



**HAL**  
open science

## 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-03065725v1>**

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.



HAL Authorization

# Near-gathering of energy-constrained mobile agents<sup>☆</sup>

Andreas Bärtschi<sup>a,\*</sup>, Evangelos Bampas<sup>b</sup>, Jérémie Chalopin<sup>c</sup>,  
Shantanu Das<sup>c</sup>, Christina Karousatou<sup>d</sup>, Matúš Mihalák<sup>e</sup>

<sup>a</sup>*Center for Nonlinear Studies, Los Alamos National Laboratory, USA*

<sup>b</sup>*Paris-Saclay University and CNRS, LRI, France*

<sup>c</sup>*CNRS, Aix-Marseille Université and Université de Toulon, LIS, France*

<sup>d</sup>*Department of Mathematics, TU Darmstadt, Germany*

<sup>e</sup>*Department 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 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

---

---

<sup>☆</sup>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. 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 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 a 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 [1].

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.

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

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

60 **Definition 1** (Near-Gathering).

61 *Instance:*  $\langle G, k, (p_i)_{i=1, \dots, k}, (b_i)_{i=1, \dots, k} \rangle$ , where  $G = (V, E, \omega)$  is an undi-  
62 rected edge-weighted graph,  $k$  denotes the total number of agents,  $p_i$  denotes  
63 the initial position of agent  $i$  and  $b_i$  denotes the total amount of energy agent  $i$   
64 initially has at its disposal.

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

68 *Goals:* (i) MINBALL: Minimize  $\text{Radius}(\mathbf{C}, \mathbf{c})$  of a smallest ball containing  
69  $\mathbf{C}$  around an optimally chosen center  $\mathbf{c}$ , where  $\text{Radius}(\mathbf{C}, \mathbf{c}) = \max_i d(\mathbf{c}, c_i)$ .  
70 We consider both the scenario with node centers only, and the scenario with  
71 arbitrary point centers.

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

73 (iii) MINSUM: Minimize  $\text{Sum}(\mathbf{C})$ , where  $\text{Sum}(\mathbf{C}) = \sum_i \sum_j d(c_i, c_j)$ .

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

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

106 For the model studied in this paper, where each agent has a distinct  
107 energy budget, the problem of *Broadcast* and *Convergecast* was studied in  
108 [22] who provided hardness results for trees and approximation algorithms  
109 for arbitrary graphs. The problem of delivering packages by multiple agents  
110 having energy constraints was studied in [23, 24, 25, 26]. All of these problems  
111 were shown to be NP-hard for general graphs even if the agents are allowed

112 to exchange energy when they meet [27, 28].

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

121 We conclude with remarks on future research opportunities, including  
122 preliminary approximation hardness results for additive approximation of  
123 MINDIAM, in Section 5. All our results – with the exception of additive  
124 approximation – hold for both node-stop as well as arbitrary-stop scenarios.

## 125 2. Preliminaries and Minimizing the Radius

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

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

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

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

150 **Definition 3** (Midpoints). *Given a set  $S$  of points in  $G$ , denote by  $G' =$   
 151  $(V', E', \omega')$  the graph we get from  $G = (V, E, \omega)$  by subdividing the edges in  
 152  $E$  with points from  $S$ , i.e.  $V' = V \cup S$ . We define the midpoint set  $M(S)$  of  
 153 points in  $G'$  – and by bijection also of  $G$  – as:*

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

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

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

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

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  $\langle G, k, (p_i)_{i=1,\dots,k}, (b_i)_{i=1,\dots,k} \rangle$ .

**Output:** Configuration  $\mathbf{C}$ , center  $\mathbf{c} \in V$  with minimum radius  $\text{Radius}(\mathbf{C}, \mathbf{c})$ .

```
1: for each  $v \in V$  do
2:   Compute  $\mathbf{C}^v := (c_1^v, \dots, c_k^v)$ ,
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  $\text{Radius}(\mathbf{C}^v, v)$ .
6: end for
7: Return  $\arg \min_{\mathbf{C}^v, v: v \in V} \text{Radius}(\mathbf{C}^v, v)$ .
```

---

177 agent  $i$  compute the minimum distance to this fixed center it can reach, given  
178 its starting position  $p_i$  and its energy budget  $b_i$ . Iterating over all possible  
179 center nodes, we find an optimal solution:

180 **Theorem 1** (MinBall, node centers). *Algorithm 1 is a polynomial-time al-*  
181 *gorithm for MINBALL with node centers.*

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

186 **Lemma 2.** *There exists an optimum solution  $(\mathbf{O}, \mathbf{o})$  for MINBALL where*  
187 *every agent  $i$  either stops on the center  $\mathbf{o}$  or on a point in  $B(i) \cup S(i)$ ,*  
188 *independent of whether  $\mathbf{o}$  is contained in  $\bigcup_i (B(i) \cup S(i))$  or not.*

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

201 *MinBall for arbitrary centers.* We now extend our approach to find optimum  
 202 MINBALL solutions for arbitrary centers. As can be seen from Lemma 2,  
 203 when testing for a fixed center  $\mathbf{c}$ , in addition to checking the points in  $B(i) \cup$   
 204  $S(i)$  we should also consider whether agent  $i$  can reach  $\mathbf{c}$  itself. As candidates  
 205 for the center  $\mathbf{c}$  we take all points in the midpoint set  $M(V \cup \bigcup_i S(i))$ , yielding  
 206 Algorithm 2:

