Near-gathering of energy-constrained mobile agents*

Andreas Bärtschi[1], Evangelos Bampas[2], Jérémie Chalopin[3], Shantanu Das[3], Christina Karousatou[4], and Matúš Mihalák[5]

[1] Center for Nonlinear Studies, Los Alamos National Laboratory, USA
baertschi@lanl.gov
[2] Laboratoire de Recherche en Informatique, Université Paris-Sud, France
evangelos.bampas@lri.fr
[3] CNRS, Aix-Marseille Université and Université de Toulon, LIS, France
jeremie.chalopin@lis-lab.fr, shantanu.das@lis-lab.fr
[4] Department of Mathematics, TU Darmstadt, Germany
karousatou@mathematik.tu-darmstadt.de
[5] Department of Data Science and Knowledge Engineering, Maastricht University, The Netherlands matus.mihalak@maastrichtuniversity.nl

Abstract. We study the task of gathering k energy-constrained mobile agents in an undirected edge-weighted graph. Each agent is initially placed on an arbitrary node and has a limited amount of energy, which constrains the distance it can move. Since this may render gathering at a single point impossible, we study three variants of *near-gathering*:
The goal is to move the agents into a configuration that minimizes either (i) the radius of a ball containing all agents, (ii) the maximum distance between any two agents, or (iii) the average distance between the agents. We prove that (i) is polynomial-time solvable, (ii) has a polynomial-time 2-approximation with a matching NP-hardness lower bound, while (iii) admits a polynomial-time $2(1-\frac{1}{k})$-approximation, but no FPTAS, unless $P = NP$. We extend some of our results to additive approximation.

Keywords: mobile agents · power-aware robots · limited battery · gathering · graph algorithms · approximation · computational complexity

1 Introduction

The problem of *gathering* is one of the fundamental problems in distributed computing with mobile entities, which includes mobile agents moving in a graph or robots moving in a continuous geometric space. In both cases, the objective is to bring together multiple autonomous agents

* This work was partially supported by the SNF (project 200021L_156620) and by the ANR (project ANCOR anr-14-CE36-0002-01), while A. Bärtschi was working at ETH Zürich, and E. Bampas and C. Karousatou were working at Aix-Marseille Université. The Los Alamos National Laboratory report number is LA-UR-19-23906.

to a single point (not predetermined). Gathering helps in coordination between the mobile agents, sharing of information between the entities, reassignment of duties among the entities, and even for protection of the agents (a group of robots gathered together is easier to protect than those dispersed in large area). Moreover, there are also theoretical reasons for studying gathering as the problem of selecting a gathering point is akin to problems of leader election and consensus in distributed systems. However, in some cases, it may be impossible to solve the problem of gathering, e.g. due to limitations in the capabilities of the agents, or due to symmetries in their perception of the environment. In some cases it may be desirable for the agents to get close to each other without actually meeting [27].

In this paper, we consider mobile agents moving on a graph, with severe limitations on their movements. We assume that the agents have limited energy resources and traversing any edge of the graph consumes some of this energy which can not be replenished. In other words, each agent has an initial energy budget which limits the total distance it can move in the graph. Under such constraints, it is not always possible to gather the agents at a single point. Thus, we consider the problem of moving the agents as close as possible to each other while respecting the movement constraints, defined below as the *near-gathering* problem.

Near-Gathering. A collection of k mobile agents is initially located at an arbitrary set of nodes of an undirected edge-weighted graph $G = (V, E, \omega)$. Each agent i, $i = 1, \dots, k$, has an energy capacity b_i, which represents the maximum distance the agent can move in the graph. The agents have *global knowledge* of the graph and are controlled by a *central entity*. The goal is to move the agents to a configuration where they are as close to each other as possible under their respective limitations of movement. Closeness criteria can be measured, e.g., as the size of the smallest region enclosing all the agents, or as the maximum or average pairwise distance between the agents. We look at each of these criteria and give a more precise definition of the problem below.

Our Model. We consider an undirected graph $G = (V, E, \omega)$, where each edge $e \in E$ has a positive weight $\omega(e) > 0$. As usual, the length of a path is the sum of the weights of its edges. We think of every edge $e = \{u, v\}$ as a segment of infinitely many points, where every point in the edge is uniquely characterized by its distance from u, which is between 0 and $w(e)$. We consider every such point to subdivide the edge $\{u, v\}$ into two edges of lengths proportional to the position of the point on the edge.

Thus, the distance $d(p, q)$ between two points p and q (nodes or points inside edges) is the length of a shortest path from p to q in G (with edges subdivided by p, q, respectively). For a point p inside an edge $e \in E$ we write $p \in G$ and $p \in seg(e)$.

A collection of k mobile agents is initially located at an arbitrary set of nodes $p_1, \ldots, p_k \in V$. Each agent i is equipped with an energy budget $b_i > 0$ and can move along edges of the graph, for a distance of at most b_i. In the *Near-Gathering* problem, the goal is to relocate every agent into a new position such that the resulting configuration minimizes one of the following objectives: (i) the radius of a smallest ball containing all agents, (ii) the maximum distance between any two agents, or (iii) the average distance between the agents (or, equivalently, the sum of all distances). We are further interested in two variants of the problem, where agents can: (I) only be relocated to *reachable nodes of the graph*, or (II) in a more general scenario, where the agents are allowed to be relocated to *reachable points* (i.e., nodes or points inside edges).

Definition 1 (Near-Gathering).

Instance: $I = \langle G, k, (p_i)_{i=1,\ldots,k}, (b_i)_{i=1,\ldots,k} \rangle$, where $G = (V, E, \omega)$ is an undirected edge-weighted graph, k denotes the total number of agents, p_i denotes the initial positions of the agents and b_i denotes the total amount of energy each agent initially has at its disposal.

Feasible solution: Any configuration $\mathbf{C} = (c_1, \ldots, c_k)$ of agent end positions c_i, in which for each agent i, $1 \leq i \leq k$, we have $d(p_i, c_i) \leq b_i$. In the node-stop variant, we additionally require $c_i \in V$.

