Distance-2-Dispersion with Termination by a Strong Team

Barun Gorain, Tanvir Kaur, Kaushik Mondal

Department of Mathematics
Indian Institute of Technology Ropar

CALDAM 2024

Outline

Introduction

The Dispersion Problem

Given an initial configuration of $k \leq n$ mobile robots on an n node anonymous graph, the robots reposition to reach a configuration where each node contains at most one robot.

Introduction

The Dispersion Problem

Given an initial configuration of $k \leq n$ mobile robots on an n node anonymous graph, the robots reposition to reach a configuration where each node contains at most one robot.

The Distance-2-Dispersion (D-2-D) Problem

Given a set of $k \geq 1$ robots placed arbitrarily in a port-labeled graph G with n nodes and m edges, the robots need to achieve a configuration by the end of the algorithm where each robot needs to settle at some node satisfying the following two conditions:

(a) no two adjacent nodes can be occupied by settled robots

(b) a robot can settle in a node where there is already a settled robot only if no more unoccupied node is present for the robot to settle satisfying condition (a)

Introduction

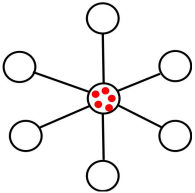


Figure: Initial Configuration

Introduction

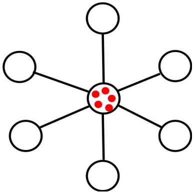


Figure: Initial Configuration

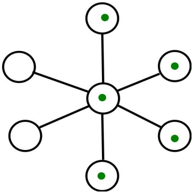


Figure: Dispersed Configuration

Introduction

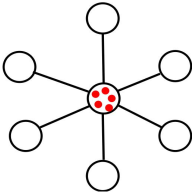


Figure: Initial Configuration

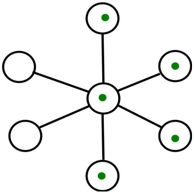


Figure: Dispersed Configuration

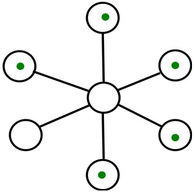


Figure: D-2-D Configuration

Existing Results

Results given by Kaur et al.*

- Lower Bound on the number of rounds: $\Omega(m\Delta)$
- Algorithm for **Rooted Configuration:**
  - Time Complexity: $2\Delta(8m - 3n + 3)$
  - Memory Complexity: $O(\log n)$
  - No global knowledge required
  - Achieves termination

*Kaur, T., Mondal, K. Distance-2-Dispersion: Dispersion with Further Constraints. NETYS. vol 14067. 2023.

Existing Results

Results given by Chand et al.[†]

- They present algorithms that compute a minimal dominating set that is also a maximal independent set with
  - Rooted Configuration: $O(m)$ time complexity
  - Arbitrary Configuration: $O(\ell\Delta\log(\lambda) + n\ell + m)$ time complexity, where Δ is the maximum degree of the graph and λ is the maximum ID-length of the robots when the robots are placed arbitrarily at ℓ nodes initially

- Both algorithms require prior knowledge of all the global parameters, including Δ, λ, and n

[†]Chand, P.K., Molla, A.R., Sivasubramaniam, S. Run for Cover: Dominating Set via Mobile Agents. ALGOWIN. vol 14061. 2023.

Our Motive:

- Study the power of a strong team
- To provide a solution for arbitrary initial configuration
- To achieve a solution without any prior knowledge of the global parameters

The Set-Up

- The Graph G:
  - n nodes, m edges
  - arbitrary
  - anonymous
  - port-labeled
  - zero-storage
- Mobile Robots:
  - $k > n$
  - unique id: $[1, n^c]$
  - face-to-face communication
  - has memory
- Synchronous rounds

Our Contribution

Configuration	Time Complexity	Memory Complexity	Global Knowledge
Rooted	$O(m)$	$O(\log n)$	No
Arbitrary	$O(pm)^*$	$O(\log n)$	No

$^*p$ is the number of nodes containing robots in the initial configuration

Our Contribution

Configuration	Time Complexity	Memory Complexity	Global Knowledge
Rooted	$O(m)$	$O(\log n)$	No
Arbitrary	$O(pm)^*$	$O(\log n)$	No

- Both algorithms achieve termination without any prior global knowledge

$^*p$ is the number of nodes containing robots in the initial configuration

The Algorithm for Rooted Configuration

High-Level Idea

- Phase 1: Dispersion on the graph
- Phase 2: Convert the dispersed configuration into the desired D-2-D one

