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

Jérémie Chalopin[a], Shantanu Das[a,*], Yann Disser[b], Arnaud Labourel[a], Matúš Mihalák[c]

[a] *Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France*
[b] *Graduate School CE, TU Darmstadt, Germany*
[c] *Department of Data Science and Knowledge Engineering, Maastricht University, Netherlands*

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 some vertex of the graph; it can move along the edges of the graph and can can pick up the package from a vertex and drop it in another vertex during the course of its movement. 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 constraint reduces the search space for feasible solutions, we show that 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 a single pickup per agent), compared to the known results for the general version of the problem.

We provide polynomial time approximation algorithms for both directed and

☆This work was partially supported by the ANR project ANCOR (anr-14-CE36-0002-01). The third author was supported by the 'Excellence Initiative' of the German Federal and State Governments and the Graduate School CE at TU Darmstadt.

*Corresponding author

Email addresses: jeremie.chalopin@lis-lab.fr (Jérémie Chalopin), shantanu.das@lis-lab.fr (Shantanu Das), disser@mathematik.tu-darmstadt.de (Yann Disser), arnaud.labourel@lis-lab.fr (Arnaud Labourel), matus.mihalak@maastrichtuniversity.nl (Matúš Mihalák)

URL: http://pageperso.lif.univ-mrs.fr/~shantanu.das/en/ (Shantanu Das)

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.

Keywords: Collaborative delivery; mobile agents; energy constrained robots; directed graphs; fixed path; approximation algorithms

---

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 e.g. [1] for a recent survey of this area of research). 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 [2, 3, 4, 5, 6, 7]. 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. [4, 5] 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 [3, 4] 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 [4]. 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 [4]). In this paper, we attempt to find the difference between the two problems in terms of approximability.

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. In this paper, we consider the agents to have *uniform budget* B. 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). 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 in G). The length of path P is the sum of the weights on the path, denoted by $w(P) = d_P(s, t)$. We denote an interval on this path as $(x, y]$ if it includes all points on P between x and y, excluding point x, but including y.

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.

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. Thus an instance of FixedPath CollaborativeDelivery is given as $(G, w, P, p_1, \ldots, p_k)$ where P is a path in G, starting at node s and ending at node t.

Related Work.

The model of energy-constrained robot was introduced by Betke et al. [8] for single agent exploration of grid graphs. Later Awerbuch et al. [9] 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. [10] 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 [11] provides a constant competitive algorithm for the same exploration problem when the value of energy budget B may be as small as the length of the smallest path that visits the 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 [12]. Dynia et al. studied the online version of the problem [7, 13]. 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. [6] 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. [14] 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. [15] studied the problem of maximizing the number of nodes explored by the agents, called the *maximal exploration* problem. For more details on tree exploration with energy constraint, see the recent thesis [16].

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 [17, 18]. Anaya et al. [2] 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. [19] 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 [20].

As mentioned before, the collaborative delivery problem was first studied by Chalopin et al. [4] 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 [5] where a quasi-pseudo-polynomial time algorithm was provided.

Bärtschi et al. [3] 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 [21] or when the agents have distinct speeds [22]. In these cases the optimization criteria is to minimize the total energy consumption and/or the total time taken for delivery. Another related work [23] studied the collective delivery problem for selfish agents that try to optimize their personal gain. See also [24] for a survey of recent results on collaborative delivery by agents with energy limitations.

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 [4], 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 $\epsilon > 0$.

2. Lower bound on approximation

In this section we prove a lower bound 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 [26]. Note the difference from the polynomially solvable $(3,3)$-Sat, where each variable appears in exactly three clauses [27].