207 **Theorem 2** (MinBall, arbitrary centers). *Algorithm 2 is a polynomial-time*  
 208 *algorithm for MINBALL with arbitrary centers.*

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

212 **Lemma 3.** *There exists an optimum solution  $(\mathbf{O}, \mathbf{o})$  for MINBALL where*  
 213 *the center  $\mathbf{o}$  is contained in  $M(V \cup \bigcup_i S(i))$ .*

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

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

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

231 Otherwise denote by  $c_a \in B(a) \cup S(a)$  the agent stopping position with  
 232 maximum distance  $d(c_a, \mathbf{c})$  among all configuration points which have a short-  
 233 est path to  $\mathbf{c}$  containing  $u$ . Analogously, denote by  $c_b \in B(b) \cup S(b)$  the  
 234 furthest agent stopping position among all configuration points which have  
 235 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  $\mathbf{C}$ , center  $\mathbf{c} \in G$  with minimum radius  $\text{Radius}(\mathbf{C}, \mathbf{c})$ .

```

1: for each  $p \in M(V \cup \bigcup_i S(i))$  do
2:   Compute  $\mathbf{C}^p := (c_1^p, \dots, c_k^p)$ ,
3:     where either  $c_i^p = p$  if  $d(p_i, p) \leq b_i$ , or otherwise
4:      $c_i^p \in \arg \min\{d(p, c_i) \mid c_i \in B(i) \cup S(i)\}$  (breaking ties arbitrarily).
5:   Compute  $\text{Radius}(\mathbf{C}^p, p)$ .
6: end for
7: Return  $\arg \min_{\mathbf{C}^p, p: p \in M(V \cup \bigcup_i S(i))} \text{Radius}(\mathbf{C}^p, p)$ .
```

---

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

### 243 3. Minimizing the Diameter

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

250 **Theorem 3** (MinDiam, 2-apx). *Algorithm 2 is a polynomial-time 2-approxi-*  
251 *mation algorithm for MINDIAM.*

252 *Proof.* Let configuration  $\mathbf{C}^* = (c_1^*, \dots, c_k^*)$  with center  $\mathbf{c}^*$  be the MINBALL  
253 solution computed by Algorithm 2. We denote the radius of  $(\mathbf{C}^*, \mathbf{c}^*)$  by  $r^* =$   
254  $\text{Radius}(\mathbf{C}^*, \mathbf{c}^*) = \max_j d(\mathbf{c}^*, c_j^*)$  and the diameter of  $\mathbf{C}^*$  by  $d^* = \text{Diam}(\mathbf{C}^*) =$   
255  $\max_{i,j} d(c_i^*, c_j^*)$ . Using the triangle inequality, we have for all configuration  
256 points  $c_i^*, c_j^*$  that  $d(c_i^*, c_j^*) \leq d(c_i^*, \mathbf{c}^*) + d(c_j^*, \mathbf{c}^*)$  and thus  $d^* \leq 2 \cdot r^*$ .

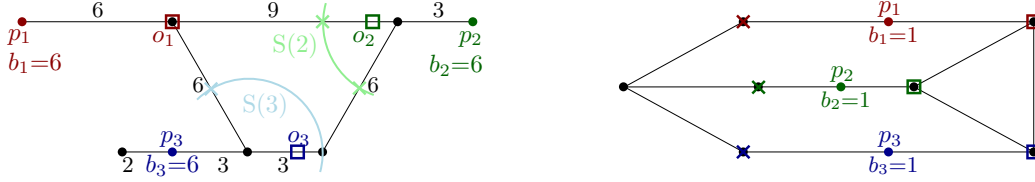


Figure 1: **(left)** MINDIAM-instance with (unique) optimum solution  $\mathbf{O} = (o_1, o_2, o_3)$  of diameter  $Diam(\mathbf{O}) = 8$ , in which we have final positions  $o_3 \notin B(3) \cup S(3)$  and  $o_2 \notin M(V \cup \bigcup_i S(i))$ .

**(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  $\square$ , with diameters 2 and 1, respectively) improves the quality of a MINDIAM solution for certain instances by a factor of 2.

257 Now let  $\mathbf{O} = (o_1, \dots, o_k)$  be an optimum configuration for MINDIAM with  
 258 diameter  $d_{OPT} := Diam(\mathbf{O}) = \max_{i,j} d(o_i, o_j)$ . We choose an arbitrary point  
 259  $o \in \mathbf{O}$  and compute the radius of a smallest ball around  $o$  containing  $\mathbf{O}$ ,  
 260  $r_o := Radius(\mathbf{O}, o) = \max_j d(o, o_j) \leq d_{OPT}$ . By Theorem 2, we have  $r^* \leq r_o$   
 261 (even though  $o$  might not have been considered as a center candidate, see  
 262 e.g. Figure 1 (left)). Combining all inequalities, we get

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

263 hence  $\mathbf{C}^*$  is a 2-approximation for MINDIAM.  $\square$

264 **Theorem 4** (MinDiam, on Trees). *Algorithm 2 is a polynomial-time algo-*  
 265 *rithm for MINDIAM on trees.*

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

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

271 We show that, indeed, Algorithm 2 computes a configuration  $\mathbf{C}^*$  with diam-  
 272 eter  $Diam(\mathbf{C}^*) = d^*$ . To this end, denote by  $a, b$  two agents giving rise to  $d^*$ ,

