Collaborative Delivery on a Fixed Path with Homogeneous Energy-Constrained Agents*

Jérémie Chalopin[1], Shantanu Das[1], Yann Disser[2], Arnaud Labourel[1], and Matúš Mihalák[3]

[1] Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
{shantanu.das,jeremie.chalopin,arnaud.labourel}@lis-lab.fr
[2] Dept. of Mathematics, TU Darmstadt, Germany
disser@mathematik.tu-darmstadt.de
[3] Dept. of Data Science and Knowledge Engg, Maastricht University, Netherlands
matus.mihalak@maastrichtuniversity.nl

Abstract. We consider the problem of collectively delivering a package from a specified source to a designated target location in a graph, using multiple mobile agents. Each agent starts from a distinct vertex of the graph, and can move along the edges of the graph carrying the package. However, each agent has limited energy budget allowing it to traverse a path of bounded length B; thus, multiple agents need to collaborate to move the package to its destination. Given the positions of the agents in the graph and their energy budgets, the problem of finding a feasible movement schedule is called the *Collaborative Delivery* problem and has been studied before.

One of the open questions from previous results is what happens when the delivery must follow a fixed path given in advance. Although this special contraint reduces the search space for feasible solutions, the problem of finding a feasible schedule remains NP hard (as the original problem). We consider the optimization version of the problem that asks for the optimal energy budget B per agent which allows for a feasible delivery schedule, given the initial positions of the agents. We show the existence of better approximations for the fixed-path version of the problem (at least for the restricted case of single pickup per agent), compared to the known results for the general version of the problem, thus answering the open question from the previous paper.

We provide polynomial time approximation algorithms for both directed and undirected graphs, and establish hardness of approximation for the directed case. Note that the fixed path version of collaborative delivery requires completely different techniques since a single agent may be used multiple times, unlike the general version of collaborative delivery studied before. We show that restricting each agent to a single pickup allows better approximations for fixed path collaborative delivery compared to the original problem. Finally, we provide a polynomial time algorithm for determining a feasible delivery strategy, if any exists, for a given budget B when the number of available agents is bounded by a constant.

* This work was partially supported by the ANR project ANCOR (anr-14-CE36-0002-01)

1 Introduction

We consider a team of mobile agents which need to collaboratively deliver a package from a source location to a destination. The difficulty of collaboration can be due to several limitations of the agents, such as limited communication, restricted vision or the lack of persistent memory, and this has been the subject of extensive research (see [21] for a recent survey). When considering agents that move physically (such as mobile robots or automated vehicles), a major limitation of the agents are their energy resources, which restricts the distance that the robot can travel. This is particularly true for small battery operated robots or drones, for which the energy limitation is the real bottleneck. We consider a set of mobile agents where each agent i has a budget B_i on the distance it can move, as in [1, 5, 11, 12, 14, 19]. We model the environment as a directed or undirected edge-weighted graph G, with each agent starting on some vertex of G and traveling along edges of G, until it runs out of energy and stops forever. In this model, the agents are obliged to collaborate as no single agent can usually perform the required task on its own.

Given a graph G with designated source and target vertices, and k agents with given starting locations and energy budgets, the decision problem of whether the agents can collectively deliver a single package from the source to the target node in G is called CollaborativeDelivery. Chalopin et al. [11, 12] showed that CollaborativeDelivery is weakly NP-hard on paths and strongly NP-hard on general graphs. When the agents are homogenous, each agent has the same uniform budget initially. The optimization version of this problem asks for the minimum energy budget B per agent, that allows a feasible schedule for delivering the package. Throughout this paper we consider agents with uniform budgets. There exist constant factor approximations [5, 11] for the optimal budget needed for solving CollaborativeDelivery.

Unlike previous papers, this paper considers a version of the problem where the package must be transported through a designated path that is provided as input to the algorithm. This is a natural assumption, e.g. for delivery of valuable packages which must go on a "safe" route, allowing them to be tracked. We call this variant FixedPath CollaborativeDelivery. Even with this additional constraint, the problem remains NP-hard for general graphs due to the result in [11]. Note that on trees, the two problems are equivalent and both problems are known to be weakly NP-hard. However, for arbitrary graphs, the two problems are quite different. In particular, in the FixedPath CollaborativeDelivery, each agent may be used multiple times, while in the original version each agent participates at most once in any optimal delivery schedule (see [11]). In this paper, we attempt to find the difference between the two problems in terms of approximability.

Our Contributions. We show that the best possible approximation of the optimal budget B for FixedPath CollaborativeDelivery is between 2 and 3 for directed graphs and at most 2.5 for undirected graphs. In contrast, the best known approximation ratio for the general version of CollaborativeDelivery is 2 for undirected graphs [11], and there is no known lower bound on approximability.

In the fixed path version of the problem agents may be used multiple times in a feasible delivery schedule, i.e., the same agent may move the package along several disjoint segments of the path. Thus, it is not surprising that our solution for FixedPath CollaborativeDelivery has a higher approximation ratio than the general version of the problem where each agent is used at most once.

For better comparison, we can make the FixedPath CollaborativeDelivery problem easier by restricting each agent to a single pickup of the package. This easier version of the problem was considered recently in [25] which provided a 3-approximation algorithm. In this paper we improve upon this and provide a 2-approximation algorithm for directed graphs and a $(2-1/2^k)$-approximation algorithm for undirected graphs. We also show that there exists no polynomial-time approximation algorithm with better approximation ratio than $\frac{3}{2}$ for directed graphs.

Finally, for the case where the number of agents k is a constant, we show that the decision version of FixedPath CollaborativeDelivery can be solved in pseudo-polynomial time. For this setting, we also provide a fully polynomial-time approximation scheme (FPTAS) giving a $(1+\epsilon)$-approximation to the optimal budget, for any $\epsilon > 0$.