> $(\leq 3,3)$-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 exactly 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 $3i-2, 3i-1, 3i$ 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 $3i-1$ does not represent any literal. The other $q \cdot t$ agents are the so-called *blockers*, defined later. We incrementally construct the fixed (s,t)-path $P = (s = v_0, v_1, \dots, v_{m+2(q+1)t})$ 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 $\varepsilon/3$

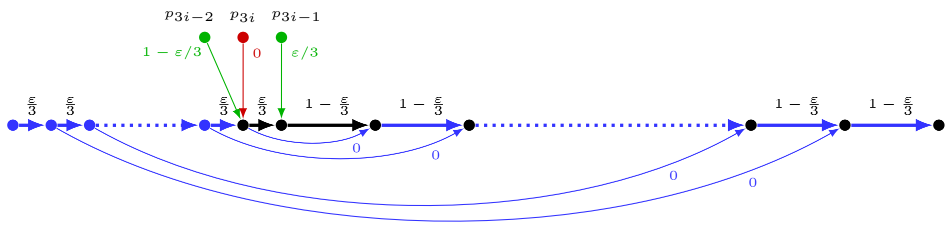

Figure 1: Illustration of the variable gadget. The horizontal arcs are part of the fixed path of the package. Colors indicate responsibilities: blue nodes are for blockers and green/red nodes contain agents associated with positive/negative literals.

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 $B = 1$.

After fixing all blockers, we can observe that any other agent, having a budget $B \leq 2-\varepsilon$ can transport the package either inside a single clause gadget or inside a single variable gadget, but not both. 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 $3i-2$ and $3i-1$ associated with the positive literals of x_i, or the agent $3i$ 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. Note that for solution to the problem the total distance travelled by the agents must be at least the length of the path P plus the distance to s from the closest agent (which denote this by D). This gives the following bound on the optimal budget per agent.

Observation 2. *The optimal budget B for* FIXEDPATH COLLABORATIVEDELIVERY *must be in the range $[D/k, D]$, where $D = \min_i d_G(p_i, s) + w(P)$.*

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 $[D/k, D]$ due to the above observation. 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.

Consider an optimal solution to the problem which moves the package on path P using a budget of B per agent. If P is of length at least $l \cdot B$ then at least l agents were used. Consider a partition of the path P into intervals of length B exactly (assuming that $w(P)$ is a multiple of B). Then, for any $x \leq l$ intervals, there must be at least x agents that pushed the package along those intervals in any optimal solution. This means that it is possible to assign agents to intervals in such a way that: (i) The agent assigned to the interval participated in moving the package on that interval, i.e. the agent is able to reach some point on the interval using at most budget B. (ii) Each agent is assigned a distinct interval.

The solution strategies that we use for the approximation algorithm would use the above idea. In particular we would try to find a matching between a subset of the agents and the intervals of the path P as described below.

Lemma 1. *Given an instance $(G, w, P, p_1, \ldots, p_k)$ of the problem for which the optimal budget is B and $B < w(P)$, let, for some $l \geq 2$, $m_0, m_1, \ldots, m_{l-1}$ be distinct points (not necessarily vertices) on the path P, such that $0 \leq d_P(s, m_0) < B$, $d_P(m_{i-1}, m_i) = B$, for $0 < i < l$, and $d_P(m_{l-1}, t) > 0$. We consider the path P as an Euclidean line and on this line, we define I_0 to be the interval [s,s] and I_i to be the interval $(m_{i-1}, m_i]$, for $0 < i < l$. Then there exists distinct agents $a_0, a_1, \ldots, a_{l-1}$ which can be matched to interval $I_0, I_1, \ldots I_{l-1}$, such that each agent a_i can reach some point in interval I_i using an energy budget of at most B.*

Proof Note that for moving the package across x segments of length B each, we need at least x agents. Consider any optimal solution for the instance and let a_0 be the agent that picks up the package at source s, which implies agent a_0 was able to reach s. If $d_P(s, m_0) > 0$, and agent a_0 moves the package over some non-zero distance in this interval, it would have depleted some of its energy; thus agent a_0 would not have enough energy to move the package over the complete interval I_1, which is of length B. Thus, at least one other agent must participate in moving the package over interval I_1, let this be agent a_1. On the other hand, if $d_P(s, m_0) = 0$, i.e. $s = m_0$, then agent a_0 can potentially move the package on the complete interval I_1; in that case it would completely exhaust its budget and there must be some other agent a_1 that picks up the package at m_1. This implies agent a_1 was able to reach point $m_1 \in I_1$. So, in both cases there is an agent a_1 that can reach I_1. Thus, the lemma holds for the base case of $l = 2$ and we can extend this argument. Suppose the lemma holds for $l = j$ and agents $a_0, \ldots a_{j-1}$ be the corresponding agents. We prove the lemma for $l = j + 1$ i.e. for j intervals I_1 to I_j.

