returned.

To apply simulated annealing to the CDP, we must define a starting point, an energy function, an acceptance probability, and a function to return random neighbors of a state. We search over the underconstrained intermediate delivery trees rather than strict delivery trees, as this allows us to develop a broader concept of neighboring solutions that is more closely tied to the approximation heuristic. We use as a starting point the tree generated by our fast multi-robot heuristic.

The energy function is the cost function of the delivery tree constructed with `deliv_tree`, and it incorporates both the total weight and depth of the tree as a function of α. The acceptance probability is 1 if $e' < e$, and $e^{\frac{e-e'}{t}}$ otherwise, a standard acceptance function frequently used in the literature.

Neighboring states are found either by *swapping* two non-robot, non-destination nodes on the intermediate delivery tree (that are not robots or f), swapping their neighbor sets, or by *grafting* one branch of the tree at a transfer point onto a neighboring node, replacing a single edge. See Fig. 3 for examples.

We run the simulated annealing algorithm for a thousand iterations. Every hundred iterations we restart from the best solution found thus far to explore the most

promising regions more thoroughly. In practice, simulated annealing improves upon the solutions of the two-approximate heuristic.

3.4 Transfers at Any Location

Next, we consider the general case of CDP-TA, where items can be transferred anywhere, rather than only at vertices in L as in the CDP-T. We show that the optimal solution for the metric CDP-T, O_v, is within a factor of two of the cost of the optimal solution to the CDP-TA, O_a, when minimizing total distance travelled.

The CDP-TA is closely related to the Steiner tree problem, in which a set of points must be connected by the shortest possible set of edges. "Helper" points may be added (transfer points) to construct a solution. The CDP-TA is different in that all the leaf nodes of a valid delivery tree must be robots or the final destination.

Claim. $O_v \leq 2O_a$

Proof. Given a delivery tree T with optimal cost O_a with transfers at any point (the transfer points are separate nodes in the tree), we construct a delivery tree T' with transfers only at vertices and cost $O_v \leq 2O_a$. Let X be the set of transfer points in T.

First, construct the forest F with vertices $X \cup N(X)$, where $N(X)$ is the set of neighboring vertices to X in T. Add all edges between the vertices of F in T to F as well. For each connected component F_i of F (which may include multiple transfer points), define S_i to be the complete graph with vertices $F_i \setminus X$. Then $w(F_i) = w(\texttt{steiner}(S_i))$, where `steiner` gives the minimum Steiner tree. If this were not the case, then either the minimum Steiner tree could be used to construct a delivery tree of lower cost, or we do not have the minimum Steiner tree.

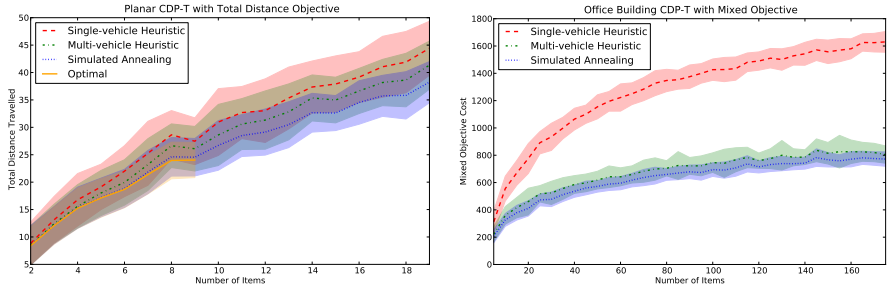
Let $S = \cup S_i$. Construct $T' = (T \setminus X) \cup \texttt{msf}(S)$, where $\texttt{msf(S)} = \cup_i \texttt{mst}(S_i)$. T' is a valid delivery tree where the items are transferred at retrieval or dropoff points.

Then $O_a = w(T) = w(T \setminus X) + w(F)$, since $T = (T \setminus X) \cup F$. Furthermore, $w(F) = \sum w(F_i) = \sum w(\texttt{steiner}(F_i \setminus X))$. Similarly, $O_v \leq w(T') = w(T \setminus X) + w(\texttt{msf}(S)) = w(T \setminus X) + \sum_i w(\texttt{mst}(F_i \setminus X))$. The weight of a minimum spanning tree is bounded by the Steiner ratio κ such that $w(\texttt{mst}(G)) \leq \kappa w(\texttt{steiner}(G))$. Hence, $O_v \leq w(T \setminus E) + \kappa \sum_i R_i \texttt{steiner}(R_i \setminus E) \leq \kappa O_a$.