The Algorithm for Rooted Configuration

- At least one robot finds no position to settle at an empty node in phase 1. Thus, it settles at the root node
- This indicates the completion of phase 1 and thus phase 2 is initiated

Description of Phase 2:

- The robot r_{min} updates $r_{min}.phase = 2$
- r_{min} begins the traversal of the graph via the tree-edges
- To gain information of the tree edges, dedicated rounds are employed

The Algorithm for Rooted Configuration

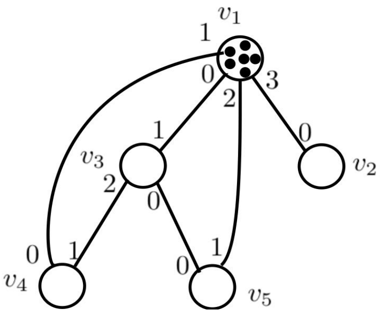
Dedicated-round	Function performed
1-dedicated round	The robots execute one round of phase 1
2-dedicated round	The robots r_i settled at an even distance from the root node move through $r_i.parent$
3-dedicated round	The robots settled at an odd distance from the root node get the information of the ports leading to their children in the DFS tree
4-dedicated round	The robots r_i settled at an odd distance from the root node move through $r_i.parent$
5-dedicated round	The robots settled at an even distance from the root node get the information of the ports leading to their children in the DFS tree
6-dedicated round	The robots execute one round of phase 2

The Algorithm for Rooted Configuration

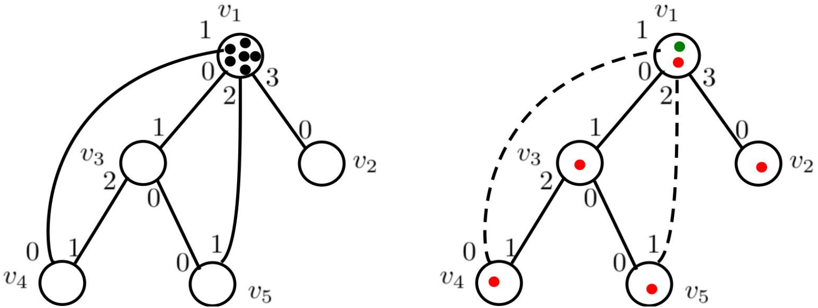
The function of the robot r_{min}

- The robot r_{min} moves through the child pointer of each settled robot unless it reaches a robot r_u with no child.
- The robot r_u decides to either settle at its original position or to move to a neighboring node that has already decided to settle there.
- The robot r_{min}, on the other hand, decides to wait for $\delta(v)$ rounds, where $\delta(v)$ is the degree of the node v where r_u was originally present.

