

Near-gathering of energy-constrained mobile agents

Andreas Bärtschi, Evangelos Bampas, Jérémie Chalopin, Shantanu Das, Christina Karousatou, Matúš Mihalák

▶ To cite this version:

Andreas Bärtschi, Evangelos Bampas, Jérémie Chalopin, Shantanu Das, Christina Karousatou, et al.. Near-gathering of energy-constrained mobile agents. Theoretical Computer Science, 2021, 849, pp.35-46. 10.1016/j.tcs.2020.10.008 . hal-03065725

HAL Id: hal-03065725 https://hal.science/hal-03065725

Submitted on 14 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Near-gathering of energy-constrained mobile agents^{$\stackrel{\bigstar}{\Rightarrow}$}

Andreas Bärtschi^{a,*}, Evangelos Bampas^b, Jérémie Chalopin^c, Shantanu Das^c, Christina Karousatou^d, Matúš Mihalák^e

^aCenter for Nonlinear Studies, Los Alamos National Laboratory, USA ^bParis-Saclay University and CNRS, LRI, France ^cCNRS, Aix-Marseille Université and Université de Toulon, LIS, France ^dDepartment of Mathematics, TU Darmstadt, Germany ^eDepartment of Data Science and Knowledge Engineering, Maastricht University, The Netherlands

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 2approximation 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

^{*}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.

^{*}Corresponding author.

Email address: baertschi@lanl.gov (Andreas Bärtschi)

1 1. Introduction

The problem of *gathering* is one of the fundamental problems in dis-2 tributed computing with mobile entities, which includes mobile agents mov-3 ing in a graph or robots moving in a continuous geometric space. In both 4 cases, the objective is to bring together multiple autonomous agents to a 5 single point (not predetermined). Gathering helps in coordination between 6 the mobile agents, sharing of information between the entities, reassignment 7 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 a large 9 area). Moreover, there are also theoretical reasons for studying gathering, 10 as the problem of selecting a gathering point is akin to problems of leader 11 election and consensus in distributed systems. However, in some cases, it 12 may be impossible to solve the problem of gathering, e.g. due to limitations 13 in the capabilities of the agents, or due to symmetries in their perception 14 of the environment. In some cases it may be desirable for the agents to get 15 close to each other without actually meeting [1]. 16

In this paper, we consider mobile agents moving on a graph, with severe 17 limitations on their movements. We assume that the agents have limited 18 energy resources and traversing any edge of the graph consumes some of 19 this energy which can not be replenished. In other words, each agent has 20 an initial energy budget which limits the total distance it can move in the 21 graph. Under such constraints, it is not always possible to gather the agents 22 at a single point. Thus, we consider the problem of moving the agents as 23 close as possible to each other while respecting the movement constraints, 24 defined below as the *near-gathering* problem. 25

Near-Gathering. A collection of k mobile agents is initially located at an 26 arbitrary set of nodes of an undirected edge-weighted graph $G = (V, E, \omega)$. 27 Each agent i, i = 1, ..., k, has an energy capacity b_i , which represents the 28 maximum distance the agent can move in the graph. The agents have global 29 knowledge of the graph and are controlled by a *central entity*. The goal is to 30 move the agents to a configuration where they are as close to each other as 31 possible under their respective limitations of movement. Closeness criteria 32 can be measured, e.g., as the size of the smallest region enclosing all the 33 agents, or as the maximum or average pairwise distance between the agents. 34 We look at each of these criteria and give a more precise definition of the 35 problem below. 36

Our Model. We consider an undirected graph $G = (V, E, \omega)$, where each 37 edge $e \in E$ has a positive weight $\omega(e) > 0$. As usual, the length of a path 38 is the sum of the weights of its edges. We think of every edge $e = \{u, v\}$ 39 as a segment of infinitely many points, where every point in the edge is 40 uniquely characterized by its distance from u, which is between 0 and w(e). 41 We consider every such point to subdivide the edge $\{u, v\}$ into two edges 42 of lengths proportional to the position of the point on the edge. Thus, the 43 distance d(p,q) between two points p and q (nodes or points inside edges) 44 is the length of a shortest path from p to q in G (with edges subdivided by 45 p, q, respectively). For a point p inside an edge $e \in E$ we write $p \in G$ and 46 $p \in seq(e).$ 47

A collection of k mobile agents is initially located at an arbitrary set 48 of nodes $p_1, \ldots, p_k \in V$. Each agent *i* is equipped with an energy budget 49 $b_i > 0$ and can move along edges of the graph, for a distance of at most 50 b_i . In the Near-Gathering problem, the goal is to relocate every agent into 51 a new position such that the resulting configuration minimizes one of the 52 following objectives: (i) the radius of a smallest ball containing all agents, (ii) 53 the maximum distance between any two agents, or (iii) the average distance 54 between the agents (or, equivalently, the sum of all distances). We are further 55 interested in two variants of the problem, where agents can: (I) only be 56 relocated to reachable nodes of the graph, or (II) in a more general scenario, 57 where the agents are allowed to be relocated to *reachable points* (i.e., nodes 58 or points inside edges). 59

⁶⁰ **Definition 1** (Near-Gathering).

Instance: $\langle G, k, (p_i)_{i=1,\dots,k}, (b_i)_{i=1,\dots,k} \rangle$, where $G = (V, E, \omega)$ is an undirected edge-weighted graph, k denotes the total number of agents, p_i denotes the initial position of agent i and b_i denotes the total amount of energy agent i 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 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.

72 (*ii*) MINDIAM: Minimize $Diam(\mathbf{C})$, where $Diam(\mathbf{C}) = \max_{i,j} d(c_i, c_j)$.

73 (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 74 scenarios (i) Gathering of mobile agents in a connected (finite or infinite) 75 graph, and (ii) Gathering of mobile robots in a (bounded or unbounded) 76 plane or other geometric spaces. In the context of distributed robotics or 77 swarm robotics [2], the problem of gathering many robots at a single point 78 has been studied as an agreement problem, where the main issue is feasibility 79 of gathering starting from arbitrary configurations [3] or gathering without 80 full knowledge of the configuration [4, 5]. The problem of *convergence* re-81 quires the robots to converge towards a point [6], without actually arriving at 82 the gathering point. When the robots are not allowed to collide, the problem 83 of moving the robots closer avoiding any collisions has been studied by Pagli 84 et al. [1]. In all these studies, the robots can move freely in any direction. 85 For mobile agents on the graph that are restricted to move along the edges, 86 gathering has been studied under different models (see e.g. [7, 8]). In par-87 ticular, the gathering of two agents, often called rendezvous, has attracted 88 a lot of attention, well documented in [9]. The problem of gathering with 89 the objective of minimizing movements has been studied in [10]. However to 90 the best of our knowledge, there have been no previous studies on gathering 91 with fixed constraints (budgets) on energy required for movements. 92

The model of energy-constrained agents was introduced in [11, 12] for 93 single agent exploration of graphs. Duncan et al. [13] consider a similar 94 model where the agent is tied with a rope of length b to the starting lo-95 cation. Multi-agent exploration under uniform energy constraint of b has 96 been studied for trees [14, 15] with the objective of minimizing the energy 97 budget per agent [16] or the number k of agents [17] required for exploration, 98 while time optimal exploration was studied by Dereniowski et al. [18] under 99 the same model. Demaine et al. [19, 20] studied problems of optimizing the 100 total or maximum energy consumption of the agents when the agents need 101 to place themselves in desired configurations (e.g. connected or independent 102 configurations); they provided approximation algorithms and inapproxima-103 bility results. Similar problems have been studied for agents moving in the 104 visibility graphs of simple polygons [21]. 105

For the model studied in this paper, where each agent has a distinct energy budget, the problem of *Broadcast* and *Convergecast* was studied in [22] 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 [23, 24, 25, 26]. 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 |27, 28|.