Case(i): Only the j agents $a_0, \ldots a_{j-1}$ move the package over intervals I_0 to I_j in the optimal solution. This is only possible if $s = m_0$ and each agent starts at the beginning of an interval. In this case the j agents would completely exhaust their total budget in moving the package and thus, a new agent a_j must pick up the package at m_j (Note that that target t is further than point m_j according to the lemma). Thus, we have agents $a_0, \ldots a_j$ that satisfy the conditions of the lemma.

Case(ii): There are $x \geq j + 1$ agents $a_0, \ldots a_x$ that participate in the optimal solution, with agents a_0 to a_{j-1} already matched to the intervals I_0 to I_{j-1}, according to the induction hypothesis. Consider the last interval I_j and let A^* be the subset of $i > 0$ agents that participated in moving the package on this particular interval. If one of these agents is unmatched, it can matched to interval I_j and we are done. Otherwise the agents in A^* are matched to i intervals, possibly including the interval $[s, s]$, so the total length of these intervals is at most $(i - 1) * B$. If we include the interval I_j of length B and consider the fact that some of the agents have to move between non-consecutive intervals incurring additional energy consumption, this implies that the total movement by all the agents that participated in these $(i + 1)$ intervals is strictly more than $i * B$. Hence, at least one other agent $a_r \notin A^*$ participated in at least one of these intervals say, I_q, where $q < j$. If we match this agent to interval I_q, then the agent a_q that was originally matched to I_q, can be matched to interval I_j. By definition $a_q \in A^*$ and thus participated in the last interval I_j, so it can reach I_j. This concludes the proof. □

The solution strategies that we use for the approximation algorithm would use the above fact. We first show that it is possible to compute in polynomial time, one such matching between a subset of agents and the segments of the path P as defined in the Lemma 1.

Lemma 2. *Consider an instance $(G, w, P, p_1, \ldots, p_k)$ of the problem for which the optimal budget is B, then given any set $I = I_0, I_1, \ldots I_l$ of segments of P satisfying the conditions of Lemma 1, there is a $O(n^3)$ algorithm to find a matching g between the a subset of agents and the segments, satisfying Lemma 1.*

Proof One can find such a matching g using the following algorithm :

1. Construct a weighted bipartite graph $H = (A \cup I, E, w_H)$ with $A = [0, k]$, $M = [0, \ell]$, $E = M \times A$ and for all $i \in M, j \in A$, $w_H(ij)$ is the smallest distance from p_j (the starting position of agent j) to some vertex in I_i (or one of the endpoints of I_i in case no vertex is located in that segment). This can be done in $O(k(m + n \log n))$ using a Dijkstra's algorithm [28] starting from each starting position of an agent. Observe that graph H has $O(n + k)$ vertices and $O(k(n + k))$ 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 [28] 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 $O(\log k((n + k)^2 \log(n + k) + k(n + k) \min(n, k)))$.

Assuming $k = O(n)$, the algorithm above has a complexity of $O(n^3)$. When the number of agents k is considerably smaller than n, the algorithm would only be faster. □

We now present the approximation algorithms for FixedPath CollaborativeDelivery in directed and undirected graphs, based on the above observations.

3.1. Directed graphs: 3-approximation

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

Proof Consider an instance $(G, w, P, p_1, \ldots, p_k)$ of FixedPath CollaborativeDelivery on directed graphs and let S be an optimal solution of this instance with uniform budget B. Let $l = \lceil w(P)/B \rceil$.

Case(i): If $B \geq w(P)$, then any agent that can reach s can transport the package to t using an additional budget of B. Since there must exist such an agent and it is possible to find such an agent in $O(k)$ time by a linear search over all agents, this give us the required approximation algorithm.