273 and let  $q_a \in B(a) \cup S(a)$ ,  $q_b \in B(b) \cup S(b)$  be two points with  $d(q_a, q_b) = d^*$ .  
 274 Since we consider tree graphs here, there is a unique shortest path from  $q_a$   
 275 to  $q_b$  and thus a unique midpoint  $\mathbf{c}^* \in G$  with  $d(\mathbf{c}^*, q_a) = d(\mathbf{c}^*, q_b) := \frac{d^*}{2}$ .  
 276 As  $\mathbf{c}^*$  is contained in  $M(V \cup \bigcup_i S(i))$ , Algorithm 2 will use  $\mathbf{c}^*$  as a center  
 277 point candidate for which it computes a configuration  $\mathbf{C}^* = (c_1^*, \dots, c_k^*)$ . By  
 278 definition, we have  $d(\mathbf{c}^*, c_a^*) = d(\mathbf{c}^*, q_a) = \frac{d^*}{2} = d(\mathbf{c}^*, q_b) = d(\mathbf{c}^*, c_b^*)$ .  
 279 It is enough to show that for all other agents  $i$  we have  $d(\mathbf{c}^*, c_i^*) \leq \frac{d^*}{2}$ ,  
 280 too. Assume for the sake of contradiction that this is not the case and that  
 281 there is an agent  $i$  with  $d(\mathbf{c}^*, c_i^*) > \frac{d^*}{2}$ . Consider the shortest  $c_i^*$ - $\mathbf{c}^*$ -path  
 282  $P_i$ , the shortest  $c_a^*$ - $\mathbf{c}^*$ -path  $P_a$  and the shortest  $c_b^*$ - $\mathbf{c}^*$ -path  $P_b$ . By definition  
 283 of  $d^*$  and  $c^*$ , the paths  $P_a$  and  $P_b$  must be interiorly disjoint,  $P_a \cap P_b =$   
 284  $\{\mathbf{c}^*\}$ . Since  $P_i$  is a path on a tree ending in the same node  $\mathbf{c}^*$ , it must  
 285 be interiorly disjoint with at least one of the two paths  $P_a, P_b$ , without loss  
 286 of generality with  $P_a$ . Because any two points in a tree are connected by  
 287 a unique path, we have  $d(c_i^*, c_a^*) = d(c_i^*, \mathbf{c}^*) + d(\mathbf{c}^*, c_a^*) > d^*$  and thus also  
 288  $\min_{q_i \in B(i) \cup S(i), q_a \in B(a) \cup S(a)} d(q_i, q_a) > d^*$ , contradicting the maximality of  $d^*$ .  
 289 Hence we have  $\text{Diam}(\mathbf{C}^*) \leq \max_{i,j} d(c_i^*, \mathbf{c}^*) + d(\mathbf{c}^*, c_j^*) = d^*$ .  $\square$

290 Replacing the computation of  $\text{Radius}(\mathbf{C}^p, p)$  in Lines 5 and 7 of Algo-  
 291 rithm 2 by a computation of  $\text{Diam}(\mathbf{C}^p)$  can improve the quality of a MIN-  
 292 DIAM solution by a factor of up to 2 for some instances, see for example  
 293 Figure 1 (right). However, this does not translate to the worst-case approxi-  
 294 mation guarantee, as one can see in the instance constructed in the following  
 295 matching approximation hardness result.

296 **Theorem 5.** *There exists no deterministic polynomial-time  $(2 - o(1))$ -appro-*  
 297 *ximation algorithm for MINDIAM, unless  $P = NP$ . This holds even in un-*  
 298 *weighted graphs with uniform budgets  $b_i = 1$ ,  $i = 1, \dots, k$ .*

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

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

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

- 323 •  $V_x = \{v_i \mid 1 \leq i \leq n\}$  are nodes corresponding to *variables*  $x_1, \dots, x_n$ ,
- 324 •  $V_\ell = \{v_i^T \mid 1 \leq i \leq n\} \cup \{v_i^F \mid 1 \leq i \leq n\}$  are nodes corresponding to  
 325 *literals*, i.e. true-value and false-value assignments of the variables  $x_i$ ,
- 326 •  $V_C = \{c_j \mid 1 \leq j \leq m\}$  are nodes corresponding to *clauses*  $C_1, \dots, C_m$ ,
- 327 •  $V_L = \{c_j^l \mid 1 \leq j \leq m, \forall l \in L\}$  are nodes corresponding to all possible  
 328 truth assignments of each clause  $C_i$ .

329 *Agents  $\mathcal{E}$  reduction idea:* On each of the nodes in  $V_x \cup V_C$  we place one  
 330 agent with a budget of  $b = 1$ , for a total of  $n + m$  agents. The main idea is to  
 331 initially space the agents by a pairwise distance of 3. We then let agents on  $V_x$   
 332 “pick the value assignment of the variables  $x_i$ ” by walking to their respective  
 333 node in  $V_\ell$ , whereas we let agents on  $V_C$  “pick the truth assignment of the  
 334 clauses  $C_j$ ” by walking to their respective node in  $V_L$ . Then a satisfiable  
 335 assignment of  $\phi$  exists, if and only if the variable agents and the clause  
 336 agents “agree in their choice” which corresponds to an optimum MINDIAM  
 337 configuration  $\mathbf{O}$  of diameter 1. Furthermore, any other configuration should  
 338 have diameter  $\geq 2$ . This gives rise to the

