Collaborative delivery by energy-sharing low-power mobile robots*

Evangelos Bampas[1], Shantanu Das[1], Dariusz Dereniowski[2], and Christina Karousatou[1]

[1] LIF, Aix-Marseille University and CNRS, Marseille, France
[2] Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Gdańsk, Poland

Abstract. We study two variants of delivery problems for mobile robots sharing energy. Each mobile robot can store at any given moment at most two units of energy, and whenever two robots are at the same location, they can transfer energy between each other, respecting the maximum capacity. The robots operate in a simple graph and initially each robot has two units of energy. A single edge traversal by an robot reduces its energy by one unit and the robot can only perform such move initially having at least one unit of energy. There are two distinguished nodes s and t in the graph and the goal for the robots is to deliver the *package* initially present on s to the node t. The package can be passed from one robot to another when they are colocated. In the first problem we study, the robots are initially placed at some given nodes of the graph and the question is whether the delivery is feasible. We prove that this problem is NP-complete. In the second problem, the initial positions of the robots are not fixed but a subset of nodes H of the graph is given as input together with an integer k, and the question is as follows: is there a placement of k robots at nodes in H such that the delivery is possible? We prove that this problem can be solved in polynomial time.

Keywords: computational complexity, energy sharing, delivery, mobile robots, power-aware

1 Introduction

We consider algorithms for coordinated tasks performed by swarms of small inexpensive robots. There has been a lot of research interest on designing teams of simple robots that can perform a given task in collaborative fashion. The task we consider is the basic operation of moving an object or a package from its source to its target destination by one or more

* Partially supported by National Science Centre (Poland) grant number 2015/17/B/ST6/01887 and the project ANR-ANCOR (anr-14-CE36-0002-01).

mobile robots. For example, the package could be a sample collected by a robotic sensor that needs to be delivered to a base station for analysis. One can imagine an automated postal delivery system where packages need to be delivered between sources and destinations using teams of robots or drones. We can model the sources and destinations as nodes of a graph and the *Delivery* problem consists of moving a single package from its source node to the target node. The main issue when using small robots is that they have a restricted supply of energy (e.g. a battery) and thus, a robot can move only a limited distance before running out of power. However, many small robots can cooperate to deliver a package from source to destination. Assuming that the robots start from different nodes of the graph, scheduling the moves of the robots for collaborative delivery is known to be a challenging task. Indeed, it was shown by Chalopin et al. [9] that even if the graph is a tree of n nodes, collaborative delivery from single source to a single destination using k robots having energy B each is NP-hard. Czyzowicz et al. [10] studied the problem when the robots can share their energy, i.e. a robot may give its unused energy to another robot. They showed that the problem can be easily solved in trees in polynomial time, but it remains NP-hard in general graphs. For all the above results, the parameters k and B have arbitrary (non-constant) values. Note that collaborative delivery for a constant value of k can be trivially solved by brute force manner in constant time. In this paper, we consider the problem when B is a small constant but k can have arbitrary values. This corresponds to the case of many small robots each operating with low power batteries such that each robot can move only for a constant number of steps in the graph. A robot with depleted energy can gather the unused energy of any other robot that it meets. However, no robot can have more than B units of energy at any time. Surprisingly, we show that even when $B = 2$, the smallest constant for which the problem is non-trivial, collaborative delivery is still NP-hard in general graphs. On the other hand, we provide an optimal polynomial-time algorithm for collaborative delivery with robots having $B = 2$ if we are allowed to choose the initial placement of robots among designated homebase nodes in the graph. Note that if we are allowed to place robots on any node of the graph, then there is a trivial optimal solution where all robots are placed on the shortest s-t path at intervals of distance 2. At the other extreme, if there is only one homebase node, then the solution is non-trivial.

Our Contributions. We completely solve the problem of collaborative delivery for $B = 2$ when energy sharing is allowed. We define two versions of the problem. In the first version of the problem called Collabora-

TIVEDELIVERY *with Fixed Placement*, the initial placement of robots (i.e. the energy distribution is given as part of the problem). In the second version of the problem called COLLABORATIVEDELIVERY *with Chosen Placement*, a set of homebase nodes is given and the algorithm may choose the distribution of robots among the homebase nodes. We show that the first version of the problem is strongly NP-hard, while the second version of the problem admits a polynomial time solution and we present such a solution strategy. Proofs are omitted due to space constraints and will appear in the full version of the paper.

