

2018

## Mixed-Weight Open Locating-Dominating Sets

Robin M. Givens

*William & Mary - Arts & Sciences*, [robinmgivens@gmail.com](mailto:robinmgivens@gmail.com)

Follow this and additional works at: <https://scholarworks.wm.edu/etd>



Part of the [Computer Sciences Commons](#)

---

### Recommended Citation

Givens, Robin M., "Mixed-Weight Open Locating-Dominating Sets" (2018). *Dissertations, Theses, and Masters Projects*. William & Mary. Paper 1550153794.

<http://dx.doi.org/10.21220/s2-4479-zx78>

This Dissertation is brought to you for free and open access by the Theses, Dissertations, & Master Projects at W&M ScholarWorks. It has been accepted for inclusion in Dissertations, Theses, and Masters Projects by an authorized administrator of W&M ScholarWorks. For more information, please contact [scholarworks@wm.edu](mailto:scholarworks@wm.edu).

Mixed-Weight Open Locating-Dominating Sets

Robin M. Givens

College of William & Mary  
Williamsburg, VA

Bachelor of Science, University of Richmond, Richmond, VA, 2006  
Master of Science, College of William & Mary, Williamsburg, VA 2014

A Dissertation presented to the Graduate Faculty  
of The College of William & Mary in Candidacy for the Degree of  
Doctor of Philosophy

Department of Computer Science

College of William & Mary  
August 2018



## APPROVAL PAGE

This Dissertation is submitted in partial fulfillment of  
the requirements for the degree of

Doctor of Philosophy



---

Robin M. Givens

Approved by the Committee, May 11, 2018



---

Committee Chair

Professor Weizhen Mao, Computer Science  
College of William & Mary



---

Chancellor Professor Rex Kincaid, Mathematics  
College of William & Mary



---

Associate Professor Robert Michael Lewis, Computer Science  
College of William & Mary



---

Professor Andreas Stathopoulos, Computer Science  
College of William & Mary



---

Associate Professor Gexin Yu, Mathematics  
College of William & Mary

## ABSTRACT

The detection and location of issues in a network is a common problem encompassing a wide variety of research areas. Location-detection problems have been studied for wireless sensor networks and environmental monitoring, microprocessor fault detection, public utility contamination, and finding intruders in buildings. Modeling these systems as a graph, we want to find the smallest subset of nodes that, when sensors are placed at those locations, can detect and locate any anomalies that arise. One type of set that solves this problem is the open locating-dominating set (OLD-set), a set of nodes that forms a unique and nonempty neighborhood with every node in the graph.

For this work, we begin with a study of OLD-sets in circulant graphs. Circulant graphs are a group of regular cyclic graphs that are often used in massively parallel systems. We prove the optimal OLD-set size for two circulant graphs using two proof techniques: the discharging method and Hall's Theorem.

Next we introduce the mixed-weight open locating-dominating set (mixed-weight OLD-set), an extension of the OLD-set. The mixed-weight OLD-set allows nodes in the graph to have different weights, representing systems that use sensors of varying strengths. This is a novel approach to the study of location-detection problems.

We show that the decision problem for the minimum mixed-weight OLD-set, for any weights up to positive integer  $d$ , is NP-complete. We find the size of mixed-weight OLD-sets in paths and cycles for weights 1 and 2. We consider mixed-weight OLD-sets in random graphs by providing probabilistic bounds on the size of the mixed-weight OLD-set and use simulation to reinforce the theoretical results.

Finally, we build and study an integer linear program to solve for mixed-weight OLD-sets and use greedy algorithms to generate mixed-weight OLD-set estimates in random geometric graphs. We also extend our results for mixed-weight OLD-sets in random graphs to random geometric graphs by estimating the probabilistic upper bound for the size of the set.

## TABLE OF CONTENTS

Acknowledgments	iv
Dedication	v
List of Tables	vi
List of Figures	ix
1 Introduction	2
1.1 Related Work and Motivation . . . . .	3
1.2 Open Locating-Dominating Sets and Related Problems . . . . .	5
1.2.1 Definitions . . . . .	5
1.2.2 Related Problems . . . . .	8
1.2.3 Focus . . . . .	9
1.3 Outline and Overview . . . . .	10
2 Open Locating-Dominating Sets in Circulant Graphs	14
2.1 Circulant Graphs and Definitions . . . . .	15
2.2 The Discharging Method . . . . .	17
2.3 The Matching Method . . . . .	21
2.4 Discussion . . . . .	27
3 Mixed-Weight Open Locating-Dominating Sets	28
3.1 Definitions and Properties . . . . .	29
3.2 A Similar Problem: Identifying Codes . . . . .	33

3.3	Discussion . . . . .	33
4	Problems and NP-Completeness	35
4.1	Minimum Size of Mixed-weight OLD-set . . . . .	36
4.2	Minimizing the Size of a Mixed-weight OLD-set . . . . .	40
4.3	Minimizing the Total Weight of a Mixed-weight OLD-set . . . . .	42
4.4	Discussion . . . . .	44
5	Mixed-Weight OLD-sets in Paths and Cycles	45
5.1	Mixed-Weight OLD-sets in Paths . . . . .	45
5.1.1	Lower Bound for Paths with Weight 1 and 2 Nodes . . . . .	46
5.1.2	Paths with All Weight 2 Nodes . . . . .	47
5.1.3	Paths with Weight 1 and 2 Nodes . . . . .	48
5.2	Mixed-Weight OLD-sets in Cycles . . . . .	50
5.2.1	Lower Bound for Cycles with Weight 1 and 2 Nodes . . . . .	50
5.2.2	Cycles with All Weight 2 Nodes . . . . .	51
5.2.3	Cycles with Weight 1 and 2 Nodes . . . . .	52
5.3	Discussion . . . . .	54
6	Mixed-Weight OLD-Sets in Random Graphs	55
6.1	Mixed-Weight OLD-set Bounds in Random Graphs . . . . .	56
6.2	Simulation Results for Random Graphs . . . . .	60
6.3	Discussion . . . . .	63
7	An Integer Linear Program for Mixed-Weight OLD-sets	64
7.1	ILP Formulation . . . . .	66
7.2	ILP Results . . . . .	70
7.3	ILP Relaxation . . . . .	71
7.4	Discussion . . . . .	72

8	Greedy Algorithms for Mixed-Weight OLD-sets	74
8.1	Greedy Algorithms . . . . .	75
8.2	Greedy Algorithm Results . . . . .	77
8.3	Greedy Hot Starts for the ILP Model . . . . .	80
8.4	Discussion . . . . .	81
9	Mixed-Weight OLD-sets in Random Geometric Graphs	82
9.1	Average Probability of an Edge in Random Geometric Graphs . . . . .	83
9.2	Estimated Random Geometric Graph Bounds . . . . .	87
9.3	Discussion . . . . .	89
10	Conclusion	90
10.1	Contributions . . . . .	90
10.2	Future Work . . . . .	92
A	Symbol Table	96

## ACKNOWLEDGMENTS

I would like to thank my advisors, Weizhen Mao, Rex Kincaid, and Gexin Yu. Thank you for working with me, and together, to make this happen. Thank you for offering me much patience, guidance, support, and nudges in the right direction.

Thank you to the rest of my committee, Andreas Stathopoulos and R. Michael Lewis, for having an interest in this topic and asking great questions.

Thank you to the Christopher Newport University Physics, Computer Science, and Engineering Department and the Randolph-Macon College Computer Science Department who allowed me to teach during this process.

Thank you to my family, friends, and classmates who continually lifted me up and believed in me. Every bit of encouragement was needed.

A special thank you to my husband, Nathaniel, who proofread my papers even when he didn't understand them and took care our kids for countless extra hours so I could work.

Finally, I would like to thank the hard working Computer Science administration team, Vanessa Godwin, Jacquelyn Johnson, and Dale Hayes.

I would like to dedicate this dissertation  
to my ever-patient husband, Nathaniel,  
to my amazing children, Sophie and Caleb,  
and in memory of my mom, Cathy Eades.

## LIST OF TABLES

5.1	One set of satisfying minimum mixed-weight OLD-sets $S$ for paths with $n = 10\ell + j \geq 8$ nodes. All nodes are weight 2, and all paths have the same initial $4(\ell - 1)$ OLD-set nodes. . . . .	47
5.2	One set of satisfying minimum mixed-weight OLD-sets $S$ for paths with $n = 10\ell + j \geq 8$ nodes that are weight 1 or 2. All paths have the same $3(\ell - 1)$ internal OLD-set nodes. Underlined nodes can be weight 1. . . . .	49
5.3	One set of satisfying minimum mixed-weight OLD-sets $S$ for cycles with $n = 10\ell + j \geq 8$ nodes. All nodes are weight 2, and all cycles have the same initial $4(\ell - 1)$ OLD-set nodes. . . . .	51
5.4	One set of satisfying minimum mixed-weight OLD-sets $S$ for cycles with $n = 10\ell + j \geq 8$ nodes that are weight 1 or 2. All cycles have the same initial $4(\ell - 1)$ OLD-set nodes. Underlined nodes can be weight 1. . . . .	53

6.1	Simulation trends for graphs $G(n, p)$ with probability of a weight 2 node $\rho$ . Both $p$ and $\rho$ were incremented by 0.05 within their range. ‘Connected’ is the percentage of graphs that were connected ‘OLD-set’ is the percentage of graphs that contained a mixed-weight OLD-set. ‘Size’ is the average size of the mixed-weight OLD-set. ‘Min. Bound’ is the average upper bound of the mixed-weight OLD-set size. ‘< Bound’ is the percentage of graphs that had a mixed-weight OLD-set less than the bound. © 2017 IEEE . . . . .	60
7.1	Percentage of graphs with a mixed-weight OLD-set and the average minimum size of the set in random geometric graphs of size $n$ with distance $r$ and probability of weight 2 node $\rho$ . ‘MW-OLD-set’ is the percentage of graphs that had a mixed-weight OLD-set. ‘Avg. Min.’ is the average minimum mixed-weight OLD-set size. . . . .	71
7.2	LP (relaxed ILP) results in random geometric graphs of size $n$ with $r, \rho \in \{0.25, 0.5, 0.75\}$ . The number of nodes with decision variables equal to 0 and 1 (‘ $s_i = 0$ ’ and ‘ $s_i = 1$ ’), and the number of nodes with decision variables equal to, less than, and greater than 0.5 (‘ $s_i = 0.5$ ’, ‘ $s_i < 0.5$ ’ and ‘ $s_i > 0.5$ ’). ‘LP Obj. Value’ is the average objective value from the LP result, and ‘Greedy Size’ is the average size of the mixed-weight OLD-set generated by greedy search through the LP result. © 2018 IEEE . . . . .	72
8.1	Average mixed-weight OLD-set size generated from each greedy algorithm by sort method. © 2018 IEEE . . . . .	78
8.2	Average time in milliseconds to generate OLD-set from greedy algorithm by sort method in Java using an Intel® Core™ i5-6500 CPU @3.2GHz with 6MB L3 cache and 16GB DDR3 RAM. . . . .	79

8.3 Average size of the mixed-weight OLD-set returned by the ILP model (optimal), by the stingy method choosing by largest outgoing ball, and by greedily selecting nodes from the results of the ILP relaxation. ‘Stingy-Diff’ and ‘Relaxed-Diff’ are the percentage differences between the stingy method and the greedy relaxed ILP method, respectively, with the optimal ILP results. © 2018 IEEE . . . . . 80

8.4 Average solve time, in milliseconds, for the ILP in AMPL with Gurobi with no hot start (‘ILP-None’), all nodes in the graph as the hot start (‘ILP-All’), and using the result of the stingy method as the hot start (‘ILP-Stingy’), and the average time to find the stingy hot start in Java using an Intel® Core™ i5-6500 CPU @3.2GHz with 6MB L3 cache and 16GB DDR3 RAM. . . . . 81

9.1 Simulation trends for graphs  $G(n, r)$  with the probability of a weight 2 node  $\rho \in \{0.25, 0.5, 0.75\}$ . Results are averaged across  $\rho$ . ‘OLD-set’ is the percentage of graphs that contained a mixed-weight OLD-set. ‘Size’ is the average size of the mixed-weight OLD-set returned by the ILP. ‘Min. Bound’ is the average estimated minimum upper bound of the mixed-weight OLD-set size. ‘< Bound’ is the percentage of graphs that had a mixed-weight OLD-set with size less than the minimum bound. ‘< 2× Bound’ is the same for twice the minimum bound. © 2018 IEEE . . . . . 88

## LIST OF FIGURES

1.1	The neighborhood of a node is all nodes adjacent to the node, not including the node. The neighborhood of a set is all nodes adjacent to the nodes in the set, not including the set. The nodes under consideration are shown in black, and the nodes in the neighborhood are circled. . . . .	5
1.2	By comparing each neighborhood intersected with the set $S$ , we can determine that $S$ is an OLD-set. The nodes in the OLD-set are shown in black. . . . .	6
1.3	Nodes $x_1$ and $x_2$ share the neighborhood $\{x_3, x_7\}$ , thus there is no OLD-set in the graph. The nodes that share a neighborhood are shown in black and the nodes in their neighborhood are circled. . . .	7
1.4	By comparing the each closed neighborhood intersected with the set $S$ , we can determine that $S$ is an ID-code. The nodes in the ID-code are shown in black. . . . .	8
1.5	By comparing each closed neighborhood intersected with the set $S$ , for nodes not in $S$ , we can determine that $S$ is an LD-set. The nodes in the LD-set are shown in black. . . . .	9
2.1	Examples of circulant graphs drawn in the typical circular way to show global behavior. . . . .	16
2.2	Examples of circulant graphs drawn in a linear way to show local behavior. . . . .	16

2.3	Examples of OLD-sets and $m$ -clusters in circulant graphs. OLD-set nodes and their cluster connections are shown in black. . . . .	17
2.4	Possible 2-clusters $\{x_1, x_2\}$ of OLD-set $S$ in $C_n(1, 2)$ . . . . .	19
2.5	Possible 3-cluster of OLD-set $S$ in $C_n(1, 2)$ . . . . .	20
2.6	Possible 4-cluster and 5-cluster of OLD-set $S$ in $C_n(1, 2)$ . . . . .	20
2.7	Three consecutive nodes of the same parity in $C_n(1, 3)$ that are not in the OLD-set are followed by four consecutive nodes in the OLD-set. . . . .	22
2.8	Consecutive nodes of the same parity in $C_n(1, 3)$ with size $k = 3, 4, 5, 6, 7, 8$ are boxed with one possible minimum OLD-set solution among their neighbors shown in black. . . . .	23
2.9	Two cases for covering the first four nodes from a set of $k$ consecutive nodes of the same parity. . . . .	25
2.10	The set $A = \{x_1, x_2, x_3\}$ is three consecutive nodes of the same parity in $C_n(1, 3)$ . . . . .	25
3.1	When using the open incoming-ball definition, a node with weight greater than 1 becomes the neighbor of other nodes. When using the open outgoing-ball definition, other nodes become the neighbor of a node with weight greater than 1. The effect of the weighted node, $w(x_1) = 2$ , is indicated by dashed arcs. . . . .	30
3.2	Mixed-Weight OLD-set Example: (a) The graph does not have an OLD-set when $w(x) = 1 \forall x$ . (b) Nodes $x_1$ and $x_3$ share open incoming-ball $\{x_2, x_4\}$ . (c) The effect of the weighted node, $w(x_3) = 2$ , is indicated by dashed arcs, and one mixed-weight OLD-set, $\{x_3, x_4, x_5, x_6\}$ , is shown with nodes in black. © 2017 IEEE (d) When $w(x_3) = 2$ , the open incoming-ball is unique for each node, allowing for a mixed-weight OLD-set. © 2017 IEEE . . . . .	32

4.1	A cycle with nodes $\{0, 1, \dots, 9\}$ and weight function $w(x) = 2$ for $x \in \{0, 5\}$ and $w(x) = 1$ otherwise, has a minimum mixed-weight OLD-set of size 6, shown in black. . . . .	36
4.2	Subgraph $\Delta$ for the MW-OLD decision problem. © 2017 IEEE . . .	37
4.3	Literal Component $G_i$ for the MW-OLD decision problem. © 2017 IEEE . . . . .	38
4.4	Clause Component $H_j$ for the MW-OLD decision problem. © 2017 IEEE . . . . .	38
4.5	A minimizing weight function for a cycle with nodes $\{0, 1, \dots, 14\}$ is $w(x) = 2$ for $x \in \{3, 5, 7, 9, 11\}$ and $w(x) = 1$ otherwise which produces a mixed-weight OLD-set of size 7, shown in black. . . . .	41
4.6	If node $v$ is given weight 2, the weight 1 nodes $x_1, x_2$ , and $x_3$ no longer need to be in the mixed-weight OLD-set. With $v$ in the mixed-weight OLD-set, the total weight is reduced by 1. Other nodes that need to be in the mixed-weight OLD-set are shown in black. . . . .	43
5.1	In this portion of a path or cycle, $w(x_4) = 2$ , and nodes $x_3, x_4, x_6, x_7$ are in the mixed-weight OLD-set and form a weakly connected 4-cluster. Nodes $x_2, x_5, x_8$ are not in the mixed-weight OLD-set but are neighbors of the cluster. . . . .	46
5.2	A potential 3-cluster in a mixed-weight OLD-set on a path when considering weights 1 and 2. . . . .	46
6.1	Random Graphs $G(n, p)$ with $n = 10$ and $p = 0.0$ , $p = 0.25$ , and $p = 0.5$ . . . . .	56

6.2	For each $n$ , the percentage of graphs containing a mixed-weight OLD-set (solid line) and containing a mixed-weight OLD-set less than the smallest possible bound (dashed line) are graphed by $\rho$ , the probability that a node is weight 2. Values were averaged across $p$ . © 2017 IEEE . . . . .	61
6.3	For each $n$ , the percentage of graphs containing a mixed-weight OLD-set (solid line) and containing a mixed-weight OLD-set less than the smallest possible bound (dashed line) are graphed by $p$ , the probability that an edge exists. Values were averaged across $\rho$ . . . . .	62
7.1	Example of a geometric graph $G(n, r)$ on a $1 \times 1$ plane with $n = 200$ nodes and adjacencies determined by distance $r = 0.15$ . . . . .	65
9.1	Four cases for the location of a node in a random geometric graph on a plane. © 2018 IEEE . . . . .	83

## Mixed-Weight Open Locating-Dominating Sets

# Chapter 1

## Introduction

Location-detection problems are found in a wide variety of areas, from finding faults in microprocessors to identifying contaminants in ventilation systems to detecting illegal logging in rain forests. Theoretical studies of these problems typically focus on minimizing the number of sensors required, or the overall cost of a system of sensors, to detect and locate anomalies. To determine the minimum number of sensors, a system is modeled as a graph  $G = (V, E)$ , where the nodes  $V$  represent locations in the network, and the edges  $E$  represent the connections between them. Sensors are placed at particular node locations that, by design, are able to uniquely detect and locate issues in the system.

Open locating-dominating sets (OLD-sets) extend basic location and detection by working under the condition that a sensor is unable to detect a problem at its own location. Sensor failure could be caused by extreme environmental conditions or by the act of a nefarious individual. This setup could also be by design, if sensors detect problems through routed messages. The mixed-weight open locating-dominating set (mixed-weight OLD-set) expands the problem even further by considering the use of sensors of varying strengths and, potentially, cost. This novel approach to location-detection problems extends the reach of theoretical solutions to systems that are not limited to a single type of sensor.