Goals: (i) MinBall: Minimize $Radius(\mathbf{C}, \mathbf{c})$ of a smallest ball containing $\mathbf{C}$ around an optimally chosen center $\mathbf{c}$, *where* $Radius(\mathbf{C}, \mathbf{c}) = \max_i d(\mathbf{c}, c_i)$. We consider both the scenario with node centers only, and the scenario with arbitrary point centers.
(ii) MinDiam: Minimize $Diam(\mathbf{C})$, where $Diam(\mathbf{C}) = \max_{i,j} d(c_i, c_j)$.
(iii) MinSum: Minimize $Sum(\mathbf{C})$, where $Sum(\mathbf{C}) = \sum_i \sum_j d(c_i, c_j)$.

Related Work. The gathering problem has been studied in two very different scenarios (i) Gathering of mobile agents in a connected (finite or infinite) graph, and (ii) Gathering of mobile robots in a (bounded or unbounded) plane or other geometric spaces. In the context of distributed robotics or swarm robotics [23], the problem of gathering many robots at a single point has been studied as an agreement problem, where the main issue is feasibility of gathering starting from arbitrary configurations [12] or gathering without full knowledge of the configuration [24,26]. The problem of *convergence* requires the robots to converge towards a point [13], without actually arriving at the gathering point. When the robots are not allowed to collide, the problem of moving the robots closer avoiding any collisions has been studied by Pagli et al. [27]. In all these studies, the robots can move freely in any direction. For mobile agents on the graph that are restricted to move along the edges, gathering has been studied under different models (see e.g. [15,28]). In particular, the gathering of two agents, often called rendezvous, has attracted a lot of attention, well documented in [1]. The problem of gathering with the objective of minimizing movements has been studied in [11]. However to the best of our knowledge, there have no previous studies on gathering with fixed constraints (budgets) on energy required for movements.

The model of energy-constrained agents was introduced in [7,3] for single agent exploration of graphs. Duncan et al. [20] consider a similar model where the agent is tied with a rope of length b to the starting location. Multi-agent exploration under uniform energy constraint of b, has been studied for trees [25,21] with the objective of minimizing the energy budget per agent [22] or the number k of agents [16] required for exploration, while time optimal exploration was studied by Dereniowski et al. [19] under the same model. Demaine et al. [17,18] studied problems of optimizing the total or maximum energy consumption of the agents when the agents need to place themselves in desired configurations (e.g. connected or independent configurations); they provided approximation algorithms and inapproximability results. Similar problems have been studied for agents moving in the visibility graphs of simple polygons [8].

For the model studied in this paper, where each agent has a distinct energy budget, the problem of *Broadcast* and *Convergecast* was studied in [2] who provided hardness results for trees and approximation algorithms for arbitrary graphs. The problem of delivering packages by multiple agents having energy constraints was studied in [9,10,5,6]. All of these problems were shown to be NP-hard for general graphs even if the agents are allowed to exchange energy when they meet [14,4].

Our Contribution and Paper Organization. In Section 2, we establish a few preliminaries and prove that MinBall is solvable in polynomial-time. In Section 3 we give a 2-approximation algorithm for MinDiam, together with a matching NP-hardness lower bound; additionally we show that MinDiam is polynomial-time solvable on tree graphs. In Section 4, we prove that MinSum admits a $2(1-\frac{1}{k})$-approximation algorithm but no FPTAS, unless P = NP. We show that the analysis of the approximation ratio of the provided algorithm is tight.

We conclude with remarks on future research opportunities, including preliminary approximation hardness results for additive approximation of MinDiam, in Section 5. All our results – with the exception of additive approximation – hold for both node-stop as well as arbitrary-stop scenarios. Omitted proofs are deferred to the full version of the paper.

2 Preliminaries and Minimizing the Radius

Preliminaries. We first point out some differences in the two scenarios we consider throughout this paper and our general approach on how to tackle and distinguish those. In the node stop scenario, where each agent i is only allowed to move to nodes v with distance $d(p_i, v) \leq b_i$, there is a finite number of feasible configurations $\mathbf{C}$. For the scenario with arbitrary final positions, where agents are also allowed to move to points p inside edges (as long as $d(p_i, p) \leq b_i$), we discretize the set of configurations. In the MinBall variant of Near-Gathering, the discretization turns out to contain at least one optimum solution, for MinDiam and MinSum it will at least contain a configuration approximating an optimum solution within a factor of 2 or $2(1 - \frac{1}{k})$, respectively. To this end, we define sets of reachable nodes and "maximally reachable" in-edge points as follows:

Definition 2 (Balls, Spheres). *For an instance $I = \langle G, k, (p_i), (b_i) \rangle$ with $i \in 1, \ldots, k$, we define*

- $B(i) := \{v \in V \mid d(p_i, v) \leq b_i\}$, *i.e. the ball containing all nodes that agent i can reach from its initial position p_i, and*
- $S(i) := \emptyset$ *for node stops, and* $S(i) := \{p \in G \mid d(p_i, p) = b_i\} \setminus B(i)$ *for arbitrary stops, i.e. the sphere of all in-edge points that agent i can reach from its initial position p_i only by spending its whole budget b_i.*

In the same spirit, we can study MinBall-Gathering for centers $\mathbf{c}$ being restricted to nodes in V, or for the continuous set of center points being allowed to be placed both on nodes as well as the inside of edges of G. To discretize this set, it will be useful to define a set of midpoints, intuitively consisting of "points m lying in the middle of a trail between points p and q":

Definition 3 (Midpoints). *Given a set S of points in G, denote by $G' = (V', E', \omega')$ the graph we get from $G = (V, E, \omega)$ by subdividing the edges in E with points from S, i.e. $V' = V \cup S$. We define the midpoint*

Algorithm 1 MinBall (node centers)