Our Model. We consider finite, connected (or strongly connected), edge-weighted graphs $G = (V, E)$ with $n = |V|$ vertices. For undirected graphs, the weight $w(e)$ of an edge $e \in E$ defines the energy required to cross the edge in either direction. For directed graphs, there may be up to two directed arcs between any pair of vertices and the weight of each arc is the energy required to traverse the arc from its tail to its head. We have k mobile agents which are initially placed on arbitrary nodes $p_1, \ldots, p_k$ of G, called the starting positions. Each agent has an initially assigned energy budget $B > 0$ which allows each agent to move along the edges of the graph for a total distance of at most B (if an agent travels only on a part of an edge, its travelled distance is downscaled proportionally to the part travelled). We say that agents have *uniform budget* B.

The agents are required to move a package from a given source node s to a target node t. An agent can pick up the package when it is at the same location as the package; we say that the agent is carrying the package. An agent carrying the package can drop it at any location that it visits, i.e., either at a node or even at a point inside an edge/arc. The agents do not need to return to their starting locations, after completing their task. We assume that the graph and the starting locations are initially known and the objective is to compute a strategy for movements of the agents which allows the delivery of the package from s to t (along a given $s-t$ path P). We denote by $d(x, y) = d_G(x, y)$ the distance between two nodes x, y in G (i.e. the sum of the weights on the shortest path from x to y). The length of path P is the sum of the weights on the path, denoted by $w(P) = d_P(s, t)$.

Definitions. Given a graph G with edge-weights w, vertices $s \neq t \in V(G)$, starting nodes $p_1, \ldots, p_k$ for the k agents, and an energy budget B, we define CollaborativeDelivery as the decision problem of whether the agents can collectively deliver the package. A solution to CollaborativeDelivery is given in the form of a *delivery schedule* which prescribes for each agent whether it moves and if so, the locations in which it has to pick up and drop off the package. A delivery schedule is *feasible* if the package can be delivered from s to t and each agent moves at most distance B.

Given (G, w, s, t) and the locations $p_1, \ldots, p_k$ of the agents in G, the optimization version of CollaborativeDelivery is to compute the minimum value of B for which there exists a feasible delivery schedule. The problem of FixedPath CollaborativeDelivery provides an additional parameter: an $(s - t)$ path P in G, and the feasible delivery schedules are restricted to those where the package travels on the given path P.

Related Work. The model of energy-constrained robot was introduced by Betke et al. [9] for single agent exploration of grid graphs. Later Awerbuch et al. [2] studied the same problem for general graphs. In both these papers, the agent is allowed to return to its starting node to refuel, and between two visits to the starting node the agent can traverse at most B edges. Duncan et al. [18] studied a similar model where the agent is tied with a rope of length B to the starting location and they optimized the exploration time, giving an $\mathcal{O}(m)$ time algorithm. A more recent paper [15] provides a constant competitive algorithm for the same exploration problem when the value of energy budget B is not much more than the distance to farthest node.

For energy-constrained agents without the option of refuelling, multiple agents may be needed to explore even graphs of restricted diameter. Given a graph G and k agents starting from the same location, each having an energy constraint of B, deciding whether G can be explored by the agents is NP-hard, even if graph G is a tree [22]. Dynia et al. studied the online version of the problem [19, 20]. They presented algorithms for exploration of trees by k agents when the energy of each agent is augmented by a constant factor over the minimum energy B required per agent in the offline solution. Das et al. [14] presented online algorithms that optimize the number of agents used for tree exploration when each agent has a fixed energy bound B. On the other hand, Dereniowski et al. [17] gave an optimal time algorithm for exploring general graphs using a large number of agents. When both k and B are fixed, Bampas et al. [3] studied the problem of maximizing the number of nodes explored by the agents, called the *maximal exploration* problem.

When multiple agents start from arbitrary locations in a graph, optimizing the total energy consumption of the agents is computationally hard for several formation problems which require the agents to place themselves in desired configurations (e.g. connected or independent configurations) in a graph [16, 10]. Anaya et al. [1] studied centralized and distributed algorithms for the information exchange by energy-constrained agents, in particular the problem of transferring information from one agent to all others (*Broadcast*) and from all agents to one agent (*Convergecast*). For both problems, they provided hardness results for trees and approximation algorithms for arbitrary graphs. Czyzowicz et al. [13] recently showed that the problems of collaborative delivery, broadcast and convergecast remain NP-hard for general graphs even if the agents are allowed to exchange energy when they meet. Further results on collective delivery with energy exchange showed that the problem remains hard even when B is a small constant [4].

As mentioned before, the collaborative delivery problem was first studied by Chalopin et al. [11] in arbitrary undirected graphs for both uniform or non-uniform budgets. When the agents have non-uniform budgets, they provided the so-called *resource-augmented algorithms* where the budgets of the agents are augmented by a small constant factor to allow polynomial time solutions for all feasible instances of the original problem. The surprising result that collaborative delivery non-uniform budgets is weakly NP-hard even for a line was proved in [12] where a quasi-pseudo-polynomial time algorithm was provided.

Bärtschi et al. [5] considered the returning version of the problem, where each agent needs to return to its starting location. They showed that, in this case, the problem can be solved in polynomial time for trees, but the problem is still NP-hard for arbitrary planar graphs. They provided 2-resource-augmented algorithm for general graphs in this setting and showed that it is the best possible solution that can be computed in polynomial time. Other variants of collaborative delivery that have been considered are when agents have distinct rate of energy consumption [6] or when the agents have distinct speeds [7]. In these cases the optimization criteria is to minimize the total energy consumption and/or the total time taken for delivery. Another related work [8] studied the collective delivery problem for selfish agents that try to optimize their personal gain.