In this chapter we discuss related work and applications in Section 1.1 and provide

definitions needed to study open locating-dominating sets in Section 1.2. In Section 1.3 we provide an outline and overview for the rest of the chapters.

## 1.1 Related Work and Motivation

Most research on location-detection problems has focused on the closed locating-dominating set and the problem of identifying codes [11, 16, 46, 56, 58, 69, 78]. The OLD-set is a more recent problem first defined in 2010 [72]. Minimum OLD-set sizes have been studied for trees [73], infinite cylinders [74], grid-like graphs [75], and infinite triangular grids [53]. A dynamic bibliography of results in identifying codes and closed and open locating-dominating sets can be found in [60].

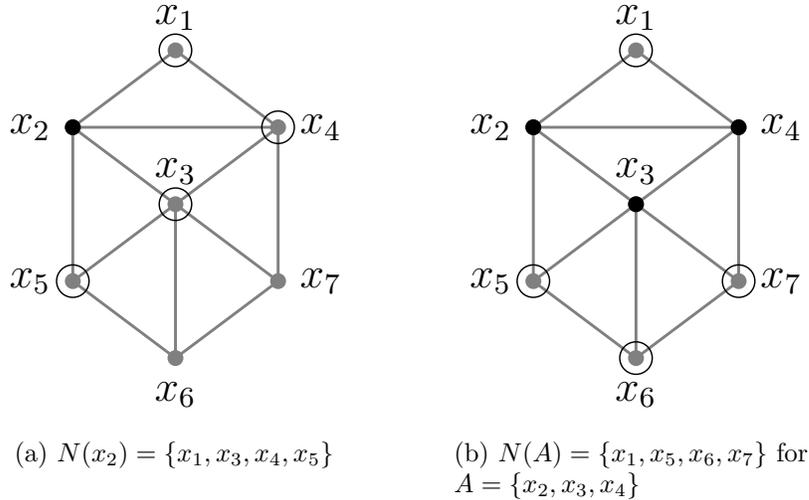
OLD-sets can determine optimal sensor placement for a variety of problems that can be modeled as a graph. A system of connected microprocessors can be modeled by an undirected graph where a processor is represented by a node and a network connection between two processors is represented by an edge. Fault detection in such networks has been studied using identifying codes and closed locating-dominating sets [11, 16, 45, 46]. Using OLD-sets we are able to consider the additional problem of a sensor failing to detect a fault in a processor where it is located. This type of sensor failure could be the result of a fault causing the sensor to fail, or, by design, when faults are only detected via routing messages directly between two connected processors.

We model the interior and exterior of a facility as a graph by dividing the physical space into regions and adding nodes to the graph representing each of these regions. An edge is added to the graph to represent two regions that would be within communication range if a sensor was placed on one of those regions. Models of this type have been used to study intruder detection [72] and survivor location in emergency situations [56, 69, 78]. The OLD-set covers the additional problem of a sensor failing to detect or locate a person at the sensor's position, a scenario that occurs when an intruder disengages a sensor or when a disaster causes the failure of a sensor.

Wireless sensor networks (WSNs) play an integral part in the monitoring and detection of changes in the environment and climate. A WSN consists of hundreds of sensors, each connected to a receiving device that has the capability to transmit, store, or manipulate data captured by the sensor. The sensors monitor the physical environment and take measurements on conditions such as temperature, humidity, pressure, and pollutants. These measurements are sent wirelessly from the receiving devices through the network to one or many controllers that store and organize the data for algorithmic interpretation. These networks can be organized in precise patterns or created at random via aircraft drops [71]. The benefits of using identifying codes in the design of WSNs, including efficiency and ease of monitoring, are discussed in [58]. WSNs are often constrained by cost, reliability of sensors, battery power, wireless communications and distance, storage limitations, and computational power [71].

WSNs are particularly useful in fields where distance and landscape make conventional networking impossible. Environmental monitoring using wireless sensor networks covers a range of important research areas including the study of glaciers [63], marine pollution [2], air pollution [52], animal behavior and welfare [61], the effect of climate change on farming [33, 22], and the detection of natural disasters [5]. The environment can be modeled similarly to a facility: divide the area into regions, where each region is represented by a node, and an edge represents two regions that would be in range of a single sensor.

Location-detection problems have also been studied for contaminant detection and source location in public utilities and building ventilation systems [6, 7]. These systems are modeled as a graph by considering the system as a network. Nodes represent potential sensor locations, and edges represent the pipes connecting those locations. Research in location-detection can also benefit the detection of criminal behavior in other environments, such as illegal logging in the rainforest [1]. OLD-sets are of particular interest for situations in which a perpetrator destroys a sensor at the location of the criminal activity.



**Figure 1.1:** The neighborhood of a node is all nodes adjacent to the node, not including the node. The neighborhood of a set is all nodes adjacent to the nodes in the set, not including the set. The nodes under consideration are shown in black, and the nodes in the neighborhood are circled.

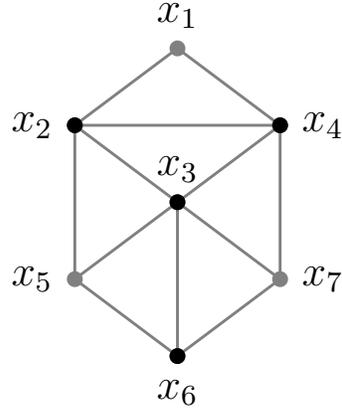
## 1.2 Open Locating-Dominating Sets and Related Problems

In this section we cover basic definitions and provide examples of open locating-dominating sets and similar sets. We begin with the definitions of a neighbor and a neighborhood. Consider an undirected graph  $G = (V, E)$  where  $V$  is the set of nodes and  $E$  is the set of edges. Figure 1.1 shows an example of the neighborhood of a node and the neighborhood of a set.

### 1.2.1 Definitions

**Definition 1.1** *The neighborhood of a node  $x$ ,  $N(x)$ , is the set of nodes that are adjacent to  $x$  in the graph, not including  $x$ , i.e.,  $x \notin N(x)$ . The neighborhood of the set  $A \subseteq V$ ,  $N(A)$ , contains all the nodes adjacent to the set and not in the set, i.e.,  $N(A) = \bigcup_{x \in A} N(x) - A$ .*

**Definition 1.2** *The neighbor of a node  $x$  or set  $A$  is any  $y$  such that  $y \in N(x)$  or  $N(A)$ .*



(a)  $S = \{x_2, x_3, x_4, x_6\}$  is an OLD-set

$x$	$N(x) \cap S$
$x_1$	$\{x_2, x_4\}$
$x_2$	$\{x_3, x_4\}$
$x_3$	$\{x_2, x_4, x_6\}$
$x_4$	$\{x_2, x_3\}$
$x_5$	$\{x_2, x_3, x_6\}$
$x_6$	$\{x_3\}$
$x_7$	$\{x_3, x_4, x_6\}$

(b)  $N(x) \cap S$  is non-empty and unique for each node in the graph

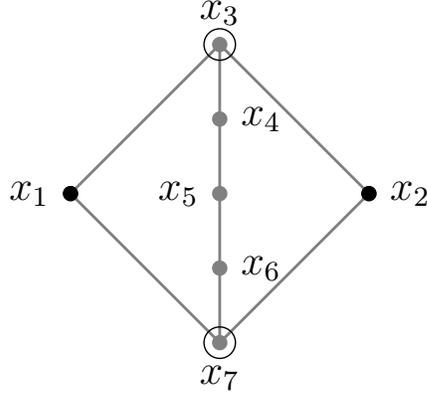
**Figure 1.2:** By comparing each neighborhood intersected with the set  $S$ , we can determine that  $S$  is an OLD-set. The nodes in the OLD-set are shown in black.

**Definition 1.3** A set  $S$  of nodes in a graph is an open locating-dominating set or OLD-set, if  $\forall x \in V, N(x) \cap S \neq \emptyset$  ( $N(x) \cap S$  is non-empty), and for any two distinct nodes  $x \neq y, N(x) \cap S \neq N(y) \cap S$  ( $N(x) \cap S$  is unique).

**Definition 1.4** The OLD-set neighborhood contains all neighbors of a node  $x$  that are in the OLD-set  $S, N(x) \cap S$ .

A set  $S$  is an OLD-set of the graph if  $\forall x \in V, x$  has at least one neighbor in  $S$  (the dominating property), and if for every pair of nodes in the graph,  $x \neq y$ , there is at least one node in  $S$  that is adjacent to either  $x$  or  $y$  but not both, in other words,  $x$  and  $y$  do not have the same set of neighbors in  $S$  (the locating property). The open property is provided by Definition 1.1 which requires that a node is excluded from its own neighborhood.

Figure 1.2 shows a graph, an OLD-set  $S$ , and the set  $N(x) \cap S$  for each node  $x$ . Determining if the set  $S$  is dominating can be done by quickly searching through the adjacencies in the graph, and is easy to see by looking at the graph. However, determining if the set is locating requires a search through every neighborhood which can be done in



**Figure 1.3:** Nodes  $x_1$  and  $x_2$  share the neighborhood  $\{x_3, x_7\}$ , thus there is no OLD-set in the graph. The nodes that share a neighborhood are shown in black and the nodes in their neighborhood are circled.

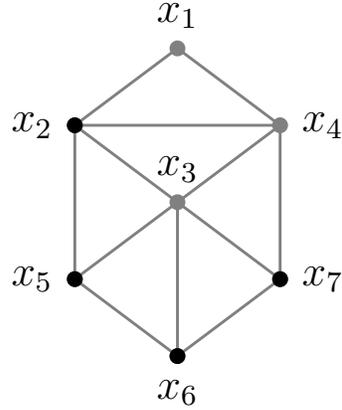
polynomial time, but is not as easy to visualize. Using a table to list  $N(x) \cap S$  for each node  $x$  makes the locating property easier to determine, as in Figure 1.2.

**Definition 1.5** A set  $S$  covers a node  $x$ , if  $S$  locates and dominates  $x$ . Thus  $N(x) \cap S \neq \emptyset$ , and  $N(x) \cap S \neq N(y) \cap S$ , for  $x \neq y$ ,  $\forall y \in V$ .

**Definition 1.6** If  $y$  is in the OLD-set neighborhood of a node  $x$ , then  $y$  dominates and locates  $x$ .

**Definition 1.7** Two nodes share OLD-set neighborhoods if  $N(x) \cap S = N(y) \cap S$  for potential OLD-set  $S$ , thus if  $S$  is an OLD-set any two nodes in the graph must not share OLD-set neighborhoods.

Two nodes will share OLD-set neighborhoods if they share neighborhoods, i.e.,  $N(x) = N(y)$ . In Figure 1.3,  $N(x_1) = N(x_2)$ , thus for any possible OLD-set  $S$ ,  $N(x_1) \cap S = N(x_2) \cap S$ . Whether a graph has an OLD-set can be answered quickly by determining if the set  $V$  is an OLD-set.  $V$  is an OLD-set if and only if  $N(x) \cap V = N(x)$  is non-empty and unique for every node  $x$ .



(a)  $S = \{x_2, x_5, x_6, x_7\}$  is an ID-code

$x$	$N[x] \cap S$
$x_1$	$\{x_2\}$
$x_2$	$\{x_2, x_5\}$
$x_3$	$\{x_2, x_5, x_6, x_7\}$
$x_4$	$\{x_2, x_7\}$
$x_5$	$\{x_2, x_5, x_6\}$
$x_6$	$\{x_5, x_6, x_7\}$
$x_7$	$\{x_6, x_7\}$

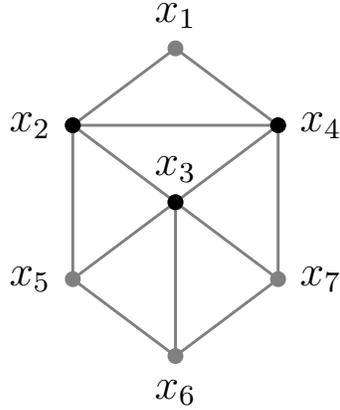
(b)  $N[x] \cap S$  is non-empty and unique for each node in the graph

**Figure 1.4:** By comparing the each closed neighborhood intersected with the set  $S$ , we can determine that  $S$  is an ID-code. The nodes in the ID-code are shown in black.

### 1.2.2 Related Problems

The OLD-set is a variation of an older problem called the *Identifying Code*, or *ID-Code*, and the related *Locating-Dominating Set*, or *LD-Set*. An ID-Code is defined similarly to an OLD-set, but on the *closed-neighborhood*,  $N[x]$ , that includes the  $x$  in its own neighborhood, i.e.,  $N[x]$  is the set including  $x$  and all the nodes adjacent to  $x$ . We note that in most literature the closed-neighborhood is typically referred to as the neighborhood, and the open-neighborhood is usually qualified, however, since the focus of this work is on OLD-sets, when we refer to the neighborhood we mean the open-neighborhood. A set of nodes,  $S$  is an ID-code if  $N[x] \cap S \neq \emptyset$  for all nodes in the graph, and for any two nodes  $x \neq y$ ,  $N[x] \cap S \neq N[y] \cap S$ . Thus any node in an ID-Code is able to locate and dominate itself.

The LD-set is also defined on the closed-neighborhood, but is only concerned with location and domination on nodes not in the set. The set  $S$  is an LD-set if  $N[x] \cap S \neq \emptyset$  for all nodes in the graph, and if  $N[x] \cap S \neq N[y] \cap S$  for any two nodes  $x \neq y$  and  $x, y \notin S$ .



(a)  $S = \{x_2, x_3, x_4\}$  is an LD-set

$x$	$N[x] \cap S$
$x_1$	$\{x_2, x_4\}$
$x_5$	$\{x_2, x_3\}$
$x_6$	$\{x_3\}$
$x_7$	$\{x_3, x_4\}$

(b)  $N[x] \cap S$  is non-empty and unique for each node not in  $S$

**Figure 1.5:** By comparing each closed neighborhood intersected with the set  $S$ , for nodes not in  $S$ , we can determine that  $S$  is an LD-set. The nodes in the LD-set are shown in black.

Figure 1.4 shows a graph, an ID-Code  $S$ , and the set  $N[x] \cap S$  for each node  $x$ . Figure 1.5 shows the same graph with an LD-set  $S$  and the set  $N[x] \cap S$  for each node  $x \notin S$ . A graph has an ID-code if and only if  $N[x]$  is unique for every node in the graph, and, in that case, the set of nodes  $V$  is an ID-code. For every graph,  $V$  is trivially an LD-set.

### 1.2.3 Focus

Location-detection problems, such as OLD-sets, ID-Codes, and LD-sets, are typically focused on finding the smallest such set in the graph with applications in the design of wireless sensor networks, fault detection in microprocessors, and contaminant detection in public utilities. Finding the minimum set allows for full coverage of the network with the least amount of overhead and cost. Although determining if a given set is an OLD-set, ID-Code, or LD-set can be done in polynomial time, finding the smallest such set is NP-complete [17, 18, 72]. The example OLD-set, ID-code, and LD-set shown in figures 1.2, 1.4, and 1.5 show minimum-sized such sets for each particular graph.

### 1.3 Outline and Overview

In this chapter we discussed the motivation to study location-detection problems and provided basic definitions and properties of open locating-dominating sets. We use OLD-sets in graphs to represent the placement of sensors in a network to locate and detect anomalies. These networks include microprocessor systems for fault detection, facilities for intruder discovery and apprehension, wireless sensor networks (WSNs) for environmental condition tracking, and public utilities for contaminant detection. OLD-sets are dominating sets, so they can detect everywhere in a graph or system, and locating sets, so they can also pinpoint the exact location of an issue. OLD-sets also have the open property that a node or sensor cannot detect or locate at its own location. This property covers situations in which a problem in the system destroys or causes the malfunction of the sensor at the problem's location. Sensor networks can also be designed with this property in mind by only detecting issues from incoming messages and data.

The focus of this thesis is the mixed-weight OLD-set, which uses weighted nodes to represent sensor strength in a network, but we begin with an exploration of the non-weighted OLD-set in circulant graphs in Chapter 2. Circulant graphs are cycle-like graphs used in multiprocessor systems. For these systems we are interested determining the minimum number of sensors required to detect and locate processor faults. We find optimal OLD-sets for two sets of circulant graphs by using two proof techniques. The first proof technique, called the discharging method, has a history in the field of location-detection [19, 20, 45, 46, 53]. For this method, we place a charge on each node in the OLD-set, and redistribute the charge throughout the graph. After redistribution, the amount of charge left in each node indicates a lower bound on the size of the OLD-set. We also introduce a new method, which we call the matching method, that uses Hall's Matching Theorem [41]. For this method, we uniquely pair nodes that are not in the OLD-set with nodes that are in the OLD-set, also providing a lower bound on the size of the OLD-set. In both cases, we are able to find OLD-sets with sizes equivalent to the lower bound, thus

providing optimally-sized OLD-sets.

In Chapter 3 we consider the mixed-weight open locating dominating-set. Mixed-weight OLD-sets extend the non-weighted OLD-set to include integer weights on nodes in the graph. These weights translate to the strength of a sensor used at a location in a system, such as in a WSN. In the theoretical context, the weight indicates the reach of a node through the graph, and nodes with greater weight become neighbors of other nodes that are further away. This chapter includes definitions necessary to study mixed-weight OLD-sets and comments on some of their interesting properties.

Chapter 4 introduces optimization problems related to the mixed-weight OLD-set and covers its NP-completeness. First, we consider the problem of finding a minimum mixed-weight OLD-set given a graph and a set of weights, and show that this problem is NP-complete. We then consider the similar, but slightly more complex problem, of finding the set of weights that provides the minimum mixed-weight OLD-set for a particular graph. Both of these problems are important considerations in location-detection as they directly translate to determining the smallest number and setup of a system of sensors. The last optimization problem is minimizing the sum of the weights of the nodes in a mixed-weight OLD-set. For non-weighted OLD-sets, this weight would always be the number of nodes in the OLD-set. However, if the cost of a sensor is proportional to the strength of the sensor, then minimizing cost of the system of sensors may become more important than minimizing the number of sensors.

In Chapter 5 we consider mixed-weight OLD-sets in paths and cycles. Paths and cycles often have polynomial solutions to problems that in general are NP-complete. Thus paths and cycles can provide a simpler way to study and use problems that are otherwise computationally difficult. We find a linear solution to the mixed-weight OLD-set problem, using weights 1 and 2, in paths and cycles. We also consider paths and cycles with all weight 2 nodes, as this is the first time weighted and mixed-weight OLD-sets have been studied.

In Chapter 6 we study bounds on the size of mixed-weight OLD-sets in random graphs

and use simulation to support the results. Random graphs estimate the distribution of WSNs that have been randomly dispersed in the field. Probabilistic upper bounds for the size of identifying codes, a problem similar to OLD-sets, but without the open property, have previously been studied in random graphs [30]. We extend those results to (mixed-weight) OLD-sets in random graphs. We show that as the number of nodes increases, the probability that a random graph has a mixed-weight OLD-set less than the bound increases.