Input: An instance $\langle G, k, (p_i)_{i\in 1,\dots,k}, (b_i)_{i\in 1,\dots,k}\rangle$.
Output: Configuration $\mathbf{C}$ and center $\mathbf{c} \in V$ with minimum radius $\mathit{Radius}(\mathbf{C}, \mathbf{c})$.

```
for each v ∈ V do
    Compute C^v := (c_1^v, ..., c_k^v), where c_i^v ∈ arg min{d(v, c_i) | c_i ∈ B(i) ∪ S(i)}
        is a point in B(i)∪S(i) minimizing the distance to v, breaking ties arbitrarily.
    Compute Radius(C^v, v).
end for
Return argmin_{C^v, v: v∈V} Radius(C^v, v).
```

set $M(S)$ of points in G' – and by bijection also of G – as:

$$\begin{aligned} M(S) := &\left\{m \in V' \mid \exists\, p, q \in S\colon d(p,m) = d(m,q)\right\} \\ &\cup \left\{m \in \mathit{seg}(e) \mid e = \{u,v\} \in E',\ \exists\, p, q \in S\colon \right. \\ &\qquad\qquad \left. d(p,u) + d(u,m) = d(m,v) + d(v,q)\right\}. \end{aligned}$$

Lemma 1. *The sets $B(i)$, $S(i)$ and $M(S)$ can be computed in time polynomial in $|V|, k$ and $|V|, |S|$, respectively.*

MinBall for node centers. Having defined balls and spheres of reachable points for the agents, we can immediately give an exhaustive search algorithm for MinBall for *centers restricted to nodes*. The main idea of Algorithm 1 is to fix a node in graph G as a *gathering point* and then for each agent i compute the minimum distance to this fixed center it can reach, given its starting position p_i and its energy budget b_i. Iterating over all possible center nodes, we find an optimal solution:

Theorem 1 (MinBall, node centers). *Algorithm 1 is a polynomial-time algorithm for* MinBall *with node centers.*

The polynomial running time follows immediately from the fact that $B(i), S(i)$ can be computed in polynomial time and have polynomial size by Lemma 1. As the algorithm iterates over all possible center nodes, we can establish correctness by characterizing optimum stopping positions:

Lemma 2. *There exists an optimum solution $(\mathbf{C}_{\text{OPT}}, \mathbf{c}_{\text{OPT}})$ for* MinBall *where every agent i either stops on $\mathbf{c}_{\text{OPT}}$ or on a point in $B(i) \cup S(i)$, independent of whether $\mathbf{c}_{\text{OPT}}$ is contained in $\bigcup_i (B(i) \cup S(i))$ or not.*

MinBall for arbitrary centers. As can be seen from Lemma 2, when testing for a fixed center $\mathbf{c}$, in addition to checking the points in $B(i) \cup S(i)$

Algorithm 2 MinBall (arbitrary centers), MinDiam (2–apx / on Trees)

Input: An instance $\langle G, k, (p_i)_{i \in 1,\ldots,k}, (b_i)_{i \in 1,\ldots,k} \rangle$.
Output: Configuration $\mathbf{C}$ and center $\mathbf{c} \in G$ with minimum radius $\mathit{Radius}(\mathbf{C}, \mathbf{c})$.

for each $p \in M\big(V \cup \bigcup_i S(i)\big)$ **do**
  Compute $\mathbf{C}^p := (c_1^p, \ldots, c_k^p)$, where either $c_i^p = p$ if $d(p_i, p) \le b_i$, or
    $c_i^p \in \arg\min\{d(p, c_i) \mid c_i \in B(i) \cup S(i)\}$ (breaking ties arbitrarily) otherwise.
  Compute $\mathit{Radius}(\mathbf{C}^p, p)$.
end for
Return $\underset{\mathbf{C}^p,\, p:\; p \in M\left(V \cup \bigcup_i S(i)\right)}{\arg\min} \mathit{Radius}(\mathbf{C}^p, p)$.

we should also consider whether agent i can reach $\mathbf{c}$ itself. As candidates for the center $\mathbf{c}$ we take all points in the midpoint set $M(V \cup \bigcup_i S(i))$:

Theorem 2 (MinBall, arbitrary centers). *Algorithm 2 is a poly-time algorithm for* MinBall *with arbitrary centers.*

As before, polynomial running time follows from the polynomial size of the candidate set $M(V \cup \bigcup_i S(i))$. Building upon Algorithm 1 and Theorem 1, it remains to show that this set contains an optimum center:

Lemma 3. *There exists an optimum solution* $(\mathbf{C}_{\mathrm{OPT}}, \mathbf{c}_{\mathrm{OPT}})$ *for* MinBall *where* $\mathbf{c}_{\mathrm{OPT}}$ *is contained in* $M(V \cup \bigcup_i S(i))$.

3 Minimizing the Diameter

In this Section, we prove that Algorithm 2, which computes an optimum solution for MinBall, also computes a 2-approximation for MinDiam. As we will show, this is likely best-possible, as there is no polynomial-time $(2 - o(1))$-approximation for MinDiam, unless P = NP. Nonetheless, for the special case of tree graphs, Algorithm 2 even computes an optimum solution for MinDiam. We start with the positive results:

Theorem 3 (MinDiam, 2-apx). *Algorithm 2 is a polynomial-time* 2-*approximation algorithm for* MinDiam.