Our Contribution and Paper Organization. In Section 2, we establish a few 113 preliminaries and prove that MINBALL is solvable in polynomial-time. In 114 Section 3 we give a 2-approximation algorithm for MINDIAM, together with 115 a matching NP-hardness lower bound; additionally we show that MINDIAM 116 is polynomial-time solvable on tree graphs. In Section 4, we prove that 117 MINSUM admits a $2(1-\frac{1}{k})$ -approximation algorithm but no FPTAS, unless 118 P = NP. We show that the analysis of the approximation ratio of the 119 provided algorithm is tight. 120

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.

¹²⁵ 2. Preliminaries and Minimizing the Radius

Preliminaries. We first point out some differences in the two scenarios we 126 consider throughout this paper and our general approach on how to tackle 127 and distinguish those. In the node stop scenario, where each agent i is only 128 allowed to move to nodes v with distance $d(p_i, v) \leq b_i$, there is a finite 129 number of feasible configurations C. For the scenario with arbitrary final 130 positions, where agents are also allowed to move to points p inside edges 131 (as long as $d(p_i, p) \leq b_i$), we discretize the set of configurations. In the 132 MINBALL variant of Near-Gathering, the discretization turns out to contain 133 at least one optimum solution, for MINDIAM and MINSUM it will at least 134 contain a configuration approximating an optimum solution within a factor 135 of 2 or $2(1-\frac{1}{k})$, respectively. To this end, we define sets of reachable nodes 136 and "maximally reachable" in-edge points as follows: 137

Definition 2 (Balls, Spheres). For an instance $\langle G, k, (p_i)_{i=1,...,k}, (b_i)_{i=1,...,k} \rangle$ with initial agent positions p_i and energy budgets b_i , we define

• $B(i) := \{v \in V \mid d(p_i, v) \leq b_i\}, i.e. the ball containing all nodes that agent i can reach from its initial position <math>p_i$, 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 **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 set M(S) of points in G' – and by bijection also of G – as:

$$M(S) := \{ m \in V' \mid \exists p, q \in S : d(p, m) = d(m, q) \}$$
$$\cup \{ m \in seg(e) \mid e = \{ u, v \} \in E', \exists p, q \in S : d(p, u) + d(u, m) = d(m, v) + d(v, q) \}.$$

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

Proof. For each agent i, we find the ball B(i) of all reachable nodes by computing a single-source shortest paths tree from p_i in $\mathcal{O}(|V|^2)$. The sphere S(i) contains at most two points per edge $e = \{u, v\}$ which can be found in constant time given knowledge of the edge weight $\omega(e)$ and the already computed node distances $d(p_i, u), d(p_i, v)$. Overall the 2k many sets B(i), S(i)are of size $\mathcal{O}(|V|)$ and $\mathcal{O}(|V|^2)$, respectively, and can be computed in time $\mathcal{O}(k|V|^2)$.

In order to compute the set M(S) of midpoints of a given set S of points 163 in G, we first compute shortest-paths trees for all points $p \in S$ to all nodes 164 $v \in V'$ in time $\mathcal{O}(|S| \cdot |V'|^2) \subseteq \mathcal{O}(|S|^3 + |S| \cdot |V|^2)$. Then we check for 165 each node $v \in V'$ whether it is contained in M(S) by iterating over all 166 pairs of points $p, q \in S$. Similarly, we check for each edge $e = \{u, v\} \in E'$ 167 and all pairs of points $p, q \in S$ in constant time (having already computed 168 the distances d(p, u), d(p, v), d(q, u) and d(q, v) whether and where there are 169 any (at most 2) midpoints $m \in seq(e)$ of p and q. Overall, M(S) is of 170 size $\mathcal{O}(|S|^2 \cdot |V'|^2) \subseteq \mathcal{O}(|S|^4 + |S|^2 \cdot |V|^2)$ and can be computed in time 171 $\mathcal{O}(|S|^4 + |S|^2 \cdot |V|^2)$ as well. 172

¹⁷³ MinBall for node centers. Having defined balls and spheres of reachable ¹⁷⁴ points for the agents, we can immediately give an exhaustive search algo-¹⁷⁵ rithm for MINBALL for centers restricted to nodes. The main idea of Algo-¹⁷⁶ rithm 1 is to fix a node in graph G as a gathering point and then for each

Algorithm 1 MINBALL (node centers)

Input: An instance $\langle G, k, (p_i)_{i=1,...,k}, (b_i)_{i=1,...,k} \rangle$. Output: Configuration C, center $\mathbf{c} \in V$ with minimum radius $Radius(\mathbf{C}, \mathbf{c})$. 1: for each $v \in V$ do 2: Compute $\mathbf{C}^v := (c_1^v, \ldots, c_k^v)$, 3: where $c_i^v \in \arg\min\{d(v, c_i) \mid c_i \in B(i) \cup S(i)\}$ is a point in 4: $B(i) \cup S(i)$ minimizing the distance to v, breaking ties arbitrarily. 5: Compute $Radius(\mathbf{C}^v, v)$. 6: end for 7: Return $\underset{\mathbf{C}^v, v: v \in V}{\operatorname{ret}}$

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 of Algorithm 1 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{O}, \mathbf{o}) for MINBALL where every agent *i* either stops on the center \mathbf{o} or on a point in $B(i) \cup S(i)$, independent of whether \mathbf{o} is contained in $\bigcup_i (B(i) \cup S(i))$ or not.

Proof. Assume that there is no such optimum solution and denote by $C^* =$ 189 $(c_1^*, c_2^*, \ldots, c_k^*)$ and \mathbf{c}^* a solution with a minimum number of points $c_i^* \notin B(i) \cup$ 190 $S(i) \cup \{\mathbf{c}^*\}$ among all optimum solutions. We take any agent a with $c_a^* \notin$ 191 $B(a) \cup S(a) \cup \{\mathbf{c}^*\}$. By definition of B(a) and S(a), c_a^* must be a point inside 192 an edge for which $d(p_a, c_a^*) < b_a$. Without loss of generality we may assume 193 that a reached c_a^* by moving along a shortest path from p_a to \mathbf{c}^* . Hence 194 it still has energy left to move further along the shortest path towards c^* . 195 We move agent a until it reaches a point in $B(a) \cup \{\mathbf{c}^*\}$ or until its energy 196 is depleted, in which case it will have reached a point in S(a). The new 197 configuration has smaller or equal radius, and also a strictly smaller number 198 of points $c_i^* \notin B(i) \cup S(i) \cup \{\mathbf{c}^*\}$, contradicting the minimality of \mathbf{C}^* . Hence 199 there is always an optimum solution adhering to Lemma 2. 200

MinBall for arbitrary centers. We now extend our approach to find optimum MINBALL solutions 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) 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 $M(V \cup \bigcup_i S(i))$, yielding Algorithm 2:

Theorem 2 (MinBall, arbitrary centers). Algorithm 2 is a polynomial-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{O}, \mathbf{o}) for MINBALL where the center \mathbf{o} is contained in $M(V \cup \bigcup_i S(i))$.