2 Lower bounds on optimal budget

In this section we prove some lower bounds on the approximation factor for any polynomial time algorithm that solves collaborative delivery with uniform budgets on a fixed path.

We give a reduction from an NP-hard variant of Sat [24]. Note the difference to the polynomially solvable $(3,3)$-Sat, where each variable appears in exactly three clauses [26].

$(\leq 3, 3)$-Sat
Input: A formula with a set of clauses C of size three over a set of variables X, where each variable appears in at most three clauses.
Problem: Is there a truth assignment of X satisfying C?

Observe that we may assume that each variable appears at most twice in positive literals and at most once in a negative literal, otherwise we can either eliminate or negate the variable.

Theorem 1. *The minimum uniform budget required to solve* FixedPath CollaborativeDelivery *on directed graphs cannot be approximated to within a factor better than* 2 *in polynomial time, unless* P = NP.

Proof. We reduce from $(\leq 3,3)$-Sat by constructing, for every sufficiently small $\varepsilon > 0$ and every instance of $(\leq 3,3)$-Sat, an instance of FixedPath CollaborativeDelivery that has a solution with budget $B \leq 2-\varepsilon$ if and only if the $(\leq 3,3)$-Sat instance has a satisfying assignment. In this case, our instance always admits a solution with budget $B = 1$. Since $(\leq 3,3)$-Sat is NP-hard, this then implies that no $(2-\varepsilon)$-approximation algorithm can exist, unless P = NP.

In the following, fix $0 < \varepsilon < 1$ and consider an instance of $(\leq 3,3)$-Sat with variables $x_1,\dots,x_t$ and clauses $C_1,\dots,C_m$. We construct a (directed) instance of FixedPath CollaborativeDelivery with $k = (3+q)t$ agents, where $q := \lceil 3/\varepsilon \rceil$, starting at vertices $p_1,\dots,p_k$. The agents $p_{3i-2}, p_{3i-1}, p_{3i}$ for $i \in \{1,\dots,t\}$ are associated with the (at most) two positive literals and the single negative literal of variable x_i, in this order, that appear in the clauses. In case variable x_i only appears in a single positive literal, the agent p_{3i-1} does not represent any literal. The other agents are so-called *blockers*. We incrementally construct the fixed s-t-path $P = (v_0, v_1, \dots, v_{m+2(q+1)t})$ that the package has to be transported along.

The first m arcs of P correspond to the clauses $C_1,\dots,C_m$. Each arc $e = (v_{j-1}, v_j)$ with $j \in \{1,\dots,m\}$ has weight $w(e) = 1$ and is associated with clause C_j. For every literal of a variable x_i that appears in C_j, we let p_{ij} denote the starting position of the (unique) agent associated with this literal, and we introduce an arc $e_{ij} = (p_{ij}, v_{j-1})$ of weight $w(e_{ij}) = 0$.

Now we add the variable gadgets to the path P. Let $q_i := m + 2(q+1)(i-1)$. The gadget associated with each variable x_i (cf. Figure 1) is the subpath $P_i = (v_{q_i},\dots,v_{q_{i+1}})$ of P consisting of $2q+2$ edges. The first q arcs have weight $\varepsilon/3$ each, the central two arcs $e_i = (v_{q_i+q}, v_{q_i+q+1})$ and $e'_i = (v_{q_i+q+1}, v_{q_i+q+2})$ have weights $w(e_i) = \varepsilon/3$ and $w(e'_i) = 1-\varepsilon/3$, and the final q arcs have weight $1-\varepsilon/3$ each. For $\ell \in \{1,\dots,q\}$, we connect the starting position of the $((i-1)q+\ell)$-th blocker to $v_{q_i+\ell-1}$ with an arc of weight 0, and we add a shortcut arc (that cannot be taken by the package) $(v_{q_i+\ell}, v_{q_{i+1}-\ell})$ of weight 0. Finally, we connect the three agents associated with variable x_i as follows: We add an arc (p_{3i-2}, v_{q_i+q}) of weight $1-\varepsilon/3$, an arc (p_{3i-1}, v_{q_i+q+1}) of weight $\varepsilon/3$, and an arc (p_{3i}, v_{q_i+q}) of weight 0.

We first claim that in every solution with $B \leq 2-\varepsilon$ we can assume that, without loss of generality, for every $i \in \{1,\dots,t\}$ and every $\ell \in \{1,\dots,q\}$, the $((i-1)q+\ell)$-th blocker transports the package across the arc $(v_{q_i+\ell-1}, v_{q_i+\ell})$, then takes the shortcut arc $(v_{q_i+\ell}, v_{q_{i+1}-\ell})$, and finally transports the package across the arc $(v_{q_{i+1}-\ell}, v_{q_{i+1}-\ell+1})$. To see this, consider the last arc $(v_{q_{i+1}-1}, v_{q_{i+1}})$ of P'_i. Since the arcs preceding the vertices v_{q_i} and $v_{q_{i+1}-1}$ along P both have length at least $1-\varepsilon/3$, no agent other than the two blockers connected to v_{q_i} and v_{q_i+1} can reach $v_{q_{i+1}-1}$ with more than $B-(1-\varepsilon/3) \leq 1-2\varepsilon/3$ budget remaining, which is insufficient to cross the last arc of P'_i. Since there is no disadvantage in using the $((i-1)q+1)$-st blocker rather than the $((i-1)q+2)$-nd, we may assume that the $((i-1)q+1)$-st blocker transports the package as claimed. By repeating this argument (slightly adapted for the iq-th blocker), we can fix all subsequent blockers, too. Note that each blocker requires only $B = 1$.