Case(ii): If $B < w(P)$ then $l = \lceil w(P)/B \rceil \geq 2$. Thus we can apply the Lemma 1 using points $m_0 = s$, and $m_i = s + i \cdot B, 0 < i < l$ on the path P. Note that the last point satisfies the property $0 < d_P(m_{l-1}, t) \leq B$. Let $a_0, \ldots a_{l-1}$ be the matching agents according to Lemma 1 (which can be computed using the algorithm from Lemma 2). Since agent a_0 can reach the source $s = m_0$ using budget B, it can transport the package to point m_1 using a budget of at most $2B$ in total. For $0 < i < l - 1$, agent a_i can reach the point m_i using a budget of at most $2B$ and thus it can transport the package from m_i to m_{i+1} using a budget of at most $3B$ in total. Similarly, the agent a_{l-1} can transport the package from m_{l-1} to the target t. This gives the required 3-approximation. □

3.2. Undirected graphs: 2.5*-approximation*

Theorem 4. *There is a* 2.5*-approximation algorithm for* FixedPath CollaborativeDelivery *on undirected graphs.*

Proof Consider an instance $(G, w, P, p_1, \ldots, p_k)$ of FixedPath CollaborativeDelivery on undirected graphs and let S be an optimal solution of this instance with uniform budget B. If $w(P) \leq 3B/2$, then any agent that reaches the vertex s can carry the package to t, using an additional budget of $3B/2$, and this immediately gives a 2.5-approximation. Thus, let us assume that $w(P) > 3B/2$ and consider $l = \lceil w(P)/B - 1/2 \rceil \geq 2$.

We define the points $m'_1, \ldots m'_l$ on P such that $m'_i + (l - i) * B = t$, for $1 \leq i \leq l$ (thus $m'_l = t$). Note that the distance from s to m'_1 is at most $3B/2$. Now let m_i be the point on path P defined as $m_i = m'_{i+1} - B/2$ for $0 \leq i < l$. Thus the point m_i is the midpoint between m'_{i-1} and m'_i for $1 < i \leq l$ and the point m_0 is at a distance at most B from s. Now we can apply Lemma 1 using points $s, m_0, \ldots m_{l-1}$ to obtain matching agents a_0 to a_{l-1}. Agent a_0 can reach the source s and thus it can transport the package to m'_1 using an additional budget of $3B/2$. Agent $a_i, 0 < i < l$ can reach the interval (m_{i-1}, m_i), and thus using an additional budget $B/2$, the agent can reach the mid-point m'_i (this may involve going back on the path). Thus agent a_i can transport the package from m'_i to m'_{i+1} using a total budget of at most $2.5 * B$ which gives the required approximation algorithm. □

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. This means that each agent that participates in the solution, moves the package over a single continuous segment of the path. In this case, we can obtain better approximations for the problem. We first present a lower bound of $\frac{3}{2}$ on the approximation ratio of optimizing FixedPath CollaborativeDelivery using the same idea as in Section 2.

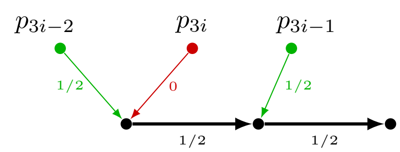

Figure 2: Illustration of the variable gadget for the case where agents cannot pickup the package more than once.

4.1. Lower bound

Theorem 5. *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

We now present approximation algorithms for FixedPath CollaborativeDelivery with the restriction of a single pickup per agent. This means that for agents with uniform budget B, any two points on the fixed path P that are separated by a distance of at least B must be served by distinct agents. This observation allows us to match the agents to specific points on the path P (as opposed to intervals on the path in the general case considered in the previous section). The rest of the algorithm is based on similar ideas as in the previous section.

Lemma 3. *Given any instance of* FixedPath CollaborativeDelivery *that admits a solution using optimal uniform budget B, under the restriction that each agent can pickup the package at most once; then given the value of B, we can compute in polynomial time a* 2*-approximate delivery strategy with a single pickup per agent. When the graph is undirected, we can compute a* $(2 - 1/2^k)$*-approximation for the same problem in polynomial time.*