Proof. Let configuration $\mathbf{C}^* = (c_1^*, \ldots, c_k^*)$ with center $\mathbf{c}^*$ be the MinBall solution computed by Algorithm 2. We denote the radius of $(\mathbf{C}^*, \mathbf{c}^*)$ by $r^* = \mathit{Radius}(\mathbf{C}^*, \mathbf{c}^*) = \max_j d(\mathbf{c}^*, c_j^*)$ and the diameter of $\mathbf{C}^*$ by $d^* := \mathit{Diam}(\mathbf{C}^*) = \max_{i,j} d(c_i^*, c_j^*)$. Using the triangle inequality, we have for all configuration points c_i^*, c_j^* that $d(c_i^*, c_j^*) \le d(c_i^*, \mathbf{c}^*) + d(c_j^*, \mathbf{c}^*)$ and thus $d^* \le 2 \cdot r^*$. Now let $\mathbf{C}_{\mathrm{OPT}} = (o_1, \ldots, o_k)$ be an optimum configuration

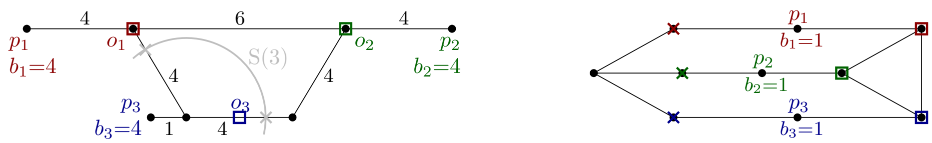

Fig. 1. (left) MinDiam-instance where agent 3's final position in the (unique) optimum solution $\mathbf{C}_{\text{Opt}} = (o_1, o_2, o_3)$ is not in $B(3) \cup S(3)$. **(right)** Replacing $\mathit{Radius}(\mathbf{C}^p, p)$ in Lines 4&6 of Algorithm 2 with $\mathit{Diam}(\mathbf{C}^p)$ (yielding configurations depicted by $\times$ vs $\square$) improves the quality of a MinDiam solution for certain instances by a factor of 2.

for MinDiam with diameter $d_{\text{Opt}} := \mathit{Diam}(\mathbf{C}_{\text{Opt}}) = \max_{i,j} d(o_i, o_j)$. We choose an arbitrary point $o \in \mathbf{C}_{\text{Opt}}$ and compute the radius of a smallest ball around o containing $\mathbf{C}_{\text{Opt}}$, $r_o = \mathit{Radius}(\mathbf{C}_{\text{Opt}}, o) = \max_j d(o, o_j) \leq d_{\text{Opt}}$. By Theorem 2, we have $r^* \leq r_o$ (even though o might not have been considered as a center candidate, see e.g. Figure 1 (left)). Combining all inequalities, we get $d^* \leq 2 \cdot r^* \leq 2 \cdot r_o \leq 2 \cdot d_{\text{Opt}}$, hence $\mathbf{C}^*$ is a 2-approximation for MinDiam. $\square$

Theorem 4 (MinDiam, on Trees). *Algorithm 2 is a polynomial-time algorithm for* MinDiam *on trees.*

Proof. First note that if there is a configuration $\mathbf{C}_{\text{Opt}}$ with maximum distance $\mathit{Diam}(\mathbf{C}_{\text{Opt}}) = 0$, it also has radius $\mathit{Radius}(\mathbf{C}_{\text{Opt}}, \mathbf{c}) = 0$ for some center $\mathbf{c}$, and thus will be found by Algorithm 2 as proven in Theorem 2. Otherwise the diameter $\mathit{Diam}(\mathbf{C}_{\text{Opt}})$ of an optimum solution $\mathbf{C}_{\text{Opt}}$ is lower bounded by the largest diameter among all optimal solutions of the instance reduced to pairs of agents i, j:

$$d^* := \begin{cases} \max\limits_{i,j} \min\limits_{q_i \in B(i),\ q_j \in B(j)} d(q_i, q_j) & \text{for the node stop scenario,} \\ \max\limits_{i,j} \quad d(p_i, p_j) - b_i - b_j & \text{for arbitrary final positions.} \end{cases}$$

We show that, indeed, Algorithm 2 computes a configuration $\mathbf{C}^*$ with $\mathit{Diam}(\mathbf{C}^*) = d^*$. To this end, denote by a, b two agents giving rise to d^*, and let $q_a \in B(a) \cup S(a)$, $q_b \in B(b) \cup S(b)$ be two points with $d(q_a, q_b) = d^*$. Since we consider tree graphs here, there is a unique shortest path from q_a to q_b and thus a unique midpoint $\mathbf{c}^* \in G$ with $d(\mathbf{c}^*, q_a) = d(\mathbf{c}^*, q_b) := \frac{d^*}{2}$. As $\mathbf{c}^*$ is contained in $M(V \cup \bigcup_i S(i))$, Algorithm 2 will use $\mathbf{c}^*$ as a center point candidate for which it computes a configuration $\mathbf{C}^* = (c_1^*, \ldots, c_k^*)$. By definition, we have $d(\mathbf{c}^*, c_a^*) = d(\mathbf{c}^*, q_a) = \frac{d^*}{2} = d(\mathbf{c}^*, q_b) = d(\mathbf{c}^*, c_b^*)$.

It is enough to show that for all other agents i we have $d(\mathbf{c}^*, c_i^*) \leq \frac{d^*}{2}$, too. Assume for the sake of contradiction that this is not the case and that there is an agent i with $d(\mathbf{c}^*, c_i^*) > \frac{d^*}{2}$. Consider the shortest c_i^*-$\mathbf{c}^*$-path P_i, the shortest c_a^*-$\mathbf{c}^*$-path P_a and the shortest c_b^*-$\mathbf{c}^*$-path P_b. By definition of d^* and c^*, the paths P_a and P_b must be interiorly disjoint, $P_a \cap P_b = \{\mathbf{c}^*\}$. Since P_i is a path on a tree ending in the same node $\mathbf{c}^*$, it must be interiorly disjoint with at least one of the two paths P_a, P_b, without loss of generality with P_a. Because any two points in a tree are connected by a unique path, we have $d(c_i^*, c_a^*) = d(c_i^*, \mathbf{c}^*) + d(\mathbf{c}^*, c_a^*) > d^*$ and thus also $\min_{q_i \in B(i) \cup S(i),\, q_a \in B(a) \cup S(a)} d(q_i, q_a) > d^*$, contradicting the maximality of d^*. Hence we have $Diam(C^*) \leq \max_{i,j} d(c_i^*, \mathbf{c}^*) + d(\mathbf{c}^*, c_j^*) = d^*$. □

Replacing the computation of $Radius(\mathbf{C}^p, p)$ in Lines 4 and 6 of Algorithm 2 by a computation of $Diam(\mathbf{C}^p)$ can improve the quality of a MinDiam solution by a factor of up to 2 for some instances, see for example Figure 1 (right). However, this does not translate to the worst-case approximation guarantee, as one can see in the instance constructed in the following matching approximation hardness result.