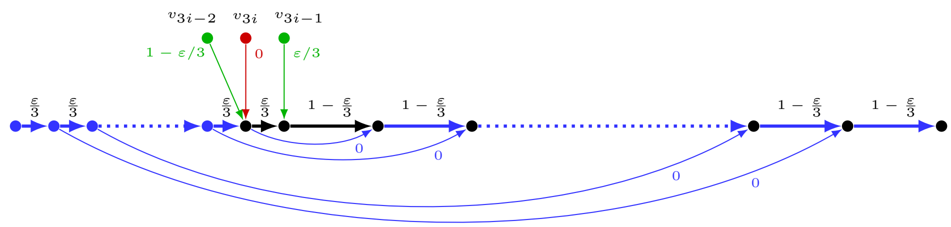

Fig. 1. Illustration of the variable gadget. Thick, horizontal arcs are part of the fixed path of the package. Colors indicate responsabilities: blue is for blockers and green/red is for agents associated with positive/negative literals.

After fixing all blockers, we can observe that every agent with budget $B \leq 2-\varepsilon$ can only transport the package along an arc inside a single clause or variable gadget: This is because transporting the package inside a clause gadget requires one unit of budget, and entering/leaving a variable gadget before or after transporting the package across one of its two central arcs also takes at least one unit of budget (all other arcs of a variable gadget are handled by blockers).

Finally, and crucially, observe that, in order to transport the package across the two central edges of the variable gadget for x_i, either the two agents p_{3i-2} and p_{3i-1} associated with the positive literals of x_i, or the agent p_{3i} associated with the negative literal are needed, since blockers cannot help (see above). We interpret the former situation as x_i being set to *false*, and the latter situation as x_i being set to *true*. Note that either assignment can be accomplished with $B = 1$.

If a variable is set to *true*, the two agents corresponding to positive literals are free to transport the package across the single (!) clause gadget each of them can reach. Otherwise, the agent corresponding to the negative literal is free to do this. In both cases, we interpret this as the clause being satisfied by the corresponding variable. Note that satisfying a clause again requires only $B = 1$.

Clearly, we can turn a satisfying assignment for $(\leq 3,3)$-SAT into a feasible solution of FIXEDPATH COLLABORATIVEDELIVERY with $B = 1$. Conversely, every feasible solution of FIXEDPATH COLLABORATIVEDELIVERY with $B \leq 2-\varepsilon$ corresponds to a satisfying assignment for $(\leq 3,3)$-SAT. Note that q is constant for fixed ε, hence our construction can be done in polynomial time.

3 Approximation algorithms for fixed path delivery

In this section, we give approximation algorithms solving FIXEDPATH COLLABORATIVEDELIVERY for both directed and undirected graphs. In the following, we assume that we are given the optimal value of B for a given instance of the problem and we provide a polynomial time algorithm to compute a delivery strategy that uses an energy budget of at most $\alpha \cdot B$ for some constant $\alpha > 1$. When B is not known, we can guess the optimal value of B by using a binary search in the interval $[D/k, D]$ where D is the length of the given fixed path plus the distance from node s to the nearest agent. The binary search terminates when we find the smallest B for which our algorithm provides a valid strategy for a budget of $\alpha \cdot B$. Clearly this provides an α-approximation algorithm for the optimization problem.

3.1 Directed graphs: 3-approximation

Theorem 2. *There is a* 3*-approximation algorithm for* FixedPath CollaborativeDelivery *on directed graphs.*

Proof. Consider an instance $(G, w, P, \{p_i \mid 1 \leq i \leq k\})$ of FixedPath CollaborativeDelivery on directed graphs and let S be an optimal solution of this instance with uniform budget B. For i from 0 to $\ell = \left\lfloor \frac{d_P(s,t)}{B} \right\rfloor$, we define m_i as the point on P at distance iB from s. Observe that $\ell = O(\min(n, k))$ since the path P is of length less or equal than kB, P has at most $n-1$ arcs and each arc in P has a weight at most B. For i from 0 to $\ell - 1$, let I_i be the interval $[m_i, m_{i+1}]$ on path P. In the solution S, there is an agent a_j starting at position p_j that moves the package from s to some point in I_0. Observe that since the length of each interval is B, for any set $\mathcal{I}$ of l intervals at least l different agents must carry the package inside $\cup_{I \in \mathcal{I}} I$, i.e., the trajectory of these agents intersects interval $\cup_{I \in \mathcal{I}} I$ in S. If the number of such agents for a set $\mathcal{I}$ is exactly l, it means each agent covers exactly an interval of size B and there is no other agent picking the package at the end of the last interval. This can only happen if $\mathcal{I} = \cup_{i=0}^{\ell-1} I_i$, $m_\ell = t$ and for all $0 \leq i \leq \ell - 1$, there is an agent at m_i. This case is easy to check and if it happens, one can construct an easy optimal solution. Hence, we can assume, w.l.o.g., that any set $\mathcal{I}$ of l intervals at least $l+1$ different agents must carry the package inside $\cup_{I \in \mathcal{I}} I$. Hence, there exists a bijection f between a set $R \subseteq [1, k] \setminus \{j\}$ and $[0, \ell - 1]$ such that for each $i \in R$, agent a_i carries the package inside interval $I_{f(i)}$ in S. Observe that $d_G(p_j, m_0) \leq B$ since agent a_j can reach $s = m_0$ with budget B in the solution S. For all $i \in R$, we have $d_G(p_i, m_{f(i)+1}) \leq 2B$ since agent a_i can reach some point in $I_{f(i)}$ with budget B in solution S and then reach $m_{f(i)+1}$ by moving inside P for a distance at most B. We can deduce that there is a bijection g between the set $R' = R \cup \{j\}$ and $[0, \ell]$ such that $d_G(p_i, m_{g(i)}) \leq 2B$. One can find such a bijection g using the following algorithm :

1. Construct a weighted bipartite graph $H = (A \cup M, E, w_H)$ with $A = [0, k]$, $M = [0, \ell]$, $E = M \times A$ and for all $i \in M, j \in A$, $w_H(ij) = d_G(m_i, p_j)$. This can be done in $O(k(m + n \log n))$ using a Dijkstra's algorithm [23] starting from each starting position of an agent. Observe that graph H has $O(\min(n, k) + k) = O(n + k)$ vertices and $O(min(n, k).k)$ edges.