We introduce an integer linear program (ILP) for mixed-weight OLD-sets and study its use in random geometric graphs in Chapter 7. ILP solvers use branch and bound to find solutions to computationally difficult problems. Without limiting time or number of computations, a solution found through an ILP model is guaranteed to be optimal. For (mixed-weight) OLD-sets we are interested in minimizing the size of the set in a graph, thus minimizing the number of sensors needed in a network, reducing initial and maintenance costs of the system. Finding a minimum-sized mixed-weight OLD-set is intractable, thus an ILP model can more quickly find a solution on a large graph in most cases. We also consider the linear programming (LP) relaxation of the integer constraint, which can potentially provide a very fast estimate of the optimal solution.

We explore several greedy algorithms and compare them to the ILP results in Chapter 8. Although ILP solutions are typically found more quickly than brute force results, quick solutions are not guaranteed. For very difficult problems, like the mixed-weight OLD-set, an ILP formulation can still take a considerable amount of time, even if significantly better than brute force. The time to solve a large ILP problem can often be shortened by providing an initial feasible solution or “hot start”. We use several greedy algorithms: a naïve approach, a method inspired by the maximal independent set problem, and a stingy method to provide the ILP model with an initial feasible, though not necessarily small, solution. We find that, on average, the stingy method, which places all the nodes into the mixed-weight OLD-set and removes nodes that are not needed one by one, significantly outperforms the other greedy methods by providing the smallest initial mixed-weight

OLD-set.

In Chapter 9 we expand the results in random graphs to estimate mixed-weight OLD-set bounds in random geometric graphs. Random geometric graphs closely resemble WSNs that are distributed at random in an environment, so their study is of particular interest. However, random geometric graphs tend to be theoretically difficult. To simplify the calculations, we solve for the average probability two nodes are adjacent in a plane. We estimate upper bounds on the size mixed-weight OLD-sets in random geometric graphs using the average probability as a substitute in the bound found for random graphs in Chapter 6.

Chapter 10 provides an overview of contributions and a description of future work. Our major contributions include the introduction of mixed-weight open locating-dominating sets with a study of their complexity and properties and the development of an integer linear program, greedy algorithms, and linear-time solutions in paths and cycles for mixed-weight OLD-sets. In the future we plan to consider non-integer weights to represent actual distances on nodes and edges, explore the maximum number of weight 1 nodes a path or cycle can have at the minimum mixed-weight OLD-set size, consider improving our greedy algorithms for other types of graphs, and expand the idea of mixed-weight nodes in identifying codes. We will also consider constructing networks so that the (mixed-weight) OLD-set is minimized. A symbol table can be found in Appendix A.

## Chapter 2

# Open Locating-Dominating Sets in Circulant Graphs

In this chapter we consider OLD-sets in circulant graphs. Circulant graphs are regular, cyclic graphs that have symmetric adjacencies, making them attractive for both design and study. Circulant graphs have been used for multiprocessor network design and other massively parallel systems [21, 59, 80]. Several topologies have been studied for fault location in other multiprocessor systems including trees, hypercubes, and meshes [11, 46]. A few results have been found for closed locating-dominating sets and identifying codes in circulant graphs [32, 62].

Proof techniques in location-detection problems have been dominated by the discharging method [51, 53, 54] and other similar methods. The discharging method was first used and made famous in the proof of the Four-Color Theorem [4]. It has recently been used to provide lower bounds on the size of an identifying code in infinite grids [19, 20]. Similarly, we use the discharging method to find the lower bound on the OLD-set size in circulant graphs of the form  $C_n(1, 2)$  and provide an OLD-set construction at that same size, thus providing the optimal OLD-set density.

Hall's Matching Theorem [41] gives the conditions necessary and sufficient to find a matching or pairing in a bipartite graph. We use Hall's theorem to show there is a

matching from nodes not in the OLD-set to nodes in the OLD-set for circulant graphs of the form  $C_n(1,3)$  by constructing a bipartite graph between the two sets. From this matching we get a lower bound on the OLD-set size in these graphs, and we also provide an OLD-set construction at that size to achieve the optimal OLD-set density.

In Section 2.1 we cover definitions needed for circulant graphs and the proofs in this chapter. In Section 2.2 we use the discharging method to show the optimal OLD-set size for  $C_n(1,2)$ , and in Section 2.3 we use Hall's Theorem to show the optimal OLD-set size for  $C_n(1,3)$ . We conclude our study of OLD-sets in circulant graphs in section 2.4. The majority of the work presented in this chapter has been submitted to *Discussiones Mathematicae Graph Theory*.

## 2.1 Circulant Graphs and Definitions

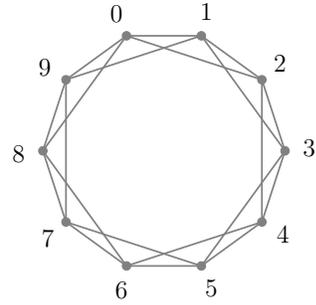
**Definition 2.1** A circulant graph  $C_n(1,t)$  is a degree four, undirected graph containing  $n$  nodes labeled  $\{0, 1, \dots, n-1\}$  where each node  $x$  is adjacent to nodes  $x \pm 1 \pmod{n}$  and  $x \pm t \pmod{n}$ .

In Figure 2.1 we see circulant graphs  $C_{10}(1,2)$  and  $C_{16}(1,3)$  with nodes drawn in the typical circular way. For graphs with  $n \gg t$ , edges are contained locally, so we can draw segments of the graph linearly to get a better view of this locality, as seen in Figure 2.2.

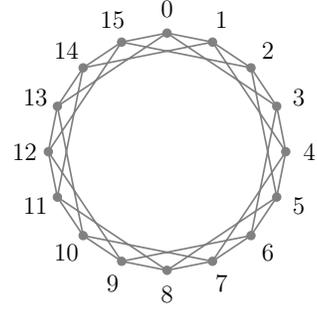
**Definition 2.2** The OLD-set density is the proportion of nodes in an OLD-set  $S$  to total nodes in the graph,  $|S|/n$ .

**Definition 2.3** The optimal OLD-set density is the achievable minimum OLD-set density in a graph, i.e., for optimal OLD-set density  $\delta$ , every OLD-set has density at least  $\delta$ , and there exists an OLD-set with density  $\delta$ .

By construction the upper bound on the minimum OLD-set density for  $C_n(1,2)$  is  $2/5$ , as seen in Figure 2.3a, and for  $C_n(1,3)$  is  $1/2$ , as seen in Figures 2.3b and 2.3c. If

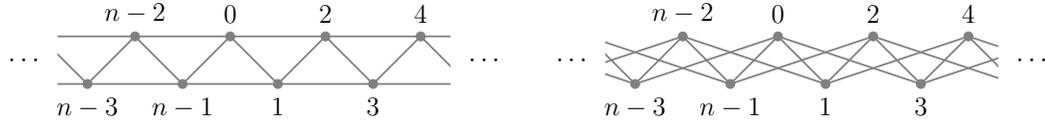


(a)  $C_{10}(1,2)$  circulant graph



(b)  $C_{16}(1,3)$  circulant graph

**Figure 2.1:** Examples of circulant graphs drawn in the typical circular way to show global behavior.



(a)  $C_n(1,2)$  drawn linearly

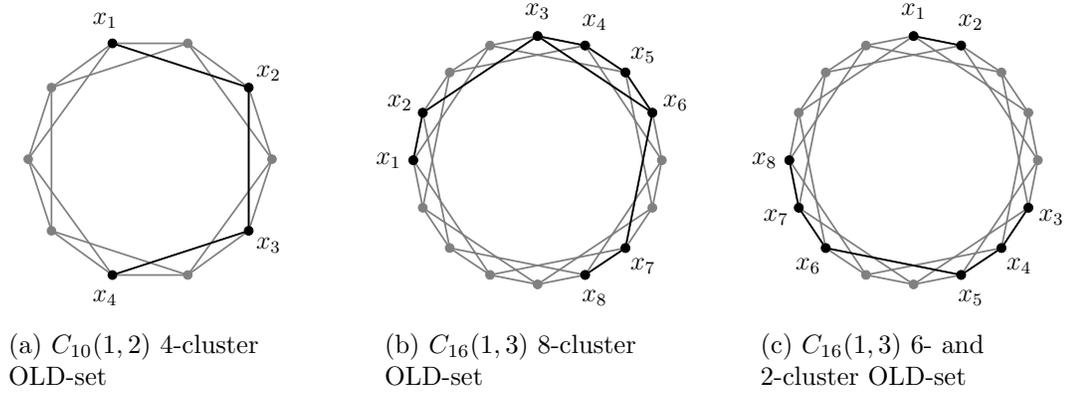
(b)  $C_n(1,3)$  drawn linearly

**Figure 2.2:** Examples of circulant graphs drawn in a linear way to show local behavior.

$\{x_1, x_2, x_3, x_4\}$  is the OLD-set shown in Figure 2.3a, then the set  $\{x_j + 10i | 1 \leq j \leq 4, 0 \leq i < k\}$  is an OLD-set in  $C_{10k}(1,2)$  of density  $2/5$ . If we consider each OLD-set in Figures 2.3b and 2.3c as  $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$ , then the set  $\{x_j + 16i | 1 \leq j \leq 8, 0 \leq i < k\}$  is an OLD-set in  $C_{16k}(1,3)$  of density  $1/2$ .

**Definition 2.4** A cluster is a connected component in the graph induced by an OLD-set  $S$ . An  $m$ -cluster is a cluster of order  $m$ . All neighbors of a cluster are not in the OLD-set.

We note that for  $m$ -clusters,  $m \geq 2$  because a 1-cluster is a single node with all of its neighbors not in the OLD-set, and thus the node in the cluster would not be covered by the OLD-set. Examples of OLD-set clusters can be seen in Figure 2.3. Throughout this Chapter we use black nodes to denote nodes known to be in the OLD-set, and circled nodes to denote nodes known not to be in the OLD-set.



**Figure 2.3:** Examples of OLD-sets and  $m$ -clusters in circulant graphs. OLD-set nodes and their cluster connections are shown in black.

## 2.2 The Discharging Method

We prove the optimal OLD-set density in  $C_n(1, 2)$  is  $\delta = 2/5$  using the discharging method. The discharging method is a proof technique that gives a particular set of nodes in a graph,  $S$ , a charge of 1, and every other node in the graph no charge. If a set of rules can be developed to redistribute the charge so that every node in the graph has a charge of at least  $f < 1$  after redistribution, then the size of  $S$  is at least  $f \cdot n$ . Using the discharging method we show that the the number of nodes in the OLD-set is at least  $2n/5$ .

**Lemma 2.1** *Any node in  $C_n(1, 2)$  that is not in the OLD-set  $S$  is the neighbor of at most two clusters.*

**Proof:** Consider node  $x \in C_n(1, 2)$  that is not in the OLD-set  $S$ . In order for  $x$  to be a neighbor of three clusters, three neighbors of  $x$  must be in different clusters. The neighborhood of  $x$  is  $N(x) = \{x - 2, x - 1, x + 1, x + 2\}$ . Note that  $x - 2$  and  $x - 1$  are neighbors,  $x - 1$  and  $x + 1$  are neighbors, and  $x + 1$  and  $x + 2$  are neighbors in  $C_n(1, 2)$ . Thus no three neighbors of  $x$  can be in  $S$  and form three different clusters, by Definition 2.4. Thus  $x$  is the neighbor of at most two clusters. ■

**Lemma 2.2** *Any  $m$ -cluster of an OLD-set  $S$  in  $C_n(1, 2)$  has at most  $m + 3$  neighbors not in  $S$ .*

**Proof:** We first note that for any  $m$ -cluster in  $C_n(1, 2)$  we can order the nodes of the cluster  $\{x_1, x_2, \dots, x_m\}$  such that  $\forall i, x_i + 1 \pmod{n} = x_{i+1}$  or  $x_i + 2 \pmod{n} = x_{i+1}$ , and thus any  $x_i$  is adjacent to  $x_{i+1}$ . If for any  $y$  in the cluster, such that  $y = x_i$  for some  $i < m$ , both  $y + 1$  and  $y + 2$  are in the cluster, then the ordering would include  $x_{i+1} = y + 1$  and  $x_{i+2} = y + 2$ . If only one of  $y + j$  for  $j = 1$  or  $2$  is in the cluster,  $x_{i+1} = y + j$ .

Consider the  $m$ -cluster  $\{x_1, x_2, \dots, x_m\}$  ordered in the way described above. Node  $x_1$  has at least 2 neighbors not in the cluster,  $x_1 - 2$  and  $x_1 - 1$ , and  $x_m$  also has at least 2 neighbors not in the cluster,  $x_m + 1$  and  $x_m + 2 \pmod{n}$ . If  $x_i + 2 = x_{i+1}$  then  $x_i + 1$  is a neighbor of the  $m$ -cluster. There are at most  $m - 1$  possibilities for  $x_i + 1$  to be a neighbor of the cluster, thus with the 2 neighbors on either end of the cluster, the cluster has at most  $m + 3$  neighbors. ■

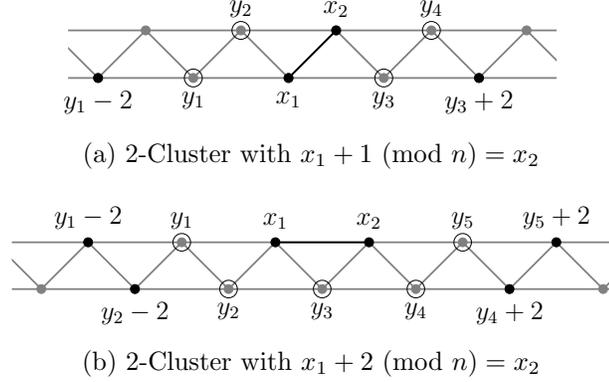
**Theorem 2.3** *The optimal OLD-set density in  $C_n(1, 2)$  is  $\delta = 2/5$ .*

**Proof:** We note that by construction the upper bound for the minimum OLD-set size is  $2/5$ . To show the lower bound is  $2/5$  we use the discharging method as follows. Let  $S$  be an OLD-set on  $C_n(1, 2)$  for  $n > 2$ . Assign a charge of 1 to nodes in the OLD-set  $S$  and a charge of 0 to all other nodes. Our goal is to redistribute the charges so each node has at least  $2/5$  charge. Redistribution Rule:

- If  $x \notin S$  is adjacent to  $k$  clusters in  $S$ , then  $x$  gets  $\frac{1}{k} \cdot \frac{2}{5}$  charge from each of the clusters.

By this rule, each node not in  $S$  will have a charge of at least  $2/5$ . For the proof we consider all the possible  $m$ -clusters in  $S$ . If each possible  $m$ -cluster in  $S$  is left with  $2m/5$  charge after redistribution, then all nodes in  $S$  will maintain  $2/5$  charge. We note that an  $m$ -cluster can give at most  $3m/5$  charge to its neighbors to maintain at least  $2m/5$  charge.

**Case 1:** 2-clusters in  $S$ .

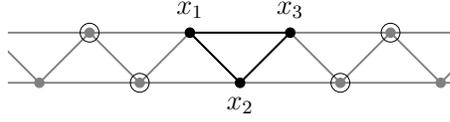


**Figure 2.4:** Possible 2-clusters  $\{x_1, x_2\}$  of OLD-set  $S$  in  $C_n(1, 2)$ .

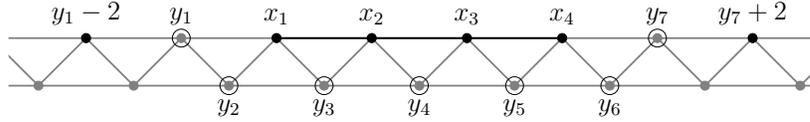
*Case 1a:* 2-cluster  $\{x_1, x_2\}$  with  $x_1 + 1 \pmod{n} = x_2$ , as in Figure 2.4a. In this case, there are 4 neighbors of the cluster  $\{y_1, y_2, y_3, y_4\}$ . Nodes  $y_2$  and  $y_3$  share OLD-set neighbors in the cluster  $\{x_1, x_2\}$ ,  $y_1$  and  $x_2$  share  $\{x_1\}$ , and  $y_4$  and  $x_1$  share  $\{x_2\}$ , so at least one node on each side of the cluster must be in the OLD-set. An OLD-set configuration that requires the cluster to give the most charge to its neighbors is the inclusion of  $y_1 - 2$  and  $y_3 + 2$ . Nodes  $y_1$ ,  $y_3$ , and  $y_4$  are each a neighbor of another cluster, and each receive  $1/5$  charge from the other cluster. Thus  $y_2$  receives  $2/5$  charge from this cluster, and  $y_1$ ,  $y_3$ , and  $y_4$  each receive  $1/5$  charge from the cluster leaving the 2-cluster with at least  $2 \cdot 2/5$  charge.

*Case 1b:* 2-cluster  $\{x_1, x_2\}$  with  $x_1 + 2 \pmod{n} = x_2$ , as in Figure 2.4b. In this case, there are 5 neighbors of the cluster  $\{y_1, y_2, y_3, y_4, y_5\}$ . Nodes  $y_2$  and  $x_2$  share an OLD-set neighborhood, so  $y_2 - 2$  must be in the OLD-set. This leaves  $y_1$  and  $y_2$  with the same OLD-set neighborhood, so  $y_1 - 2$  must be in the OLD-set. Nodes  $x_1$ ,  $y_4$ , and  $y_5$  follow similarly on the other side requiring  $y_4 + 2$  and  $y_5 + 2$  to also be in the OLD-set. Therefore  $y_1$ ,  $y_2$ ,  $y_4$  and  $y_5$  each receive  $1/5$  charge from other clusters. Thus the cluster  $\{x_1, x_2\}$  gives  $2/5$  charge to  $y_3$  and  $1/5$  charge to its remaining 4 neighbors, leaving the cluster with at least  $2 \cdot 2/5$  charge.

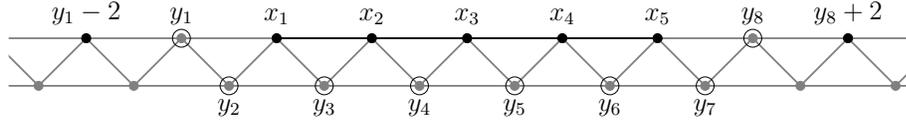
**Case 2:** 3-clusters in  $S$ . For any 3-cluster  $\{x_1, x_2, x_3\}$  with  $x_1$  adjacent to  $x_2$  and  $x_2$



**Figure 2.5:** Possible 3-cluster of OLD-set  $S$  in  $C_n(1, 2)$ .



(a) 4-cluster



(b) 5-cluster

**Figure 2.6:** Possible 4-cluster and 5-cluster of OLD-set  $S$  in  $C_n(1, 2)$ .

adjacent to  $x_3$ , if the 3-cluster does not form a triangle, i.e.,  $x_1$  is not adjacent to  $x_3$ , then  $x_1$  and  $x_3$  share OLD-set neighborhood,  $\{x_2\}$ . Thus there is only one possible 3-cluster shape in  $S$  as shown in Figure 2.5. This cluster has 4 neighbors and a total charge of 3. The 4 neighbors need at most a charge of  $8/5$  which would leave the cluster with at least  $3 \cdot 2/5$  charge.