Related Work. Betke et al. [6] considered for the first time energy-constrained robots in the context of exploration of grid graphs by a robot who can return to its starting node for refueling. Awerbuch et al. [2] studied the same problem for general graphs. Duncan et al. [16] studied a similar model where the robot is tethered to its starting position with a rope of fixed length and they optimized the exploration time. Several other papers have considered robots with limited energy, or the goal of minimizing spent energy or maximum displacement of robots, e.g. in the context of exploration [19, 17, 18, 12, 14, 20], formation [13, 7], coverage [11, 15], and broadcast/convergecast [10, 1] problems.

In what concerns specifically collaborative delivery by energy-aware mobile robots, the problem has been considered in some recent works under various assumptions. In [8], the authors assume robots with limited energy that is consumed as they move. They prove that the problem of deciding whether delivery is feasible is NP-hard even if the robots are initially collocated, and they provide a 2-approximation algorithm for the optimization version of finding the minimum initial energy that can be given to all robots so that delivery becomes feasible, as well as exact, approximation, and resource-augmented algorithms for variants of the problem. In [9], the authors show that the problem is weakly NP-hard even on the line (with initially dispersed robots), and provide a quasi-, pseudo-polynomial algorithm under the assumption of integer numerical values in the problem instance. In [3], the authors consider the variant of the delivery probem in which the robots have to return to their respective starting positions and they prove that this problem is NP-hard for planar graphs but can be solved efficiently on trees and lines, in contrast to the non-returning version which is NP-hard on lines. They also give resource-augmented algorithms for returning delivery in general graphs and prove tight lower bounds on the resource augmentation for both the returning and the non-returning variant. In [4], the robots do not have a limited energy source, but instead they have different rates of energy consumption

and the goal is to find a delivery schedule that minimizes the total energy spent. Moreover, there are several messages that need to be delivered from their respective source to their respective target. The authors study separately three subtasks that need to be solved in order to compute the optimal solution (collaboration of different robots on the same message, planning for a robot that works on multiple messages, and assignment of messages to robots) and they provide a polynomial-time (nonconstant) approximation algorithm for the problem. In this setting, [5] studies the design of truthful mechanisms in a game-theoretic model where the rate of energy consumption is information private to each robot.

The only previous work that considers *energy sharing* by mobile robots is [10], where this feature is introduced in the context of the delivery and convergecast problems. The authors show that both problems can be solved efficiently in trees, whereas they are NP-complete in general undirected and directed graphs. It is important to note that, in the model of [10], a robot may store an *unlimited* amount of energy as a result of receiving energy from other robots it encounters. In other words, there is no battery capacity constraint for the robots, in contrast to the model that we study in the present paper.

2 Preliminaries

We now define precisely the *collaborative delivery* problem for a collection of energy sharing mobile robots. Given a simple undirected graph $G = (V, E)$, with two special nodes s (source) and t (target), and a collection of k mobile robots located initially in specific nodes of the graph, the objective is to decide whether there is schedule of robot moves that can deliver a package from s to t. Each robot has a constant energy budget and we denote its value by B. Traversing each edge consumes one unit of energy, thus, a fully charged robot can move a distance of B before running out of energy. When two robots are at the same node, one robot can transfer to the other robot, any integral part of its energy, with the only constraint that no robot can have more than B units of energy at any time.

In this paper, $B = 2$ for all robots. There is a unique package initially at node s that needs to be moved to node t. To simplify the discussion, we will assume that the system is synchronous (any synchronous strategy can also be implemented in an asynchronous system using appropriate waits). A robot r located at a node v at time j and having some positive energy, can perform any subset of the following actions:

– Pick up the package, if the package is present at v at time j.
– Transfer one unit of energy to another robot r' that is located at v at time j, if r' is not fully charged.
– Move to a neighboring node u, consuming one unit of energy and arriving at u at time $j + 1$.

A solution strategy is a sequence of steps as above, such that after the last step, the package is located at node t.

In the general version of the problem described below, the position of the robots is given by an adversary.

*Problem 1 (*CollaborativeDelivery *with Fixed Placement).* CDX
Instance: $\langle G, s, t, k, h\rangle$, where $G = (V, E)$ is a simple undirected graph, $s, t \in V$ are, respectively, the source and target nodes, $h : \{1, \ldots, k\} \to V$ is the placement function that specifies the initial positions of the $k \geq 1$ robots.
Question: Does there exist a solution strategy for moving the package from s to t, when each robot start with $B = 2$ units of energy?

If the placement of robots among homebase nodes can be chosen by the algorithm, then we have the following version of the problem:

*Problem 2 (*CollaborativeDelivery *with Chosen Placement).* CDC
Instance: $\langle G, s, t, k, H\rangle$, where $G = (V, E)$ is a simple undirected graph, $s, t \in V$ are, respectively, the source and target nodes, $k \geq 1$ is the number of mobile robots and $H \subset V$ is the set of homebase nodes.
Question: Does there exist a placement function $p : \{1, \ldots, k\} \to H$ and a corresponding solution strategy for moving the package from s to t, when the i-th robot starts at node $p(i)$ with $B = 2$ units of energy?