Proof. Given any optimum configuration $\mathbf{C} = (c_1, \ldots, c_k)$ with center $\mathbf{c} \notin M(V \cup \bigcup_i S(i))$ and agent stopping positions c_i adhering to Lemma 2, we can directly construct an optimum solution (\mathbf{O}, \mathbf{o}) for which $\mathbf{o} \in M(V \cup \bigcup_i S(i))$. Let $G' = (V', E', \omega')$ be the graph we get from $G = (V, E, \omega)$ by subdividing the edges in E with points from $\bigcup_i S(i)$.

Let $e = \{u, v\} \in E'$ be the edge-subdivision containing $\mathbf{c}, \mathbf{c} \in seg(e)$, and denote by A_u, A_v the set of agents *i* with stopping positions $c_i = \mathbf{c}$ that entered *e* via *u* or *v*, respectively. Without loss of generality, each agent $i \in A_u \cup A_v$ has reached **c** along a shortest p_i -**c**-path and, since $c_i = \mathbf{c} \notin S(i)$, has a remaining energy of $b_i - d(p_i, c_i) > 0$.

We first assume that $A_u \cup A_v$ contains all k agents. In this case we move the center \mathbf{c} and all agent stopping positions \mathbf{C} to u, yielding a new center node $\mathbf{c}^* := u \in V \cup \bigcup_i S(i)$ and configuration $\mathbf{C}^* = (\mathbf{c}^*, \dots, \mathbf{c}^*)$ with radius $Radius(\mathbf{C}^*, \mathbf{c}^*) = 0$. Note that for each agent $i \in A_u$ we have $d(p_i, \mathbf{c}^*) < d(p_i, \mathbf{c})$ and for each agent $j \in A_v$ we have – since there is no point $p \in seg(\{\mathbf{c}, u\})$ with $p \in B(j) \cup S(j)$ – that $d(p_j, \mathbf{c}^*) \leq b_j$. Hence \mathbf{C}^* is a feasible configuration and $(\mathbf{C}^*, \mathbf{c}^*)$ an optimum solution.

Otherwise denote by $c_a \in B(a) \cup S(a)$ the agent stopping position with maximum distance $d(c_a, \mathbf{c})$ among all configuration points which have a shortest path to \mathbf{c} containing u. Analogously, denote by $c_b \in B(b) \cup S(b)$ the furthest agent stopping position among all configuration points which have a shortest path to \mathbf{c} containing v. Since $\mathbf{c} \notin M(V \cup \bigcup_i S(i))$, we know Algorithm 2 MINBALL (arbitrary centers), MINDIAM (2–apx / on Trees)

Input: An instance $\langle G, k, (p_i)_{i=1,\dots,k}, (b_i)_{i=1,\dots,k} \rangle$. **Output:** Configuration C, center $c \in G$ with minimum radius Radius(C, c). 1: for each $p \in M(V \cup \bigcup_i S(i))$ do Compute $\mathbf{C}^p := (c_1^p, \ldots, c_k^p),$ 2: where either $c_i^p = p$ if $d(p_i, p) \leq b_i$, or otherwise 3: $c_i^p \in \arg\min\{d(p, c_i) \mid c_i \in B(i) \cup S(i)\}$ (breaking ties arbitrarily). 4: Compute $Radius(\mathbf{C}^p, p)$. 5: 6: end for 7: Return arg min $Radius(\mathbf{C}^p, p).$ $\mathbf{C}^p, p: p \in M(V \cup \bigcup_i S(i))$

that $d(\mathbf{c}, c_a) \neq d(\mathbf{c}, c_b)$. Thus we can move \mathbf{c} together with all agent stopping positions $c_i = \mathbf{c}$ (of agents $i \in A_u \cup A_v$ that have stopped on \mathbf{c}) by a small distance of $\varepsilon > 0$ towards the further of the two positions c_a, c_b . This still gives a feasible solution $(\mathbf{C}^*, \mathbf{c}^*)$ that has strictly smaller radius $Radius(\mathbf{C}^*, \mathbf{c}^*) = \max\{d(\mathbf{c}, c_a), d(\mathbf{c}, c_b)\} - \varepsilon$, contradicting the optimality of (\mathbf{C}, \mathbf{c}) . The cases where only c_a or only c_b is defined can be treated analogously.

²⁴³ 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 $r^* =$ *Radius* $(\mathbf{C}^*, \mathbf{c}^*) = \max_j d(\mathbf{c}^*, c_j^*)$ and the diameter of \mathbf{C}^* by $d^* = Diam(\mathbf{C}^*) =$ $\max_{i,j} d(c_i^*, c_j^*)$. Using the triangle inequality, we have for all configuration points c_i^*, c_j^* that $d(c_i^*, c_j^*) \leq d(c_i^*, \mathbf{c}^*) + d(c_j^*, \mathbf{c}^*)$ and thus $d^* \leq 2 \cdot r^*$.



Figure 1: (left) MINDIAM-instance with (unique) optimum solution $\mathbf{O} = (o_1, o_2, o_3)$ of diameter $Diam(\mathbf{O}) = 8$, in which we have final positions $o_3 \notin B(3) \cup S(3)$ and $o_2 \notin M(V \cup \bigcup_i S(i))$.

(right) Replacing $Radius(\mathbf{C}^p, p)$ in Lines 5&7 of Algorithm 2 with $Diam(\mathbf{C}^p)$ (yielding configurations depicted by \times vs \Box , with diameters 2 and 1, respectively) improves the quality of a MINDIAM solution for certain instances by a factor of 2.

Now let $\mathbf{O} = (o_1, \ldots, o_k)$ be an optimum configuration for MINDIAM with diameter $d_{\text{OPT}} := Diam(\mathbf{O}) = \max_{i,j} d(o_i, o_j)$. We choose an arbitrary point $o \in \mathbf{O}$ and compute the radius of a smallest ball around o containing \mathbf{O} , $r_o := Radius(\mathbf{O}, o) = \max_j d(o, o_j) \leq d_{\text{OPT}}$. 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

$$d^* \le 2 \cdot r^* \le 2 \cdot r_o \le 2 \cdot d_{\mathrm{OPT}},$$

hence C^* is a 2-approximation for MINDIAM.