2. Compute a maximal matching in H that minimizes the maximum weight. For each weight ω, one can compute in time $O((n+k)^2 \log(n+k) + k(n+k)\min(n,k))$ a maximal matching [23] in the graph H without edges of weight greater than ω. Hence, one can decide if there is a maximal matching in H with maximum weight ω and by using binary search, one can compute a maximal matching in H which minimizes the maximum weight, in time $O(\log k((n+k)^2 \log(n+k) + k(n+k)\min(n,k)))$.
3. For each edge ij in the matching, we fix $g(i) = j$. This gives us a bijection g between some set R' of size $\ell + 1$ and M. This bijection minimizes the maximal distance $d_G(p_i, m_{g(i)})$ and this value must be less than $2B$ since there is at least one such bijection.

From such a bijection g, we can deduce a 3-approximated solution of our instance: for each $i \in [0, \ell]$, agent $a_{g^{-1}(i)}$ moves to point m_i (cost less than $2B$) and then carries the package to point m_{i+1} if $i < \ell$ or t otherwise (cost less than B).

3.2 Undirected graphs: 2.5-approximation

Theorem 3. *There is a* 2.5*-approximation algorithm for* FixedPath CollaborativeDelivery *on undirected graphs.*

Proof. The proof is similar to that of Theorem 2, the intervals are slightly different in order to use the possibility for an agent to move on the path P in both directions.

Consider an instance $(G, w, P, \{p_i \mid 1 \le i \le k\})$ of FixedPath CollaborativeDelivery on undirected graphs and let S be an optimal solution with budget B. For i from 0 to $\ell = \left\lfloor \frac{d_P(s,t)}{B} \right\rfloor$, we define m_i as the point on P at distance iB from s (same definition as in proof of Theorem 2). For i from 1 to ℓ, we define m'_i as the point on P at distance $iB - B/2$ from s. We set $m'_0 = s$. Let $\ell' = \ell + 1$ and $m'_{\ell'}$ be the point of P at distance $\ell B + B/2$ from s, if $d_P(m_\ell, t) > \frac{B}{2}$, and let $\ell' = \ell$ and $m'_{\ell'} = t$ otherwise. For i from 0 to $\ell' - 1$, let I_i be the interval $[m'_i, m'_{i+1}]$ on path P. Observe that $|I_0| = B/2$, and for each $i \in [1, \ell' - 1]$, $|I_i| = B$. Hence the union of l intervals have a length strictly greater than $(l-1)B$. With a similar argument as proof of Theorem 2, there exists a bijection f between a set $R \subseteq [1, k]$ and $[1, \ell' - 1]$ such that for each $i \in R$, agent a_i carries the package inside interval $I_{f(i)}$ in S. The starting position p_i of agent a_i is at distance at most B from some point $s_{f(i)}$ in $I_{f(i)}$. Observe that for all $i \in [0, \ell]$, we have $d_P(m_i, m'_i) \le B/2$ and $d_P(m_i, m'_{i+1}) \le B/2$. It follows that every point in I_i and so s_i is at distance at most $B/2$ of m_i. By the triangular inequality, we have that for all $i \in [0, \ell]$ $d_P(p_i, m_{f(i)}) \le \frac{3}{2}B$. One can find a bijection f having this property with the same algorithm as in the proof of Theorem 2.

From a bijection f, we can deduce a 2.5-approximated solution of our instance: for each $i \in [0, \ell]$, agent $a_{f^{-1}(i)}$ moves to point m_i (cost less or equal than $\frac{3}{2}B$) and then carries the package to point m_{i+1} if $i < \ell$ or t otherwise (cost less or equal than B).

4 Special case: single pickup per agent

In this section, we consider a slightly easier version of the problem when each agent can pickup the package at most once. We first present a lower bound of $\frac{3}{2}$ on the approximation ratio of optimizing FixedPath CollaborativeDelivery.

4.1 Lower bound

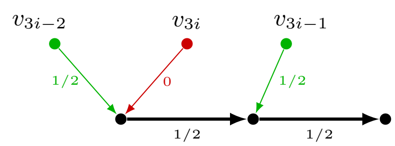

Fig. 2. Illustration of the clause gadget for the case where agents cannot pickup the package more than once.

Theorem 4. *The minimum uniform budget required to solve* FixedPath CollaborativeDelivery *on directed graphs cannot be approximated to within a factor better than* 1.5 *in polynomial time, unless* P = NP*, even when each agent may pickup the package at most once.*

Proof. We use the same construction as in the proof of Theorem 1, but we set $\varepsilon = 3/2$ and $q = 0$ (cf. Fig. 2). All claims in the proof of Theorem 1 remain valid for any $B < 3/2$. Note that, since we eliminated all blockers, no agent has to pickup the package more than once in the optimum solution.

4.2 Approximation algorithm for single pickup per agent

Lemma 1. *Given any instance of the decision problem for* FixedPath CollaborativeDelivery *that admits a solution where each agent can pickup the package at most once; then we can compute in polynomial time a* 2*-approximate delivery strategy. When the graph is undirected, we can compute a* $(2 - 1/2^k)$*-approximation in polynomial time.*

Proof. Suppose there exists a feasible solution S for the problem using uniform budget B and single pickup per agent. Consider the fixed $(s - t)$ path P and partition it into segments using the points $X = (m_1, m_2 \ldots m_l = t)$ on P, such that $l = \lceil w(P)/B \rceil$, the length of segment (m_i, m_{i+1}) is B, $\forall 1 \leq i < l$, and the length of the first segment $(s, m_1) \leq B$. We have the following observations for strategy S: (1) Any agent that moves the package over point m_i in strategy S must have enough energy to reach point m_i, and (2) Any single agent can not transport the package over two distinct points in X since the distance between two points is at least B.