A path with nodes $v_1, \ldots, v_n$ and edges $\{v_i, v_{i+1}\}$, $i \in \{1, \ldots, n-1\}$, is denoted by $(v_1, \ldots, v_n)$. The path graph with n vertices and $n-1$ edges is denoted by P_n. The length of the shortest path between two nodes u, v of a graph G is denoted by $d_G(u, v)$, or simply $d(u, v)$ when there is no potential for confusion. If a robot is initially placed on a node v, then we write $a(v)$ to refer to this robot at any point of a strategy.

We obtian the following regarding the hardness of problem CDX:

Theorem 1. CDX *is* NP*-complete in the class of graphs with degree bounded by* 5 *and with each node being initially occupied by at most one robot.*

Theorem 2. CDX *is* NP*-complete in the class of graphs with diameter at most* 42 *with each node being initially occupied by at most one robot.*

3 An efficient algorithm for CollaborativeDelivery *with Chosen Placement*

Let $I = \langle G, s, t, H \rangle$ be an instance of CDC, with $G = (V, E)$ and $H \subseteq V$. Recall that a solution to I is a strategy that enables a group of energy-exchanging robots starting from some or all of the nodes in H with battery capacity $B = 2$ to transfer the package from s to t. The cost of a solution is the total initial energy of the robots that are placed on nodes in H. If $u \in V$, we denote by h_u the node in H that is closest to u, breaking ties arbitrarily, and we denote by $z(u)$ the distance from u to h_u in G, i.e., $z(u) = d_G(h_u, u)$. Let $\hat{G}$ be a weighted complete digraph with vertex set V and arc set E', and let the weight of an arc $e = (u, v) \in E'$ be $w(e) = 2^{z(u)+d_G(u,v)-1}$. If P is a directed path in $\hat{G}$, the *weight of* P is the sum of the weights of the arcs in P. This section is devoted to the proof of the following theorem:

Theorem 3. *An optimal solution to* $I = \langle G, s, t, H \rangle$ *has cost* $d_{\hat{G}}(s, t)$.

It suffices to show that there exists a directed s-t path in $\hat{G}$ with weight smaller than or equal to the cost of the optimal solution for I and, additionally, that every directed s-t path in $\hat{G}$ corresponds to some solution for I with cost equal to the weight of the path. The latter claim is given in the following lemma:

Lemma 1. *For every directed s-t path in $\hat{G}$, there exists a solution for I with cost equal to the weight of the path.*

In the rest of this section, we derive some structural properties of optimal solutions for I, which permit us to prove the former claim (cf. Lemma 5). In Section 3.1 we introduce our main tool in the analysis: an energy flow hypergraph that provides a way of presenting solutions to CDC. Then, Sections 3.2 and 3.3 give a series of properties of this hypergraph, allowing us to finish the proof of Theorem 3 in Section 3.4. We assume that $s \neq t$, otherwise Theorem 3 holds trivially.

3.1 The energy flow hypergraph

Given a solution $\mathcal{S}$ for I with cost $X > 0$, we represent $\mathcal{S}$ by a triple $\mathcal{S} = (\mathcal{V}, \mathcal{E}, \tilde{\mathcal{E}})$, where $(\mathcal{V}, \mathcal{E}) = \mathcal{H}$ is a directed hypergraph that represents the flow and eventual consumption of energy units (cf. Definition 1 below) and $\tilde{\mathcal{E}} \subseteq \mathcal{E}$ corresponds to the package moves under $\mathcal{S}$ (cf. Definition 2

below). The nodes of $\mathcal{H}$ are the energy arrival and extinction events of $\mathcal{S}$, as specified below.

We assume that the units of energy that are initially present at nodes in H receive distinct identities from 1 to X. We distinguish two types of *events* during the delivery under $\mathcal{S}$. An *arrival* event occurs whenever a robot with two units of energy i, j with $i < j$ moves from some node u of G to a neighbor v at time step t. We say that the unit of energy j is *wasted by* i during the event and that the unit of energy i *arrives* at v at time $t+1$. We denote this event as $(i, t+1, v)$. An *extinction* event occurs whenever a robot with one unit of energy i moves from some node u of G to a neighbor v at time step t. We say that the unit of energy i *wastes itself* during the event. We denote this event as $(\perp_i, t+1, v)$.

We also consider as *arrival* events the appearance of the X units of energy at the homebases at time 0, and we denote them as $(i, 0, h_i)$, for $1 \leq i \leq X$, where $h_i \in H$ is the homebase where energy unit i was placed.