**Case 3:**  $m$ -clusters in  $S$  with  $m = 4, 5$ . For  $m = 4, 5$ , a cluster begins with  $m$  charge and can give  $3m/5$  charge to its neighbors. Each neighbor needs  $2/5$  charge, so with  $3m/5$  charge available, a cluster can provide total charge for up to  $\lfloor 3m/2 \rfloor$  neighbors. The only  $m$ -clusters,  $m = 4, 5$ , with more than  $\lfloor 3m/2 \rfloor = 6, 7$ , respectively, neighbors are shown in Figure 2.6. Both of these clusters are of the form  $\{x_1, x_2, \dots, x_m\}$  where  $x_i + 2 \pmod n = x_{i+1}$  for  $i < m$  and have  $m + 3$  neighbors,  $\{y_1, y_2, \dots, y_{m+3}\}$ . In these clusters,  $y_1$  and  $y_2$  share the same OLD-set neighborhood  $\{x_1\}$ , so  $y_1 - 2$  must be in the OLD-set. Similarly,  $y_{m+2}$  and  $y_{m+3}$  share  $\{x_m\}$ , and  $y_{m+3} + 2$  must be in the OLD-set. Therefore other clusters give  $1/5$  to  $y_1$  and  $y_{m+3}$ . And these clusters give  $1/5$  charge to

$y_1$  and  $y_{m+3}$  and  $2/5$  charge to the remaining  $m + 1$  neighbors, leaving the clusters with charge  $m - 2 \cdot (1/5) - (m + 1) \cdot 2/5 \geq m \cdot 2/5$  charge.

**Case 4:**  $m$ -clusters in  $S$  with  $m \geq 6$ . The cluster begins with charge  $m$  and has at most  $m + 3$  neighbors. Thus to cover all of its neighbors, a cluster would give out at most  $2(m + 3)/5 = (2m + 6)/5 \leq 3m/5$  charge leaving the cluster with at least  $m \cdot 2/5$  charge.

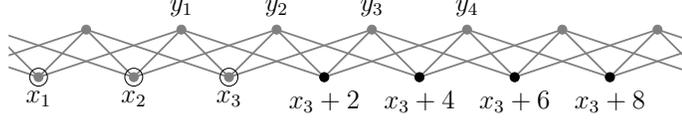
Thus for any  $m$ -cluster in  $S$ , the cluster maintains at least  $2m/5$  charge after discharging, leaving each node in the cluster with at least charge  $2/5$ . Therefore an OLD-set on  $C_n(1, 2)$  needs at least  $2n/5$  nodes, and by construction we know that an OLD-set needs at most  $2n/5$  nodes, as seen in Figure 2.3a. Thus the optimal OLD-set density in  $C_n(1, 2)$  is  $\delta = 2/5$ . ■

## 2.3 The Matching Method

We prove the optimal OLD-set density for  $C_n(1, 3)$  is  $\delta = 1/2$  using Hall's Theorem. Hall's Theorem states that bipartite graphs with nodes partitioned into sets  $R$  and  $S$  have a matching of size  $|R|$  if and only if for every subset  $A \subseteq R$ ,  $|A| \leq |N(A)|$ . We use Hall's Theorem to show there is a matching from nodes not in an OLD-set to nodes in the OLD-set for  $C_n(1, 3)$ . If such a matching exists, then the number of nodes in the OLD-set is at least  $n/2$ .

For the rest of this section we will consider  $x$  and  $y$  nodes such that  $x = 2i$  and  $y = 2i + 1$  for  $0 \leq i \leq n/2$ , so that  $x$  and  $y$  may be neighbors of each other, but  $x$  and  $y$  do not share neighbors. We say that two nodes have different parity if their label is a different parity, so all nodes  $x$  and  $y$  have different parity. We note that for any node  $x$  in  $C_n(1, 3)$ ,  $x$  does not have the same parity as its neighbors,  $x \pm 1$ ,  $x \pm 3$ . In particular, two nodes of different parity do not share neighbors.

**Lemma 2.4** *If  $|x_1 - x_2| \geq 8$ , i.e.,  $x_1$  and  $x_2$  have three or more nodes of the same parity between them, then they do not share neighbors.*



**Figure 2.7:** Three consecutive nodes of the same parity in  $C_n(1,3)$  that are not in the OLD-set are followed by four consecutive nodes in the OLD-set.

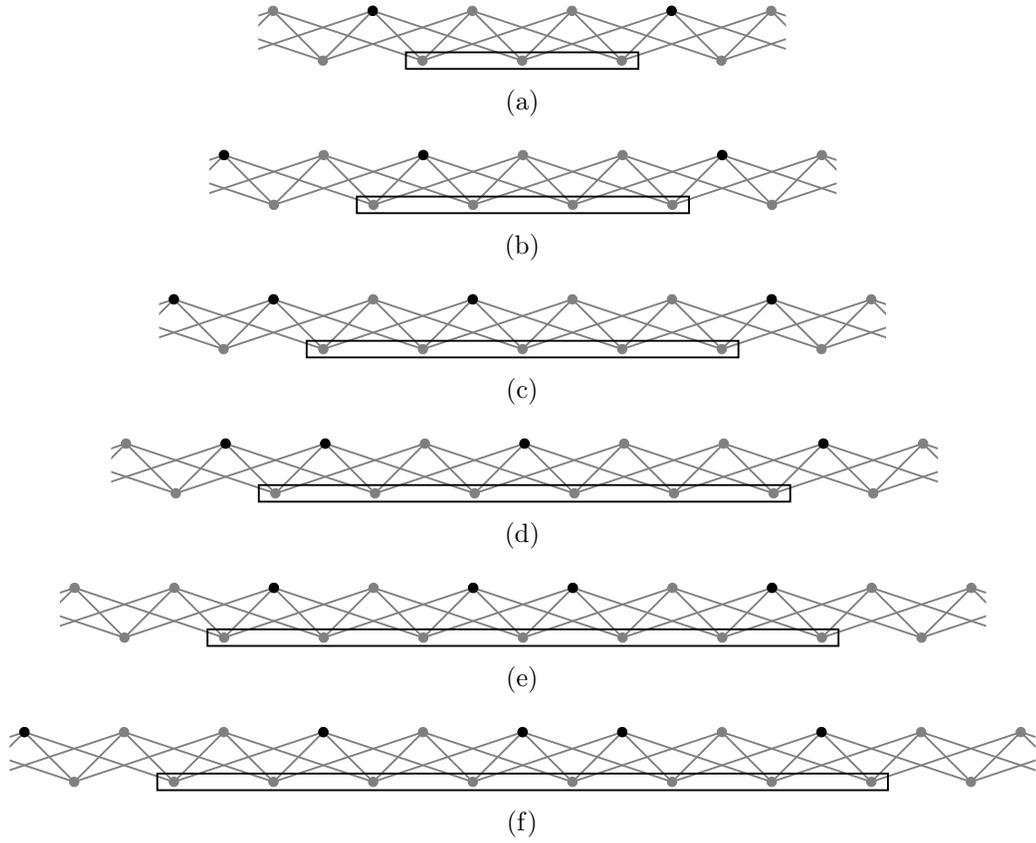
**Proof:** If  $|x_1 - x_2| \geq 8$  and  $x_1 < x_2$ , then the largest neighbor of  $x_1$  is  $x_1 + 3$  and the smallest neighbor of  $x_2$  is  $x_2 - 3$ . However,  $x_1 + 3 \leq x_2 - 5$ , therefore  $x_1$  and  $x_2$  do not share neighbors. ■

**Lemma 2.5** *If  $A = \{x_1, x_2, \dots, x_k\}$  such that  $x_i = x_{i-1} + 2$ , then  $|N(A)| = |A| + 3$ . Subsequently, if  $|A| = k \geq 4$ , there is a set  $B = \{y_1, y_2, \dots, y_{k-3}\}$  of size  $k - 3$  such that  $y_i = y_{i-1} + 2$  and  $N(B) = A$ .*

**Proof:** If  $A = \{x_1, x_2, \dots, x_k\}$  such that  $x_i = x_{i-1} + 2$ , then  $N(A) = \{x_1 - 3, x_1 - 1, x_1 + 1, \dots, x_1 + 2k - 1, x_1 + 2k + 1\} = \{y_{n-3}, y_{n-2}, y_{n-1}, y_1, \dots, y_k\}$  and  $|N(A)| = |A| + 3$ . We note that  $y_i = y_{i-1} + 2$ , the set  $B = \{y_1, y_2, \dots, y_{k-3}\}$  has neighbors  $N(B) = \{x_1, x_2, \dots, x_k\} = A$ , and  $|B| = k - 3$ . ■

**Lemma 2.6** *If nodes  $x_1, x_2$ , and  $x_3$ , such that  $x_3 = x_2 + 2$  and  $x_2 = x_1 + 2$ , are not in an OLD-set, then  $x_3 + i$  for  $i = 2, 4, 6, 8$  must be in the OLD-set. It follows that the four preceding nodes of the same parity must also be in the OLD-set.*

**Proof:** As seen in Figure 2.7 neighbor  $y_1$  only has one possible node in the OLD-set, so  $x_3 + 2$  must be in the OLD-set. Nodes  $y_1$  and  $y_2$  cannot share OLD-set neighborhood  $\{x_3 + 2\}$ , so  $x_3 + 4$  must be in the OLD-set. Nodes  $y_2$  and  $y_3$  cannot share OLD-set neighborhood  $\{x_3 + 2, x_3 + 4\}$ , so  $x_3 + 6$  must be in the OLD-set. Finally, nodes  $y_3$  and  $y_4$  cannot share OLD-set neighborhood  $\{x_3 + 2, x_3 + 4, x_3 + 6\}$ , so  $x_3 + 8$  must be in the OLD-set. The proof follows similarly for the four preceding nodes of the same parity. ■



**Figure 2.8:** Consecutive nodes of the same parity in  $C_n(1,3)$  with size  $k = 3, 4, 5, 6, 7, 8$  are boxed with one possible minimum OLD-set solution among their neighbors shown in black.

**Lemma 2.7** *If  $A = \{x_1, x_2, \dots, x_k\}$  such that  $x_i = x_{i-1} + 2$  for  $i < k$ , and  $S$  is an OLD-set on  $C_n(1,3)$ , then  $|N(A) \cap S| \geq \lfloor \frac{k}{2} \rfloor + 1$ .*

**Proof:**

For  $k = 1$ , the node must be dominated by  $S$ , so one neighbor must be in  $S$ . For  $k = 2$ , one node in  $S$  cannot locate two nodes, so 2 neighbors must be in  $S$ . Figure 2.8 shows examples of minimum OLD-sets neighborhoods for  $k = 3, 4, \dots, 8$  which are 2 neighbors in  $S$  for  $k = 3$ , 3 neighbors for  $k = 4$ , 4 neighbors for  $k = 5, 6, 7$  and 5 neighbors for  $k = 8$ .

Assume for all  $8 \leq \ell < k$  that  $\ell$  nodes  $x_i = x_{i-1} + 2$  for  $i < \ell$  have  $\lfloor \frac{\ell}{2} \rfloor + 1$  neighbors in the OLD-set. Consider  $k$  nodes  $A = \{x_1, x_2, \dots, x_k\}$  and by induction  $\{x_3, \dots, x_k\}$  contains

$\lfloor \frac{k-1}{2} \rfloor + 1 = \lfloor \frac{k}{2} \rfloor$  neighbors in  $S$ . Nodes  $x_1$  and  $x_2$  introduce two new neighbors to  $N(A)$ , namely  $\{x_1 - 3, x_1 - 1\}$ . We may assume that neither of the neighbors is in  $S$ .

**Case 1:** Neighbor  $x_1 + 1$  is in the OLD-set.

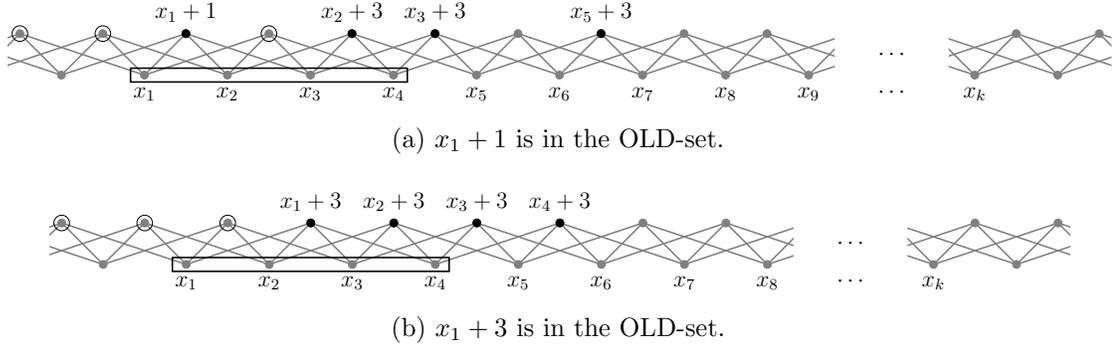
Suppose  $x_1 + 1$  is in the OLD-set as seen in Figure 2.9a. Nodes  $x_1$  and  $x_2$  cannot share OLD-set neighborhood  $\{x_1 + 1\}$ , so  $x_2 + 3$  must be in the OLD-set. Nodes  $x_2$  and  $x_3$  cannot share OLD-set neighborhood  $\{x_1 + 1, x_2 + 3\}$ , so  $x_3 + 3$  must be in the OLD-set. And nodes  $x_4$  and  $x_5$  cannot share OLD-set neighborhood  $\{x_2 + 3, x_3 + 3\}$ , so  $x_5 + 3$  must be in the OLD-set. The OLD-set nodes  $\{x_1 + 1, x_2 + 2, x_3 + 3, x_5 + 3\}$  cover nodes  $\{x_1, x_2, \dots, x_7\}$ . Consider the set  $\{x_9, x_{10}, \dots, x_k\}$  which is size  $k - 8$  and does not have neighbors in the set of OLD-set nodes  $\{x_1 + 1, x_2 + 2, x_3 + 3, x_5 + 3\}$ . This set of  $k - 8$  nodes needs at least  $\lfloor \frac{k-8}{2} \rfloor + 1$  of its neighbors in the OLD-set. Thus the set of  $k$  nodes need at least  $4 + \lfloor \frac{k-8}{2} \rfloor + 1 = \lfloor \frac{k}{2} \rfloor + 1$  neighbors in the OLD-set.

**Case 2:** Neighbor  $x_1 + 3$  is in the OLD-set.

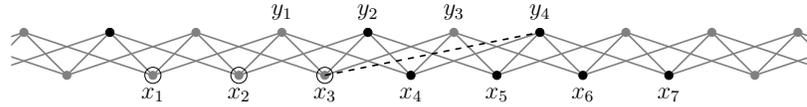
If  $x_1 + 3$  is in the OLD-set, then the three neighbors of  $x_1$  are not in the OLD-set, and so the next four nodes of the same parity, including  $x_1 + 3$ , must be in the OLD-set as shown in Figure 2.9b. The first seven nodes of the  $k$  total nodes are covered by these four OLD-set neighbors. The remaining  $k - 7$  nodes do not currently share any OLD-set neighbors with the first seven nodes, but will require  $\lfloor \frac{k-7}{2} \rfloor + 1$  neighbors in the OLD-set to cover them. Thus the set of  $k$  nodes need at least  $4 + \lfloor \frac{k-7}{2} \rfloor + 1 \geq \lfloor k/2 \rfloor + 1$  neighbors in the OLD-set. ■

**Theorem 2.8** *The optimal OLD-set density in  $C_n(1, 3)$  is  $\delta = 1/2$ .*

**Proof:** We note that by construction the upper bound for the minimum OLD-set size is  $1/2$ . To show the lower bound is  $1/2$ , we show there is a matching from nodes not in the OLD-set to nodes in the OLD-set for  $C_n(1, 3)$ . We build an auxiliary bipartite graph from  $C_n(1, 3)$ ,  $B = (R, S)$ , where  $R = V - S$ , with edges  $E(B) = \{(u, v) | u \in R, v \in S, (u, v) \in E(C_n(1, 3))\} \cup \{(x, x + 5) | x - 4, x - 2, x \notin S, x + 5 \in S\}$ . We only need to consider  $A \subseteq R$  such that the nodes in  $A$  share neighbors. If  $A$  has sets of nodes that do



**Figure 2.9:** Two cases for covering the first four nodes from a set of  $k$  consecutive nodes of the same parity.



**Figure 2.10:** The set  $A = \{x_1, x_2, x_3\}$  is three consecutive nodes of the same parity in  $C_n(1, 3)$ .

not share neighbors,  $A$  can be separated into sets of nodes with disjoint neighbor sets, and the size of the neighbors of  $A$  in the OLD-set will be the sum of the size of the neighbors in the OLD-set of the individual sets.

**Case 1:**  $|A| = 1$ . The bipartite graph contains  $A$  in one partition,  $N(A) \cap S$  in the other, and all edges between those sets that occur in the graph. For  $|A| = 1$ , the node in  $A$  must be covered by at least 1 node in the OLD-set, thus  $|N(A) \cap S| \geq |A|$ .

**Case 2:**  $|A| = 2$ . The bipartite graph contains  $A$  in one partition,  $N(A) \cap S$  in the other, and all edges between those sets that occur in the graph. For  $|A| = 2$ , each node must be covered by 1 node, but they cannot be covered by the same node or they would not be distinguishable by the OLD-set. Thus they must be covered by at least 2 nodes and  $|N(A) \cap S| \geq |A|$ .

**Case 3:**  $|A| \geq 3$ .

*Case 3a:*  $A$  is  $\{x_1, x_2, x_3\}$  with  $x_i = x_{i-1} + 2$ . The bipartite graph contains  $A$  in one

partition,  $(N(A) \cap S) \cup \{x_3 + 5\}$  in the other, all edges between those sets that occur in the graph, and edge  $(x_3, x_3 + 5)$  if  $x_3 + 5 \in S$ . By Lemma 2.7 we know that the three nodes can be covered by two OLD-set nodes. However, because  $A$  only contains nodes not in the OLD-set, by Lemma 2.6 the next four nodes of the same parity,  $\{x_4, x_5, x_6, x_7\}$  must be in the OLD-set, as shown in Figure 2.10. Thus  $x_3 + 5$  will not be in  $N(A) \cap S$  for any  $A$ . If  $A$  only has 2 neighbors in the OLD-set in  $C_n(1, 3)$ , it is also the case that  $x_3 + 5$  must be in the OLD-set in order to cover  $x_4 = x_3 + 2$ , otherwise  $x_3$  and  $x_4$  share OLD-set neighborhoods. For any subset  $A = \{x_1, x_2, x_3\}$  with  $x_i = x_{i-1} + 2$ , if  $x_3 + 5 \in S$ , the bipartite graph includes an edge from  $x_3$  to  $x_3 + 5$ , also shown in Figure 2.10. If  $x_3 + 5 \notin S$ , a third neighbor of  $A$  must be in  $S$ . Thus  $|N(A) \cap S| \geq |A|$ .

*Case 3b:*  $A = \{x_1, x_2, \dots, x_{|A|}\}$  such that  $x_i + 2j \pmod n = x_{i+1}$  for  $j > 0$  and at least one  $j > 1$ . We note that if a subset of  $A$  has two nodes  $x_i$  and  $x_{i+1}$  that do not share neighbors,  $A$  can be split into at least two different sets with disjoint neighbors. Then the size of the OLD-set neighborhood  $A$  would be the sum of the size of the OLD-set neighbors of the individual sets. Because of this consideration,  $j < 4$  by Lemma 2.4. And by Lemma 2.6,  $A$  will not contain three nodes as in Case 3a as these nodes would not share neighbors with any other node not in the OLD-set.