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

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

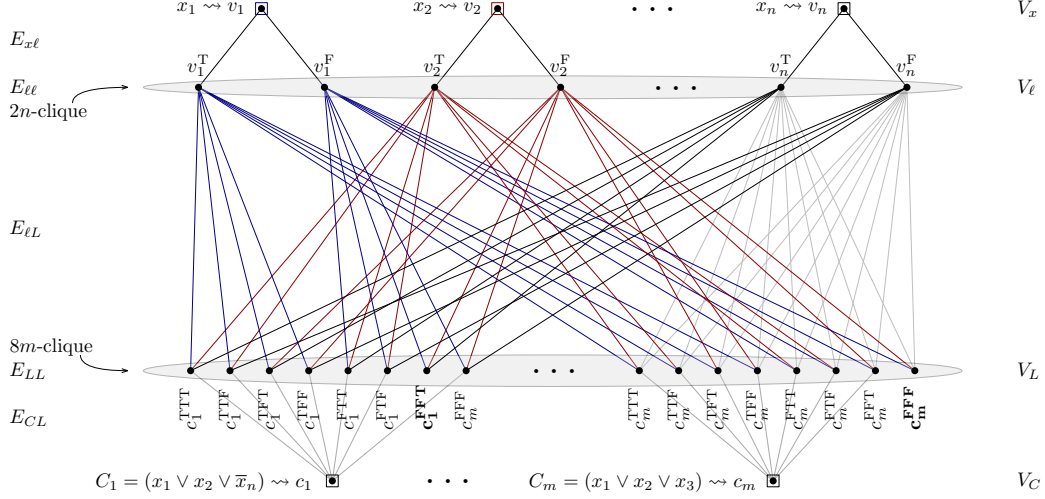


Figure 2: A part of an instance of MINDIAM, constructed from the 3-SAT instance  $C_1 \wedge \dots \wedge C_m$  with variables  $x_1, \dots, x_n$ , displaying the connections between nodes  $v_1, v_2, v_n, c_1$  and  $c_m$ . Notice that nodes  $c_1^{\text{FFT}}$  and  $c_m^{\text{FFF}}$  are not connected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

- 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 •  $E_{\ell L} = \{\{v_i^l, c_j^l\} \mid i \leq n, j \leq m: v_i^l \in \{v_i^T, v_i^F\} \subset V_\ell, c_j^l \in V_L, \text{ such that}$   
346 *either*  $x_i$  does not appear in  $C_j$ ,  
347 *or*  $x_i$  appears in  $C_j$  and  $v_i^l$  agrees with  $c_j^l\}$   
348 are edges connecting unrelated literals and clause truth-assignments,  
349 as well as matching literals and clause truth-assignments.
- 350 •  $E_{\ell\ell} = \{\{u, v\} \mid u, v \in V_\ell\}$  and  $E_{LL} = \{\{u, v\} \mid u, v \in V_L\}$  are edges  
351 pairwise connecting nodes in  $V_\ell$ , and nodes in  $V_L$ , respectively.

352 Figure 2 shows a part of an instance of MINDIAM which is constructed  
353 from an instance of 3SAT as described above. Before giving a proof of The-  
354 orem 5, we argue that no agent would stop in the middle of an edge:

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

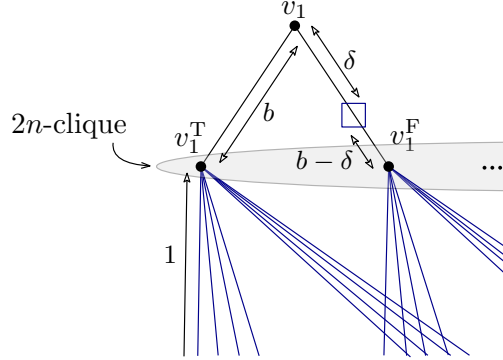


Figure 3: A configuration  $\mathbf{C}$  where agent  $a(v_1)$ , depicted by  $\square$ , stops at distance  $\delta$  from its starting node  $v_1$ : Moving fully to  $v_1^F$  will only decrease the diameter  $\text{Diam}(\mathbf{C})$ .

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

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

374 *Proof. (Theorem 5).* Based on the preceding construction of MINDIAM in-  
 375 stances from 3SAT instances and the structural Lemma 4, we now give a  
 376 proof of Theorem 5.

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

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

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

412 Let us examine the maximum distance of any two agents in this final con-  
 413 figuration. Recall that the set of edges  $E_{CL}$  connects each clause node to  
 414 nodes corresponding to all possible satisfying assignments for this clause. As  
 415 a result, nodes  $c_y$  and  $c_y^{\text{FTF}}$  are not connected by an edge. Moreover, the  
 416 shortest path between nodes  $c_y$  and  $c_y^{\text{FTF}}$  is equal to 2 (via edges in  $E_{CL}$  and  
 417  $E_{LL}$ ). Therefore, agent  $a(c_y)$  cannot reach node  $c_y^{\text{FTF}}$ . Any other node  $c_y^l$ ,  
 418 where  $l \in L \setminus \{\text{FTF}\}$ , to which agent  $a(c_y)$  could relocate, corresponds to

419 a truth assignment to  $x_r$ ,  $x_s$  and  $x_t$  where at least one of the variables has  
420 the opposite value of its assignment. Say that  $a(c_y)$  chooses to move to node  
421  $c_y^{\text{TF}}$ , then  $a(c_y)$  will have a distance of 2 from agent  $a(v_r)$  since  $a(v_r)$  has  
422 moved to node  $v_r^{\text{F}}$ . Recall that node  $v_r^{\text{F}}$  is not connected by an edge to node  
423  $c_y^{\text{TF}}$ , since  $x_r$  appears in  $C_y$  but  $v_r^{\text{F}}$  does not agree with  $c_y^{\text{TF}}$ . Therefore,  
424 agents  $a(c_y)$  and  $a(v_r)$  will have a distance of 2.