Case (i): In strategy S, the agent that picks up the package at s is not the same agent that moves the package over m_1. In this case, there exists a matching between the agents and the points $X^+ = (s = m_0, m_1, m_2, \dots m_l = t)$ such that each agent can reach the point that it is mapped to. We call any such matching a type M_0 matching. Case(ii): In strategy S, a single agent delivers the package from s to m_1 with its original energy budget B. In this case, there exists a matching between the agents and the points in X (w.l.o.g. agent a_i is mapped to point p_i), such that, agent a_1 has enough energy to move the package from s to m_1 and $\forall i > 1$, agent a_i can reach m_i, using budget B. We call any such matching a type M_1 matching. Note that if S is a feasible solution to the problem using single pickup per agent, then there exists a matching of type M_0 or M_1. If we can find such a matching, then, using budget B per agent, we can move the package to point p_1 and move each agent a_i to the respective point m_i in path P. If the budget of each agent is augmented by factor 2, then using the additional budget B, the agent a_i that is mapped to point m_i can actually deliver the package to the next point m_{i+1}. This gives a 2-approximate solution to the problem (for directed and undirected graphs).

For undirected graphs, we will now construct a delivery strategy where each agent has a budget $2B - B/2^l$. As per previous discussion, using the original budget B each agent a_i can reach point m_i and the package can be moved to point m_1. Each agent a_i now has available energy budget of at least $B - B/2^l$ after arriving at the designated point m_i.

Consider the points $m'_i = m_i + B - (2^i - 1)B/2^l$, $1 \leq i \leq l-1$. The agent a_1 delivers the package from point m_1 to m'_1. For $1 < i < l$, each agent a_i located at point m_i returns to m'_{i-1} to pick up the package and then moves the package to point m'_i. This requires an additional budget of $B - (2^i - 1)B/2^l + 2 \times 2^i B/2^l$ $= B(1 - 1/2^l)$, for each of these agents. Finally, note that the distance between point m'_{l-1} and the target $t = m_l$ is at most $B/2 - B/2^l$, and so the agent a_l can move from m_l to m'_{l-1} to pick up the package and deliver it to the target, using $2 \times (B/2 - B/2^l) < B(1 - 1/2^l)$ additional energy.

Since $k \geq l$, this provides a $(2 - 1/2^k)$-approximate solution strategy for any instance which has a feasible solution using k agents and a single pickup per agent.

The computation of the schedule requires constructing a bipartite graph between k agents and at most k points, and then solving maximum matching in this bipartite graph. The former task requires $O(n^3)$ time using an all-pair shortest path algorithm to compute distances in the original graph. The second task of computing the matching requires at most $O(k^2)$ time.

As in the previous section, we use a binary search to find the smallest B for which there exists a matching of type M_0 or M_1 from the above lemma. This gives us a $(2 - 1/2^k)$-approximate (respectively 2-approximate) solution to the optimization problem for undirected (resp. directed) graphs. Hence we have the following results:

Theorem 5. *The minimum uniform budget required to solve* FixedPath CollaborativeDelivery *with single pickup per agent on undirected graphs can be approximated to a factor* $(2 - 1/2^k)$*, in polynomial time.*

Theorem 6. *The minimum uniform budget required to solve* FixedPath CollaborativeDelivery *with single pickup per agent on directed graphs can be approximated to a factor* 2*, in polynomial time.*

5 Delivery with few agents

In this section we consider the special case when agents are allowed to exchange the package at vertices only. Using the dynamic programming technique, we design an algorithm that for a given B, computes whether there exists a feasible schedule with uniform budget B, and has a running time that is exponential in k and pseudo-polynomial in n (the run-time will depend on B). To find a minimum B such that there exists a feasible schedule, we can use binary search on B, which adds multiplicative $\log B$ increase to the run-time.

We keep a boolean table $T_v[j|p_1^v, \ldots, p_k^v|B_1^v, \ldots, B_k^v]$ denoting whether there exists a feasible schedule that delivers the package from s to vertex v on the path P such that

1. the last agent that delivers the package to vertex v is agent a_j,
2. the positions of the k agents, when the package arrives at v, are $p_1^v, \ldots, p_k^v$, and
3. the remaining budgets of the agents are $B_1^v, \ldots, B_k^v$.

We initialize $T_s[0|p_1, \ldots, p_k|B, \ldots, B] = \texttt{TRUE}$ and initialize $T_s[...] = \texttt{FALSE}$ for all other values of j and p_i^s and B_i^s, $i = 1, \ldots, k$. Here, $j = 0$ denotes that no agent has been used yet. We also abuse the notation and use p_0 to denote s. Clearly, $T_v[j|p_1^v, \ldots, p_k^v|B_1^v, \ldots, B_k^v] = \texttt{TRUE}$ if and only if $p_j^v = v$, and there exists a vertex u on the path P before vertex v and an agent's index $j' \neq j$ such that there is a feasible schedule where agent a_j walks from position p_j^u to pick-up the package at vertex u from agent $a_{j'}$ and carries it from vertex u to vertex v. I.e., we have $T_v[j|p_1^v, \ldots, p_k^v|B_1^v, \ldots, B_k^v] = \texttt{TRUE}$ if and only if there exists u and j' and an entry in the table T such that $T_u[j'|p_1^u, \ldots, p_k^u|B_1^u, \ldots, B_k^u] = \texttt{TRUE}$ and $p_j^v = v$, $p_{j'}^v = p_{j'}^u = u$, $p_i^v = p_i^u$ for every $i \neq j, j'$, $B_j^v = B_j^u - d(p_j^u, u) - d_P(u, v)$, and $B_i^v = B_i^u$ for every $i \neq j$. Recall that $d_P(u, v)$ denotes the distance from u to v on the path P.