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 **O** with maximum distance $Diam(\mathbf{O}) = 0$, it also has radius $Radius(\mathbf{O}, \mathbf{o}) = 0$ for some center **o**, and thus will be found by Algorithm 2 as proven in Theorem 2. Otherwise the diameter $Diam(\mathbf{O})$ of an optimum solution **O** is lower bounded by the largest diameter among all optimal solutions of the instance reduced to pairs of agents i, j:

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

We show that, indeed, Algorithm 2 computes a configuration \mathbf{C}^* with diameter $Diam(\mathbf{C}^*) = d^*$. 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 $d(\mathbf{c}^*, c_i^*) \leq \frac{d^*}{2}$. 279 too. Assume for the sake of contradiction that this is not the case and that 280 there is an agent *i* with $d(\mathbf{c}^*, c_i^*) > \frac{d^*}{2}$. Consider the shortest c_i^* - \mathbf{c}^* -path 281 P_i , the shortest c_a^* - \mathbf{c}^* -path P_a and the shortest c_b^* - \mathbf{c}^* -path P_b . By definition 282 of d^* and c^* , the paths P_a and P_b must be interiorly disjoint, $P_a \cap P_b =$ 283 $\{\mathbf{c}^*\}$. Since P_i is a path on a tree ending in the same node \mathbf{c}^* , it must 284 be interiorly disjoint with at least one of the two paths P_a, P_b , without loss 285 of generality with P_a . Because any two points in a tree are connected by 286 a unique path, we have $d(c_i^*, c_a^*) = d(c_i^*, \mathbf{c}^*) + d(\mathbf{c}^*, c_a^*) > d^*$ and thus also 287 $\min_{q_i \in B(i) \cup S(i), q_a \in B(a) \cup S(a)} d(q_i, q_a) > d^*$, contradicting the maximality of d^* . 288 Hence we have $Diam(C^*) \leq \max_{i,j} d(c_i^*, \mathbf{c}^*) + d(\mathbf{c}^*, c_j^*) = d^*$. 289

Replacing the computation of $Radius(\mathbf{C}^p, p)$ in Lines 5 and 7 of Algorithm 2 by a computation of $Diam(\mathbf{C}^p)$ can improve the quality of a MIN-DIAM 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 (2-o(1))-approximation algorithm for MINDIAM, unless P = NP. This holds even in unweighted graphs with uniform budgets $b_i = 1, i = 1, ..., k$.

We will prove Theorem 5 by a reduction from 3SAT along the following 299 lines: First, given any 3SAT instance, we construct a MINDIAM instance 300 with variable agents and clause agents. Next, we present a structural result 301 (Lemma 4), from which we can infer that each variable agent will always move 302 to either a node representing its positive literal or a node representing its 303 negative literal; similarly, we infer that each clause agent will move to a node 304 representing a possible truth assignment of the respective clause. Finally, 305 we prove Theorem 5 by showing that satisfiable 3SAT instances admit a 306 MINDIAM solution of diameter 1, while unsatisfiable 3SAT instances result 307 in instances with optimum MINDIAM solutions of diameter at least 2. 308

Reduction. Let ϕ be an arbitrary boolean formula in conjunctive normal 309 form, where each clause contains 3 different literals, and let x_1, \ldots, x_n be 310 the *n* many variables and C_1, \ldots, C_m be the *m* many clauses of ϕ . We show 311 that any polynomial-time (2 - o(1))-approximation algorithm for MINDIAM 312 can be used to decide whether ϕ is satisfiable. From ϕ , we construct an 313 instance $\langle G, k, (p_i)_{i=1,\dots,k}, (b)_{i=1,\dots,k} \rangle$ with k agents of uniform budget b = 1314 and a graph $G = (V, E, \omega)$ with uniform edge weights $\omega = 1$ in the following 315 manner. 316

Set of nodes V: Using T = true and F = false, we first define the set of all possible truth assignments of a clause C containing 3 literals, L := $\{\text{TTT, TTF, TFT, TFF, FTT, FTF, FFT, FFF}\}$. Note that every clause C is satisfiable by exactly 7 of the 8 possible truth assignments in L (e.g. $x_1 \lor x_2 \lor$ \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

•
$$V_x = \{v_i \mid 1 \le i \le n\}$$
 are nodes corresponding to variables x_1, \ldots, 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 1325 *literals*, i.e. true-value and false-value assignments of the variables x_i ,

• $V_C = \{c_j \mid 1 \le j \le m\}$ are nodes corresponding to clauses C_1, \ldots, C_m ,

327 328 • $V_L = \{c_j^l \mid 1 \le j \le m, \forall l \in L\}$ 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 329 agent with a budget of b = 1, for a total of n + m agents. The main idea is to 330 initially space the agents by a pairwise distance of 3. We then let agents on V_x 331 "pick the value assignment of the variables x_i " by walking to their respective 332 node in V_{ℓ} , whereas we let agents on V_C "pick the truth assignment of the 333 clauses C_i " by walking to their respective node in V_L . Then a satisfiable 334 assignment of ϕ exists, if and only if the variable agents and the clause 335 agents "agree in their choice" which corresponds to an optimum MINDIAM 336 configuration **O** of diameter 1. Furthermore, any other configuration should 337 have diameter ≥ 2 . This gives rise to the 338

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

•
$$E_{x\ell} = \{\{v_i, v_i^{\mathrm{T}}\}, \{v_i, v_i^{\mathrm{F}}\} \mid 1 \leq i \leq n : v_i \in V_x, v_i^{\mathrm{T}}, v_i^{\mathrm{F}} \in V_\ell\}$$
 are edges
connecting each variable node x_i to its two literal nodes,



Figure 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 (\Box).

342	• $E_{CL} = \{\{c_j, c_j^l\} \mid 1 \leq j \leq m : c_j \in V_C, c_j^l \in V_L, c_j^l \text{ satisfies } C_j\}$
343	are edges connecting each clause node c_j with all nodes representing
344	satisfying assignments for clause C_j ,

345	• $E_{\ell L} = \{\{v'_i, c'_j\} \mid i \le n, j \le m : v'_i \in \{v^{\mathrm{T}}_i, v^{\mathrm{F}}_i\} \subset V_{\ell}, c'_j \in V_L, \text{ such that} \}$
346	either x_i does not appear in C_j ,
347	or x_i appears in C_j and v'_i agrees with c^l_j
348	are edges connecting unrelated literals and clause truth-assignments,
349	as well as matching literals and clause truth-assignments.

350 351 • $E_{\ell\ell} = \{\{u, v\} \mid u, v \in V_{\ell}\}$ 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 giving a proof of Theorem 5, we argue that no agent would stop in the middle of an edge:

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



Figure 3: A configuration **C** where agent $a(v_1)$, depicted by \Box , stops at distance δ from its starting node v_1 : Moving fully to $v_1^{\rm F}$ will only decrease the diameter $Diam(\mathbf{C})$.

Proof. Consider an agent $a(v_i)$ which corresponds to the variable x_i and 358 without loss of generality suppose that $a(v_i)$ chooses to move towards node 359 $v_i^{\rm F}$. Assume now, for the sake of contradiction, that agent $a(v_i)$ has stopped 360 at distance $0 \leq \delta < b = 1$ on the edge $(v_i, v_i^{\rm F})$, subdividing the edge into 361 two segments $(v_i + \delta, v_i)$ and $(v_i + \delta, v_i^{\rm F})$. (With $\delta = 0$ indicating that $a(v_i)$ 362 stayed on v_i without moving at all). In this case, $a(v_i)$ has spent δ units of 363 energy and has $b - \delta$ units of energy remaining (see Figure 3). Agent $a(v_i)$ is 364 connected with the rest of the agents through two possible paths: The first 365 one is through the segment $(v_i + \delta, v_i^F)$ of length $b - \delta > 0$, the other one is 366 through the path $(v_i + \delta, v_i), (v_i, v_i^{\mathrm{T}})$ of length $\delta + b \ge 1$. 367

It is easy now to notice that if $a(v_i)$ moves to node $v_i^{\rm F}$ (recall that it has $b - \delta$ units of energy remaining to do so), its distance to the other agents can only be reduced, as the contribution of the distance through $v_i^{\rm F}$ is now 0 and the contribution through node $v_i^{\rm T}$ is now 1. The same argument holds for the agents that correspond to the clauses. Hence moving all agents completely down to V_ℓ , V_L results in a configuration of non-increased diameter.