If our distance function is a metric, than $\kappa = 2$ [17], and the optimal solution to the CDP-T is two-approximate to the CDP-TA. If the distance function is Euclidean, κ is conjectured to be $\frac{2}{\sqrt{3}} \approx 1.1547$. This conjecture is believed to be true, but is still open and unproven [12]. If it holds, then for the Euclidean case CDP-T is a $\frac{2}{\sqrt{3}}$-approximation to CDP-TA. Hence, our two-approximate heuristic for the CDP-T provides a four-approximate heuristic for metric CDP-TA and a $\frac{4}{\sqrt{3}} \approx 2.31$-approximate for the Euclidean CDP-TA, assuming the conjecture regarding the Euclidean Steiner ratio holds.

4 Simulation Results

To validate our approach, we tested the algorithms in two scenarios. We varied the number of items to retrieve, and created 50 random problem instances for each set



(a) (b)

Fig. 4 Results for the CDP-T algorithms with (a) three robots and up to twenty items to retrieve under the total distance objective function in the planar domain, and (b) 25 robots and up to 175 items to retrieve under the mixed objective function in the office building domain. Lines indicate the mean objective function cost, and the filled area shows the standard deviation across trials. Darker areas indicate overlap.

of parameters. We solved each problem optimally (when feasible), with the single and multi-robot two-approximation algorithms, and with simulated annealing.

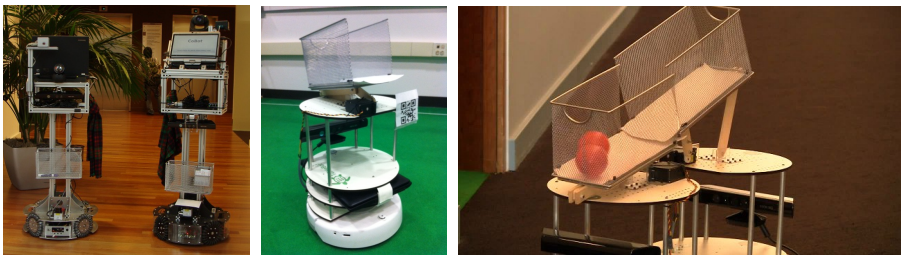
In the first scenario, we chose random pickup, delivery, and starting points from a plane, using a Euclidean distance function. Figure 4a shows results for three robots and up to twenty items. Each phase of the algorithm gives an improvement. The simulated annealing solution is near-optimal, when the optimal solution is found.

In the second scenario, points were chosen randomly from four floors of an office building in a simulation of the mail collection task. The distance function was an empirical estimate of robot travel times. The objective was to minimize a weighted sum of the moment energy cost and the delivery time ($\alpha = 0.5$). Figure 4b shows the results with 25 robots and up to 175 items. Each level of the algorithm offers a significant improvement, particularly the multi-vehicle heuristic, since it achieves much better depths than the single robot approximation. We are unable to find the optimal solutions for problems of this size.

On an Intel 2.83 GHz Core2 Quad CPU, the two heuristics ran in under a second for all instances of up to 500 vertices (these instances are not shown). Simulated annealing ran in under fifteen seconds for every instance. The optimal MIP formulations were solved with `lpsolve`, with each attempt aborted after thirty seconds. These results show that the approximation algorithms can solve large problems quickly, and provide near-optimal solutions when comparison is possible.

5 Illustrative Deployments

We have deployed our CDP-T heuristic on two robot platforms: the CoBot and CreBot robots (see Fig. 5). The CoBots autonomously navigate over large distances indoors, and with them we show that transferring items can reduce energy consumption and delivery time. The CoBots cannot physically transfer items, and so they ask humans to help. With the CreBots, we demonstrate that autonomous transfers are feasible by presenting a method to transfer items with a tilting tray.



(a) (b) (c)

Fig. 5 (a) Two CoBot robots. (b) A CreBot robot, with a Create base, laptop computer, Kinect RGB-D camera, tilting tray, and QR code for alignment. (c) One CreBot transfers an item to another during a collection and delivery task.