Theorem 5. *There exists no deterministic polynomial-time* $\big(2 - o(1)\big)$*-approximation algorithm for* MinDiam*, unless* $P = NP$*. This holds even in unweighted graphs with uniform budgets* $b_i = 1$*,* $i = 1, \dots, k$*.*

Proof (Sketch). We prove Theorem 5 by a reduction from 3Sat to MinDiam: Let ϕ be an arbitrary boolean formula in conjunctive normal form, where each clause contains 3 different literals, and let $x_1, \dots, x_n$ be the n many variables and $C_1, \dots, C_m$ be the m many clauses of ϕ. We show that any polynomial-time $(2 - o(1))$-approximation algorithm for MinDiam can be used to decide whether ϕ is satisfiable. From ϕ, we construct an instance $I = \langle G, k, (p_i)_{i \in 1,\dots,k}, (b)_{i \in 1,\dots,k} \rangle$ with k agents of uniform budget $b = 1$ and a graph $G = (V, E, \omega)$ with uniform edge weights $\omega = 1$ in the following manner.

Set of nodes V: Using $\mathrm{T} = \mathit{true}$ and $\mathrm{F} = \mathit{false}$, we first define the set of all possible truth assignments of a clause C containing 3 literals, $L := \{\mathrm{TTT}, \mathrm{TTF}, \mathrm{TFT}, \mathrm{TFF}, \mathrm{FTT}, \mathrm{FTF}, \mathrm{FFT}, \mathrm{FFF}\}$. Note that every clause C is satisfiable by exactly 7 of the 8 possible truth assignments in L (e.g. $x_1 \vee x_2 \vee \overline{x}_n$ is satisfied by $x_1, x_2, x_n \in L \setminus \{FFT\}$). Now, let $V := V_x \cup V_\ell \cup V_C \cup V_L$, where

- $V_x = \{v_i \mid 1 \leq i \leq n\}$ are nodes corresponding to *variables* $x_1, \dots, x_n$,
- $V_\ell = \{v_i^{\mathrm{T}} \mid 1 \leq i \leq n\} \cup \{v_i^{\mathrm{F}} \mid 1 \leq i \leq n\}$ are nodes corresponding to *literals*, i.e. true-value and false-value assignments of the variables x_i,
- $V_C = \{c_j \mid 1 \leq j \leq m\}$ are nodes corresponding to *clauses* $C_1, \dots, C_m$,

– $V_L = \{c_j^l \mid 1 \le j \le m,\ \forall l \in L\}$ are nodes corresponding to all possible truth assignments of each clause C_i.

Agents & reduction idea: On each of the nodes in $V_x \cup V_C$ we place one agent with a budget of $b = 1$, for a total of $n + m$ agents. The main idea is to initially space the agents by a pairwise distance of 3. We then let agents on V_x "pick the value assignment of the variables x_i" by walking to their respective node in V_ℓ, whereas we let agents on V_C "pick the truth assignment of the clauses C_j" by walking to their respective node in V_L. Then a satisfiable assignment of ϕ exists, if and only if the variable agents and the clause agents "agree in their choice" which corresponds to an optimum MinDiam configuration $\mathbf{C}_{\text{Opt}}$ of diameter 1. Furthermore, any other configuration should have diameter ≥ 2. This gives rise to the

Set of edges $E := E_{x\ell} \cup E_{\ell L} \cup E_{CL} \cup E_{\ell\ell} \cup E_{LL}$, where:

– $E_{x\ell} = \{\{v_i, v_i^{\text{T}}\}, \{v_i, v_i^{\text{F}}\} \mid 1 \le i \le n\colon v_i \in V_x,\ v_i^{\text{T}}, v_i^{\text{F}} \in V_\ell\}$ are edges connecting each variable node x_i to its two literal nodes,
– $E_{CL} = \{\{c_j, c_j^l\} \mid 1 \le j \le m\colon c_j \in V_C,\ c_j^l \in V_L,\ c_j^l \text{ satisfies } C_j\}$ are edges connecting each clause node c_j with all nodes representing satisfying assignments for clause C_j,
– $E_{\ell L} = \{\{v_i, c_j^l\} \mid i \le n,\ j \le m\colon v_i \in \{v_i^{\text{T}}, v_i^{\text{F}}\} \subset V_x,\ c_j^l \in V_l$, such that
  - either x_i does not appear in C_j, or
  - x_i appears in C_j and v_i agrees with $c_j^l\}$

  are edges connecting unrelated literals and clause truth-assignments, as well as matching literals and clause truth-assignments.
– $E_{\ell\ell} = \{\{u, v\} \mid u, v \in V_\ell\}$ and $E_{LL} = \{\{u, v\} \mid u, v \in V_L\}$ are edges pairwise connecting nodes in V_ℓ, and nodes in V_L, respectively.