Proof. (Theorem 5). Based on the preceding construction of MINDIAM instances from 3SAT instances and the structural Lemma 4, we now give a
proof of Theorem 5.

³⁷⁷ \Rightarrow We first show that if ϕ is satisfiable then there exists a configuration C ³⁷⁸ of diameter $Diam(\mathbf{C}) = 1$. Since ϕ is satisfiable we have a truth assignment ³⁷⁹ to the variables which satisfies every clause of ϕ . For each variable x_i , we let ³⁸⁰ agent $a(v_i)$ move to node v_i^{T} if $x_i = true$ and to node v_i^{F} otherwise. Next, for

each clause C_i , we let agent $a(c_i)$ move to the node c_i^l , which corresponds to 381 the correct true/false-assignment picked by the three agents of the variables 382 in C_j . Note that both types of moves can be done with an energy of b = 1. 383 Let us examine the maximum distance of any two agents in this final config-384 uration. Notice that all agents $\{a(v_i) \mid v_i \in V_x\}$ moved to nodes in V_{ℓ} . By 385 construction, they are pairwise connected with an edge in $E_{\ell\ell}$. Similarly, the 386 agents $\{a(c_j) \mid c_j \in V_C\}$ have moved to nodes in V_L and are thus connected 387 by edges belonging to E_{LL} . It remains to compute the distance between the 388 variable agents (located in V_{ℓ}) and the clause agents (in V_L). Each agent 389 $a(c_i)$ by construction has distance equal to 1 from the three agents on nodes 390 that correspond to the truth assignment of the variables contained in clause 391 C_i , namely through an edge of $E_{\ell L}$. Moreover, each agent $a(c_i)$ has distance 392 1 from the nodes that belong to V_{ℓ} which correspond to the truth assignment 393 to variables that are *not* contained in clause c_i . Therefore, the maximum 394 distance between any two agents is equal to 1. 395

 \Leftarrow We now show that if ϕ is not satisfiable then every solution to MIN-396 DIAM is of size greater than or equal to 2. According to Lemma 4, we may 397 assume without loss of generality that no agent stops inside an edge nor 398 stays on its starting position. If ϕ is not satisfiable, then for every possible 399 truth assignment to the variables, there exists at least one clause in ϕ that is 400 not satisfied. Let us note here that in any optimum solution to MINDIAM, 401 the final positions of the agents that are initially located in variable nodes 402 corresponds to a truth assignment to the variables. Therefore, any final con-403 figuration will correspond to a truth assignment to the variables which will 404 not satisfy ϕ . Consider now an arbitrary final configuration of an instance 405 of MINDIAM. For the corresponding truth assignment to the variables, let 406 us assume that the clause that is not satisfied is $C_y = (x_r \vee \overline{x}_s \vee x_t)$. We 407 can show that similar arguments hold for any unsatisfied clause (irrespective 408 of whether the literals in the clause are positive or negative). If C_y is not 409 satisfied, this implies that agents $a(v_r), a(v_s)$ and $a(v_t)$ are located in nodes 410 $v_r^{\rm F}, v_s^{\rm T}$ and $v_t^{\rm F}$, respectively. 411

Let us examine the maximum distance of any two agents in this final configuration. Recall that the set of edges E_{CL} connects each clause node to nodes corresponding to all possible satisfying assignments for this clause. As a result, nodes c_y and c_y^{FTF} are not connected by an edge. Moreover, the shortest path between nodes c_y and c_y^{FTF} is equal to 2 (via edges in E_{CL} and E_{LL}). Therefore, agent $a(c_y)$ cannot reach node c_y^{FTF} . Any other node c_y^l , where $l \in L \setminus \{\text{FTF}\}$, to which agent $a(c_y)$ could relocate, corresponds to a truth assignment to x_r , x_s and x_t where at least one of the variables has the opposite value of its assignment. Say that $a(c_y)$ chooses to move to node c_y^{TTF} , then $a(c_y)$ will have a distance of 2 from agent $a(v_r)$ since $a(v_r)$ has moved to node v_r^{F} . Recall that node v_r^{F} is not connected by an edge to node c_y^{TTF} , since x_r appears in C_y but v_r^{F} does not agree with c_y^{TTF} . Therefore, agents $a(c_y)$ and $a(v_r)$ will have a distance of 2.

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.

429 4. Minimizing the Average Distance

In this Section we describe and analyze an algorithm for minimizing the 430 average pairwise distance between agents. We complement its approximation 431 ratio of $2(1-\frac{1}{k})$ with a tight analysis and rule out an FPTAS for MINSUM. 432 The main idea of the presented Algorithm 3 for MINSUM is similar to the 433 idea of Algorithm 2 for MINDIAM. We fix a point p in the graph G as a 434 gathering point and move each agent i as close as possible to p with respect 435 to its energy constraint, breaking ties arbitrarily. Algorithm 3 exhaustively 436 tests all points in $V \cup \bigcup_i S(i)$ as possible gathering points and selects the 437 point p with a configuration $\mathbf{C} = (c_1, \ldots, c_k)$ of minimum sum of pairwise 438 distances between the agents, $Sum(\mathbf{C}) = \sum_{i} \sum_{j} d(c_i, c_j)$. The choice of the 439 search space for gathering points is based on a characterization of optimum 440 solutions, similar in look and proof to Lemmata 2 and 3: 441

Lemma 5. There exists an optimum solution O for MINSUM where every agent stops on a point in $V \cup \bigcup_i S(i)$.

Proof. Assume for the sake of contradiction that in every optimum configuration $\mathbf{C} = (c_1, \ldots, c_j)$, there is at least one agent j which stops on a point $c_j \notin V \cup \bigcup_i S(i)$. Define by \mathbb{O} the set of all optimum solutions, and with $\mathbb{O}' \subseteq \mathbb{O}$ its subset of configurations with a minimum number of agents jsuch that $c_j \notin V \cup \bigcup_i S(i)$. We denote by $\mathbf{C}^* = (c_1^*, c_2^*, \ldots, c_k^*)$ a configuration with a maximum number of agents stopping on any common point $c^* \notin V \cup \bigcup_i S(i)$, among all optimum configurations of \mathbb{O}' .