425 Since a polynomial-time  $(2-o(1))$ -approximation algorithm for MINDIAM  
426 could distinguish between instances with an optimum solution with diameter  
427 1 and instances with an optimum solution with diameter 2, it would also be  
428 able to decide whether  $\phi$  is satisfiable or not.  $\square$

#### 429 4. Minimizing the Average Distance

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

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

444 *Proof.* Assume for the sake of contradiction that in every optimum configu-  
445 ration  $\mathbf{C} = (c_1, \dots, c_j)$ , there is at least one agent  $j$  which stops on a point  
446  $c_j \notin V \cup \bigcup_i S(i)$ . Define by  $\mathbb{O}$  the set of all optimum solutions, and with  
447  $\mathbb{O}' \subseteq \mathbb{O}$  its subset of configurations with a *minimum* number of agents  $j$   
448 such that  $c_j \notin V \cup \bigcup_i S(i)$ . We denote by  $\mathbf{C}^* = (c_1^*, c_2^*, \dots, c_k^*)$  a configu-  
449 ration with a *maximum* number of agents stopping on any *common point*  
450  $c^* \notin V \cup \bigcup_i S(i)$ , 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  $c^* \notin V$ ,  $c^* \in \text{seg}(e)$  for some edge  $\{u, v\}$ . Denote by  $A_u$  the set of

454 all agents  $j$  for which  $c_j^* \in \text{seg}(e)$  and  $c_j^*$  is between  $u$  and  $c^*$  in  $\text{seg}(e)$ , or for  
 455 which there is a shortest path from  $c_j^*$  to  $c^*$  going through  $u$ . Denote by  $A_v$   
 456 the set of all agents  $j \notin A_= \cup A_u$  (for which, there must be a shortest path  
 457 from  $c_j^*$  to  $c^*$  going through  $v$ , or for which  $c_j^* \in \text{seg}(e)$  and  $c_j^*$  is between  $v$   
 458 and  $c^*$  in  $\text{seg}(e)$ ).

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

- 462 • The pairwise distances in  $A_=$ , in  $A_u$ , and in  $A_v$  individually and the  
 463 distances between agents in  $A_u$  and agents in  $A_v$  stay the same.
- 464 • The distances between agents in  $A_=$  and in  $A_u$  decrease by  $\varepsilon$  each.
- 465 • The distances between agents in  $A_=$  and in  $A_v$  increase by at most  $\varepsilon$ .

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

477 **Theorem 6** (MinSum,  $2(1 - \frac{1}{k})$ -apx). *Algorithm 3 is a polynomial-time*  
 478  *$2(1 - \frac{1}{k})$ -approximation algorithm (and the approximation ratio is tight).*

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

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

---

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


---

**Input:** An instance  $\langle G, k, (p_i)_{i=1,\dots,k}, (b_i)_{i=1,\dots,k} \rangle$ .

**Output:** Configuration  $\mathbf{C}$  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**
  - 2:     Compute  $\mathbf{C}^p := (c_1^p, \dots, c_k^p)$ ,
  - 3:         where either  $c_i^p = p$  if  $d(p_i, p) \leq b_i$ , or otherwise
  - 4:          $c_i^p \in \arg \min \{d(p, c_i) \mid c_i \in B(i) \cup S(i)\}$  (breaking ties arbitrarily).
  - 5:     Compute  $Sum(\mathbf{C}^p)$ .
  - 6: **end for**
  - 7: Return  $\arg \min_{\mathbf{C}^p: p \in V \cup \bigcup_i S(i)} Sum(\mathbf{C}^p)$ .
- 

485 Consider now the configuration  $\mathbf{C}^o = (c_1^o, \dots, c_k^o)$  which Algorithm 3 com-  
486 puted 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,  
487 we have  $s^* \leq s^o$ . Furthermore,  $o$  is reachable by at least one agent  $a$ , thus  
488 by Step 2 we also have  $c_a^o = o$ . Finally, as Step 2 moves agents as close to  
489  $o$  as possible, we have  $d(o, c_j^o) \leq d(o, o_j)$ . Using the triangle inequality, we  
490 rewrite  $s^o$  to get

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

491 (*Lower bound*). To see that the above analysis is tight, we construct in-  
492 stances  $\langle G, k, (p_i)_{i=1,\dots,k}, (b_i)_{i=1,\dots,k} \rangle$  with  $k$  agents of uniform budget  $b = 1$ ,  
493 for which the tie-breaking in Line 3 of Algorithm 3 leads to a configuration  
494  $\mathbf{C}$  with  $Sum(\mathbf{C}) = 2(1 - \frac{1}{k}) \cdot Sum(\mathbf{O})$ . An example of such an instance with  
495 three agents of uniform budget  $b = 1$  is given in Figure 4. We now give a

496 construction for an arbitrary number of agents  $k$  and an evaluation for each  
 497  $p \in V \cup \bigcup_i S(i)$ :