We are now ready to define the *energy flow hypergraph* $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ in terms of the arrival and extinction events as follows:

Definition 1 (Energy flow hypergraph). *The vertex set $\mathcal{V}$ contains all of the arrival and extinction events, as specified above. The hyperarc set $\mathcal{E}$ contains $(\{(i, t_1, u), (j, t_2, u)\}, \{(i, t, v)\})$ if $t \geq 1$, the unit of energy i came from node u during the event (i, t, v), $j > i$ is the unit of energy consumed during the event (i, t, v), and in addition $t_1 < t$, $t_2 < t$, and i (resp. j) is not involved in any other events between times t_1 (resp. t_2) and t. Furthermore, $\mathcal{E}$ contains $(\{(i, t_1, u)\}, \{(\perp_i, t, v)\})$ if $t_1 < t$ and i is not involved in any other events between times t_1 and t.*

Definition 2 (Item moves). *The set $\tilde{\mathcal{E}}$ is defined as the subset of $\mathcal{E}$ that contains all of the hyperarcs that correspond to package moves. More precisely, a hyperarc $(\{(i, t_1, u), (j, t_2, v)\}, \{(i, t, v)\}) \in \tilde{\mathcal{E}}$ if the robot that arrived with energy unit i at v at time t was carrying the package. Similarly, a hyperarc $(\{(i, t_1, u)\}, \{(\perp_i, t, v)\}) \in \tilde{\mathcal{E}}$ if the robot that arrived with zero energy at v at time t (having wasted energy unit i) was carrying the package.*

We illustrate the energy flow hypergraph $\mathcal{H}$ corresponding to a simple solution for a CDC instance in Figure 1. Note that the cost of $\mathcal{S}$ is given by the total number of energy units that "arrive" at nodes in H at time 0, which corresponds to the number of nodes of the form $(\cdot, 0, \cdot)$ in $\mathcal{H}$.

By construction, there is no cycle in $\mathcal{H}$. Moreover, the head of every hyperarc of $\mathcal{H}$ has size 1 and every node of $\mathcal{H}$ is contained in at most one

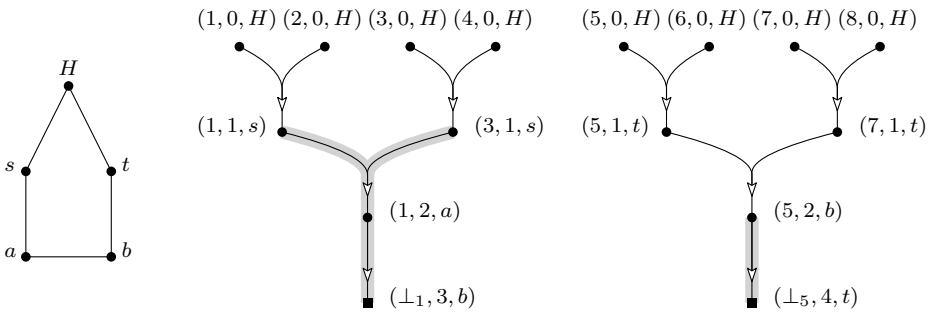

Fig. 1. An energy flow hypergraph constructed for a graph G shown on the left. The hypergraph consists of two components (the hyperarcs that correspond to package moves are highlighted): the first component dictates two robots to move from H to s and then one of those robots picks up the package and travels along path (s, a, b); the second component makes two robots to move from H to t and then one of them goes to b, picks up the package and returns to t.

hyperarc head and in at most one hyperarc tail. Therefore, $\mathcal{H}$ consists of a number of independent components $\mathcal{H}_1, \ldots, \mathcal{H}_\sigma$, each of which has a tree-like structure, as in Figure 1.

Notation If $e \in \mathcal{E}$, we denote by $\mathsf{head}(e)$ the unique node that is in the head of e. If $v \in \mathcal{V}$, we denote by $\mathfrak{g}(v)$ the node of G that is involved in the event v. If $e \in \mathcal{E}$, then we denote by $\mathfrak{g}_{\mathsf{tail}}(e)$ the node of G that is involved in the events in the tail of e (recall that, by definition of the hypergraph $\mathcal{H}$, all events in the tail of e must involve the same node of G), and by $\mathfrak{g}_{\mathsf{head}}(e)$ the node of G that is involved in the unique event in the head of e (i.e., $\mathfrak{g}_{\mathsf{head}}(e) = \mathfrak{g}(\mathsf{head}(e))$).