⁴⁵¹ Denote by $A_{=}$ the set of all agents j with stopping point $c_{j}^{*} = c^{*}$. Since ⁴⁵² $c^{*} \notin V \cup \bigcup_{i} S(i)$, we must have $d(p_{j}, c_{j}^{*}) < b_{j}$ for all agents $j \in A_{=}$. Further-⁴⁵³ more, since $c^{*} \notin V$, $c^{*} \in seg(e)$ for some edge $\{u, v\}$. Denote by A_{u} the set of all agents j for which $c_j^* \in seg(e)$ and c_j^* is between u and c^* in seg(e), or for which there is a shortest path from c_j^* to c^* going through u. Denote by A_v the set of all agents $j \notin A_= \cup A_u$ (for which, there must be a shortest path from c_j^* to c^* going through v, or for which $c_j^* \in seg(e)$ and c_j^* is between vand c^* in seg(e)).

Without loss of generality, we assume $|A_u| \ge |A_v|$. Now consider what happens to $Sum(\mathbf{C}^*)$ when we move all stopping points c_j^* of agents $j \in A_=$ by an $\varepsilon > 0$ towards u:

- The pairwise distances in $A_{=}$, in A_{u} , and in A_{v} individually and the distances between agents in A_{u} and agents in A_{v} stay the same.
- The distances between agents in A_{\pm} and in A_u decrease by ε each.
- The distances between agents in A_{\pm} and in A_v increase by at most ε .

Overall, $Sum(\mathbf{C}^*)$ changes under the moving operation by a total value of 466 at most $2\varepsilon \cdot |A_{\pm}| \cdot |A_{\nu}| - 2\varepsilon \cdot |A_{\pm}| \cdot |A_{\mu}| \leq 0$. Hence we can move the stopping 467 points c_i^* of agents $j \in A_{=}$ until we reach (i) a point $p \in V \cup \bigcup_i S(i)$ or until 468 we reach (ii) the stopping point $p = c_a^*$ of another agent a, whichever comes 469 first. In either case, we still have for all agents $j \in A_{=}$ that $d(p_j, p) \leq b_j$. 470 Furthermore, in the first case we have found a feasible configuration with 471 a smaller number of agents j such that $c_j \notin V \cup \bigcup_i S(i)$, contradicting the 472 minimality of \mathbb{O}' -configurations among configurations in \mathbb{O} . In the second 473 case, we have found a feasible configuration with a larger number of agents 474 stopping on the same point p, contradicting the maximality of \mathbf{C}^* among 475 configurations in \mathbb{O}' . 476

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). Let $\mathbf{C}^* = (c_1^*, \ldots, c_k^*)$ denote the configuration computed by Algorithm 3. We denote with $s^* := Sum(\mathbf{C}^*)$ the sum of all pairwise agent distances in \mathbf{C}^* . Furthermore, let $\mathbf{O} = (o_1, \ldots, o_k)$ 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 $s_{\mathrm{OPT}} = Sum(\mathbf{O}) = \sum_i \sum_j d(o_i, o_j)$. Choosing a point $o \in \arg\min_{o_i \in \mathbf{O}} \sum_j d(o_i, o_j)$ we get

$$\sum_{j} d(o, o_{j}) = \frac{1}{k} \cdot k \sum_{j} d(o, o_{j}) \leq \frac{1}{k} \cdot \sum_{i} \sum_{j} d(o_{i}, o_{j}) = \frac{1}{k} \cdot s_{\text{OPT}}.$$
 (1)

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

Input: An instance $\langle G, k, (p_i)_{i=1,\dots,k}, (b_i)_{i=1,\dots,k} \rangle$. **Output:** Configuration C with $Sum(\mathbf{C}) \leq 2(1 - \frac{1}{k}) \cdot \min_{\text{feasible } \mathbf{C}'} Sum(\mathbf{C}')$. 1: for each $p \in V \cup \bigcup_i S(i)$ do Compute $\mathbf{C}^p := (c_1^p, \ldots, c_k^p),$ 2: where either $c_i^p = p$ if $d(p_i, p) \leq b_i$, or otherwise 3: $c_i^p \in \arg\min\{d(p, c_i) \mid c_i \in B(i) \cup S(i)\}$ (breaking ties arbitrarily). 4: Compute $Sum(\mathbf{C}^p)$. 5:6: end for $Sum(\mathbf{C}^p).$ 7: Return arg min $\mathbf{C}^p : p \in V \cup \bigcup_i S(i)$

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 $s^{o} := Sum(\mathbf{C}^{o}) = \sum_{i} \sum_{j} d(c_{i}^{o}, c_{j}^{o})$. 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

$$\begin{split} s^* &\leq s^o = \sum_i \sum_j d(c_i^o, c_j^o) \\ &\leq 2 \sum_j d(c_a^o, c_j^o) + \sum_{i \neq a} \sum_{\substack{j \neq a \\ j \neq i}} (d(c_i^o, o) + d(o, c_j^o)) \\ &= 2 \sum_j d(o, c_j^o) + (k-2) \sum_{i \neq a} d(c_i^o, o) + (k-2) \sum_{j \neq a} d(o, c_j^o) \\ &= (2k-2) \sum_j d(o, c_j^o) \\ &\leq 2(k-1) \sum_j d(o, o_j) \stackrel{(1)}{\leq} 2(1 - \frac{1}{k}) \cdot s_{\text{OPT}}. \end{split}$$

(Lower bound). To see that the above analysis is tight, we construct instances $\langle G, k, (p_i)_{i=1,...,k}, (b)_{i=1,...,k} \rangle$ with k agents of uniform budget b = 1, for which the tie-breaking in Line 3 of Algorithm 3 leads to a configuration **C** with $Sum(\mathbf{C}) = 2(1 - \frac{1}{k}) \cdot Sum(\mathbf{O})$. An example of such an instance with three agents of uniform budget b = 1 is given in Figure 4. We now give a construction for an arbitrary number of agents k and an evaluation for each $p \in V \cup \bigcup_i S(i)$:

Define $G = (V, E, \omega)$ with uniform edge weights $\omega = 1$ as follows: For 498 each agent *i*, we connect its starting position p_i to nodes $u_{i,j}, \forall 1 \leq j \leq k$. 499 Furthermore, we connect each node $u_{i,i}$ to all nodes $u_{j,i}$, $\forall 1 \leq j \leq k$. We also 500 add edges $\{u_{i,i}, u_{j,j}\}, \forall 1 \leq i < j \leq k$, such that the k nodes $u_{1,1}, \ldots, u_{k,k}$ 501 induce a k-clique, see Figure 4. As each pair of agents $i \neq j$ has original 502 distance $d(p_i, p_j) = 3$, every configuration C must have $Sum(C) \ge k(k-1)$, 503 with equality only for the k-clique $\mathbf{O} = (u_{1,1}, \ldots, u_{k,k})$. For every agent i, we 504 have $B(i) \cup S(i) = \{p_i, u_{i,1}, \dots, u_{i,i}, \dots, u_{i,k}\}$ and thus $V \cup \bigcup_i S(i) = V$. We 505 analyze the configurations \mathbf{C}^p and $Sum(\mathbf{C}^p) = \sum_i \sum_j d(c_i^p, c_j^p)$ computed in 506 Lines 2–5 for each round $p \in V$: 507

• $p = u_{a,a}$: Agent a will move to $c_a^p = u_{a,a}$, while every other agent j is indifferent between $u_{j,j}$ and $u_{j,a}$ and thus might move to $c_j^p = u_{j,a}$. In this case, agents $i \neq j$ have distance $d(c_i^p, c_j^p) = 1$ if i = a or j = a and $d(c_i^p, c_j^p) = 2$ otherwise, giving $Sum(\mathbf{C}^p) = 2(k-1)\cdot 1 + (k-1)(k-2)\cdot 2 =$ $2(k-1)^2 = 2\frac{k-1}{k}k(k-1) = 2(1-\frac{1}{k})\cdot Sum(\mathbf{O}).$

• $p = u_{a,b}$ (for $a \neq b$): Agent *a* will move to $c_a^p = u_{a,b}$ and agent *b* to $c_b^p = u_{b,b}$. Every other agent *j* is indifferent between $u_{j,j}$ and $u_{j,b}$, having both distance 2 to $u_{a,b}$. In case they each choose $c_j^p = u_{j,b}$, we get a configuration \mathbf{C}^p which is symmetric to the previous case.