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

- 508 •  $p = u_{a,a}$ : Agent  $a$  will move to  $c_a^p = u_{a,a}$ , while every other agent  $j$  is  
 509 indifferent between  $u_{j,j}$  and  $u_{j,a}$  and thus might move to  $c_j^p = u_{j,a}$ . In  
 510 this case, agents  $i \neq j$  have distance  $d(c_i^p, c_j^p) = 1$  if  $i = a$  or  $j = a$  and  
 511  $d(c_i^p, c_j^p) = 2$  otherwise, giving  $Sum(\mathbf{C}^p) = 2(k-1) \cdot 1 + (k-1)(k-2) \cdot 2 =$   
 512  $2(k-1)^2 = 2 \frac{k-1}{k} k(k-1) = 2(1 - \frac{1}{k}) \cdot Sum(\mathbf{O})$ .
- 513 •  $p = u_{a,b}$  (for  $a \neq b$ ): Agent  $a$  will move to  $c_a^p = u_{a,b}$  and agent  $b$  to  
 514  $c_b^p = u_{b,b}$ . Every other agent  $j$  is indifferent between  $u_{j,j}$  and  $u_{j,b}$ , having  
 515 both distance 2 to  $u_{a,b}$ . In case they each choose  $c_j^p = u_{j,b}$ , we get a  
 516 configuration  $\mathbf{C}^p$  which is symmetric to the previous case.
- 517 •  $p = p_a$ : Agent  $a$  will stay on  $p_a$ , while every other agent  $j$  is indifferent  
 518 between  $u_{j,j}$  and  $u_{j,a}$ . In case they each choose  $c_j^p = u_{j,a}$ , we get a  
 519 configuration  $\mathbf{C}^p$  where any two agents  $i, j$  have distance 2., giving  
 520  $Sum(\mathbf{C}^p) = 2k(k-1) = 2 \cdot Sum(\mathbf{O})$ .

521 Hence the approximation analysis of Algorithm 3 is tight. □

522 **Theorem 7.** *There is no FPTAS for MINSUM, unless  $P = NP$ .*

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

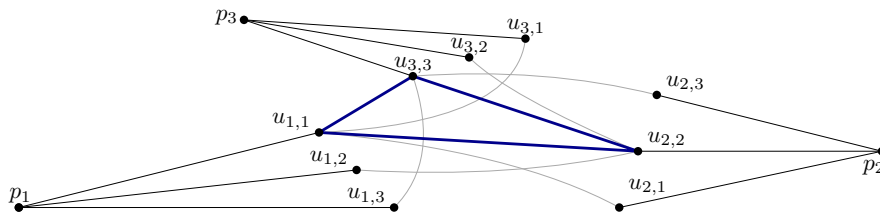


Figure 4: Lower bound construction for Algorithm 3 with  $k = 3$  agents: The only optimum MINSUM configuration is  $\mathbf{O} = (u_{1,1}, u_{2,2}, u_{3,3})$  with  $Sum(\mathbf{O}) = k(k-1)$ . In the tie-breaking in Line 4, agents  $i$  will generally be indifferent between the options  $u_{i,i}$ 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

530 Summing up the pairwise distances we get for (i) that  $Sum(\mathbf{C}^*) = k(k -$   
531  $1)$ , while for (ii) we have  $Sum(\mathbf{C}) \geq k(k - 1) + 1$ . The existence of an  
532 FPTAS, using  $\varepsilon \leq \frac{1}{k^2}$ , means that we can approximate  $Sum(\mathbf{C}^*)$  to within  
533  $(1 + \frac{1}{k^2}) \cdot k(k - 1) = k^2 - k + 1 - \frac{1}{k} < k(k - 1) + 1 \leq Sum(\mathbf{C})$ . Hence we could  
534 distinguish the existence of a solution  $\mathbf{C}^*$  from any other solution and thus  
535 decide satisfiability of  $\phi$  in time  $poly(k, \frac{1}{1/k^2}) = poly(k)$ , in contradiction to  
536 the assumption  $P \neq NP$ .  $\square$

## 537 5. Additive Approximation and Conclusion

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

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

551 **Theorem 8.** *There exists no deterministic polynomial-time additive*  
552  *$+(2 \max_i b_i - o(1))$ -approximation algorithm for MINDIAM with node stops,*  
553 *and no deterministic polynomial-time additive  $+(\frac{4}{3} \max_i b_i - o(1))$ -approximation*

554 *mation algorithm for MINDIAM with arbitrary stops, unless  $P = NP$ .*  
 555 *(Proof at end of Section.)*

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

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

569 We finish by proving the additive approximation hardness results in Theo-  
 570 rem 8 by a similar reduction from 3SAT as the one given for the multiplicative  
 571  $(2 - o(1))$ -approximation hardness of MINDIAM. Instead of a self-contained  
 572 proof, we describe all necessary adaptations we make in the proof of Theorem 5.

573 *Proof.* Given an arbitrary 3SAT formula  $\phi$  with  $n$  variables and  $m$  clauses, we  
 574 first make a one-to-one copy  $\phi'$  of all its variables and all its clauses. Clearly,  
 575  $\phi$  is satisfiable if and only if  $\Phi := \phi \wedge \phi'$  is satisfiable. We now construct  
 576 an instance  $\langle G, k, (p_i)_{i=1, \dots, k}, (b)_{i=1, \dots, k} \rangle$  with  $k = 2n + 2m$  agents of uniform  
 577 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:  $\Phi$  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  $\phi$  or of  $\phi'$ .