Both the CoBots and CreBots autonomously localize and navigate in the building using software developed for the CoBots [5]. A centralized web server accepts user requests and sends schedules to the robots [7]. The robots collect and deliver items by arriving at an office and asking a human to place or remove the items.

5.1 Transfers with CoBots

We examine three scenarios where two CoBots were deployed to collect and deliver objects in an office building. The problem setups and the planned paths, both with transfers and without, are shown in Fig. 6. For each scenario, we recorded the time it took the robot(s) to complete the task and the total combined distance traveled by all of the robots. To reduce the variance in our experiments, we assumed that humans were always immediately available to place items in the CoBots' baskets and to help transfer items. However, the times recorded with transfers do include the time to ask and thank a human for their help, and are shown in Table 1.

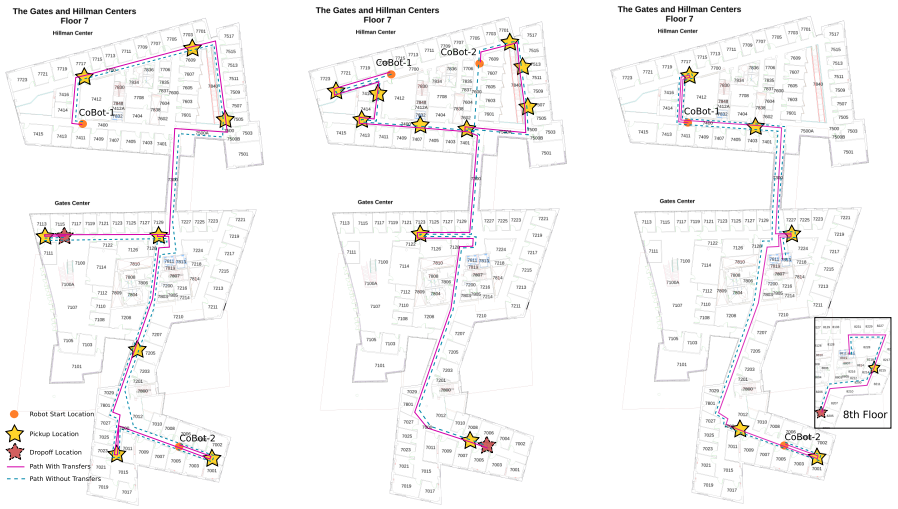
	With Transfers		Single Robot	
Scenario	Time (min.)	Dist.(m)	Time (min.)	Dist. (m)
1	4:22	229.35	8:18	287.12
2	5:52	220.68	7:55	238.66
3	7:00	230.56	9:36	272.37

Table 1 Deployment Results for Selected Scenarios

In all three scenarios, using multiple robots with transfers reduced both the time to complete the task and the total distance traveled. This may not be the case in all scenarios, depending on the problem setup. However, we have demonstrated that in many scenarios, transferring items between robots can and does reduce the energy consumed and the time taken to complete the task.

5.2 Autonomous Transfers

We have designed the CreBots to transfer items autonomously. They consist of an iRobot Create as a base, with Willow Garage Turtlebot shelves, a laptop, Kinect RGB-D camera, and a custom-built tilting tray on top to transfer items (see Fig. 5b).



(a) (b) (c)

Fig. 6 The paths planned by the approximation algorithm and taken by the robots, with and without transfers, for (a) Scenario 1, (b) Scenario 2, and (c) Scenario 3.

Unlike the CoBots, the CreBots transfer items autonomously. To do so, two CreBots head towards the same location based on their localization information. The robots stop either when they reach the destination, or are blocked within two meters of the destination (presumably by the other robot). Next, the robots send each other their localization positions wirelessly and turn to face each other.

Localization is accurate and robust enough for a rough alignment, but not precise enough to transfer objects based solely on localization. For fine alignment, the robots are each equipped with a QR code which is detected by the Kinect and used for precise docking. The transferrer advances slowly until it comes within the range it can detect the QR code using an off the shelf library, which measures the QR code's bounding box. The transferrer computes the distance and angle to the QR code from its known size and camera parameters, then rotates in place to align with the QR code. The transferrer continues to move forwards until its bump sensor is triggered, and dumps its load. When dumping, the tilting tray shakes back and forth to ensure that all the objects are dislodged. The CreBots have successfully executed collection and delivery tasks with transfers.[1]

6 Conclusion

We have introduced the Collection and Delivery Problem with Transfers, and shown that the solution comes in the form of a delivery tree. We solved the CDP-T optimally with an MIP, and bounded the cost with transfers anywhere in terms of the cost when transfers occur only at vertices. We introduced a two-approximate algorithm

[1] Video available at http://youtu.be/pzXv7p_aZhE.

under the minimum length objective, and proposed a heuristic and metaheuristic to find solutions of lower depth while maintaining the bound on total length. Furthermore, we demonstrated the effectiveness of these heuristics on large real-world problem instances, and showed the feasibility of transfers between physical robots.