• $p = p_a$: Agent *a* will stay on p_a , while every other agent *j* is indifferent between $u_{j,j}$ and $u_{j,a}$. In case they each choose $c_j^p = u_{j,a}$, we get a configuration \mathbf{C}^p where any two agents *i*, *j* have distance 2., giving $Sum(\mathbf{C}^p) = 2k(k-1) = 2 \cdot Sum(\mathbf{O}).$

⁵²¹ Hence the approximation analysis of Algorithm 3 is tight.

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 $poly(k, \frac{1}{\varepsilon})$. We reuse the reduction to 3SAT already given in Theorem 5. Recall from its proof that (i) the underlying 3SATformula 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.



Figure 4: Lower bound construction for Algorithm 3 with k = 3 agents: The only optimum MINSUM configuration is $\mathbf{O} = (u_{1,1}, u_{2,2}, u_{3,3})$ with $Sum(\mathbf{O}) = k(k-1)$. In the tie-breaking in Line 4, agents *i* will generally be indifferent between the options $u_{i,i}$ and $u_{i,j}$, leading to a configuration \mathbf{C} with $Sum(\mathbf{C}) = 2(k-1)^2 = 2(1-\frac{1}{k}) \cdot Sum(\mathbf{O})$.

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

537 5. Additive Approximation and Conclusion

In this paper, we explored the task of *Near-Gathering* a group of energy-538 constrained agents, whose movements are restricted by their energy budget. 539 We showed how to compute, in polynomial time, an optimum solution for 540 MINBALL (minimizing the radius of a smallest ball containing all agents), a 541 2-approximation for MINDIAM (minimizing the maximum distance between 542 any two agents), and a $2(1-\frac{1}{k})$ -approximation for MINSUM (minimizing the 543 average distance between any two agents). For MINDIAM, we provided a 544 matching hardness result, while for MINSUM, we ruled out the existence of 545 an FPTAS, unless P = NP. Hence for future work, a major open problem is 546 to improve upon the (in)approximability of MINSUM. 547

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. There exists 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, unless P = NP. (Proof at end of Section.)

This is surprising for two reasons. On the one hand, not moving the agents 556 at all is already an additive $+(2 \max_i b_i)$ -approximation. On the other hand, 557 this is the only result in this paper, in which the two scenarios of (I) node 558 stops and (II) arbitrary stops *differ*. The difference in the hardness result 559 boils down to the loss of Lemma 4 in the adaption of the proof of Theorem 5, 560 which we can only fully salvage for the case of node stops. Does this mean 561 that there is a polynomial-time $+(2 \max_i b_i - o(1))$ -approximation for the 562 scenario with arbitrary final positions? This remains completely open. 563

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.

We finish by proving the additive approximation hardness results in Theorem 8 by a similar reduction from 3SAT as the one given for the multiplicative (2 - o(1))-approximation hardness of MINDIAM. Instead of a self-contained proof, we describe all necessary adaptions we make in the proof of Theorem 5.

Proof. Given an arbitrary 3SAT formula ϕ with n variables and m clauses, we first make a one-to-one copy ϕ' of all its variables and all its clauses. Clearly, ϕ is satisfiable if and only if $\Phi := \phi \wedge \phi'$ is satisfiable. We now construct an instance $\langle G, k, (p_i)_{i=1,\dots,k}, (b)_{i=1,\dots,k} \rangle$ with k = 2n + 2m agents of uniform budget b and a graph $G = (V, E, \omega)$ in the same manner as for Theorem 5.

⁵⁷⁸ We add weights to the edge of G in the following manner: the weight of ⁵⁷⁹ each edge in $E_{x\ell} \cup E_{CL}$ is b and the weight of each edge in $E_{\ell\ell} \cup E_{\ell L} \cup E_{LL}$ is 2b. ⁵⁸⁰ Overall, the main reduction idea is now the following: Φ shall be satisfiable if ⁵⁸¹ and only if there is an optimum solution of diameter 2b. Furthermore, from ⁵⁸² any configuration with a "good" additive approximation and small diameter, ⁵⁸³ we can infer either a satisfiable assignment of ϕ or of ϕ' .

We first show that if Φ is satisfiable, then there exists an optimum configuration **C** of diameter $Diam(\mathbf{C}) = 2b$. This follows immediately from the proof of Theorem 5. Since we increased the weight of all relevant edges by a factor or 2b, we get with the same reasoning an optimum configuration **C** of diameter $Diam(\mathbf{C}) = 2b$ (instead of the previously shown 1). We now show that if Φ is not satisfiable, then every solution to MIN-DIAM with node stops is of size greater than or equal to 2b + 2b = 4b. The difficulty lies in the fact that Lemma 4 is no longer valid, since the node triples $v_i, v_i^{\mathrm{T}}, v_i^{\mathrm{F}}$ no longer form an equilateral triangle; instead, we have $\omega(\{v_i^{\mathrm{T}}, v_i^{\mathrm{F}}\}) = 2b$, while v_i is connected to $v_i^{\mathrm{T}}, v_i^{\mathrm{F}}$ with two edges of weight b. Similarly, edges in E_{LL} have weight 2b while edges in E_{CL} have weight b.

We now observe the following: If in a configuration \mathbf{C} there are two agents 595 i, j which stay on their starting position p_i, p_j , then they must have a dis-596 tance of $d(p_i, p_j) \ge b + 2b + b = 4b$. Otherwise, there is at most one agent 597 staying at its starting position. Thus in at least one of the subgraphs induced 598 by ϕ and ϕ' , respectively, we can assume that all agents move to nodes in 599 $V_{\ell} \cup V_L$. Repeating the arguments given in the proof of Theorem 5, since ϕ 600 and ϕ' are not satisfiable, there must be two agents in V_{ℓ} and V_L which are 601 not connected by an edge and thus have distance at least 2b + 2b = 4b. 602