Proof Suppose there exists a feasible solution S for the problem using uniform budget B and a single pickup per agent. Consider the fixed (s, t) path P and partition it into segments using the points $X = (m_1, m_2 \dots m_l = t)$ on P, such that $l = \lceil w(P)/B \rceil$ and, the length of the first segment $d_P(s, m_1) \leq B$, the length of segment (m_i, m_{i+1}) is B, $\forall 1 \leq i < l$. We have the following observations for strategy S: (1) Any agent that moves the package over point $m_i \in X$ in strategy S must have enough energy to reach point m_i, and (2) Any single agent cannot transport the package over two distinct points in X since the distance between these 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 $X^+ = (s = m_0, m_1, m_2, \dots m_l)$ 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 m_i), such that, agent a_1 has enough energy to move the package from s to m_1 and $\forall 1 < i < l$, 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 a single pickup per agent and uniform budget B, 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 m_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 $2B - B/2^k$. We consider the same two cases as before.
Case (i): The delivery strategy S uses at least $l + 1$ agents and there is a a type M_0 matching between the agents and $l + 1$ points $s = m_0, m_1, m_2, \dots m_l = t$. Consider the points $m'_i = m_i + B - (2^i - 1)B/2^{l+1}$, $0 \le i \le l - 1$. The agent a_0 can the package from point $s = m_0$ to m'_0 using additional budget of $B(1 - 1/2^{l+1})$. For $0 < i < l$, 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 $B - (2^i - 1)B/2^{l+1} + 2 \times 2^i B/2^{l+1} = B(1 - 1/2^{l+1})$, for each of these agents. Finally, note that the distance between point m'_{l-1} and the target $t = m_l$ is at most $B/2 - B/2^{l+1}$, 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 $2 \times (B/2 - B/2^{l+1}) < B(1 - 1/2^{l+1}) \le B(1 - 1/2^k)$ additional energy.
Case (ii): The delivery strategy S uses l agents and there is a a type M_1 matching between the agents and l points $s = m_1, m_2, \dots m_l = t$. Consider the points $m'_i = m_i + B - (2^i - 1)B/2^l$, $1 \le i \le l - 1$. 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 $2 \times (B/2 - B/2^l) < B(1 - 1/2^l) \le B(1 - 1/2^k)$ additional energy.

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. Similar to the proof of Lemma 2, these computations can be performed in $O(n^3)$ 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 can state the following theorem:

Theorem 6. *The minimum uniform budget required to solve* FixedPath CollaborativeDelivery *with a single pickup per agent on directed (and undirected) graphs can be approximated to a factor* 2 *(respectively* $(2-1/2^k)$*), in polynomial time.*

5. Fixed Path Delivery with $O(1)$ agents

In this section we consider the FixedPath CollaborativeDelivery problem with only a few agents, i.e., when k is a small constant. Further we will assume in this section that the agents are allowed to exchange the package at vertices only. Recall that if there is a single agent ($k=1$) then the problem can be solved trivially (by simply computing the shortest path from the agent to the source). However for $k>1$ agents, the problem is weakly NP-hard.

Theorem 7. FixedPath CollaborativeDelivery *is (weakly) NP-hard for* $k=2$ *agents even if the agents are restricted to pickup the package only at vertices of* G*.*

Proof Consider a complete graph on n vertices where the fixed path P is $(s=v_1,v_2,v_3\dots v_n=t)$, $k=2$ and both the agents are initially located at the source. We show a reduction from the NP-complete problem *Subset-Sum*: Given a set X of n integers $a_1,a_2,\dots a_n$ whose sum is $2S$, does there exist a subset of X whose sum is exactly S?

We construct the instance of FixedPath CollaborativeDelivery by assigning weights $a_1,a_2,\dots,a_n$ to the edges $(v_1,v_2),(v_2,v_3),\dots(v_{n-1},v_n)$ of the path P and we assign weight zero to all other edges of the complete graph. Finally we assign a budget of $B=S$ to each agent. It is easy to see that there is a feasible delivery schedule by the two agents if and only if each agent can move on a subset of edges whose sum of weights equals S, which is equivalent to finding a subset of sum S for the instance of subset sum. □

Given an instance of the decision problem for a specific B, we can design a dynamic programming algorithm that 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).

Theorem 8. *There is an algorithm that decides whether there exists a feasible schedule restricted to pickup at vertices, for* FixedPath CollaborativeDelivery *with uniform budget* B *in undirected or directed graphs. The algorithm runs in* $O(k\cdot n^{k+2}\cdot B^k)$ *time.*

Proof The algorithm works as follows. We keep a boolean table whose entries are of the form $T_v[j|p_1^v, \ldots, p_k^v|B_1^v, \ldots, B_k^v]$ 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 $O(k \cdot n^{k+2} \cdot B^k)$. □

By using the data rounding technique, we turn the developed algorithm into a fully polynomial-time approximation scheme (FPTAS).