Figure 2 shows a part of an instance of MinDiam which is constructed from an instance of 3Sat as described above. Before continuing with our proof we need to argue that no agent would stop in the middle of an edge:

Lemma 4. *For any configuration $\mathbf{C}' = (c'_1, \dots, c'_k)$ with an agent i for which $c'_i \notin \{V_\ell \cup V_L\}$, there exists another configuration $\mathbf{C}'' = (c''_1, \dots, c''_k)$ with diameter $Diam(\mathbf{C}'') \le Diam(\mathbf{C}')$ for which $\forall i\colon c''_i \in \{V_\ell \cup V_L\}$.*

$\Rightarrow$ Continuing with our proof of Theorem 5, we first show that if ϕ is satisfiable then there exists a configuration $\mathbf{C}$ of diameter $Diam(\mathbf{C}) = 1$. Since ϕ is satisfiable we have a truth assignment to the variables which satisfies every clause of ϕ. For each variable x_i, we let agent $a(v_i)$ move to node v_i^{T} if $x_i = \mathit{true}$ and to node v_i^{F} otherwise. Next, for each clause C_j, we let agent $a(c_j)$ move to the node c_i^l, which corresponds to the correct

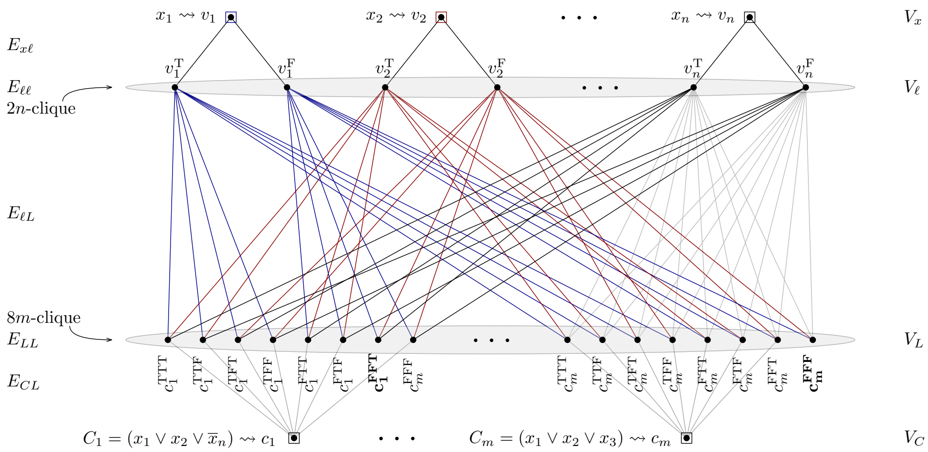

Fig. 2. A part of an instance of MinDiam, constructed from the 3-SAT instance $C_1 \wedge \cdots \wedge C_m$ with variables $x_1, \ldots, x_n$, displaying the connections between nodes v_1, v_2, v_n, c_1 and c_m. Notice that nodes c_1^{FFT} and c_m^{FFF} are not connected to nodes c_1 and c_m, respectively. The location of mobile agents is denoted by squares ($\square$).

true/false-assignment picked by the three agents of the variables in C_j. Note that both types of moves can be done with an energy of $b = 1$. By construction of the instance, the maximum distance of any two agents in this final configuration is 1.

$\Leftarrow$ We now show that if ϕ is not satisfiable then every solution to MinDiam is of size greater than or equal to 2.
By Lemma 4, we may assume that every agent starting on some node $v_i \in V_x$ moves to one of the nodes $v_i^{\mathrm{T}}, v_i^{\mathrm{F}}$, and every agent starting on some node $c_j \in V_C$ moves to one of the nodes c_j^l, $l \in L$ (otherwise, if an agent does not move, its distance is clearly at least 2 from any other agent). Therefore, by inspection of the final positions of agents starting in V_x, every MinDiam solution corresponds to a truth assignment. Since ϕ is not satisfiable, this truth assignment must leave at least one clause C_y, involving variables x_r, x_s, x_t, unsatisfied. By construction of the instance, and in particular in view of the fact that the edge $\{c_y, c_y^{l^\star}\}$ is missing (where $l^\star$ is the assignment to x_r, x_s, x_t falsifying C_y), the agent that started on c_y cannot move to $c_y^{l^\star}$, and thus it will have a distance of 2 in the final configuration from at least one of the agents starting on v_r, v_s, v_t.

Since a polynomial-time $(2 - o(1))$-approximation algorithm for MinDiam could distinguish between instances with an optimum solution with diameter 1 and instances with an optimum solution with diameter 2, it would also be able to decide whether ϕ is satisfiable of not. $\square$

Algorithm 3 MinSum $(2(1-\frac{1}{k})$–apx$)$

Input: An instance $\langle G, k, (p_i)_{i\in 1,\ldots,k}, (b_i)_{i\in 1,\ldots,k}\rangle$.
Output: Configuration $\mathbf{C}$ with $\mathit{Sum}(\mathbf{C}) \le 2(1-\frac{1}{k}) \cdot \min_{\text{feasible } \mathbf{C}'} \mathit{Sum}(\mathbf{C}')$.

for each $p \in V \cup \bigcup_i S(i)$ **do**
  Compute $\mathbf{C}^p := (c_1^p, \ldots, c_k^p)$, where either $c_i^p = p$ if $d(p_i, p) \le b_i$, or
    $c_i^p \in \arg\min\{d(p, c_i) \mid c_i \in B(i) \cup S(i)\}$ (breaking ties arbitrarily) otherwise.
  Compute $\mathit{Sum}(\mathbf{C}^p)$.
end for
Return $\underset{\mathbf{C}^p\,:\,p\in V\cup\bigcup_i S(i)}{\arg\min} \mathit{Sum}(\mathbf{C}^p)$.

4 Minimizing the Average Distance

In this Section we describe and analyze an algorithm for minimizing the average pairwise distance between agents. We complement its approximation ratio of $2(1-\frac{1}{k})$ with a tight analysis and rule out an FPTAS for MinSum. The main idea of the presented Algorithm 3 for MinSum is similar to the idea of Algorithm 2 for MinDiam. We fix a point p in the graph G as a gathering point and move each agent i as close as possible to p with respect to its energy constraint, breaking ties arbitrarily. Algorithm 3 exhaustively tests all points in $V \cup \bigcup_i S(i)$ as possible gathering points and selects the point p for with a configuration $\mathbf{C} = (c_1, \ldots, c_k)$ of minimum sum of pairwise distances between the agents, $\mathit{Sum}(\mathbf{C}) = \sum_i \sum_j d(c_i, c_j)$. The choice of the search space for gathering points is based on a characterization of optimum solutions, similar in look to Lemmata 2 and 3:

Lemma 5. *There exists an optimum solution* $\mathbf{C}_{\text{OPT}}$ *for* MinSum *where every agent stops on a point in* $V \cup \bigcup_i S(i)$.