 \Leftarrow We now show that if Φ is not satisfiable then every solution to MIN-603 DIAM with arbitrary stops is of size greater than or equal to $2b + \frac{4}{3}b = \frac{10}{3}b$. 604 As in the case of node stops, we note that if in a configuration \mathbf{C} there 605 are two agents i, j which move away from their starting positions p_i, p_j by 606 a distance of at most $\delta := \frac{b}{3}$, then they must have a distance of at least 607 $d(p_i, p_j) - \delta - \delta = b + 2b + b - 2\delta = \frac{10}{3}b$. Otherwise, there is at most one 608 agent staying closer than δ to its starting position. Thus in at least one of 609 the subgraphs induced by ϕ and ϕ' , respectively, each agent moves either to 610 a vertex in $V_{\ell} \cup V_L$ or to a point of an edge of $E_{x\ell} \cup E_{CL}$ at distance at least 611 $\delta = \frac{b}{3}$ from its starting position. Repeating the arguments given in the proof 612 of Theorem 5, since ϕ and ϕ' are not satisfiable, there must be a variable 613 agent $a(v_i)$ moving towards a node $u \in V_{\ell}$ and a clause agent $a(c_i)$ moving 614 towards a node $v \in V_L$ which are not connected, i.e. $\{u, v\} \notin E_{\ell L}$. Thus 615 $a(v_i)$ and $a(c_i)$ are connected with a shortest path going via v_i or c_i and 616 hence have a distance of at least $\delta + b + 2b = \frac{10}{3}b$. 617

Since a polynomial-time additive +(2b - o(1))-approximation algorithm 618 for MINDIAM with node stops could distinguish between instances with an 619 optimum solution with diameter 2b and instances with an optimum solution 620 with diameter 4b, it would also be able to decide whether Φ is satisfiable of 621 not. Similarly a polynomial-time additive $+(\frac{4}{3}b - o(1))$ -approximation algo-622 rithm for MINDIAM with arbitrary stops could distinguish between instances 623 with an optimum solution with diameter 2b and instances with an optimum 624 solution with diameter $\frac{10}{3}b$. This completes the proof. 625

- [1] L. Pagli, G. Prencipe, G. Viglietta, Getting close without touching:
 Near-gathering for autonomous mobile robots, Distributed Computing
 28 (5) (2015) 333–349. doi:10.1007/s00446-015-0248-5.
- [2] P. Flocchini, G. Prencipe, N. Santoro, Distributed Computing by Mo bile 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.
- [3] M. Cieliebak, P. Flocchini, G. Prencipe, N. Santoro, Distributed computing by mobile robots: Gathering, SIAM Journal on Computing 41 (4)
 (2012) 829–879. doi:10.1137/100796534.
- [4] P. Flocchini, G. Prencipe, N. Santoro, P. Widmayer, Gathering of asynchronous robots with limited visibility, Theoretical Computer Science 337 (1-3) (2005) 147–168. doi:10.1016/j.tcs.2005.01.001.
- [5] J. Lin, A. S. Morse, B. D. O. Anderson, The multi-agent rendezvous
 problem. parts 1 and 2., SIAM Journal on Control and Optimization
 46 (6) (2007) 2096–2147.
- [6] R. Cohen, D. Peleg, Convergence properties of the gravitational algorithm in asynchronous robot systems, SIAM Journal on Computing 34 (2005) 1516–1528. doi:10.1137/S0097539704446475.
- [7] J. Czyzowicz, A. Labourel, A. Pelc, How to meet asynchronously (almost) everywhere, ACM Transactions on Algorithms 8 (4) (2012) 37:1– 37:14. doi:10.1145/2344422.2344427.
- [8] A. Pelc, Deterministic rendezvous in networks: A comprehensive survey, Networks 59 (3) (2012) 331–347. doi:10.1002/net.21453.
- [9] S. Alpern, S. Gal, The Theory of Search Games and Rendezvous,
 Kluwer, 2003. doi:10.1007/b100809.
- [10] S. Cicerone, G. D. Stefano, A. Navarra, Gathering of robots on meetingpoints: feasibility and optimal resolution algorithms, Distributed Computing 31 (1) (2018) 1–50. doi:10.1007/s00446-017-0293-3.
- [11] M. Betke, R. L. Rivest, M. Singh, Piecemeal learning of an unknown
 environment, Machine Learning 18 (2) (1995) 231–254. doi:10.1007/
 BF00993411.

- [12] B. Awerbuch, M. Betke, R. L. Rivest, M. Singh, Piecemeal graph exploration by a mobile robot, Information and Computation 152 (2) (1999)
 155–172. doi:10.1006/inco.1999.2795.
- [13] C. A. Duncan, S. G. Kobourov, V. S. A. Kumar, Optimal Constrained
 Graph Exploration, ACM Transactions on Algorithms 2 (3) (2006) 380–
 402. doi:10.1145/1159892.1159897.
- [14] P. Fraigniaud, L. Gasieniec, D. R. Kowalski, A. Pelc, Collective tree
 exploration, Networks 48 (3) (2006) 166–177. doi:10.1002/net.20127.
- [15] 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. doi:10.1007/
 11682127_24.
- [16] 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. doi:10.1007/
 978-3-540-72951-8_5.
- [17] 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. doi:10.1007/978-3-319-25258-2_
 25.
- [18] D. Dereniowski, Y. Disser, A. Kosowski, D. Pająk, P. Uznański, Fast collaborative graph exploration, Information and Computation 243 (2015)
 37-49. doi:https://doi.org/10.1016/j.ic.2014.12.005.
- [19] E. D. Demaine, M. Hajiaghayi, H. Mahini, A. S. Sayedi-Roshkhar,
 S. Oveisgharan, M. Zadimoghaddam, Minimizing movement, ACM
 Transactions on Algorithms 5 (3) (2009) 30:1–30:30. doi:10.1145/
 1541885.1541891.
- [20] E. D. Demaine, M. Hajiaghayi, D. Marx, Minimizing movement: Fixed-parameter tractability, ACM Transactions on Algorithms 11 (2) (2014)
 14:1-14:29. doi:10.1145/2650247.

- [21] 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. doi:10.1007/978-3-642-45346-5_6.
- [22] 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. doi:10.1007/s00453-014-9939-8.
- 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. doi:
 10.1007/978-3-642-45346-5_9.
- [24] 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. doi:10.1007/978-3-662-43951-7_36.
- [25] A. Bärtschi, J. Chalopin, S. Das, Y. Disser, B. Geissmann, D. Graf,
 A. Labourel, M. Mihalák, Collaborative Delivery with EnergyConstrained Mobile Robots, in: 23rd International Colloquium on Structural Information and Communication Complexity SIROCCO'16, 2016,
 pp. 258–274. doi:10.1007/978-3-319-48314-6_17.
- [26] A. Bärtschi, J. Chalopin, S. Das, Y. Disser, B. Geissmann,
 D. Graf, A. Labourel, M. Mihalák, Collaborative delivery with energyconstrained mobile robots, Theoretical Computer ScienceTo appear. See
 also SIROCCO'16 (2017). doi:10.1016/j.tcs.2017.04.018.
- [27] 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, 2016. doi:10.1007/978-3-319-48314-6_18.
- [28] E. Bampas, S. Das, D. Dereniowski, C. Karousatou, Collaborative delivery by energy-sharing low-power mobile robots, in: International Symposium on Algorithms and Experiments for Sensor Systems, Wireless

Networks and Distributed Robotics ALGOSENSORS'17, 2017, pp. 1–

12. doi:10.1007/978-3-319-72751-6_1.