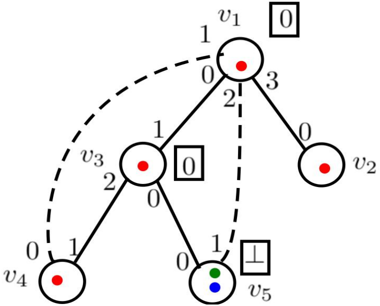
Example



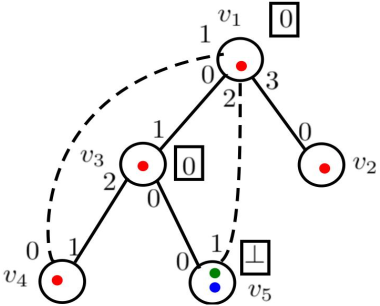
Example



Example

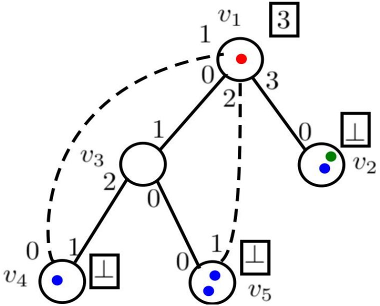


Example

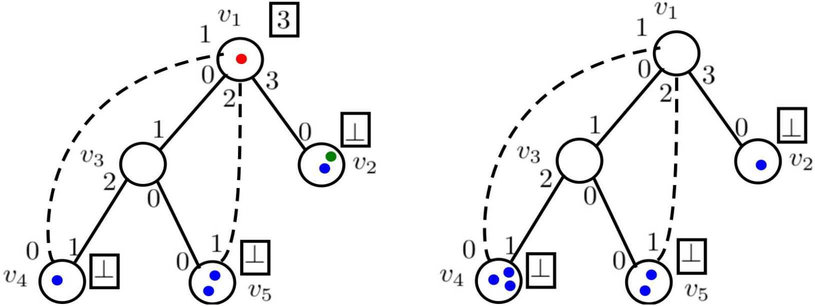


v_1
0
1
0
3
2
1
0
v_3
0
v_2
2
0
0
1
1
⊥
0
v_4
0
v_5
v_1
0
1
0
3
2
1
0
v_3
2
v_2
2
0
0
1
1
⊥
0
v_4
⊥
0
v_5

Example



Example



Theorem

Let G be a port-labeled, undirected, and connected graph with n nodes, m edges, and maximum degree Δ. Let $k(> n)$ robots be initially placed at a single node of the graph G and robots have no prior knowledge of any of the global parameters. The robots achieve D-2-D on the graph with termination in $O(m)$ rounds with $O(\log n)$ memory per robot.

The Algorithm for Arbitrary Configuration

The Initial Configuration:

- $k > n$ robots are initially placed at p distinct nodes, where $1 < p \leq n$

High-Level Idea of the Algorithm:

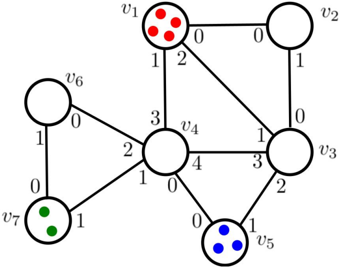
- The Algorithm runs in two phases
- Phase 1: Dispersion on the graph
- Phase 2: Convert the dispersed configuration into the desired D-2-D one
- Target: Merge DFSs in such a way that at the end there exists exactly one DFS tree in the graph. Finally, convert the dispersed configuration into D-2-D one

The Algorithm for Arbitrary Configuration

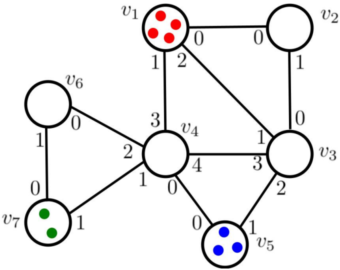
Overview

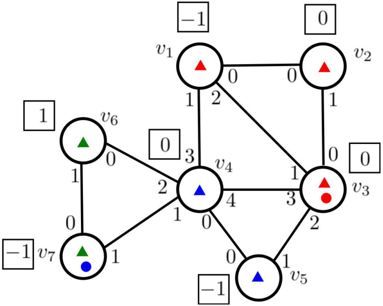
- Each DFS traversal is associated with a *label* that corresponds to the minimum ID robot of the respective group
- When groups of unsettled robots from two or more DFSs meet at a common node: the minimum id robot from the minimum labeled DFS (say l) settles at the current node and the unsettled robots from the other DFSs join the DFS l
- If the unsettled robots of DFS l meets with a settled robot of DFS m:
    - If $l > m$: The unsettled robots of DFS l follows to reach the head of DFS m and extend its DFS further. During this, if the unsettled robots meet another robot with even smaller *label*, then they are subsumed into that DFS
    - If $l < m$: The settled robot of DFS m becomes a part of the DFS l by changing its *label* and the parent pointer