If $v \in \mathcal{V}$, let Δv denote the subgraph of $\mathcal{H}$ induced by the ancestors of v and v itself. Let $\mathsf{height}(v)$ denote the number of hyperarcs in the longest path that terminates at v. If $e \in \mathcal{E}$, we abuse the notation slightly and we denote by $\mathsf{height}(e)$ the height of $\mathsf{head}(e)$. If $v, v' \in \mathcal{V}$, we write $v \prec v'$ if v is an ancestor of v' and we write $v \sqsubset v'$ if v precedes v' temporally, i.e., $v = (i, t, x)$ and $v' = (i', t', x')$ with $t < t'$. Note that $v \prec v'$ implies $v \sqsubset v'$. As above, we extend the notation to arcs and we write $e \prec e'$ if $\mathsf{head}(e) \prec \mathsf{head}(e')$ and $e \sqsubset e'$ if $\mathsf{head}(e) \sqsubset \mathsf{head}(e')$.

If Δv contains x nodes of the form $(\cdot, 0, \cdot)$, then we say that Δv *incurs a cost of* x. This represents the energy units used by the solution in order to generate the event v. We also say that a component $\mathcal{H}_i$ incurs a cost equal to the cost incurred by its maximal node (under $\prec$). The cost of $\mathcal{S}$ is the sum of the costs incurred by the components of $\mathcal{H}$.

3.2 Properties of optimal solutions

The goal of this section is to prove a property of optimal solutions that can be informally stated as follows: every component of the hypergraph corresponding to the solution contains exactly one chain of item moves and the last hyperarc of this chain is an extinction event of the component.

Proposition 1. *For every solution $\mathcal{S} = (\mathcal{V}, \mathcal{E}, \tilde{\mathcal{E}})$ in which there exist arcs $f, g \in \tilde{\mathcal{E}}$ with $f \prec g$, there exists a solution $\mathcal{S}' = (\mathcal{V}, \mathcal{E}, \tilde{\mathcal{E}}')$ with $\tilde{\mathcal{E}}' = \tilde{\mathcal{E}} \setminus \{e : f \sqsubset e \sqsubset g\} \cup \{e : f \prec e \prec g\}$.*

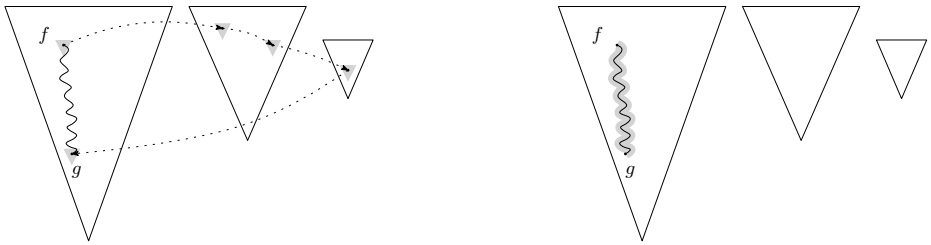

Fig. 2. Illustration of Proposition 1. Triangles represent components of the solution hypergraph. The solution $\mathcal{S}$ is shown on the left (in which some arcs of the path from f to g are not package moves — those package moves can be possibly in different components; the dotted arrows represent the time succession between package moves that take the package from f to g) and the corresponding $\mathcal{S}'$ is shown on the right.

By repeated application of Proposition 1, we obtain the following:

Corollary 1. *For every solution $\mathcal{S} = (\mathcal{V}, \mathcal{E}, \tilde{\mathcal{E}})$, there exists a solution $\mathcal{S}' = (\mathcal{V}, \mathcal{E}, \tilde{\mathcal{E}}')$ and a partition of $\tilde{\mathcal{E}}'$ into sets $\tilde{\mathcal{E}}'_1, \ldots, \tilde{\mathcal{E}}'_\tau$ such that, for every i, the hyperarcs of $\tilde{\mathcal{E}}'_i$ form a chain in $\mathcal{H}$ and, for every $e \in \tilde{\mathcal{E}}'_i$ and $e' \in \tilde{\mathcal{E}}'_j$ with $i < j$, we have $e \sqsubset e'$, $e \not\prec e'$, and $e' \not\prec e$.*

Note that Proposition 1 and Corollary 1 apply to *any* solution (not necessarily an optimal one). Furthermore, in both statements, the obtained solution $\mathcal{S}'$ has the same energy flow hypergraph as $\mathcal{S}$, and therefore it has the same cost as $\mathcal{S}$.