At the end, when the whole table is computed, we check whether there is an entry at target vertex t such that $T_t[\ldots] = \texttt{TRUE}$, in which case there is a feasible schedule for the uniform budget B, and there is no feasible schedule otherwise. To compute the feasible schedule, standard bookkeeping techniques can be applied. There are $n \cdot n^k \cdot B^k$ entries in T that need to be computed. To compute one entry
$T_v[j|p_1^v, \ldots, p_k^v|B_1^v, \ldots, B_k^v]$, we need to check the existence of j' and u with the above mentioned properties, which can be done in time $O(k \cdot n)$. Hence, the total run-time of the alorithm is $O(k \cdot n^{k+2} \cdot B^k)$. Thus, we have shown the following:

Theorem 7. *There is an algorithm that decides whether a feasible schedule for uniform budget B exists and runs in $O(k \cdot n^{k+2} \cdot B^k)$ time.*

By using the data rounding technique, we turn the developed algorithm into a fully polynomial-time approximation scheme (FPTAS). Let $\epsilon > 0$ be a (small) error margin for which we want to design a $(1+\epsilon)$-approximation algorithm (for computing a minimum feasible uniform budget B).

We define an alternative weight unit $\mu := \epsilon \frac{w(P)/k+X}{m^2}$, where $w(P)$ is the weight of the fixed path P, X is the minimum distance of any agent to the path P, and m is the number of edges of the graph G. We measure the weights $w(e)$ in the integer multiples of μ, rounded-up, i.e., we define $\bar{w}(e) := \lceil w(e)/\mu \rceil$.

We solve the problem in the new edge weights $\bar{w}(e)$ using the dynamic programming approach, where we also measure budget in multiples of μ. Let $\bar{B}$ be the computed optimum uniform budget for the modified edge-weights. Our algorithm returns $B^A = \bar{B} \cdot \mu$ as the solution for the original edge-weights. Let $\bar{P}_1, \ldots, \bar{P}_k$ be the walks that the k agents walk in the optimum solution for the modified edge-weights. Hence, $\bar{B} = \max_i\{\bar{w}(\bar{P}_i)\}$, and thus $\bar{B} \cdot \mu = \max_i\{\bar{w}(\bar{P}_i) \cdot \mu\}$. Observe also that B^A is a feasible budget, since every path $\bar{P}_i$ can be walked with budget B^A, since the original length of $\bar{P}_i$ is $w(\bar{P}_i) \le \mu \cdot \bar{w}(\bar{P}_i) \le \mu \bar{B}$.

Let B^* be the optimum budget for the original edge-weights, and let $P_1^*, \ldots, P_k^*$ be the walks of the k agents in some optimum solution. Hence, $B^* = \max_i\{w(P_i)\}$. We now argue that B^A is not much larger than B^*. We have $B^A = \mu \cdot \bar{B} = \mu \cdot \max_i\{\bar{w}(\bar{P}_i)\} \overset{(1)}{\le} \mu \cdot \max_i\{\bar{w}(P_i^*)\} = \max_i\{\mu \cdot \bar{w}(P_i^*)\} \overset{(2)}{\le} \max_i\{w(P_i^*) + m^2\mu\} = m^2\mu + \max_i\{w(P_i^*)\} = m^2\mu + B^* = m^2\left(\epsilon \frac{w(P)/k+X}{m^2}\right) + B^* \overset{(3)}{\le} \epsilon \cdot B^* + B^* = (1+\epsilon)B^*$. Here, inequality (1) is because $\max_i \bar{P}_i$ is the optimum feasible solution in weights $\bar{w}$; inequality (2) follows because any walk appears at most m times on the path P, and between any two appearances, the walk contains at most m edges (this part of the walk is a simple path), inequality (3) follows because B^* needs to be at least $w(P)/k + X$ (the average traveled distance per agent on P plus the distance to get from the initial position to the path P).

We now analyze the run-time of the algorithm. Observe first that $B^* \le \min_i d(p_i, s) + w(P) \le (X + w(P)) + w(P) \le 2(X + w(P))$. Therefore, measured in the units μ, we search for $\bar{B}$ in the range between 1 and $2(X + w(P))/\mu \le \frac{2m^2k}{\epsilon}$. Hence, one run of the dynamic programming on the modified weights takes time $O(k \cdot n^{k+2} \cdot (\frac{2m^2k}{\epsilon})^k)$. Using the binary search to find optimum $\bar{B}$ adds a multiplicative logarithmic factor of $\log\left(\frac{2m^2k}{\epsilon}\right)$. Thus, we have shown the following.

Theorem 8. *For any $\epsilon > 0$, there is an algorithm that computes a feasible uniform budget B that is at most $(1+\epsilon)B^*$, where B^* is the optimum uniform budget, and runs in $O\left(k \cdot n^{k+2} \cdot (\frac{2m^2k}{\epsilon})^k \log\left(\frac{2m^2k}{\epsilon}\right)\right)$ time.*

Corollary 1. *There exists an FPTAS for the variant where the number of agents is constant.*

6 Conclusions

The problem of collectively delivering a package by mobile agents is a difficult problem even when the path for moving the package is given in advance. However, for the case of single pickup per agent, we were able to find better approximation algorithms for the fixed path version of collaborative delivery. These results leave many open questions: how to reduce the gap between the upper and lower bounds for the various versions of the problem? How to extend the results to agents with *non-uniform* budgets and find resource-augmented algorithms for fixed path delivery? Finally, what is the effect of restricting package handovers to nodes only and not anywhere inside the edges.