Theorem 9. *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.*

Proof 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 $w(\bar{P}_i) \leq \mu \cdot \bar{w}(\bar{P}_i) \leq \mu\bar{B}$.

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 $B^A = \mu \cdot \bar{B} = \mu \cdot \max_i\{\bar{w}(\bar{P}_i)\} \overset{(1)}{\leq} \mu \cdot \max_i\{\bar{w}(P_i^*)\} = \max_i\{\mu \cdot \bar{w}(P_i^*)\} \overset{(2)}{\leq} \max_i\{w(P_i^*) + m^2\mu\} = m^2\mu + \max_i\{w(P_i^*)\} = m^2\mu + B^* = m^2\left(\epsilon \frac{w(P)/k+X}{m^2}\right) + B^* \overset{(3)}{\leq} \epsilon \cdot B^* + B^* = (1+\epsilon)B^*$. 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 $B^* \leq \min_i d(p_i, s) + w(P) \leq (X + w(P)) + w(P) \leq 2(X + w(P))$. Therefore, measured in the units μ, we search for $\bar{B}$ in the range between 1 and $2(X + w(P))/\mu \leq \frac{2m^2k}{\epsilon}$. 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 the minimum such $\bar{B}$ adds a multiplicative logarithmic factor of $\log\left(\frac{2m^2k}{\epsilon}\right)$. This proves the theorem. □

Thus, we have shown the following.

Corollary 1. *There exists an FPTAS for* FixedPath CollaborativeDelivery *restricted to pickup at vertices, when 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. We presented approximation algorithms and lower bounds of approximation for the fixed path version of the problem. These results leave some gaps and we would like to reduce the gap between the upper and lower bounds for the various versions of the problem. We also considered the special case of fixed path delivery with a single pickup per agent, and we were able to find better approximation algorithms for this case compared to the best known algorithm for collaborative delivery without fixed path. This seems to suggest that the fixed path version admits better approximation than the general version, when each agent is restricted to a single pickup. However to prove this we need to find lower bounds on the approximation factor of collaborative delivery. Another possible extension to this work is to consider agents with *non-uniform* budgets and find resource-augmented algorithms for fixed path delivery. Finally, we would like to analyse more precisely the effect of restricting package handovers to nodes only and not anywhere inside the edges.

References

[1] P. Flocchini, G. Prencipe, N. Santoro (Eds.), Distributed Computing by Mobile Entities, Current Research in Moving and Computing, Vol. 11340 of Lecture Notes in Computer Science, Springer, 2019. doi:10.1007/978-3-030-11072-7.

[2] J. Anaya, J. Chalopin, J. Czyzowicz, A. Labourel, A. Pelc, Y. Vaxès, Convergecast and broadcast by power-aware mobile agents, Algorithmica 74 (1) (2016) 117–155.

[3] A. Bärtschi, J. Chalopin, S. Das, Y. Disser, B. Geissmann, D. Graf, A. Labourel, M. Mihalák, Collaborative delivery with energy-constrained mobile robots, Theoretical Computer Science.

[4] J. Chalopin, S. Das, M. Mihalák, P. Penna, P. 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, 2013, pp. 111–122.

[5] J. Chalopin, R. Jacob, M. Mihalák, P. Widmayer, Data delivery by energy-constrained mobile agents on a line, in: 41st International Colloquium on Automata, Languages, and Programming ICALP'14, 2014, pp. 423–434.

[6] S. Das, D. Dereniowski, C. Karousatou, Collaborative exploration by energy-constrained mobile robots, in: 22th International Colloquium on Structural Information and Communication Complexity SIROCCO'15, 2015, pp. 357–369.

[7] M. Dynia, M. Korzeniowski, C. Schindelhauer, Power-aware collective tree exploration, in: 19th International Conference on Architecture of Computing Systems, ARCS'06, 2006, pp. 341–351.

[8] M. Betke, R. L. Rivest, M. Singh, Piecemeal learning of an unknown environment, Machine Learning 18 (2) (1995) 231–254.

[9] B. Awerbuch, M. Betke, R. L. Rivest, M. Singh, Piecemeal graph exploration by a mobile robot, Information and Computation 152 (2) (1999) 155–172.