Let  $S'$  be the set of OLD-set nodes  $\{x'_1, x'_2, \dots, x'_j\}$  such that  $A \cup S'$  is the set of all nodes  $\{x_1 + 2i \mid 0 \leq i \leq \frac{x_{|A|} - x_1}{2}\}$ . We note that  $|A \cup S'| = k \geq 4$  and  $A \cup S'$  is a neighborhood to  $k - 3$  consecutive nodes of the same parity by Lemma 2.5. Thus at least  $\lfloor \frac{k-3}{2} \rfloor + 1$  nodes of  $A \cup S'$  must be in the OLD-set, and  $|A| \leq k - (\lfloor \frac{k-3}{2} \rfloor + 1)$ . We note that because each  $x_i$  in  $A$  shares at least one neighbor with  $x_{i+1}$ ,  $N(S') \subset N(A)$  and thus  $N(A) = N(A \cup S')$ . At least  $\lfloor \frac{k}{2} \rfloor + 1$  neighbors of  $A \cup S'$  must be in the OLD-set, and thus at least  $\lfloor \frac{k}{2} \rfloor + 1$  neighbors of  $A$  must be in the OLD-set. For  $k$ , even or odd, we find that  $|A| \leq k - (\lfloor \frac{k-3}{2} \rfloor + 1) = \lfloor \frac{k}{2} \rfloor + 1 = |N(A \cup S') \cap S| = |N(A) \cap S|$ .

Thus we have created a bipartite graph from  $R$  to  $S$  such that for all  $A \subseteq R$ ,  $|N(A) \cap S| \geq |A|$ . By Hall's Theorem there is a matching from  $R$  to  $S$  of size  $|R|$ , and therefore an OLD-set on  $C_n(1, 3)$  needs at least  $n/2$  nodes. By construction we know that an OLD-set

needs at most  $n/2$  nodes. Thus the optimal OLD-set density in  $C_n(1, 3)$  is  $\delta = 1/2$ . ■

## 2.4 Discussion

We derived the optimal minimum OLD-set density for  $C_n(1, 2)$  and  $C_n(1, 3)$  using two proof techniques, the discharging method and the matching method, which is based on Hall's Theorem. Previously the discharging method has been used in graph coloring and identifying code problems. We showed that it is a valid method to find lower-bounds on the size of OLD-sets in circulant graphs. To the best of our knowledge this is the first time Hall's Theorem has been used to prove a result for open locating-dominating sets. Recently, a result using matching theory in a different context was published in [28]. This further illustrates that matching is a valuable proof technique in addressing location-detection problems. These methods have the potential to find OLD-set size bounds in other circulant graphs and topologies used in microprocessor systems. Finding these bounds will help determine best practices for fault detection and design in networked microprocessors.

## Chapter 3

# Mixed-Weight Open Locating-Dominating Sets

The mixed-weight open locating-dominating set (mixed-weight OLD-set) models a system in which sensors of different strengths, and potentially different costs, are strategically placed throughout the system. The mixed-weight OLD-set extends the open locating-dominating set by allowing multiple integer weights to be given to nodes. An increase in the weight expands the reach of the node by an equivalent number of edges in the graph. For instance, a weight 1 node can reach its neighbors, a weight 2 node can reach its neighbors and its neighbors' neighbors, and so on. Wireless sensor networks often use multiple types of sensors, such as in systems that monitor natural habitats [61]. Mixed-weight OLD-sets can aid in the development and cost management of these networks.

Mixed-weight OLD-sets are related to the weighted  $d$ -identifying code and  $d$ -locating-dominating set, where all nodes receive the same weight  $d$ . Weighted identifying codes and locating-dominating sets have been studied for paths and cycles [8, 15, 25, 26, 39, 44, 70], hypercubes [45, 46, 47, 51, 68], and grids [50, 66, 76]. The mixed-weight OLD-set is also similar to an OLD-set on a directed graph: increased weight can be represented by arcs to other nodes. Identifying codes have also been studied for directed graphs in [57]. We note that, to our knowledge, there is no literature for weighted OLD-sets or OLD-sets in

directed graphs. The weighted OLD-set is a special case of the mixed-weight OLD-set, therefore this is the first time weighted or mixed-weight OLD-sets have been studied.

In Section 3.1 we introduce and provide definitions for the mixed-weight OLD-set, and in Section 3.2 we discuss mixed-weight identifying codes, a related problem. We conclude the chapter in Section 3.3. The majority of the work presented in this chapter has been published in [34] © 2017 IEEE.

### 3.1 Definitions and Properties

We begin with the basic definition of distance between nodes in a graph, and build to the definition of a mixed-weight OLD-set.

**Definition 3.1** *The geodesic distance or distance between two nodes  $x$  and  $y$ ,  $d(x, y)$ , is the length of the shortest path between node  $x$  and node  $y$ , where each edge is considered to be length 1 if the graph is unweighted.*

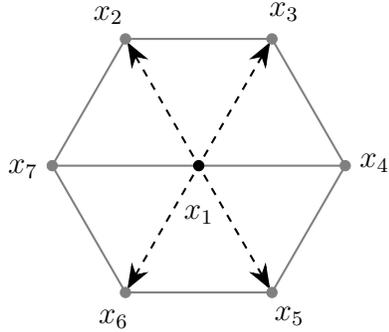
**Definition 3.2** *The weight function,  $w(x) \geq 1 \forall x \in V$ , defines the distance marker given to each node in the graph.*

**Definition 3.3** *For a particular node  $x$ ,  $w(x)$  is the weight of the node. For a set of nodes  $A \subset V$ , the weight of the set  $w(A)$  is the sum of all the weights of nodes in that set,  $\sum_{x \in A} w(x)$ .*

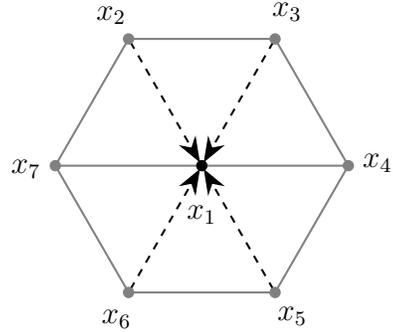
**Definition 3.4** *The open outgoing-ball,  $B^-(x)$  is the set of all nodes within a distance of  $w(x)$  from  $x$  but not including  $x$ , i.e.,  $B^-(x) = \{y \in V | 0 < d(x, y) \leq w(x)\}$ .*

**Definition 3.5** *The open incoming-ball,  $B^+(x)$  is the set of all nodes that contain  $x$  in their open outgoing-ball, i.e.,  $B^+(x) = \{y \in V | 0 < d(x, y) \leq w(y)\}$ .*

**Definition 3.6** *The mixed-weight open locating-dominating set, mixed-weight OLD-set, or MW-OLD-set of a graph  $G$  is a set of nodes  $S \subseteq V$  such that  $B^+(x) \cap S$  is nonempty and unique for every  $x \in V$ .*



(a) Visualization of  $x_1$  in the open incoming-balls of other nodes,  $w(x_1) = 2$ . © 2017 IEEE



(b) Visualization of  $x_1$  in the open outgoing-balls of other nodes,  $w(x_1) = 2$ .

**Figure 3.1:** When using the open incoming-ball definition, a node with weight greater than 1 becomes the neighbor of other nodes. When using the open outgoing-ball definition, other nodes become the neighbor of a node with weight greater than 1. The effect of the weighted node,  $w(x_1) = 2$ , is indicated by dashed arcs.

The case in which  $w(x) = 1$  for all  $x \in V$  defines the non-weighted OLD-set. For the mixed-weight OLD-set, we say a node  $y$  is a *neighbor* of node  $x$  if  $y \in B^+(x)$ . In this case neighbors are not symmetric. We similarly say a set  $S$  covers a node  $x$  if the set locates and dominates  $x$  given the neighbors in  $B^+(x)$ . We note that the weight of a node  $x$  not in OLD-set  $S$  does not affect  $S$  or the size of  $S$ , both theoretically and in the application of sensor monitoring.

**Definition 3.7** *The total weight of the OLD-set  $S$ ,  $w(S)$ , is the sum of all the weights of all nodes in the OLD-set.*

The mixed-weight OLD-set can also be defined on the open outgoing-ball,  $B^-(x)$ . These two definitions have similar properties but different applications. For the mixed-weight OLD-set defined on the open outgoing-ball, we say a node  $y$  is a *neighbor* of node  $x$  if  $y \in B^-(x)$ . Figure 3.1 gives a visual comparison of weighted nodes based on the incoming-ball and outgoing-ball definitions, where neighbors defined by weights are represented by arcs from nodes that are neighbors.

In Figure 3.1a, using the open incoming-ball definition, a node with weight greater than 1 becomes the neighbor of other nodes. In a WSN, a weighted node under this definition would represent a sensor that has a stronger antenna and can therefore receive data from transmitters that are further away. Similarly, this could represent stronger video monitors that are able to clearly capture visual data at longer distances. Thus a weighted node represents a sensor that is able to detect problems in the system at further distances.

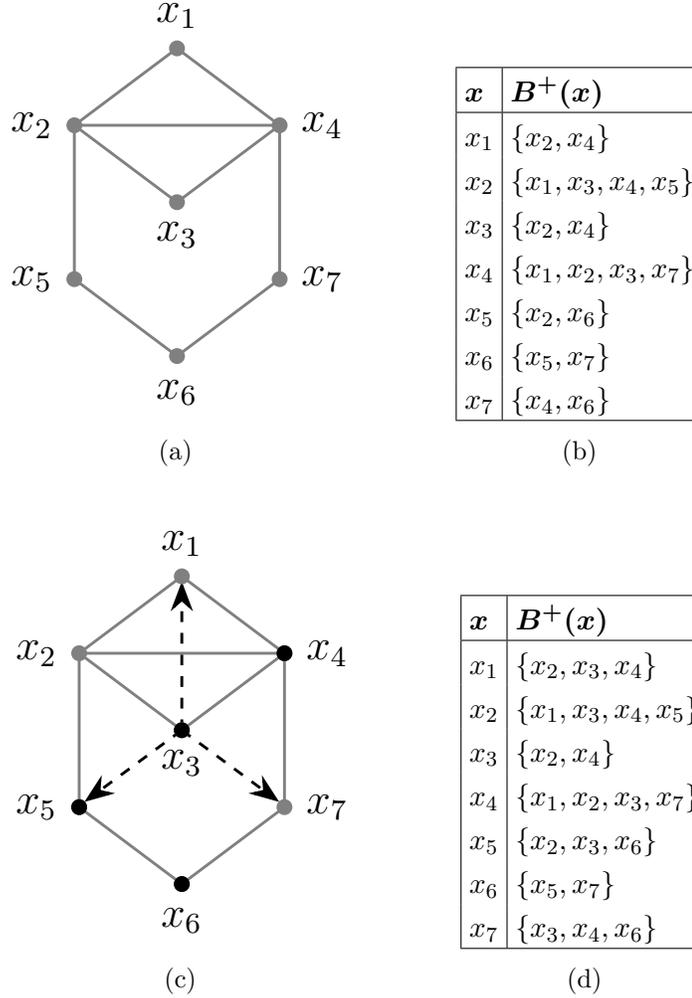
In Figure 3.1b, using the open outgoing-ball definition, other nodes become the neighbor of a node with weight greater than 1. In this case, a sensor represented by a weighted node would have stronger transmitting ability, with the ability to transmit data further through the system. For the remainder of this work we will consider the mixed-weight OLD-set on the open incoming-ball.

We will now discuss necessary conditions for the existence of mixed-weight OLD-sets. For graph  $G = (V, E)$  and weight function  $w(x)$ , a mixed-weight OLD-set  $S \subseteq V$  exists if and only if  $\forall x, y \in V, x \neq y, B^+(x) \neq B^+(y)$ , i.e., if the open incoming-ball is unique for all nodes in the graph. If a mixed-weight OLD-set exists, then  $S = V$  is a mixed-weight OLD-set. For every connected graph with  $|V| > 1$ , there exists a weight function such that the graph contains a mixed-weight OLD-set.

**Definition 3.8** *The eccentricity of  $x$  is the greatest distance between  $x$  and any other node in the graph, i.e.,  $\varepsilon(x) = \max\{d(x, y) | y \in V, y \neq x\}$ .*

The weight function  $w(x) = \varepsilon(x)$  creates the equivalent of a complete graph, and therefore the set  $S = V$  is always an OLD-set for this weight function. Thus it is possible for a mixed-weight OLD-set to exist even if a non-weighted OLD-set does not exist. For instance, if two nodes,  $x$  and  $y$ , share the same open incoming-ball, and  $x$  is given weight  $w$  so that it can be in the open incoming-ball of  $y$ , then a mixed-weight OLD-set exists for any weight function where  $w(x) \geq w$ .

**Example 3.1** *Figure 3.2 shows an example of a mixed-weight OLD-set in a graph that does not contain an OLD-set. In Figure 3.2a, nodes  $x_1$  and  $x_3$  share open incoming-ball*



**Figure 3.2:** Mixed-Weight OLD-set Example: (a) The graph does not have an OLD-set when  $w(x) = 1 \forall x$ . (b) Nodes  $x_1$  and  $x_3$  share open incoming-ball  $\{x_2, x_4\}$ . (c) The effect of the weighted node,  $w(x_3) = 2$ , is indicated by dashed arcs, and one mixed-weight OLD-set,  $\{x_3, x_4, x_5, x_6\}$ , is shown with nodes in black. © 2017 IEEE (d) When  $w(x_3) = 2$ , the open incoming-ball is unique for each node, allowing for a mixed-weight OLD-set. © 2017 IEEE

$\{x_2, x_4\}$  when  $w(x) = 1 \forall x \in V$ . If node  $x_3$  is given weight 2, as in Figure 3.2c, then every open incoming-ball becomes unique, allowing for a mixed-weight OLD-set. One minimum sized mixed-weight OLD-set is  $\{x_3, x_4, x_5, x_6\}$ .

## 3.2 A Similar Problem: Identifying Codes

An *identifying code* (*ID-code*) is a set of nodes,  $S \subseteq V$ , such that every node in the graph has a unique and non-empty set of neighbors in the  $S$ , where a node  $x$  is considered a neighbor of itself.

**Definition 3.9** *The closed incoming-ball,  $B^+[x]$ , is the set of all nodes within a distance of  $w(x)$  from  $x$ , i.e.,  $B^+[x] = \{y \in V | d(x, y) \leq w(x)\}$ .*

**Definition 3.10** *The mixed-weight identifying code or mixed-weight ID-code is a set of nodes  $S \subseteq V$  such that the closed incoming-ball is non-empty and unique for every  $x \in V$ .*

**Definition 3.11** *Two nodes,  $x$  and  $y$ , in a graph are twins if they are adjacent and have the same closed incoming-ball. We say a graph is twin-free if it contains no twins.*

For given graph  $G = (V, E)$  and weight function  $w(x)$ , a mixed-weight ID-code  $S \subseteq V$  exists if and only if  $\forall x, y \in V$  and  $x \neq y$ ,  $B^+[x] \neq B^+[y]$ , i.e., if the closed incoming-ball is unique for all nodes in the graph. A mixed-weight ID-code exists on a graph  $G = (V, E)$  if and only if  $G$  is twin-free. Thus, if a non-weighted ID-code does not exist, then no mixed-weight ID-code exists. The existence of an OLD-set does not imply the existence of an ID-code, and vice versa.

## 3.3 Discussion

The mixed-weight OLD-set addresses a unique problem in wireless sensor networks and other monitoring systems. Previously no model has been provided for systems that use sensors of varying strengths. For mixed-weight OLD-sets, we model the strength of a sensor in a network as a weighted node in a graph. This weight is representative of the distance a sensor can monitor in the field. In this chapter we provided definitions and demonstrated properties of mixed-weight OLD-sets. We also introduced the related

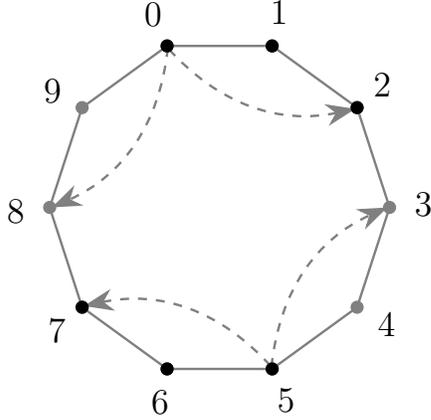
problem of mixed-weight identifying codes. This novel approach to location-detection problems provides a way to represent a broader range of sensor networks.

## Chapter 4

# Problems and NP-Completeness

In this chapter we introduce three optimization problems based on the mixed-weight OLD-set. These problems are finding the minimum size of an mixed-weight OLD-set for a graph and weight function, finding the weight function that produces the minimum mixed-weight OLD-set for a graph, and finding the minimum total weight of the mixed-weight OLD-set in a graph. We show that the decision problem for determining the minimum size of the mixed-weight OLD-set is NP-complete for all weight functions with weights  $\leq d$ , for any positive integer  $d$ . The non-weighted OLD-set decision problem was shown to be NP-complete in [72], and the weighted ID-code and LD-set decision problems, where all nodes receive the same weight  $d$ , were shown to be NP-complete in [13].

In Section 4.1 we show that finding the minimum mixed-weight OLD-set for a graph and weight function is NP-complete. We discuss finding the minimum mixed-weight OLD-set for a graph by considering every possible weight function in Section 4.2. In Section 4.3 we introduce the problem of finding the minimum total weight, which is related to the problem of reducing the total cost of a sensor network. We conclude in Section 4.4. The work presented in this chapter has been published in [34] © 2017 IEEE.



**Figure 4.1:** A cycle with nodes  $\{0, 1, \dots, 9\}$  and weight function  $w(x) = 2$  for  $x \in \{0, 5\}$  and  $w(x) = 1$  otherwise, has a minimum mixed-weight OLD-set of size 6, shown in black.

#### 4.1 Minimum Size of Mixed-weight OLD-set

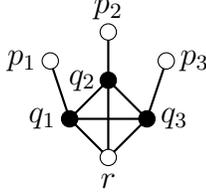
Consider a graph  $G = (V, E)$  and a weight function  $w(x)$ . We want to determine the smallest set  $S$  that is a mixed-weight OLD-set. Alternatively, we can ask the question, for graph  $G = (V, E)$ , weight function  $w(x)$ , and positive integer  $k \leq |V|$ , is there a mixed-weight OLD-set  $S$  where  $|S| \leq k$ ?

**Example 4.1** Consider a cycle of size 10 with nodes labeled  $\{0, 1, 2, \dots, 9\}$  and weight function  $w(x) = 2$  for  $x \in \{0, 5\}$  and  $w(x) = 1$  otherwise, as seen in Figure 4.1. The minimum size of the OLD-set is 6 and one satisfying OLD-set is  $S = \{0, 1, 2, 5, 6, 7\}$ .

The decision problem for finding the minimum mixed-weight OLD-set in a graph is as follows.

**NAME:** Mixed-weight open locating-dominating set (MW-OLD).

**INSTANCE:** A connected graph  $G = (V, E)$ , weight function  $w(x) \leq d, \forall x$ , where  $d$  is a positive integer, and  $k \leq |V|$  is a positive integer.