References

1. Alton, K., Mitchell, I.: Efficient dynamic programming for optimal multi-location robot rendezvous. In: IEEE Conference on Decision and Control, pp. 2794–2799. IEEE (2008)
2. Bansal, N., Blum, A., Chawla, S., Meyerson, A.: Approximation algorithms for deadline-tsp and vehicle routing with time-windows. In: Proceedings of the ACM symposium on Theory of computing, pp. 166–174. ACM (2004)
3. Berbeglia, G., Cordeau, J., Gribkovskaia, I., Laporte, G.: Static pickup and delivery problems: a classification scheme and survey. Top **15**(1), 1–31 (2007)
4. Bhattacharya, B., Hu, Y.: Approximation algorithms for the multi-vehicle scheduling problem. Algorithms and Computation pp. 192–205 (2010)
5. Biswas, J., Coltin, B., Veloso, M.: Corrective gradient refinement for mobile robot localization. In: Proc. of IEEE Conf. on Intelligent Robots and Systems (IROS), pp. 73–78. IEEE (2011)
6. Charikar, M., Raghavachari, B.: The finite capacity dial-a-ride problem. In: Proc. of 39th Annual Symposium on Foundations of Computer Science, 1998, pp. 458–467. IEEE (1998)
7. Coltin, B., Veloso, M., Ventura, R.: Dynamic user task scheduling for mobile robots. In: Proc. of the Work. on Automated Action Planning for Autonomous Mobile Robots, AAAI (2011)
8. Cortés, C., Matamala, M., Contardo, C.: The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. European Journal of Operational Research **200**(3), 711–724 (2010)
9. Gerkey, B., Matarić, M.: A formal analysis and taxonomy of task allocation in multi-robot systems. The International Journal of Robotics Research **23**(9), 939–954 (2004)
10. Gørtz, I., Nagarajan, V., Ravi, R.: Minimum makespan multi-vehicle dial-a-ride. Algorithms-ESA 2009 pp. 540–552 (2009)
11. Gupta, A., Hajiaghayi, M., Nagarajan, V., Ravi, R.: Dial a ride from k-forest. ACM Transactions on Algorithms (TALG) **6**(2), 41 (2010)
12. Ivanov, A., Tuzhilin, A.: The steiner ratio gilbert–pollak conjecture is still open. Algorithmica pp. 1–3 (2011)
13. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science **220**(4598), 671 (1983)
14. Mitrovic-Minic, S., Laporte, G.: The pickup and delivery problem with time windows and transshipment. Information Systems and Operational Research **44**(3), 217–228 (2006)
15. Nakao, Y., Nagamochi, H.: Worst case analysis for a pickup and delivery problem with single transfer. Numerical Optimization methods, theory and applications **1584**, 142–148 (2008)
16. Parragh, S., Doerner, K., Hartl, R.: A survey on pickup and delivery problems. Journal für Betriebswirtschaft **58**(2), 81–117 (2008)
17. Takahashi, H., Matsuyama, A.: An approximate solution for the steiner problem in graphs. Math. Japonica **24**(6), 573–577 (1980)
18. Thangiah, S., Fergany, A., Awan, S.: Real-time split-delivery pickup and delivery time window problems with transfers. Central European Journal of Op. Research **15**(4), 329–349 (2007)
19. Toth, P., Vigo, D.: The vehicle routing problem, vol. 9. Soc. for Industrial Mathematics (2002)
20. Waisanen, H., Shah, D., Dahleh, M.: Fundamental performance limits for multi-stage vehicle routing problems. Operations Research (2007)

Acknowledgements This research was partially sponsored by the Office of Naval Research under grant number N00014-09-1-1031, and by the National Science Foundation under award number NSF IIS-1012733 . The views and conclusions expressed are those of the authors only.

Special thanks to Thomas Charley for designing and building the CreBots' tilting trays.