584  $\Rightarrow$  We first show that if  $\Phi$  is satisfiable, then there exists an optimum  
 585 configuration  $\mathbf{C}$  of diameter  $Diam(\mathbf{C}) = 2b$ . This follows immediately from  
 586 the proof of Theorem 5. Since we increased the weight of all relevant edges  
 587 by a factor of  $2b$ , we get with the same reasoning an optimum configuration  
 588  $\mathbf{C}$  of diameter  $Diam(\mathbf{C}) = 2b$  (instead of the previously shown 1).

589  $\Leftarrow$  We now show that if  $\Phi$  is not satisfiable, then every solution to MIN-  
590 DIAM with node stops is of size greater than or equal to  $2b + 2b = 4b$ .  
591 The difficulty lies in the fact that Lemma 4 is no longer valid, since the  
592 node triples  $v_i, v_i^T, v_i^F$  no longer form an equilateral triangle; instead, we have  
593  $\omega(\{v_i^T, v_i^F\}) = 2b$ , while  $v_i$  is connected to  $v_i^T, v_i^F$  with two edges of weight  $b$ .  
594 Similarly, edges in  $E_{LL}$  have weight  $2b$  while edges in  $E_{CL}$  have weight  $b$ .  
595 We now observe the following: If in a configuration  $\mathbf{C}$  there are two agents  
596  $i, j$  which stay on their starting position  $p_i, p_j$ , then they must have a dis-  
597 tance of  $d(p_i, p_j) \geq b + 2b + b = 4b$ . Otherwise, there is at most one agent  
598 staying at its starting position. Thus in at least one of the subgraphs induced  
599 by  $\phi$  and  $\phi'$ , respectively, we can assume that all agents move to nodes in  
600  $V_\ell \cup V_L$ . Repeating the arguments given in the proof of Theorem 5, since  $\phi$   
601 and  $\phi'$  are not satisfiable, there must be two agents in  $V_\ell$  and  $V_L$  which are  
602 not connected by an edge and thus have distance at least  $2b + 2b = 4b$ .

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

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

- 626 [1] L. Pagli, G. Prencipe, G. Viglietta, Getting close without touching:  
627 Near-gathering for autonomous mobile robots, *Distributed Computing*  
628 28 (5) (2015) 333–349. doi:10.1007/s00446-015-0248-5.
- 629 [2] P. Flocchini, G. Prencipe, N. Santoro, *Distributed Computing by Mo-*  
630 *bile Entities*, *Current Research in Moving and Computing*, Vol. 11340  
631 of *Lecture Notes in Computer Science*, Springer, 2019. doi:10.1007/  
632 978-3-030-11072-7.
- 633 [3] M. Cieliebak, P. Flocchini, G. Prencipe, N. Santoro, *Distributed com-*  
634 *puting by mobile robots: Gathering*, *SIAM Journal on Computing* 41 (4)  
635 (2012) 829–879. doi:10.1137/100796534.
- 636 [4] P. Flocchini, G. Prencipe, N. Santoro, P. Widmayer, *Gathering of asyn-*  
637 *chronous robots with limited visibility*, *Theoretical Computer Science*  
638 337 (1-3) (2005) 147–168. doi:10.1016/j.tcs.2005.01.001.
- 639 [5] J. Lin, A. S. Morse, B. D. O. Anderson, *The multi-agent rendezvous*  
640 *problem. parts 1 and 2.*, *SIAM Journal on Control and Optimization*  
641 46 (6) (2007) 2096–2147.
- 642 [6] R. Cohen, D. Peleg, *Convergence properties of the gravitational algo-*  
643 *rithm in asynchronous robot systems*, *SIAM Journal on Computing* 34  
644 (2005) 1516–1528. doi:10.1137/S0097539704446475.
- 645 [7] J. Czyzowicz, A. Labourel, A. Pelc, *How to meet asynchronously (al-*  
646 *most) everywhere*, *ACM Transactions on Algorithms* 8 (4) (2012) 37:1–  
647 37:14. doi:10.1145/2344422.2344427.
- 648 [8] A. Pelc, *Deterministic rendezvous in networks: A comprehensive survey*,  
649 *Networks* 59 (3) (2012) 331–347. doi:10.1002/net.21453.
- 650 [9] S. Alpern, S. Gal, *The Theory of Search Games and Rendezvous*,  
651 Kluwer, 2003. doi:10.1007/b100809.
- 652 [10] S. Cicerone, G. D. Stefano, A. Navarra, *Gathering of robots on meeting-*  
653 *points: feasibility and optimal resolution algorithms*, *Distributed Com-*  
654 *puting* 31 (1) (2018) 1–50. doi:10.1007/s00446-017-0293-3.
- 655 [11] M. Betke, R. L. Rivest, M. Singh, *Piecemeal learning of an unknown*  
656 *environment*, *Machine Learning* 18 (2) (1995) 231–254. doi:10.1007/  
657 BF00993411.