References

1. Julian Anaya, Jérémie Chalopin, Jurek Czyzowicz, Arnaud Labourel, Andrzej Pelc, and Yann Vaxès. Convergecast and broadcast by power-aware mobile agents. *Algorithmica*, 74(1):117–155, 2016.
2. Baruch Awerbuch, Margrit Betke, Ronald L. Rivest, and Mona Singh. Piecemeal graph exploration by a mobile robot. *Information and Computation*, 152(2):155–172, 1999.
3. Evangelos Bampas, Jérémie Chalopin, Shantanu Das, Jan Hackfeld, and Christina Karousatou. Maximal exploration of trees with energy-constrained agents. *CoRR*, abs/1802.06636, 2018.
4. Evangelos Bampas, Shantanu Das, Dariusz Dereniowski, and Christina Karousatou. Collaborative delivery by energy-sharing low-power mobile robots. In *13th International Symposium on Algorithms and Experiments for Wireless Sensor Networks (ALGOSENSORS)*, pages 1–12, 2017.
5. Andreas Bärtschi, Jérémie Chalopin, Shantanu Das, Yann Disser, Barbara Geissmann, Daniel Graf, Arnaud Labourel, and Matús Mihalák. Collaborative delivery with energy-constrained mobile robots. *Theoretical Computer Science*, 2017.
6. Andreas Bärtschi, Jérémie Chalopin, Shantanu Das, Yann Disser, Daniel Graf, Jan Hackfeld, and Paolo Penna. Energy-efficient delivery by heterogeneous mobile agents. In *34th Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 10:1–10:14, 2017.
7. Andreas Bärtschi, Daniel Graf, and Matús Mihalák. Collective Fast Delivery by Energy-Efficient Agents. In *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*, volume 117 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 56:1–56:16, 2018.
8. Andreas Bärtschi, Daniel Graf, and Paolo Penna. Truthful Mechanisms for Delivery with Agents. In *17th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2017)*, volume 59 of *OpenAccess Series in Informatics (OASIcs)*, pages 2:1–2:17, 2017.
9. Margrit Betke, Ronald L. Rivest, and Mona Singh. Piecemeal learning of an unknown environment. *Machine Learning*, 18(2):231–254, 1995.
10. Davide Bilò, Yann Disser, Luciano Gualà, Matúš Mihalák, Guido Proietti, and Peter Widmayer. Polygon-constrained motion planning problems. In *9th International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics ALGOSENSORS'13*, pages 67–82, 2013.

11. Jérémie Chalopin, Shantanu Das, Matúš Mihalák, Paolo Penna, and Peter Widmayer. Data delivery by energy-constrained mobile agents. In *9th International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics ALGOSENSORS'13*, pages 111–122, 2013.
12. Jérémie Chalopin, Riko Jacob, Matúš Mihalák, and Peter Widmayer. Data delivery by energy-constrained mobile agents on a line. In *41st International Colloquium on Automata, Languages, and Programming ICALP'14*, pages 423–434, 2014.
13. Jurek Czyzowicz, Krzysztof Diks, Jean Moussi, and Wojciech Rytter. Communication problems for mobile agents exchanging energy. In *23rd International Colloquium on Structural Information and Communication Complexity SIROCCO'16*, 2016.
14. Shantanu Das, Dariusz Dereniowski, and Christina Karousatou. Collaborative exploration by energy-constrained mobile robots. In *22th International Colloquium on Structural Information and Communication Complexity SIROCCO'15*, pages 357–369, 2015.
15. Shantanu Das, Dariusz Dereniowski, and Przemyslaw Uznanski. Energy constrained depth first search. *CoRR*, abs/1709.10146, 2017.
16. Erik D. Demaine, Mohammadtaghi Hajiaghayi, Hamid Mahini, Amin S. Sayedi-Roshkhar, Shayan Oveisgharan, and Morteza Zadimoghaddam. Minimizing movement. *ACM Trans. Algorithms*, 5(3):1–30, 2009.
17. Dariusz Dereniowski, Yann Disser, Adrian Kosowski, Dominik Pająk, and Przemysław Uznański. Fast collaborative graph exploration. *Information and Computation*, 243:37–49, 2015.
18. Christian A. Duncan, Stephen G. Kobourov, and V. S. Anil Kumar. Optimal constrained graph exploration. In *12th ACM Symposium on Discrete Algorithms, SODA'01*, pages 807–814, 2001.
19. Miroslaw Dynia, Miroslaw Korzeniowski, and Christian Schindelhauer. Power-aware collective tree exploration. In *19th International Conference on Architecture of Computing Systems, ARCS'06*, pages 341–351, 2006.
20. Miroslaw Dynia, Jakub Łopuszański, and Christian Schindelhauer. Why robots need maps. In *14th International Colloquium on Structural Information and Communication Complexity, SIROCCO'07*, pages 41–50, 2007.
21. Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. *Distributed Computing by Oblivious Mobile Robots.* Morgan & Claypool, 2012.
22. Pierre Fraigniaud, Leszek Gąsieniec, Dariusz R. Kowalski, and Andrzej Pelc. Collective tree exploration. *Networks*, 48(3):166–177, 2006.
23. Michael L. Fredman and Robert Endre Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, 34(3):596–615, 1987.
24. M. R. Garey and D. S. Johnson. *Computers and intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., New York, 1979.
25. Aristotelis Giannakos, Mhand Hifi, and Gregory Karagiorgos. Data delivery by mobile agents with energy constraints over a fixed path. *CoRR*, abs/1703.05496, 2017.
26. Craig A. Tovey. A simplified np-complete satisfiability problem. *Discrete Applied Mathematics*, 8(1):85–89, 1984.