**Figure 4.2:** Subgraph  $\Delta$  for the MW-OLD decision problem. © 2017 IEEE

**QUESTION:** Is there a mixed-weight OLD-set  $S \subseteq V$  of size at most  $k$ ?

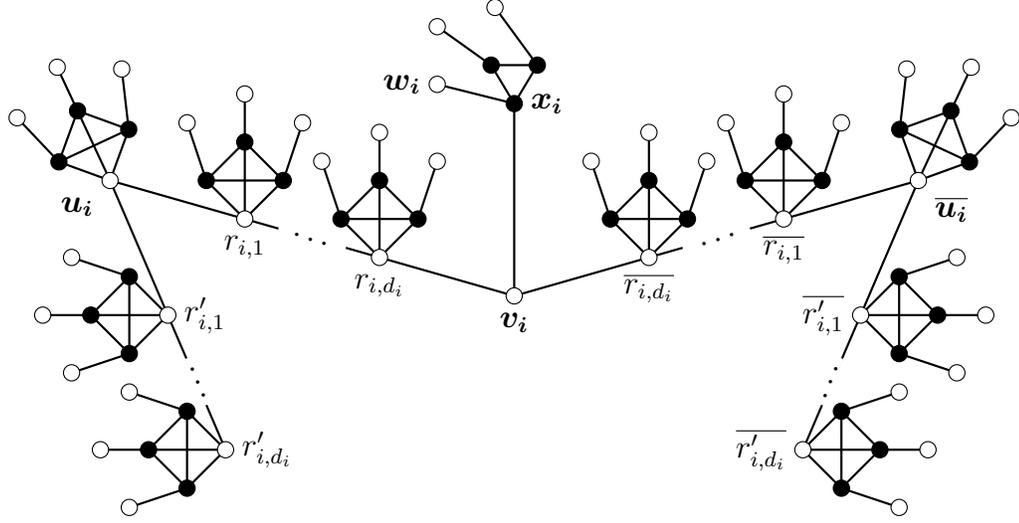
We prove that this decision problem is NP-complete, requiring the following lemmas.

**Lemma 4.1** *Let  $P = \{p_1, p_2, p_3\}$ ,  $Q = \{q_1, q_2, q_3\}$ , and  $\Delta$  be a subgraph of a graph  $G = V, E$  with node set  $V_\Delta = P \cup Q \cup \{r\}$  and edge set  $E_\Delta = \{(q_i, p_i), (q_i, q_j), (q_i, r) | 1 \leq i, j \leq 3, i \neq j\}$ , as show in Figure 4.2, and suppose that  $r$  is the only node in  $V_\Delta$  that is adjacent to another node in  $G$ . If all nodes in  $V - (P \cup Q)$  are not in the open incoming-ball of the nodes in  $P$ , then  $Q$  must be in any OLD-set of  $G$ .*

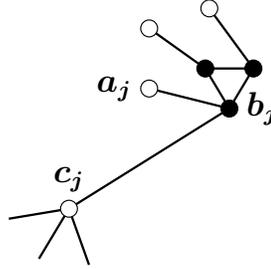
**Proof:** The nodes in  $P$  are endpoints, therefore  $Q$  must be in the OLD-set of  $G$  so that the OLD-set is open-dominating. ■

**Lemma 4.2** *Let  $P$ ,  $Q$ , and  $\Delta$  be as defined in Lemma 4.1. If  $r$  is the only node in  $V - (P \cup Q)$  that is in the open incoming-ball of the nodes in  $P$ , then  $Q$  must be in the minimum OLD-set of  $G$ .*

**Proof:** First, we note that if  $Q$  is in the OLD-set, then all of the nodes in  $\Delta$  are covered. WLOG, suppose that  $q_1$  is not in the OLD-set. Then  $p_3$  must be in the OLD-set, so that  $p_2$  and  $q_3$  are covered,  $p_2$  must be in the OLD-set so that  $p_3$  and  $q_2$  are covered, and  $r$  must be in the OLD-set so that  $p_1$  is covered. Thus in the minimum OLD-set of  $G$ ,  $Q$  must be in the OLD-set. ■



**Figure 4.3:** Literal Component  $G_i$  for the MW-OLD decision problem.  
 © 2017 IEEE



**Figure 4.4:** Clause Component  $H_j$  for the MW-OLD decision problem.  
 © 2017 IEEE

**Lemma 4.3** *Let  $P$ ,  $Q$ , and  $\Delta$  be as defined in Lemma 4.1. If  $x \neq r$  is the only node in  $V - (P \cup Q)$  that is in the open incoming-ball of the nodes in  $P$ , then  $Q$  must be in the minimum OLD-set of  $G$ .*

**Proof:** First, we note that if  $Q$  is in the OLD-set, then all of the nodes in  $\Delta$  are covered. WLOG, suppose that  $q_1$  is not in the OLD-set. Then  $r$  must be in the OLD-set, so that  $p_2$ ,  $p_3$ ,  $q_2$ , and  $q_3$  are covered, and  $x$  must be in the OLD-set so that  $p_1$  is covered. Thus in the minimum OLD-set of  $G$ ,  $Q$  must be in the OLD-set. ■

We now prove the decision problem for finding the minimum mixed-weight OLD-set

in a graph with weight function  $w(x) \leq d$ ,  $\forall x \in V$ , where  $d$  is a positive integer, is NP-complete.

**Theorem 4.4** *MW-OLD is NP-complete.*

**Proof:** First we show that MW-OLD is in NP. Given a subset  $S \subseteq V$ , constructing  $B^+(x) \cap S$  for each  $x \in V$  can be done in polynomial time on  $|V|$  since finding the shortest path and the intersection are both polynomial time. Then, determining that  $B^+(x) \cap S$  is unique and nonempty for  $x \in V$  can also be done in polynomial time on  $|V|$ .

Next, we polynomially reduce 3-SAT to MW-OLD, inspired by the proof of OLD in [72]. Given an instance of 3-SAT with literals  $U = \{u_1, u_2 \dots u_N\}$  and clauses  $C = \{c_1, c_2 \dots c_M\}$  we construct a literal component  $G_i$ ,  $1 \leq i \leq N$ , for each literal as seen in Figure 4.3, and a clause component  $H_j$ ,  $1 \leq j \leq M$ , for each clause as shown in Figure 4.4. Let  $\Delta$  be a subgraph as shown in Figure 4.2.

Let  $d_i = i \pmod{d}$ , and let  $r_{i,0}, r'_{i,0} = u_i$  and  $\overline{r_{i,0}}, \overline{r'_{i,0}} = \overline{u_i}$ . In the literal component  $G_i$ , there is a set of  $d_i$   $\Delta$  subgraphs between  $v_i$  and  $u_i$ , with  $v_i$  adjacent to  $r_{i,d_i}$ , and  $r_{i,k}$  adjacent to  $r_{i,k+1}$  for  $0 \leq k < d_i$ . There is another set of  $d_i$   $\Delta$  subgraphs between  $v_i$  and  $\overline{u_i}$  with adjacency defined similarly for  $\overline{r_{i,k}}$  for  $0 \leq k < d_i$ . There are two more sets of  $d_i$   $\Delta$  subgraphs, where  $r'_{i,k}$  is adjacent to  $r'_{i,k+1}$ , and  $\overline{r'_{i,k}}$  is adjacent to  $\overline{r'_{i,k+1}}$ ,  $0 \leq k < d_i$ . Node  $v_i$  is also adjacent to the subgraph containing  $x_i$  and  $w_i$ , and nodes  $u_i$  and  $\overline{u_i}$  are each a part of their own  $\Delta$  subgraph.

In the clause component  $H_j$ , node  $c_j$  is adjacent to subgraph containing  $b_j$  and  $a_j$ . For clause  $C_j$ ,  $1 \leq j \leq M$ , with literals  $u_{j,1}, u_{j,2}, u_{j,3}$ , connect node  $c_j$  to  $r_{j,\ell}$  where  $r_{j,\ell}$  is  $r'_{i,d_i}$  if  $u_{j,\ell} = u_i$  or  $r_{j,\ell}$  is  $\overline{r'_{i,d_i}}$  if  $u_{j,\ell} = \overline{u_i}$ , for  $1 \leq \ell \leq 3$ .

We also construct weight function  $w(x) = d_i + 1$ , i.e.  $1 \leq w(x) \leq d$ , if  $x$  is a node representing the literal  $u_i$  or its complement  $\overline{u_i}$ , and  $w(x) = 1$  otherwise. Given this graph and weight function, we have constructed a graph  $G = (V, E)$  with  $|V| = 7M + 21N + 28 \sum_{i=1}^N d_i$  and  $|E| = 10M + 27N + 40 \sum_{i=1}^N d_i$ , and this can be done in polynomial time. We also set  $k = 3M + 10N + 3 \sum_{i=1}^N d_i$ .

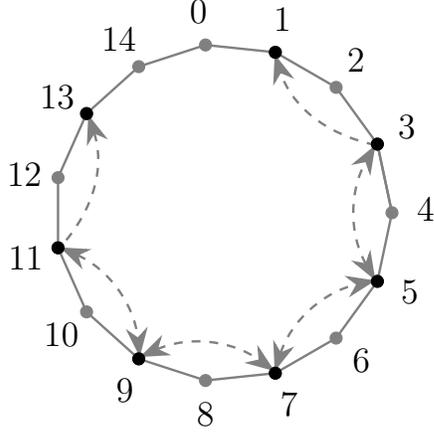
If  $S \subseteq V$  is a mixed-weight OLD-set on  $G$ , then  $S$  must contain every node adjacent to a node of degree 1, to satisfy the dominating property of the OLD-set. By construction, if a literal or its complement is in the OLD-set, then, by Lemmas 4.1, 4.2, and 4.3, for nodes in  $G$  that form a  $\Delta$  subgraph it is best to include the nodes equivalent to  $q_1$ ,  $q_2$ , and  $q_3$ , as shown in Figure 4.2, in the OLD-set. Thus a minimum construction of  $S$  must contain all the black nodes shown in Figures 4.3 and 4.4 for a total of  $3M + 9N + 3 \sum_{i=1}^N d_i$ . To satisfy the locating property of the OLD-set, we must have  $B^+(w_i) \neq B^+(v_i)$ , so at least one of  $u_i$ ,  $\overline{u_i}$ ,  $r_{i,d_i}$ , and  $\overline{r_{i,d_i}}$  must be in  $S$ . Similarly,  $B^+(a_j) \neq B^+(c_j)$ , so at least one of  $u_{j,1}, u_{j,2}, u_{j,3}, r_{j,1}, r_{j,2}, r_{j,3}$  must be in  $S$ . Thus given  $k = 3M + 10N + 3 \sum_{i=1}^N d_i$ , it is clear that C is satisfiable if and only if  $|S| = k$ . ■

**Corollary 4.5** *The decision problem for finding the minimum mixed-weight OLD-set in a graph with weight function with  $d$  possible weights, including weight 1, i.e.  $w(x) \in \{w_1, w_2, \dots, w_d\}$ ,  $\forall x \in V$ , where  $w_h$  is a positive integer for  $1 \leq h \leq d$  and  $\exists h$  where  $w_h = 1$ , is NP-complete.*

**Proof:** Define  $d_i = w_{i'} - 1$ , where  $i' = i \pmod{d} + 1$ , so  $d_i \in \{w_1 - 1, w_2 - 1, \dots, w_d - 1\}$ . Define the literal components  $G_i$ , the clause components  $H_j$ , and the weight function  $w(x)$  as above, so  $w(x) \in \{w_1, w_2, \dots, w_d\}$  if  $x$  is a node representing the literal  $u_i$  or its complement  $\overline{u_i}$ , and  $w(x) = 1$  otherwise. The rest of the proof follow similarly to the proof in Theorem 4.4. ■

## 4.2 Minimizing the Size of a Mixed-weight OLD-set

To minimize the size of a mixed-weight OLD-set, we are given a graph  $G = (V, E)$ , and we want to determine a weight function  $w(x)$  that gives a minimum sized mixed-weight OLD-set. Alternatively, we can ask the question, for graph  $G = (V, E)$  and positive integers  $j, k \leq |V|$ , is there a weight function  $w(x)$  with  $\max(w(x)) = j$ ,  $\forall x \in V$ , that allows for a mixed-weight OLD-set  $S$  such that  $|S| \leq k$ ?



**Figure 4.5:** A minimizing weight function for a cycle with nodes  $\{0, 1, \dots, 14\}$  is  $w(x) = 2$  for  $x \in \{3, 5, 7, 9, 11\}$  and  $w(x) = 1$  otherwise which produces a mixed-weight OLD-set of size 7, shown in black.

**Example 4.2** Consider a cycle of size 15 with nodes labeled  $\{0, 1, 2, \dots, 14\}$ . One minimizing weight function for  $\max(w(x)) = 2, \forall x \in V$ , is  $w(x) = 2$  for  $x \in \{3, 5, 7, 9, 11\}$  and  $w(x) = 1$  otherwise, as seen in Figure 4.5. The minimum OLD-set size is 7, for example,  $S = \{1, 3, 5, 7, 9, 11, 13\}$ . The minimum OLD-set when  $w(x) = 2$  for all nodes is also size 7, and the minimum OLD-set with  $w(x) = 1 \forall x$  is 10. Thus in this particular graph, a mixed-weight OLD-set provides a smaller sized OLD-set than using all weight 1 nodes, and provides the same minimum sized OLD-set when using all weight 2 nodes.

A particular application of this problem is in the design of WSNs. If we want to make the network as small as possible in order to reduce administrative cost, then a multitude of potential weight functions should be considered. Given that Theorem 4.4 shows that with a single weight function the problem is NP-complete, solving the problem with all potential weight functions will also be NP-complete.

We note that increasing the weight of nodes does not necessarily reduce the size of the mixed-weight OLD-set. For instance, increasing the weight to the eccentricity of each node,  $w(x) = \varepsilon(x) \forall x$ , which creates the equivalent of a complete graph, would result in a smallest mixed-weight OLD-set of size  $|V| - 1$ . A complete graph requires  $|V| - 1$  nodes

in an OLD-set because every subset with fewer than  $|V| - 1$  nodes is in the neighborhood of at least two nodes.

### 4.3 Minimizing the Total Weight of a Mixed-weight OLD-set

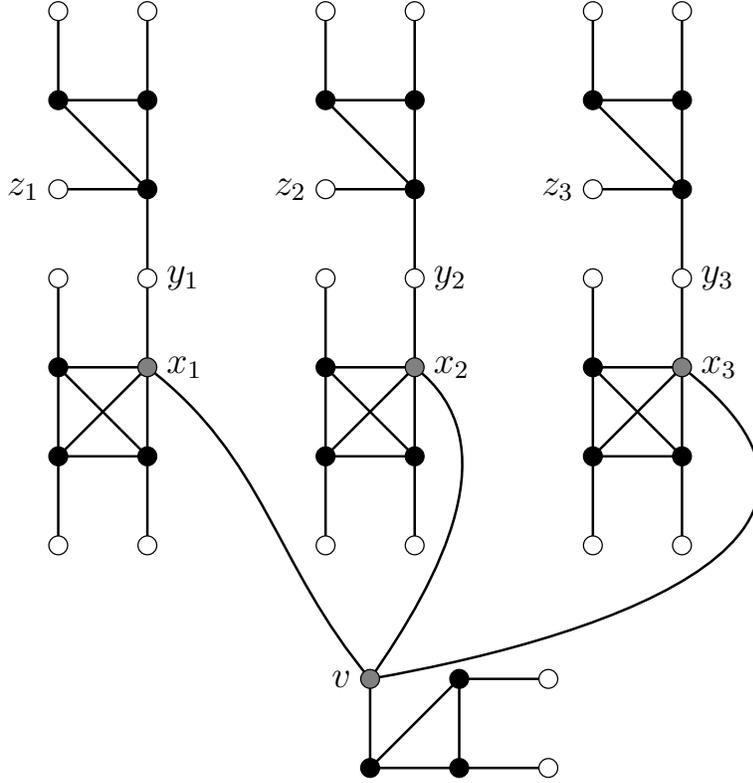
To minimize the total weight of a mixed-weight OLD-set, we are given a graph  $G = (V, E)$ , and we want to determine weight function  $w(x)$  that gives a minimum total weight of all the nodes in the mixed-weight OLD-set. Alternatively, we can ask the question, for graph  $G = (V, E)$  and  $k \leq |V|^2$ , is there a weight function  $w(x)$  that allows for a mixed-weight OLD-set  $S$  such that  $w(S) \leq k$ ?

It is possible to reduce  $w(S)$  by increasing the weight of particular nodes in the graph, but this is not possible for all graphs. An example of reducing the total weight is seen in Figure 4.6. If all nodes are weight 1, each  $x_i$  must be in the OLD-set so that the  $y_i$  and  $z_i$  do not share OLD-set neighborhoods, resulting in  $w(S) = 24$ . However, if  $w(v) = 2$ , then the  $x_i$  do not need to be in the OLD-set if  $v$  is in the OLD-set, resulting in  $w(S) = 23$ . In both scenarios the OLD-set described is minimally sized.

This problem is also applicable to WSNs. The total cost of the network could be reduced by replacing several cheaper, weaker sensors with a single stronger sensor. This could be extended by considering that the actual cost of a sensor is not necessarily proportional to its strength. In that case a cost function based on the weight could be introduced to the problem, with the goal of reducing the total cost of the system.

The following proposition gives bound on the total weight of an mixed-weight OLD-set in a path.

**Proposition 4.6** *A path with  $n$  nodes does not have a mixed-weight OLD-set with total weight  $w(S) < \lceil (2/3)n \rceil$ .*



**Figure 4.6:** If node  $v$  is given weight 2, the weight 1 nodes  $x_1$ ,  $x_2$ , and  $x_3$  no longer need to be in the mixed-weight OLD-set. With  $v$  in the mixed-weight OLD-set, the total weight is reduced by 1. Other nodes that need to be in the mixed-weight OLD-set are shown in black.

**Proof:** From [72] we know that in a path with unweighted nodes, for every six consecutive nodes in the path, at least four of those nodes must be in the OLD-set resulting in  $|S| = w(S) \geq \lceil (2/3)n \rceil$ . If there exists a weight function  $w(x) \leq 2$  such that  $w(S) < \lceil (2/3)n \rceil$ , then at least one set of six consecutive nodes in the path  $X = \{x_i, x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4}, x_{i+5}\}$  must have  $w(X \cap S) = 3$ . It cannot be the case that  $|X \cap S| = 3$  weight 1 nodes. If  $X \cap S$  contains one weight 2 node and one weight 1 node, then there are only 3 unique and nonempty subsets of  $X \cap S$ . Thus there are not enough unique, non-empty subsets to locate and dominate all six nodes in  $X$ . This result follows similarly for  $w(x) \leq d$  for  $d > 2$ . ■

## 4.4 Discussion

In this chapter we introduced optimization problems related to the mixed-weight open locating-dominating set. Similar to other location-detection problems, we showed that finding the minimum mixed-weight OLD-set is NP-complete for all weight functions with weights up to positive integer  $d$ . We also introduced the problem of finding the minimum total weight of nodes in the mixed-weight OLD-set, which is related to reducing the total cost of a sensor network. Determining the smallest number of sensors and reducing the overall cost of system of sensors is an important problem in fields that use networks of sensors, such as wireless sensor networks. This is the first time optimization problems have been considered for systems that have sensors with different strengths and costs.