Theorem 6 (MinSum, $2(1-\frac{1}{k})$-apx). *Algorithm 3 is a polynomial-time* $2(1-\frac{1}{k})$*-approximation algorithm (and the approximation ratio is tight).*

Proof (Upper bound only). Let $\mathbf{C}^* = (c_1^*, \ldots, c_k^*)$ denote the configuration computed by Algorithm 3. We denote with $s^* := \mathit{Sum}(\mathbf{C}^*)$ the sum of all pairwise agent distances in $\mathbf{C}^*$. Furthermore, let $\mathbf{C}_{OPT} = (o_1, \ldots, o_k)$ be an optimum MinSum solution in which each agent j stops on a point $o_j \in V \cup \bigcup_i S(i)$ and let $s_{\text{OPT}} = \mathit{Sum}(\mathbf{C}_{\text{OPT}}) = \sum_i \sum_j d(o_i, o_j)$. Choosing a point $o \in \arg\min_{o_i \in \mathbf{C}_{\text{OPT}}} \sum_j d(o_i, o_j)$ we get

$$\sum_j d(o, o_j) = \frac{1}{k} \cdot k \sum_j d(o, o_j) \le \frac{1}{k} \cdot \sum_i \sum_j d(o_i, o_j) = \frac{1}{k} \cdot s_{\text{OPT}}.$$

Consider now the configuration $\mathbf{C}^o = (c_1^o, \ldots, c_k^o)$ which Algorithm 3 computed for point o in Step 2 and let $s^o := \mathit{Sum}(\mathbf{C}^o) = \sum_i \sum_j d(c_i^o, c_j^o)$. Clearly, we have $s^* \leq s^o$. Furthermore, o is reachable by at least one agent a, thus by Step 2 we also have $c_a^o = o$. Finally, as Step 2 moves agents as close to o as possible, we have $d(o, c_j^o) \leq d(o, o_j)$. Using the triangle inequality, we rewrite s^o to get

$$\begin{aligned} s^* \leq s^o &= \sum_i \sum_j d(c_i^o, c_j^o) \leq 2 \sum_j d(c_a^o, c_j^o) + \sum_{i \neq a} \sum_{\substack{j \neq a \\ j \neq i}} d(c_i^o, o) + d(o, c_j^o) \\ &= 2 \sum_j d(o, c_j^o) + (k-2) \sum_{i \neq a} d(c_i^o, o) + (k-2) \sum_{j \neq a} d(o, c_j^o) \\ &= (2k-2) \sum_j d(o, c_j^o) \leq (2k-2) \sum_j d(o, o_j) \leq 2(1 - \tfrac{1}{k}) s_{\mathrm{OPT}}. \quad \square \end{aligned}$$

Theorem 7. *There is no FPTAS for* MinSum, *unless P = NP.*

Proof. Assume for the sake of contradiction that there is a polynomial-time approximation scheme for MinSum which for all $\varepsilon > 0$ computes a $(1+\varepsilon)$-approximation in time $\mathit{poly}(k, \frac{1}{\varepsilon})$. We reuse the reduction to 3Sat already given in Theorem 5. Recall from its proof that (i) the underlying 3Sat-formula ϕ is satisfiable if and only if there is a Near-Gathering solution $\mathbf{C}^*$ in which all agents have pairwise distance 1, and that (ii) any other solution $\mathbf{C}$ has at least one pair of agents with distance 2.

Summing up the pairwise distances we get for (i) that $\mathit{Sum}(\mathbf{C}^*) = k(k-1)$, while for (ii) we have $\mathit{Sum}(\mathbf{C}) \geq k(k-1)+1$. The existence of an FPTAS, using $\varepsilon \leq \frac{1}{k^2}$, means that we can approximate $\mathit{Sum}(\mathbf{C}^*)$ to within $(1 + \frac{1}{k^2}) \cdot k(k-1) = k^2 - k + 1 - \frac{1}{k} < k(k-1) + 1 \leq \mathit{Sum}(\mathbf{C})$. Hence we could distinguish the existence of a solution $\mathbf{C}^*$ from any other solution and thus decide satisfiability of ϕ in time $\mathit{poly}(k, \frac{1}{1/k^2}) = \mathit{poly}(k)$, in contradiction to the assumption P $\neq$ NP. $\square$

5 Additive Approximation and Conclusion

In this paper, we explored the task of *Near-Gathering* a group of energy-constrained agents, whose movements are restricted by their energy budget. We showed how to compute, in polynomial time, an optimum solution for MinBall (minimizing the radius of a smallest ball containing all agents), a 2-approximation for MinDiam (minimizing the maximum distance between any two agents), and a $2(1-\frac{1}{k})$-approximation for MinSum (minimizing the average distance between any two agents). For MinDiam, we provided a matching hardness result, while for MinSum, we ruled out the existence of an FPTAS, unless P = NP. Hence for future work, a major open problem is to improve upon the (in)approximability of MinSum.

A second possible research direction for Near-Gathering is an analysis of additive approximation. For this, we briefly review how we can reuse our hardness construction of multiplicative approximation of MinDiam:

Theorem 8. *Unless $P = NP$, there is no deterministic polynomial-time additive $+(2\max_i b_i - o(1))$-approximation algorithm for* MinDiam *with node stops, and no deterministic polynomial-time additive $+(\frac{4}{3}\max_i b_i - o(1))$-approximation algorithm for* MinDiam *with arbitrary stops.*

This is surprising for two reasons. On the one hand, *not moving the agents at all* is already an additive $+(2\max_i b_i)$-approximation. On the other hand, this is the only result in this paper, in which the *two scenarios* of (I) node stops and (II) arbitrary stops *differ*. The difference in the hardness result boils down to the loss of Lemma 4 in the adaption of the proof of Theorem 5, which we can only fully salvage for the case of node stops. Does this mean that there is a polynomial-time $+(2\max_i b_i - o(1))$-approximation for the scenario with arbitrary final positions? This remains completely open.