[10] C. A. Duncan, S. G. Kobourov, V. S. A. Kumar, Optimal constrained graph exploration, in: 12th ACM Symposium on Discrete Algorithms, SODA'01, 2001, pp. 807–814.

[11] S. Das, D. Dereniowski, P. Uznanski, Energy constrained depth first search, CoRR abs/1709.10146. arXiv:1709.10146.
URL `http://arxiv.org/abs/1709.10146`

[12] P. Fraigniaud, L. Gąsieniec, D. R. Kowalski, A. Pelc, Collective tree exploration, Networks 48 (3) (2006) 166–177.

[13] M. Dynia, J. Łopuszański, C. Schindelhauer, Why robots need maps, in: 14th International Colloquium on Structural Information and Communication Complexity, SIROCCO'07, 2007, pp. 41–50.

[14] D. Dereniowski, Y. Disser, A. Kosowski, D. Pająk, P. Uznański, Fast collaborative graph exploration, Information and Computation 243 (2015) 37–49.

[15] E. Bampas, J. Chalopin, S. Das, J. Hackfeld, C. Karousatou, Maximal exploration of trees with energy-constrained agents, CoRR abs/1802.06636. arXiv:1802.06636.

[16] C. Karousatou, Distributed algorithms for energy constrained mobile agents, Ph.D. thesis, Aix-Marseille Université, Marseille, France (2017).

[17] E. D. Demaine, M. Hajiaghayi, H. Mahini, A. S. Sayedi-Roshkhar, S. Oveisgharan, M. Zadimoghaddam, Minimizing movement, ACM Trans. Algorithms 5 (3) (2009) 1–30.

[18] D. Bilò, Y. Disser, L. Gualà, M. Mihalák, G. Proietti, P. Widmayer, Polygon-constrained motion planning problems, in: 9th International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics ALGOSENSORS'13, 2013, pp. 67–82.

[19] J. Czyzowicz, K. Diks, J. Moussi, W. Rytter, Communication problems for mobile agents exchanging energy, in: 23rd International Colloquium on Structural Information and Communication Complexity SIROCCO'16, Vol. 9988 of Lecture Notes in Computer Science, 2016, pp. 275–288.

[20] E. Bampas, S. Das, D. Dereniowski, C. Karousatou, Collaborative delivery by energy-sharing low-power mobile robots, in: 13th International Symposium on Algorithms and Experiments for Wireless Sensor Networks (ALGOSENSORS), 2017, pp. 1–12. doi:10.1007/978-3-319-72751-6_1.

[21] A. Bärtschi, J. Chalopin, S. Das, Y. Disser, D. Graf, J. Hackfeld, P. Penna, Energy-efficient delivery by heterogeneous mobile agents, in: 34th Symposium on Theoretical Aspects of Computer Science (STACS), 2017, pp. 10:1–10:14. doi:10.4230/LIPIcs.STACS.2017.10.

[22] A. Bärtschi, D. Graf, M. Mihalák, Collective Fast Delivery by Energy-Efficient Agents, in: 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018), Vol. 117 of LIPIcs, 2018, pp. 56:1–56:16. doi:10.4230/LIPIcs.MFCS.2018.56.

[23] A. Bärtschi, D. Graf, P. Penna, Truthful Mechanisms for Delivery with Agents, in: 17th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2017), Vol. 59 of OASIcs, 2017, pp. 2:1–2:17. doi:10.4230/OASIcs.ATMOS.2017.2.

[24] A. Bärtschi, Efficient delivery with mobile agents, Ph.D. thesis, ETH Zurich, Zurich (2017). doi:10.3929/ethz-b-000232464.

[25] A. Giannakos, M. Hifi, G. Karagiorgos, Data delivery by mobile agents with energy constraints over a fixed path, CoRR abs/1703.05496. arXiv:1703.05496.

[26] M. R. Garey, D. S. Johnson, Computers and intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman & Co., New York, 1979.

[27] C. A. Tovey, A simplified np-complete satisfiability problem, Discrete Applied Mathematics 8 (1) (1984) 85–89.

[28] M. L. Fredman, R. E. Tarjan, Fibonacci heaps and their uses in improved network optimization algorithms, J. ACM 34 (3) (1987) 596–615. doi:10.1145/28869.28874.