Lemma 2. *For every optimal solution $\mathcal{S} = (\mathcal{V}, \mathcal{E}, \tilde{\mathcal{E}})$ and for every component $\mathcal{H}_i$ of $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ with maximum (under $\prec$) hyperarc r_i, we have $r_i \in \tilde{\mathcal{E}}$ and $\mathsf{head}(r_i)$ is an extinction event.*

By applying Corollary 1 to an arbitrary optimal solution, we obtain the following corollary in view of Lemma 2:

Corollary 2. *There exists an optimal solution $\mathcal{S} = (\mathcal{V}, \mathcal{E}, \tilde{\mathcal{E}})$ such that every component $\mathcal{H}_i$ of $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ with maximum (under $\prec$) hyperarc r_i contains exactly one chain of package moves whose last hyperarc is r_i and, in addition, $\mathsf{head}(r_i)$ is an extinction event.*

3.3 Canonical nodes

Given a solution $\mathcal{S} = (\mathcal{V}, \mathcal{E}, \tilde{\mathcal{E}})$ for I, let $v \in \mathcal{V}$ such that Δv does not contain any hyperarc in $\tilde{\mathcal{E}}$. Intuitively, if v is not an extinction event, then the sole function of Δv in the solution is to bring one unit of energy to $\mathfrak{g}(v)$. It thus makes sense that, if $\mathcal{S}$ is optimal, then the energy units that participate in the events of Δv travel along shortest paths from their respective homebases to $\mathfrak{g}(v)$. If v satisfies these conditions, then we say that v is *canonical*. The following definition captures this notion:

Definition 3 (Canonical nodes). *Given a solution $\mathcal{S} = (\mathcal{V}, \mathcal{E}, \tilde{\mathcal{E}})$, a node $v \in \mathcal{V}$ is called* canonical *if either $\mathsf{height}(v) = 0$, or $\mathsf{height}(v) = h+1$ for some $h \geq 0$ (in this case, $v = \mathsf{head}(e)$ for some $e \in \mathcal{E}$) and all of the following hold:* (i) *for every node u in the tail of e, u is canonical and $\mathsf{height}(u) = h$,* (ii) *$e \notin \tilde{\mathcal{E}}$, and* (iii) *$z(\mathfrak{g}(v)) = 1 + z(\mathfrak{g}_{\mathsf{tail}}(e))$.*

The two propositions below follow easily by induction on the height of v. Recall that, by definition of $\mathcal{H}$, if $\mathsf{height}(v) = 0$ then $\mathfrak{g}(v) \in H$.

Proposition 2. *If v is canonical, then $z(\mathfrak{g}(v)) = \mathsf{height}(v)$.*

Proposition 3. *If v is canonical and it is not an extinction event, then the cost incurred by Δv is $2^{\mathsf{height}(v)}$.*

We can now prove that, in every optimal solution, every node v whose Δv does not contain any package move is canonical.

Lemma 3. *For every optimal solution $\mathcal{S} = (\mathcal{V}, \mathcal{E}, \tilde{\mathcal{E}})$ and for every $v \in \mathcal{V}$, v is canonical or Δv contains a hyperarc in $\tilde{\mathcal{E}}$.*

3.4 Completing the proof of Theorem 3

In the following, let $\mathcal{S}^\star = (\mathcal{V}, \mathcal{E}, \tilde{\mathcal{E}})$ be an optimal solution as guaranteed by Corollary 2, with the maximum number σ of components of $\mathcal{H} = (\mathcal{V}, \mathcal{E})$. Let $(\mathcal{H}_i)_{i=1,\ldots,\sigma}$ be an enumeration of the components of $\mathcal{H}$ in temporal order of their extinction events. For $i \in \{1, \ldots, \sigma\}$, component $\mathcal{H}_i$ is responsible for moving the package along a path $P_i = (u_{i,0}, u_{i,1}, \ldots, u_{i,\rho_i})$ in G, where $u_{1,0} = s$, $u_{\sigma,\rho_\sigma} = t$, and $u_{i,\rho_i} = u_{i+1,0}$ (for $i < \sigma$).

Lemma 4. *For every i, j in the ranges $1 \leq i \leq \sigma$ and $0 \leq j < \rho_i - 1$, $z(u_{i,j+1}) = 1 + z(u_{i,j})$.*

Corollary 3. *The cost incurred by component $\mathcal{H}_i$ is $2^{z(u_{i,0})+\rho_i-1}$.*

Let Q be the directed s-t path in $\hat{G}$ that consists of the arcs $(u_{1,0}, u_{1,\rho_1})$, $(u_{2,0}, u_{2,\rho_2}), \ldots, (u_{\sigma,0}, u_{\sigma,\rho_\sigma})$. By definition of $\hat{G}$, the i-th arc of Q has weight $2^{z(u_{i,0})+d_G(u_{i,0},u_{i,\rho_i})-1}$. However, P_i is a path in G from $u_{i,0}$ to u_{i,ρ_i} and its length is ρ_i. Therefore, $d_G(u_{i,0}, u_{i,\rho_i}) \leq \rho_i$. In view of Corollary 3, we conclude that the weight of the i-th arc of Q is at most equal to the cost of $\mathcal{H}_i$ and thus the total weight of the arcs of Q is at most equal to the cost of $\mathcal{S}^\star$. We have proved the following lemma, which concludes the proof of Theorem 3:

Lemma 5. *There exists a directed s-t path in $\hat{G}$ with weight at most equal to the cost of an optimal solution to I.*

4 Concluding remarks

Our work reveals an interesting differentiation in the complexity of the collaborative delivery problem by robots with battery capacity $B = 2$, depending on whether the energy allocation to the homebases (starting nodes of robots) is given as part of the input on the one hand, or the allocation can be chosen as part of the solution on the other hand.

As we showed, the problem with fixed allocation of energy units to the homebases is NP-complete. However, we proved in Section 3 that the delivery problem in which one is given the total available energy and is asked if it is possible to distribute this energy to robots at the homebases in order to achieve delivery is solvable in polynomial time. In fact, what we proved is that the underlying optimization problem, i.e., finding the minimum amount of energy that can be distributed to the homebases so that delivery is feasible, is solvable in polynomial time by reduction to a shortest path computation in a complete directed graph.

A natural question is how to handle greater battery capacities $B \geq 3$. While we expect that our NP-completeness reduction generalized to $B \geq 3$, the situation is less clear when it comes to the question of computing the energy allocation to the homebases as part of the solution. A straightforward adaptation of our algorithm from Section 3 would be to reduce the problem to computing the shortest s-t path in a directed graph $\hat{G}$ similar to the one we construct in Section 3, except that the weight of an arc (u, v) would be equal to the minimum amount of energy required

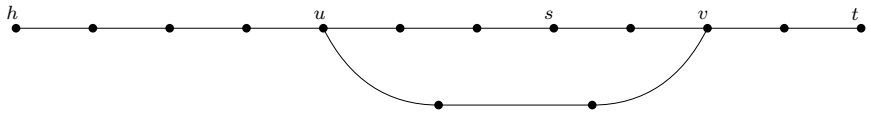

Fig. 3. An example of a graph in which the straightforward adaptation of our algorithm from Section 3 to $B = 3$ does not give an optimal solution. Here, $H = \{h\}$ and the shortest s-t path in $\hat{G}$ is $s \to v \to t$, with each arc having a weight of 41 for a total cost of 82. However, the optimal solution has a cost of 81: 27 fully charged robots start from h and they reach u with 16 remaining energy units in total. At u, the robots split into two groups with 8 units of energy each. The first group goes to s and then to v from the top branch, picking up the package from s on the way. The second group goes to v from the bottom branch, picks up the package, and continues until t.

by robots with capacity B to traverse the path $h_u \rightsquigarrow u \rightsquigarrow v$, where h_u is the nearest homebase to u. Unfortunately, this algorithm is no longer guaranteed to produce an optimal solution for $B \geq 3$ (see Figure 3).

The reason is that several nice properties of the optimal solutions for $B = 2$ no longer hold for $B \geq 3$. In particular, since the hyperarcs in the energy flow hypergraph can now have up to two nodes in their heads, each component can now have more than one bottommost nodes and it can contain more than one chains of package moves. This is exactly the case in the example of Figure 3, where the energy flow hypergraph of the optimal solution has only one component, which contains two chains of package moves. Furthermore, it is no longer the case that all of the nodes in the tail of a given hyperarc have the same height. This can be seen even in cases where the optimal solution consists of only one component with only one chain of package moves (see Figure 4).

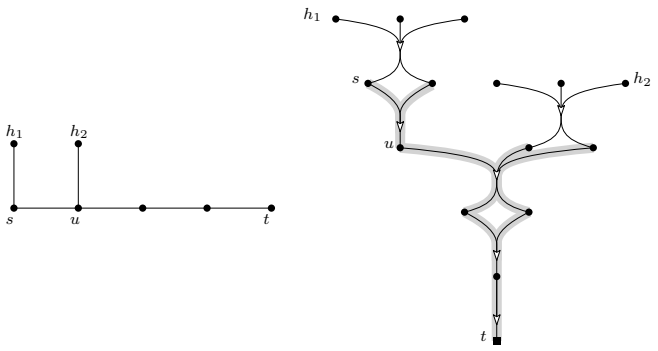

Fig. 4. An example of a graph (*left*, with $H = \{h_1, h_2\}$) in which the energy flow hypergraph of the optimal solution (*right*) contains a hyperarc with nodes of different heights in its tail.