Finally, we aim to study the reverse problem of *Spreading* energy-constrained mobile agents, with the respective goals of (i) maximizing the radius of a smallest ball containing all agents, (ii) maximizing the minimum distance between any two agents, and (iii) maximizing the average distance between any two agents.

References

1. Alpern, S., Gal, S.: The Theory of Search Games and Rendezvous. Kluwer (2003)
2. 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)
3. Awerbuch, B., Betke, M., Rivest, R.L., Singh, M.: Piecemeal graph exploration by a mobile robot. Information and Computation **152**(2), 155–172 (1999)
4. Bampas, E., Das, S., Dereniowski, D., Karousatou, C.: Collaborative delivery by energy-sharing low-power mobile robots. In: ALGOSENSORS'17. pp. 1–12 (2017)
5. Bärtschi, A., Chalopin, J., Das, S., Disser, Y., Geissmann, B., Graf, D., Labourel, A., Mihalák, M.: Collaborative Delivery with Energy-Constrained Mobile Robots. In: SIROCCO'16. pp. 258–274 (2016)
6. Bärtschi, A., Chalopin, J., Das, S., Disser, Y., Geissmann, B., Graf, D., Labourel, A., Mihalák, M.: Collaborative delivery with energy-constrained mobile robots. Theoretical Computer Science (2017), to appear. See also SIROCCO'16

7. Betke, M., Rivest, R.L., Singh, M.: Piecemeal learning of an unknown environment. Machine Learning **18**(2), 231–254 (1995)
8. Bilò, D., Disser, Y., Gualà, L., Mihalák, M., Proietti, G., Widmayer, P.: Polygon-Constrained Motion Planning Problems. In: ALGOSENSORS'13. pp. 67–82 (2013)
9. Chalopin, J., Das, S., Mihalák, M., Penna, P., Widmayer, P.: Data Delivery by Energy-Constrained Mobile Agents. In: ALGOSENSORS'13. pp. 111–122 (2013)
10. Chalopin, J., Jacob, R., Mihalák, M., Widmayer, P.: Data Delivery by Energy-Constrained Mobile Agents on a Line. In: ICALP'14. pp. 423–434 (2014)
11. Cicerone, S., Stefano, G.D., Navarra, A.: Gathering of robots on meeting-points: feasibility and optimal resolution algorithms. Distributed Computing **31**(1), 1–50 (2018)
12. Cieliebak, M., Flocchini, P., Prencipe, G., Santoro, N.: Distributed computing by mobile robots: Gathering. SIAM Journal on Computing **41**(4), 829–879 (2012)
13. Cohen, R., Peleg, D.: Convergence properties of the gravitational algorithm in asynchronous robot systems. SIAM Journal on Computing **34**, 1516–1528 (2005)
14. Czyzowicz, J., Diks, K., Moussi, J., Rytter, W.: Communication Problems for Mobile Agents Exchanging Energy. In: SIROCCO'16 (2016)
15. Czyzowicz, J., Labourel, A., Pelc, A.: How to meet asynchronously (almost) everywhere. ACM Transactions on Algorithms **8**(4), 37:1–37:14 (2012)
16. Das, S., Dereniowski, D., Karousatou, C.: Collaborative Exploration by Energy-Constrained Mobile Robots. In: SIROCCO'15. pp. 357–369 (2015)
17. Demaine, E.D., Hajiaghayi, M., Mahini, H., Sayedi-Roshkhar, A.S., Oveisgharan, S., Zadimoghaddam, M.: Minimizing movement. ACM Transactions on Algorithms **5**(3), 30:1–30:30 (2009)
18. Demaine, E.D., Hajiaghayi, M., Marx, D.: Minimizing movement: Fixed-parameter tractability. ACM Transactions on Algorithms **11**(2), 14:1–14:29 (2014)
19. Dereniowski, D., Disser, Y., Kosowski, A., Pająk, D., Uznański, P.: Fast collaborative graph exploration. Information and Computation **243**, 37–49 (2015)
20. Duncan, C.A., Kobourov, S.G., Kumar, V.S.A.: Optimal Constrained Graph Exploration. ACM Transactions on Algorithms **2**(3), 380–402 (2006)
21. Dynia, M., Korzeniowski, M., Schindelhauer, C.: Power-Aware Collective Tree Exploration. In: ARCS'06. pp. 341–351 (2006)
22. Dynia, M., Łopuszański, J., Schindelhauer, C.: Why robots need maps. In: SIROCCO'07. pp. 41–50 (2007)
23. Flocchini, P., Prencipe, G., Santoro, N.: Distributed Computing by Mobile Entities, Current Research in Moving and Computing, Lecture Notes in Computer Science, vol. 11340. Springer (2019)
24. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Gathering of asynchronous robots with limited visibility. Theoretical Computer Science **337**(1-3), 147–168 (2005)
25. Fraigniaud, P., Gasieniec, L., Kowalski, D.R., Pelc, A.: Collective tree exploration. Networks **48**(3), 166–177 (2006)
26. Lin, J., Morse, A.S., Anderson, B.D.O.: The multi-agent rendezvous problem. parts 1 and 2. SIAM Journal on Control and Optimization **46**(6), 2096–2147 (2007)
27. Pagli, L., Prencipe, G., Viglietta, G.: Getting close without touching: Near-gathering for autonomous mobile robots. Distributed Computing **28**(5), 333–349 (2015)
28. Pelc, A.: Deterministic rendezvous in networks: A comprehensive survey. Networks **59**(3), 331–347 (2012)