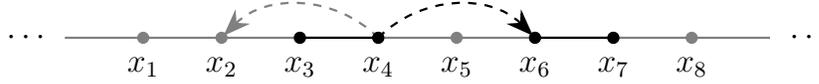
## Chapter 5

# Mixed-Weight OLD-sets in Paths and Cycles

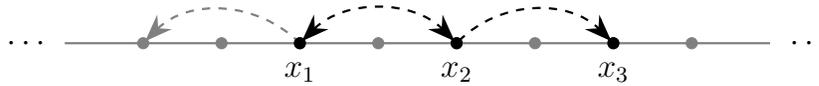
Given the simplicity of paths and cycles, they often fall into a special subset of an NP-complete problem that can be solved in polynomial time. For example, minimum-sized identifying codes [12] and minimum-sized open locating-dominating sets [72] in paths can both be found in linear time. Identifying codes and locating-dominating sets have also been studied in cycles [8, 49]. Extensive work has been done for the weighted  $d$ -identifying code and  $d$ -locating-dominating set, where all nodes receive the same weight  $d$ , in paths and cycles [8, 15, 25, 26, 39, 44, 70]. We will study the mixed-weight OLD-set in paths and cycles with a focus on weights 1 and 2. This will be the first time paths and cycles have been studied in weighted or mixed-weight OLD-sets. We cover mixed-weight OLD-sets with weights 1 and 2 in paths in Section 5.1, and in cycles in Section 5.2. We conclude in Section 5.3.

### 5.1 Mixed-Weight OLD-sets in Paths

Consider a path  $P_n$  with  $n$  nodes, labeled  $\{0, 1, 2, \dots, n-1\}$ , and its corresponding weight function  $w(x) \geq 1$ . We consider the mixed-weight OLD-set problem for weights



**Figure 5.1:** In this portion of a path or cycle,  $w(x_4) = 2$ , and nodes  $x_3, x_4, x_6, x_7$  are in the mixed-weight OLD-set and form a weakly connected 4-cluster. Nodes  $x_2, x_5, x_8$  are not in the mixed-weight OLD-set but are neighbors of the cluster.



**Figure 5.2:** A potential 3-cluster in a mixed-weight OLD-set on a path when considering weights 1 and 2.

1 and 2. Let  $OLD(P_n, w)$  be the size of the smallest mixed-weight OLD-set on  $P_n$  with weight function  $w(x) \leq 2$ . We show that  $OLD(P_n, w) \geq \frac{2n}{5}$ . We then find the minimum-sized mixed-weight OLD-sets in paths with all weight 2 nodes and paths with weight 1 and 2 nodes.

### 5.1.1 Lower Bound for Paths with Weight 1 and 2 Nodes

We find a lower bound on the mixed-weight OLD-set in cycles  $C_n$  to be  $\frac{2n}{5}$ . We note that this bound cannot be reached in paths.

For mixed-weight OLD-sets, we define an  $m$ -cluster as a weakly connected component of order  $m$  in the graph induced by an OLD-set  $S$ , where node  $x$  with  $w(x) = 2$  is considered to have arcs from  $x$  to  $x - 2$  and  $x + 2$ , as seen in Figure 5.1. We define a *neighbor* of a cluster as a node that has at least one node in its open incoming-ball that is in the cluster. All neighbors of a cluster cannot be in the OLD-set.

**Theorem 5.1**  $OLD(P_n, w) \geq \frac{2n}{5}$  where  $w(x) \leq 2 \forall x$ .

**Proof:** This proof follows similarly to the proof of Theorem 2.3 using the definition of an  $m$ -cluster and open incoming-ball for mixed-weight OLD-sets. However, with weights 1 and 2, there is one 3-cluster with more than  $\lfloor 3m/2 \rfloor = 4$  neighbors,  $\{x_1, x_2, x_3\}$  with

$n \geq 8$	$ S $	$S = \{10i + 1, 10i + 3, 10i + 5, 10i + 7 \mid 0 \leq i < \ell\} \cup$
$10\ell, 10\ell + 1$	$4\ell + 1$	$\{10\ell - 1\}$
$10\ell + 2, 10\ell + 3$	$4\ell + 2$	$\{10\ell - 1, 10\ell + 1\}$
$10\ell + 4, 10\ell + 5$	$4\ell + 3$	$\{10\ell - 1, 10\ell + 1, 10\ell + 3\}$
$10\ell + 6, 10\ell + 7$	$4\ell + 4$	$\{10\ell - 1, 10\ell + 1, 10\ell + 4, 10\ell + 5\}$
$10\ell + 8, 10\ell + 9$	$4\ell + 4$	$\{10\ell + 1, 10\ell + 3, 10\ell + 5, 10\ell + 7\}$

**Table 5.1:** One set of satisfying minimum mixed-weight OLD-sets  $S$  for paths with  $n = 10\ell + j \geq 8$  nodes. All nodes are weight 2, and all paths have the same initial  $4(\ell - 1)$  OLD-set nodes.

$x_i = x_{i-1} + 2$  and at least one of  $x_1$  or  $x_3$  having a weight of 2. WLOG, suppose  $w(x_1) = 2$ . If  $w(x_2) = 1$ , then  $x_1$  will not have any nodes in its mixed-weight neighborhood. If  $w(x_2) = w(x_3) = 2$ , then  $x_1$  and  $x_3$  will share mixed-weight neighborhood  $\{x_2\}$ . If  $w(x_2) = 2$  and  $w(x_3) = 1$ , then the neighbors of the cluster are  $x_1 - 2$ ,  $x_1 - 1$ ,  $x_1 + 1$ ,  $x_2 + 1$ , and  $x_3 + 1$ , as seen in Figure 5.2. In order for nodes  $x_1$  and  $x_3$  to not share mixed-weight neighborhood  $\{x_2\}$ ,  $x_2 + 3$  must be in the OLD-set with  $w(x_3 + 2) = 2$ . In this case,  $x_3 + 1$  will be the neighbor of two clusters. Thus  $x_3 + 1$  will receive at most  $\frac{1}{5}$  charge from the cluster and the remaining four neighbors will receive at most  $\frac{2}{5}$  charge, leaving the cluster with at least  $\frac{6}{5}$  charge. ■

### 5.1.2 Paths with All Weight 2 Nodes

For the minimum sized mixed-weight OLD-sets in paths, we begin with all weight 2 nodes. Let  $OLD(P_n, w_2)$  be the size of the smallest mixed-weight OLD-set on  $P_n$  with weight function  $w(x) = 2$ . We say a graph is *minimally covered* by a minimum-sized mixed-weight OLD-set.

Table 5.1 shows one set of satisfying minimum mixed-weight OLD-sets, where all nodes are weight 2, for paths with  $n \geq 8$  nodes. We note that  $\{1, 2, 4, 5\}$  is a similar sized minimum mixed-weight OLD-set for  $n = 6, 7$ .

**Theorem 5.2**  $OLD(P_n, w_2) > \frac{2n}{5}$  for  $n \geq 6$ , where  $w(x) = 2, \forall x$ . Specifically, for  $n = 10\ell + j$  nodes,  $|S| = 4\ell + j'$  where  $j' = \lfloor j/2 \rfloor + 1$  for  $0 \leq j \leq 7$ , and  $j' = \lfloor j/2 \rfloor$  for  $j = 8, 9$ .

**Proof:** Each of the sets in Table 5.1 can be easily verified as mixed-weight OLD-sets. For  $8 \leq n \leq 15$ , the minimum mixed-weight OLD-set sizes can be verified as a minimum by exhaustive search. We note that  $\{1, 2, 4, 5\}$  is a similar sized minimum mixed-weight OLD-set for  $n = 6, 7$ . We now show that the sizes given are minimum by induction, and assume the sizes given in Table 5.1 are minimum for all paths up to some  $k \geq 15$ .

For  $k + 1 = 10\ell + j$  nodes, with nodes labeled  $0, 1, \dots, k$ , consider the first  $k - 9 = 10(\ell - 1) + j$  and the last 10 nodes. The first  $k - 9$  nodes can be minimally covered by  $4(\ell - 1) + j'$  nodes where  $j' = \lfloor j/2 \rfloor + 1$  for  $0 \leq j \leq 7$ , and  $j' = \lfloor j/2 \rfloor$  for  $j = 8, 9$ .

Of the  $4(\ell - 1) + j'$  nodes, the last node in the mixed-weight OLD set must be  $k - 10$ ,  $k - 11$ , or  $k - 12$ . If  $k - 10$  is the last node, then the  $4(\ell - 1) + j'$  nodes can cover up to node  $k - 8$ , leaving at least 8 nodes in the path to be covered. If  $k - 11$  is the last node, then the  $4(\ell - 1) + j'$  nodes can cover up to node  $k - 9$ , leaving at least 9 nodes in the path to be covered. If  $k - 12$  is the last node, then the  $4(\ell - 1) + j'$  nodes can cover up to node  $k - 10$ , leaving 10 nodes in the path to be covered. Thus at least 4 nodes from the last 10 nodes in the path must be in the OLD-set to cover at least 8 nodes. Therefore the mixed-weight OLD-set on the entire path at least  $4\ell + j'$  nodes where  $j' = \lfloor j/2 \rfloor + 1$  for  $0 \leq j \leq 7$ , and  $j' = \lfloor j/2 \rfloor$  for  $j = 8, 9$ . ■

### 5.1.3 Paths with Weight 1 and 2 Nodes

We find a minimum-sized mixed-weight OLD-set in  $P_n$  over all possible weight functions  $w(x) \leq 2$ , and discuss the number of nodes in the minimum mixed-weight OLD-set that can be weight 1. Table 5.2 shows one set of satisfying minimum mixed-weight OLD-sets, where nodes are weight 1 and 2, for paths with  $n \geq 8$  nodes. We note that  $\{1, 2, 4, 5\}$  with  $w(4) = 2$  is a similar sized minimum mixed-weight OLD-set for  $n = 6, 7$ .

$n \geq 8$	$ S $	$S = \{10i + 3, 10i + 5, 10i + 7   0 \leq i < \ell\} \cup$
$10\ell$	$4\ell + 1$	$\{\underline{1}, \underline{10\ell - 2}\} \cup \{10i + 9   0 \leq i < \ell - 1\}$
$10\ell + 1$	$4\ell + 1$	$\{\underline{1}\} \cup \{10i + 9   0 \leq i < \ell\}$
$10\ell + 2$	$4\ell + 2$	$\{\underline{1}, \underline{10\ell}\} \cup \{10i + 9   0 \leq i < \ell\}$
$10\ell + 3$	$4\ell + 2$	$\{\underline{1}, \underline{10\ell + 1}\} \cup \{10i + 9   0 \leq i < \ell\}$
$10\ell + 4$	$4\ell + 3$	$\{\underline{1}, \underline{10\ell + 1}, \underline{10\ell + 2}\} \cup \{10i + 9   0 \leq i < \ell\}$
$10\ell + 5$	$4\ell + 3$	$\{\underline{1}, 10\ell + 1, \underline{10\ell + 3}\} \cup \{10i + 9   0 \leq i < \ell\}$
$10\ell + 6$	$4\ell + 4$	$\{\underline{1}, 10\ell + 1, \underline{10\ell + 4}, \underline{10\ell + 5}\} \cup \{10i + 9   0 \leq i < \ell\}$
$10\ell + 7$	$4\ell + 4$	$\{\underline{1}, 10\ell + 3, \underline{10\ell + 4}, \underline{10\ell + 5}\} \cup \{10i + 9   0 \leq i < \ell\}$
$10\ell + 8$	$4\ell + 4$	$\{10\ell + 3, 10\ell + 5, \underline{10\ell + 6}\} \cup \{10i + 1   0 \leq i \leq \ell\}$
$10\ell + 9$	$4\ell + 4$	$\{10\ell + 3, 10\ell + 5, 10\ell + 7\} \cup \{10i + 1   0 \leq i \leq \ell\}$

**Table 5.2:** One set of satisfying minimum mixed-weight OLD-sets  $S$  for paths with  $n = 10\ell + j \geq 8$  nodes that are weight 1 or 2. All paths have the same  $3(\ell - 1)$  internal OLD-set nodes. Underlined nodes can be weight 1.

**Theorem 5.3**  $OLD(P_n, w) > \frac{2n}{5}$  for  $n \geq 6$ , where  $w(x) \leq 2 \forall x$ . Specifically, for  $n = 10\ell + j$  nodes,  $|S| = 4\ell + j'$  where  $j' = \lfloor j/2 \rfloor + 1$  for  $0 \leq j \leq 7$ , and  $j' = \lfloor j/2 \rfloor$  for  $j = 8, 9$ .

**Proof:** Each of the sets in Table 5.2 can be easily verified as mixed-weight OLD-sets. For  $8 \leq n \leq 15$ , the mixed-weight OLD-sets can be verified as a minimum by exhaustive search. We note that the minimum mixed-weight OLD-set for  $n = 6, 7$  is size 4. For example  $\{1, 2, 4, 5\}$  with  $w(4) = 2$  is a mixed-weight OLD-set for  $n = 6, 7$ .

The remainder of the proof follows similarly to the proof of Theorem 5.2, allowing for the last node in the mixed-weight OLD-set for the first  $k - 9$  nodes to be weight 1 or 2. ■

Maximizing the number of weight 1 nodes in the mixed-weight OLD-set is of interest for a system in which we would want to minimize the number of sensors, such as in a WSN, and well as the number of strong, and potentially costly, sensors. Although Table 5.2 shows an example of minimum-sized mixed-weight OLD-sets for every path, it does not necessarily show the minimum-sized mixed-weight OLD-set with the maximum number of weight 1 nodes. For instance, when  $n = 10\ell + 6$  such that  $\ell > 1$ , you can choose integers  $\ell_1, \ell_2 > 0$  such that  $\ell_1 + \ell_2 = \ell$ . We know the path with  $n = 10\ell + 6$  can be minimally

covered by  $4\ell + 4$  nodes, and our example shows three weight 1 nodes. However, the first  $10\ell_1 + 3$  nodes can be minimally covered by  $4\ell_1 + 2$  nodes with two weight 1 nodes, and the last  $10\ell_2 + 3$  nodes can be minimally covered by  $4\ell_2 + 2$  nodes with two weight 1 nodes. Thus the path with  $n = 10\ell + 6$ , with  $\ell > 1$ , nodes can be minimally covered by  $4\ell_1 + 2 + 4\ell_2 + 2 = 4\ell + 4$  nodes with four weight 1 nodes in the mixed-weight OLD-set.

**Example 5.1** *Consider a path  $P_n$  with  $n = 36$  nodes. The minimum OLD-set size is 16, and the example set in Table 5.2 shows three weight 1 nodes. However, the first 13 nodes can be minimally covered by 6 nodes with two weight 1 nodes, and the last 23 nodes can be covered by 10 nodes with two weight 1 nodes. Thus we can find a minimally sized mixed-weight OLD-set with four weight 1 nodes in a path of length 36.*

## 5.2 Mixed-Weight OLD-sets in Cycles

Consider a cycle  $C_n$  with  $n$  nodes, labeled  $\{0, 1, 2, \dots, n - 1\}$ , and its corresponding weight function  $w(x) \geq 1$ . We consider the mixed-weight OLD-set problem for possible weights 1 and 2. Let  $OLD(C_n, w)$  be the size of the smallest mixed-weight OLD-set on  $C_n$  with weight function  $w(x) \leq 2$ . We show that  $OLD(C_n, w) \geq \frac{2n}{5}$ , and find minimum-sized mixed-weight OLD-sets in cycles with all weight 2 nodes and cycles with weight 1 and 2 nodes.

### 5.2.1 Lower Bound for Cycles with Weight 1 and 2 Nodes

Similar to paths, we find a lower bound on the mixed-weight OLD-set in cycles  $C_n$  to be  $\frac{2n}{5}$ . We note that this bound can be reached in cycles with  $n = 10\ell$ .

**Theorem 5.4**  $OLD(C_n, w) \geq \frac{2n}{5}$  where  $w(x) \leq 2 \forall x$ .

**Proof:** This proof follows similarly to the proof of Theorem 5.1. ■

$n \geq 8$	$ S $	$S = \{10i + 1, 10i + 3, 10i + 5, 10i + 7 \mid 0 \leq i < \ell\} \cup$
$10\ell$	$4\ell$	$\emptyset$
$10\ell + 1$	$4\ell + 1$	$\{10\ell - 2\}$
$10\ell + 2$	$4\ell + 1$	$\{10\ell - 1\}$
$10\ell + 3, 10\ell + 4$	$4\ell + 2$	$\{10\ell + 1, 10\ell + 2\}$
$10\ell + 5$	$4\ell + 3$	$\{10\ell + 1, 10\ell + 3, 10\ell + 4\}$
$10\ell + 6$	$4\ell + 3$	$\{10\ell + 1, 10\ell + 3, 10\ell + 5\}$
$10\ell + j, j = 7, 8, 9$	$4\ell + 4$	$\{10\ell + 1, 10\ell + 3, 10\ell + 5, 10\ell + 6\}$

**Table 5.3:** One set of satisfying minimum mixed-weight OLD-sets  $S$  for cycles with  $n = 10\ell + j \geq 8$  nodes. All nodes are weight 2, and all cycles have the same initial  $4(\ell - 1)$  OLD-set nodes.

### 5.2.2 Cycles with All Weight 2 Nodes

For the minimum sized mixed-weight OLD-sets in cycles, we begin with all weight 2 nodes. Recall that a graph is *minimally covered* by a minimum-sized mixed-weight OLD-set. We note that a cycle with all weight 2 nodes is equivalent to the circulant graph  $C_n(1, 2)$ .

Table 5.3 shows one set of satisfying minimum mixed-weight OLD-sets, where all nodes are weight 2, for cycles with  $n \geq 8$  nodes. We note that  $\{1, 2, 4, 5\}$  is a similar sized minimum mixed-weight OLD-set for  $n = 7$ . When all nodes are weight 2, a cycle of size 6 does not have an OLD-set.

**Theorem 5.5**  $OLD(C_n, w_2) > \frac{2n}{5}$  for  $n \geq 7$ , where  $w(x) = 2, \forall x$ . Specifically, for  $n = 10\ell + j$  nodes,  $|S| = 4\ell + j'$  where  $j' = \lfloor (j + 1)/2 \rfloor$  for  $0 \leq j \leq 8$ , and  $j' = \lfloor j/2 \rfloor$  for  $j = 9$ .

**Proof:** Each of the sets in Table 5.3 can be easily verified as mixed-weight OLD-sets. For  $8 \leq n \leq 16$ , the mixed-weight OLD-sets can be verified as a minimum by exhaustive search. We note that  $\{1, 2, 4, 5\}$  is a similar sized minimum mixed-weight OLD-set for  $n = 7$ . We now show that the sizes given are minimum by induction, and assume the sizes given in Table 5.3 are minimum for all cycles up to some  $k \geq 16$ .