- 658 [12] B. Awerbuch, M. Betke, R. L. Rivest, M. Singh, Piecemeal graph explo-  
659 ration by a mobile robot, *Information and Computation* 152 (2) (1999)  
660 155–172. doi:10.1006/inco.1999.2795.
- 661 [13] C. A. Duncan, S. G. Kobourov, V. S. A. Kumar, Optimal Constrained  
662 Graph Exploration, *ACM Transactions on Algorithms* 2 (3) (2006) 380–  
663 402. doi:10.1145/1159892.1159897.
- 664 [14] P. Fraigniaud, L. Gasieniec, D. R. Kowalski, A. Pelc, Collective tree  
665 exploration, *Networks* 48 (3) (2006) 166–177. doi:10.1002/net.20127.
- 666 [15] M. Dynia, M. Korzeniowski, C. Schindelhauer, Power-Aware Collective  
667 Tree Exploration, in: 19th International Conference on Architecture  
668 of Computing Systems ARCS’06, 2006, pp. 341–351. doi:10.1007/  
669 11682127\_24.
- 670 [16] M. Dynia, J. Łopuszański, C. Schindelhauer, Why robots need maps,  
671 in: 14th International Colloquium on Structural Information and Com-  
672 munication Complexity, SIROCCO’07, 2007, pp. 41–50. doi:10.1007/  
673 978-3-540-72951-8\_5.
- 674 [17] S. Das, D. Dereniowski, C. Karousatou, Collaborative Exploration  
675 by Energy-Constrained Mobile Robots, in: 22th International Col-  
676 loquium on Structural Information and Communication Complexity  
677 SIROCCO’15, 2015, pp. 357–369. doi:10.1007/978-3-319-25258-2\_  
678 25.
- 679 [18] D. Dereniowski, Y. Disser, A. Kosowski, D. Pająk, P. Uznański, Fast col-  
680 laborative graph exploration, *Information and Computation* 243 (2015)  
681 37–49. doi:https://doi.org/10.1016/j.ic.2014.12.005.
- 682 [19] E. D. Demaine, M. Hajiaghayi, H. Mahini, A. S. Sayedi-Roshkhar,  
683 S. Oveisgharan, M. Zadimoghaddam, Minimizing movement, *ACM*  
684 *Transactions on Algorithms* 5 (3) (2009) 30:1–30:30. doi:10.1145/  
685 1541885.1541891.
- 686 [20] E. D. Demaine, M. Hajiaghayi, D. Marx, Minimizing movement: Fixed-  
687 parameter tractability, *ACM Transactions on Algorithms* 11 (2) (2014)  
688 14:1–14:29. doi:10.1145/2650247.

- 689 [21] D. Bilò, Y. Disser, L. Gualà, M. Mihalák, G. Proietti, P. Widmayer,  
690 Polygon-Constrained Motion Planning Problems, in: 9th International  
691 Symposium on Algorithms and Experiments for Sensor Systems, Wire-  
692 less Networks and Distributed Robotics ALGOSENSORS'13, 2013, pp.  
693 67–82. doi:10.1007/978-3-642-45346-5\_6.
- 694 [22] J. Anaya, J. Chalopin, J. Czyzowicz, A. Labourel, A. Pelc, Y. Vaxès,  
695 Convergecast and Broadcast by Power-Aware Mobile Agents, *Algorith-*  
696 *mica* 74 (1) (2016) 117–155. doi:10.1007/s00453-014-9939-8.
- 697 [23] J. Chalopin, S. Das, M. Mihalák, P. Penna, P. Widmayer, Data Delivery  
698 by Energy-Constrained Mobile Agents, in: 9th International Symposium  
699 on Algorithms and Experiments for Sensor Systems, Wireless Networks  
700 and Distributed Robotics ALGOSENSORS'13, 2013, pp. 111–122. doi:  
701 10.1007/978-3-642-45346-5\_9.
- 702 [24] J. Chalopin, R. Jacob, M. Mihalák, P. Widmayer, Data Delivery by  
703 Energy-Constrained Mobile Agents on a Line, in: 41st International Col-  
704 loquium on Automata, Languages, and Programming ICALP'14, 2014,  
705 pp. 423–434. doi:10.1007/978-3-662-43951-7\_36.
- 706 [25] A. Bärtschi, J. Chalopin, S. Das, Y. Disser, B. Geissmann, D. Graf,  
707 A. Labourel, M. Mihalák, Collaborative Delivery with Energy-  
708 Constrained Mobile Robots, in: 23rd International Colloquium on Struc-  
709 tural Information and Communication Complexity SIROCCO'16, 2016,  
710 pp. 258–274. doi:10.1007/978-3-319-48314-6\_17.
- 711 [26] A. Bärtschi, J. Chalopin, S. Das, Y. Disser, B. Geissmann,  
712 D. Graf, A. Labourel, M. Mihalák, Collaborative delivery with energy-  
713 constrained mobile robots, *Theoretical Computer Science*To appear. See  
714 also SIROCCO'16 (2017). doi:10.1016/j.tcs.2017.04.018.
- 715 [27] J. Czyzowicz, K. Diks, J. Moussi, W. Rytter, Communication Prob-  
716 lems for Mobile Agents Exchanging Energy, in: 23rd International  
717 Colloquium on Structural Information and Communication Complex-  
718 ity SIROCCO'16, 2016. doi:10.1007/978-3-319-48314-6\_18.
- 719 [28] E. Bampas, S. Das, D. Dereniowski, C. Karousatou, Collaborative deliv-  
720 ery by energy-sharing low-power mobile robots, in: International Sym-  
721 posium on Algorithms and Experiments for Sensor Systems, Wireless

722 Networks and Distributed Robotics ALGOSENSORS'17, 2017, pp. 1–  
723 12. doi:10.1007/978-3-319-72751-6\_1.