An Illustrative Example

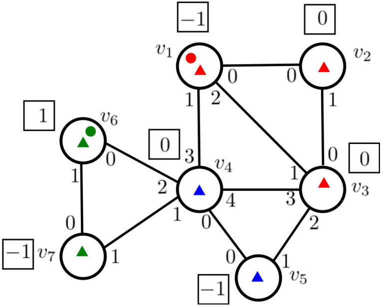


An Illustrative Example

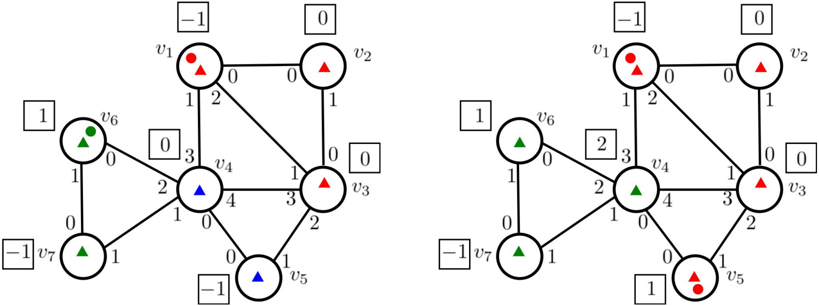




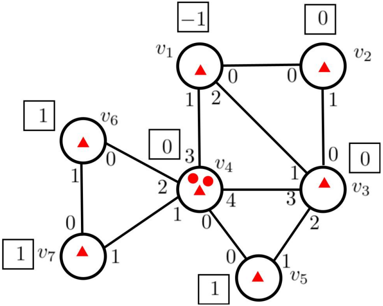
An Illustrative Example



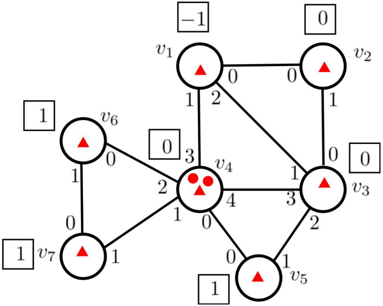
An Illustrative Example

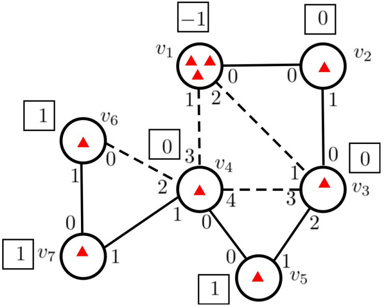


An Illustrative Example



An Illustrative Example





Theorem

Let G be a port-labeled, undirected, and connected graph with n nodes, m edges, and maximum degree Δ. Let $k(k > n)$ robots be arbitrarily placed on it at p multiplicity nodes and the robots have no prior knowledge of any of the global parameters n, m, k, p, and Δ. Our algorithm solves D-2-D with termination in $O(pm)$ rounds, and $O(\log n)$ memory is required by each robot to run the algorithm.

Future Works

- It would be interesting to do a lower-bound study of D-2-D starting with $k(> n)$ robots
- Analysing and improving the time and memory complexity to solve the problem of D-2-D with $k \leq n$ mobile robots
- Solving D-2-D in the presence of faults is another direction of further study

References I

J. Augustine and W. K. Moses Jr.
Dispersion of mobile robots: A study of memory-time trade-offs.
In *ICDCN*, pages 1:1–1:10, 2018.

P. K. Chand, A. R. Molla, and S. Sivasubramaniam.
Run for cover: Dominating set via mobile agents.
CoRR, abs/2309.02200, 2023.

T. Kaur and K. Mondal.
Distance-2-dispersion: Dispersion with further constraints.
In *NETYS*, pages 157–173, 2023.

A. D. Kshemkalyani and F. Ali.
Efficient dispersion of mobile robots on graphs.
In *ICDCN*, pages 218–227, 2019.

References II

A. D. Kshemkalyani, A. R. Molla, and G. Sharma.
Fast dispersion of mobile robots on arbitrary graphs.
In *ALGOSENSORS*, pages 23–40, 2019.

A. D. Kshemkalyani and G. Sharma.
Near-optimal dispersion on arbitrary anonymous graphs.
In *OPODIS*, pages 8:1–8:19, 2021.

A. R. Molla, K. Mondal, and W. K. Moses Jr.
Fast deterministic gathering with detection on arbitrary graphs: The power of many robots.
In *IPDPS*, pages 47–57, 2023.

D. Pattanayak, G. Sharma, and P. S. Mandal.
Dispersion of mobile robots tolerating faults.
In *ICDCN*, pages 133–138, 2021.

References III

S. Pramanick, S. V. Samala, D. Pattanayak, and P. S. Mandal.
Distributed algorithms for filling mis vertices of an arbitrary graph by myopic luminous robots.
Theoretical Computer Science, 978:114187, 2023.

T. Shintaku, Y. Sudo, H. Kakugawa, and T. Masuzawa.
Efficient dispersion of mobile agents without global knowledge.
In *SSS*, pages 280–294, 2020.

Y. Sudo, M. Shibata, J. Nakamura, Y. Kim, and T. Masuzawa.
Near-linear time dispersion of mobile agents.
CoRR, abs/2310.04376, 2023.


Thank you


thank you