References

1. Anaya, J., Chalopin, J., Czyzowicz, J., Labourel, A., Pelc, A., Vaxès, Y.: Convergecast and broadcast by power-aware mobile agents. Algorithmica 74(1), 117–155 (2016)
2. Awerbuch, B., Betke, M., Rivest, R.L., Singh, M.: Piecemeal graph exploration by a mobile robot. Information and Computation 152(2), 155–172 (1999)
3. Bärtschi, A., Chalopin, J., Das, S., Disser, Y., Geissmann, B., Graf, D., Labourel, A., Mihalák, M.: Collaborative delivery with energy-constrained mobile robots. In: Suomela, J. (ed.) Structural Information and Communication Complexity - 23rd International Colloquium, SIROCCO 2016, Helsinki, Finland, July 19-21, 2016, Revised Selected Papers. Lecture Notes in Computer Science, vol. 9988, pp. 258–274 (2016)
4. Bärtschi, A., Chalopin, J., Das, S., Disser, Y., Graf, D., Hackfeld, J., Penna, P.: Energy-efficient delivery by heterogeneous mobile agents. In: Vollmer, H., Vallée, B. (eds.) 34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany. LIPIcs, vol. 66, pp. 10:1–10:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2017)
5. Bärtschi, A., Graf, D., Penna, P.: Truthful mechanisms for delivery with mobile agents. CoRR abs/1702.07665 (2017)
6. Betke, M., Rivest, R.L., Singh, M.: Piecemeal learning of an unknown environment. Machine Learning 18(2), 231–254 (1995)
7. Bilò, D., Disser, Y., Gualà, L., Mihalák, M., Proietti, G., Widmayer, P.: Polygon-constrained motion planning problems. In: 9th International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics ALGOSENSORS'13. pp. 67–82 (2013)
8. Chalopin, J., Das, S., Mihalák, M., Penna, P., Widmayer, P.: 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. pp. 111–122 (2013)
9. Chalopin, J., Jacob, R., Mihalák, M., Widmayer, P.: Data delivery by energy-constrained mobile agents on a line. In: 41st International Colloquium on Automata, Languages, and Programming ICALP'14. pp. 423–434 (2014)
10. Czyzowicz, J., Diks, K., Moussi, J., Rytter, W.: Communication problems for mobile agents exchanging energy. In: 23rd International Colloquium on Structural Information and Communication Complexity SIROCCO'16 (2016)
11. Czyzowicz, J., Kranakis, E., Krizanc, D., Lambadaris, I., Narayanan, L., Opatrny, J., Stacho, L., Urrutia, J., Yazdani, M.: On minimizing the maximum sensor movement for barrier coverage of a line segment. In: 8th International Conference on Ad-Hoc, Mobile and Wireless Networks, ADHOC-NOW'09. pp. 194–212 (2009)
12. Das, S., Dereniowski, D., Karousatou, C.: Collaborative exploration by energy-constrained mobile robots. In: 22th International Colloquium on Structural Information and Communication Complexity SIROCCO'15. pp. 357–369 (2015)
13. Demaine, E.D., Hajiaghayi, M., Mahini, H., Sayedi-Roshkhar, A.S., Oveisgharan, S., Zadimoghaddam, M.: Minimizing movement. ACM Trans. Algorithms 5(3), 1–30 (2009)
14. Dereniowski, D., Disser, Y., Kosowski, A., Pająk, D., Uznański, P.: Fast collaborative graph exploration. Information and Computation 243, 37–49 (2015)

15. Dobrev, S., Durocher, S., Hesari, M.E., Georgiou, K., Kranakis, E., Krizanc, D., Narayanan, L., Opatrny, J., Shende, S.M., Urrutia, J.: Complexity of barrier coverage with relocatable sensors in the plane. Theoretical Computer Science 579, 64–73 (2015)
16. Duncan, C.A., Kobourov, S.G., Kumar, V.S.A.: Optimal constrained graph exploration. In: 12th ACM Symposium on Discrete Algorithms, SODA'01. pp. 807–814 (2001)
17. Dynia, M., Korzeniowski, M., Schindelhauer, C.: Power-aware collective tree exploration. In: 19th International Conference on Architecture of Computing Systems, ARCS'06. pp. 341–351 (2006)
18. Dynia, M., Łopuszański, J., Schindelhauer, C.: Why robots need maps. In: 14th International Colloquium on Structural Information and Communication Complexity, SIROCCO'07. pp. 41–50 (2007)
19. Fraigniaud, P., Gąsieniec, L., Kowalski, D.R., Pelc, A.: Collective tree exploration. Networks 48(3), 166–177 (2006)
20. Ortolf, C., Schindelhauer, C.: Online multi-robot exploration of grid graphs with rectangular obstacles. In: 24th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA'12. pp. 27–36 (2012)