For  $k + 1 = 10\ell + j$  nodes, with nodes labeled  $0, 1, \dots, k$ , consider the first  $k - 10 = 10(\ell - 1) + j - 1$  nodes as a path segment, with 11 nodes remaining in the cycle. Considering the nodes as a path segment, we can use Theorem 5.2 to find the minimum number of mixed-weight OLD-set nodes needed to cover the nodes in the segment. If  $j = 0$ , then the first  $k - 10 = 10(\ell - 1) - 1 = 10(\ell - 2) + 9$  can be minimally covered by  $4(\ell - 2) + 4 = 4(\ell - 1)$  nodes. If  $1 \leq j \leq 8$ , then the  $k - 10 = 10(\ell - 1) + j - 1$  nodes can be minimally covered by  $4(\ell - 1) + \lfloor (j - 1)/2 \rfloor + 1 = 4(\ell - 1) + \lfloor (j + 1)/2 \rfloor$  nodes. If  $j = 9$ , then the  $k - 10 = 10(\ell - 1) + j - 1$  nodes can be minimally covered by  $4(\ell - 1) + \lfloor (j - 1)/2 \rfloor = 4(\ell - 1) + \lfloor j/2 \rfloor$  nodes. Thus the first  $k - 10 = 10(\ell - 1) + j - 1$  nodes, can be minimally covered by  $4(\ell - 1) + j'$  nodes, where  $j' = \lfloor (j + 1)/2 \rfloor$  for  $0 \leq j \leq 8$ , and  $j' = \lfloor j/2 \rfloor$  for  $j = 9$ .

The first node in the mixed-weight OLD-set from the  $k - 10 = 10(\ell - 1) + j - 1$  nodes is either node 0, 1, or 2. The last node in the mixed-weight OLD-set from the  $k - 10 = 10(\ell - 1) + j - 1$  nodes is either node  $k - 11$ ,  $k - 12$ , or  $k - 13$ .

If 0 is the first node and  $k - 13$  is the last node, then the mixed-weight OLD-set from the  $k - 10$  nodes can cover up to  $k - 6$  nodes, leaving at least 7 nodes in the path to be covered. If 0 is the first node and  $k - 2$  is the last node, then the mixed-weight OLD-set from the  $k - 10$  nodes can cover up to  $k - 7$  nodes, leaving at least 8 nodes in the path to be covered, etc.

Thus, from Theorem 5.2 at least 4 nodes from the last 11 nodes must be in the mixed-weight OLD-set to cover a path segment of at least 7 nodes. Therefore there are at least  $4(\ell - 1) + j' + 4 = 4\ell + j'$  nodes, where  $j' = \lfloor (j + 1)/2 \rfloor$  for  $0 \leq j \leq 8$ , and  $j' = \lfloor j/2 \rfloor$  for  $j = 9$ , in the mixed-weight OLD-set. ■

### 5.2.3 Cycles with Weight 1 and 2 Nodes

We find a minimum mixed-weight OLD-set in  $C_n, S$ , over all possible weight functions  $w(x) \leq 2$ . We also find the minimum number of nodes in  $S, M$ , that must be weight 2. Table 5.4 shows one set of satisfying minimum mixed-weight OLD-sets, where nodes are

$n \geq 8$	$ S $	$S = \{10i + 1, 10i + 3, 10i + 5, 10i + 7   0 \leq i < \ell\} \cup$
$10\ell$	$4\ell$	$\emptyset$
$10\ell + 1$	$4\ell + 1$	$\{\underline{10\ell - 1}\}$
$10\ell + 2$	$4\ell + 1$	$\{10\ell - 1\}$
$10\ell + 3$	$4\ell + 2$	$\{\underline{10\ell}, \underline{10\ell + 1}\}$
$10\ell + 4$	$4\ell + 2$	$\{10\ell + 1, 10\ell + 2\}$
$10\ell + 5$	$4\ell + 3$	$\{10\ell_1, \underline{10\ell + 2}, \underline{10\ell + 3}\}$
$10\ell + 6$	$4\ell + 3$	$\{10\ell_1, \underline{10\ell + 2}, 10\ell + 3\}$
$10\ell + 7$	$4\ell + 4$	$\{10\ell_1, \underline{10\ell + 2}, \underline{10\ell + 3}, \underline{10\ell + 6}\}$
$10\ell + 8$	$4\ell + 4$	$\{10\ell + 1, 10\ell + 3, 10\ell + 5, \underline{10\ell + 6}\}$
$10\ell + 9$	$4\ell + 4$	$\{10\ell + 1, 10\ell + 3, 10\ell + 5, 10\ell + 6\}$

**Table 5.4:** One set of satisfying minimum mixed-weight OLD-sets  $S$  for cycles with  $n = 10\ell + j \geq 8$  nodes that are weight 1 or 2. All cycles have the same initial  $4(\ell - 1)$  OLD-set nodes. Underlined nodes can be weight 1.

weight 1 and 2, for cycles with  $n \geq 8$  nodes. We note that  $\{1, 2, 4, 5\}$  with  $w(4) = 2$  is a similar sized minimum mixed-weight OLD-set for  $n = 6, 7$ .

**Theorem 5.6**  $OLD(P_n, w) > \frac{2n}{5}$  for  $n \geq 6$ , where  $w(x) \leq 2 \forall x$ . Specifically, for  $n = 10\ell + j$  nodes,  $|S| = 4\ell + j'$  where  $j' = \lfloor (j+1)/2 \rfloor$  for  $0 \leq j \leq 8$ , and  $j' = \lfloor j/2 \rfloor$  for  $j = 9$ .

**Proof:** Each of the mixed-weight OLD-sets in Table 5.4 can be easily verified. For  $8 \leq n \leq 15$ , the mixed-weight OLD-sets can be verified as a minimum by exhaustive search. We note that the minimum mixed-weight OLD-set for  $n = 6, 7$  is size 4. For example  $\{1, 2, 4, 5\}$  with  $w(4) = 2$  is a mixed-weight OLD-set for  $n = 6, 7$ .

The remainder of the proof follows similarly to the proof of Theorem 5.5, allowing for the first and last nodes in the mixed-weight OLD-set for the first  $k - 10$  nodes to be weight 1 or 2. ■

### 5.3 Discussion

Paths and cycles provide interesting insight into the way location-detection sets behave in the simplest of cases. Often, these graphs provide special case polynomial solutions to NP-complete problems. In this chapter we were able to show a general lower bound of  $\frac{2n}{5}$  for paths and cycles of length  $n$  with nodes of weights 1 and 2. We also found the optimal mixed-weight OLD-set for every path and cycle, and provided a linear solution at the optimal size. These results could help estimate mixed-weight OLD-sets in larger graphs that have sections that behave like paths and cycles

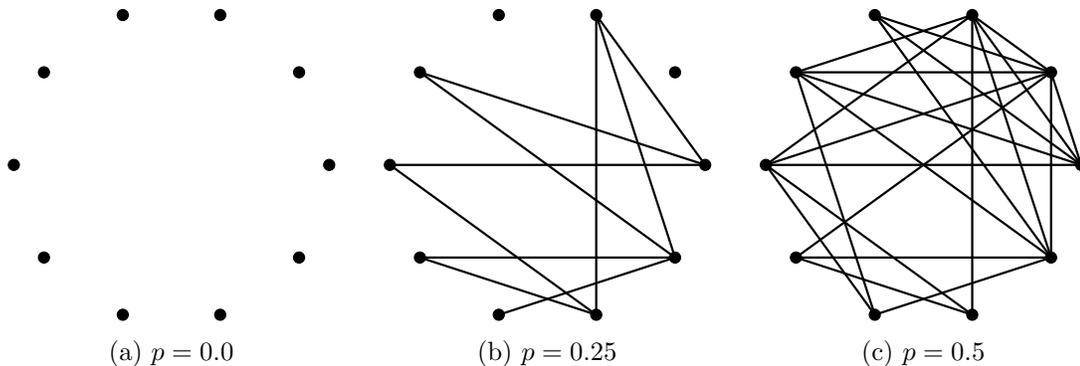
## Chapter 6

# Mixed-Weight OLD-Sets in Random Graphs

The structure of wireless sensor networks (WSNs) is often unknown until deployment making random graphs an important area of study. Random graphs  $G(n, p)$  are graphs with  $n$  nodes where every edge exists with probability  $p$  and does not exist with probability  $1 - p$ . When generating such a graph, the probability of obtaining a particular graph with  $m$  edges is  $p^m(1 - p)^{\binom{n}{2} - m}$ , which is the probability that  $m$  particular edges exist and the remaining edges do not exist [10]. Figure 6.1 shows random graphs with 10 nodes.

Random graphs have been studied in areas where the structure of the graph is unknown such as neural networks, social connections, the World Wide Web [9] and WSNs [24]. For many WSNs, sensors are dispersed from aircraft and then connect to each other after distribution [3], thus the structure of the network is unknown until after deployment. We study the mixed-weight OLD-set problem in random graphs with particular focus on bounds on the size of the mixed-weight OLD-set.

Identifying codes were studied in random graphs in [30]. We expand those results to include (mixed-weight) open locating-dominating sets and provide simulation results to show the accuracy of the theoretical bounds. To our knowledge, this is the first time OLD-sets have been studied in random graphs.



**Figure 6.1:** Random Graphs  $G(n, p)$  with  $n = 10$  and  $p = 0.0$ ,  $p = 0.25$ , and  $p = 0.5$ .

In Section 6.1 we show that almost every random graph has a mixed-weight OLD-set and provide a probabilistic bound for the size of the set. In Section 6.2 we provide simulation results to support our probabilistic bounds. We conclude our study of mixed-weight OLD-sets in random graphs in Section 6.3. The work presented in this chapter has been published in [34] © 2017 IEEE.

## 6.1 Mixed-Weight OLD-set Bounds in Random Graphs

We study the relationship between the probability of an edge existing in a random graph and the existence of a mixed-weight OLD-set in the following lemma. We also determine a bound on the size of a mixed-weight OLD-set in a random graph for weights 1 and 2, for weights  $a$  and  $b$ , where  $a$  and  $b$  are positive integers, and for weights  $1, 2, \dots, k$  in the subsequent theorems. For random graphs, almost every graph in  $G(n, p)$  has property  $X$  if, as  $n$  goes to infinity, the probability that  $G(n, p)$  has property  $X$  goes to 1.

**Lemma 6.1** *Almost every graph in  $G(n, p)$  has a mixed-weight OLD-set  $S$  with at most  $|S| \leq \frac{(2+\varepsilon)\log n}{\log 1/q}$  nodes, where  $q$  is the probability that a node is either in or not in both open incoming-balls of two other nodes. Values  $p$ ,  $1 - p$ , and  $\varepsilon$  are  $\omega(1/\log n)$ , i.e., the values are asymptotically greater than  $1/\log n$ .*

**Proof:** We note that this proof follows directly from the proof of identifying codes in  $G(n, p)$  in [30]. Let  $p'$  be the probability that any two nodes  $x \neq y$  are not in the open incoming-ball of each other. Consider a subset of nodes in a graph of  $G(n, p)$ ,  $S$ , with cardinality  $|S| = c$ , where  $c$  is the minimum mixed-weight OLD-set size. Let  $A_{x,y}(S)$  be the event that  $\{B^+(x) \cap S = B^+(y) \cap S\}$ , i.e.,  $A_{x,y}(S)$  is the event that two nodes share an open incoming-ball in  $S$  and thus  $S$  is not a mixed-weight OLD-set.

The probability that  $S$  is not a locating set is

$$Pr \left( \bigcup_{x,y} A_{x,y}(S) \right) \leq \sum_{x \neq y} Pr(A_{x,y}(S))$$

For nodes  $x \neq y$ , if  $x, y \in S$  then  $Pr(A_{x,y}(S)) = p'q^{c-2}$ , the probability that  $x$  and  $y$  are not neighbors and are adjacent to the same set of nodes in the set  $S$ . If, WLOG,  $x \in S$  and  $y \notin S$ , then  $Pr(A_{x,y}(S)) = p'q^{c-1}$  by the same reasoning. If  $x, y \notin S$ , then  $Pr(A_{x,y}(S)) = q^c$ , the probability that the two nodes are adjacent to the same set of nodes in  $S$ . There are  $\binom{c}{2}$  possible choices for  $x, y \in S$ ,  $c(n-c)$  possible choices for  $x \in S$  and  $y \notin S$ , and  $\binom{n-c}{2}$  possible choices for  $x, y \notin S$ . Thus

$$Pr \left( \bigcup_{x,y} A_{x,y}(S) \right) \leq \binom{c}{2} p'q^{c-2} + c(n-c)p'q^{c-1} + \binom{n-c}{2} q^c \quad (6.1)$$

$$\begin{aligned} &\leq c^2 p'q^{c-2} + c(n-c)p'q^{c-1} + (n-c)^2 q^c \\ &= (n-c)^2 q^c \left[ 1 + \frac{cp'}{(n-c)q} + \frac{c^2 p'}{(n-c)^2 q^2} \right] \end{aligned} \quad (6.2)$$

If  $c = n$ , then as  $n$  tends to  $\infty$ , equation (6.1) tends to 0, thus almost every graph in  $G(n, p)$  has a mixed-weight OLD-set. As  $n$  tends to  $\infty$ , the bracketed portion of equation (6.2) tends to 1, thus we consider  $(n-c)^2 q^c$  from equation (6.2). Let  $c = \frac{(2+\varepsilon) \log n}{\log 1/q}$ .

$$\begin{aligned}
(n-c)^2 q^c &= e^{2\log(n-c) + c\log q} \\
&= e^{2\log(n-c) - (2+\varepsilon)\log n} \\
&= \frac{(n-c)^2}{n^{2+\varepsilon}} \\
&\leq n^{-\varepsilon}
\end{aligned}$$

Since  $\varepsilon = \omega(1/\log n)$ , as  $n$  tends to  $\infty$ , the probability that  $S$  is not a mixed-weight OLD-set tends to 0. ■

**Theorem 6.2** *Almost every graph in  $G(n, p)$  has a mixed-weight OLD-set  $S$  with weights 1 and 2 with  $|S| \leq \frac{(2+\varepsilon)\log n}{\log 1/q}$  for*

$$q = (p + \varrho(1-p)p^2)^2 + (1-p)^2(1-p^2\varrho)^2,$$

where  $\varrho$  is the probability that a node is weight 2. Values  $p$ ,  $1-p$ , and  $\varepsilon$  are  $\omega(1/\log n)$ , i.e., the values are asymptotically greater than  $1/\log n$ .

**Proof:** Let  $q$  be the probability that a node  $z$  is either in or not in both open incoming-balls of two other nodes  $x \neq y$ . If  $z$  is in the open incoming-ball of a node  $x$ , then either  $z$  is adjacent to  $x$ , with a probability of  $p$ , or  $z$  is weight 2, not adjacent to  $x$ , and there exists a path of length 2 between  $z$  and  $x$ , with probability  $\varrho(1-p)p^2$ . If  $z$  is not in the open incoming-ball of a node  $x$ , then  $z$  is not adjacent to  $x$ , with probability  $1-p$ , and if  $z$  is weight 2, then there is no path of length 2 between  $z$  and  $x$ , with probability  $1-p^2$ . So the probability that  $z$  is in the open incoming-ball of  $x$  is  $p + \varrho(1-p)p^2$ , and the probability that  $z$  is not in the open incoming-ball of  $x$  is  $(1-p)((1-\varrho) + \varrho(1-p^2)) = (1-p)(1-p^2\varrho)$ . ■

**Theorem 6.3** *Almost every graph in  $G(n, p)$  has a mixed-weight OLD-set  $S$  with weights*

$a$  and  $b$  with  $|S| \leq \frac{(2+\varepsilon)\log n}{\log 1/q}$  for

$$q = \left( \sum_{i \in \{a,b\}} \varrho_i \sum_{j=1}^i p^j (1 - p^{j-1}) \right)^2 + \left( \sum_{i \in \{a,b\}} \varrho_i \prod_{j=1}^i (1 - p^j) \right)^2,$$

where  $\varrho_a, \varrho_b > 0$  is the probability that a node is weight  $a, b$ , and  $\varrho_a + \varrho_b = 1$ . Values  $p$ ,  $1 - p$ , and  $\varepsilon$  are  $\omega(1/\log n)$ , i.e., the values are asymptotically greater than  $1/\log n$ .

**Proof:** Let  $q$  be the probability that a node  $z$  is either in or not in both open incoming-balls of two other nodes  $x \neq y$ . If  $z$  is in the open incoming-ball of a node  $x$ , then if  $z$  is weight  $i$  where  $i = a$  or  $i = b$ , there is a path of length at most  $i$  between  $x$  and  $z$  with probability  $\sum_{j=1}^i p^j (1 - p^{j-1})$ . If  $z$  is not in the open incoming-ball of a node  $x$ , then if  $z$  is weight  $i$ , there is no path of length at most  $i = a, b$  between  $x$  and  $z$  with probability  $\prod_{j=1}^i (1 - p^j)$ . So  $\sum_{i \in \{a,b\}} \varrho_i \sum_{j=1}^i p^j (1 - p^{j-1})$  is the probability that  $z$  is in the open incoming-ball of  $x$ , and  $\sum_{i \in \{a,b\}} \varrho_i \prod_{j=1}^i p^j (1 - p^{j-1})$  is the probability that  $z$  is not in the open incoming-ball of  $x$ . ■

**Theorem 6.4** *Almost every graph in  $G(n, p)$  has a mixed-weight OLD-set  $S$  with  $|S| \leq \frac{(2+\varepsilon)\log n}{\log 1/q}$  for*

$$q = \left( \sum_{i=1}^k \varrho_i \sum_{j=1}^i p^j (1 - p^{j-1}) \right)^2 + \left( \sum_{i=1}^k \varrho_i \prod_{j=1}^i (1 - p^j) \right)^2,$$

where  $\varrho_i \geq 0$  is the probability that a node is weight  $i$ , and  $\sum_{i=1}^k \varrho_i = 1$ . Values  $p$ ,  $1 - p$ , and  $\varepsilon$  are  $\omega(1/\log n)$ , i.e., the values are asymptotically greater than  $1/\log n$ .

**Proof:** Let  $q$  be the probability that a node  $z$  is either in or not in both open incoming-balls of two other nodes  $x \neq y$ . If  $z$  is in the open incoming-ball of a node  $x$ , then if  $z$  is weight  $i \leq k$ , there is a path of length at most  $i$  between  $x$  and  $z$  with probability  $\sum_{j=1}^i p^j (1 - p^{j-1})$ . If  $z$  is not in the open incoming-ball of a node  $x$ , then if  $z$  is weight  $i \leq k$ , there is no path of length at most  $i$  between  $x$  and  $z$  with probability  $\prod_{j=1}^i (1 - p^j)$ .

Simulation Setup			Simulation Results				
$n$	Values of $p$	Values of $\varrho$	Connected	OLD-set	Size	Min. Bound	<Bound
10	[0.45, 0.55]	[0.35, 0.65]	97.5%	70.1%	5.51	8.39	69.2%
15	[0.40, 0.60]	[0.35, 0.65]	99.7%	79.5%	6.40	9.85	77.2%
20	[0.35, 0.65]	[0.35, 0.65]	99.9%	88.8%	6.86	11.18	86.3%
25	[0.35, 0.65]	[0.35, 0.65]	99.9%	93.3%	7.03	11.89	92.1%

**Table 6.1:** Simulation trends for graphs  $G(n, p)$  with probability of a weight 2 node  $\varrho$ . Both  $p$  and  $\varrho$  were incremented by 0.05 within their range. ‘Connected’ is the percentage of graphs that were connected ‘OLD-set’ is the percentage of graphs that contained a mixed-weight OLD-set. ‘Size’ is the average size of the mixed-weight OLD-set. ‘Min. Bound’ is the average upper bound of the mixed-weight OLD-set size. ‘< Bound’ is the percentage of graphs that had a mixed-weight OLD-set less than the bound. © 2017 IEEE

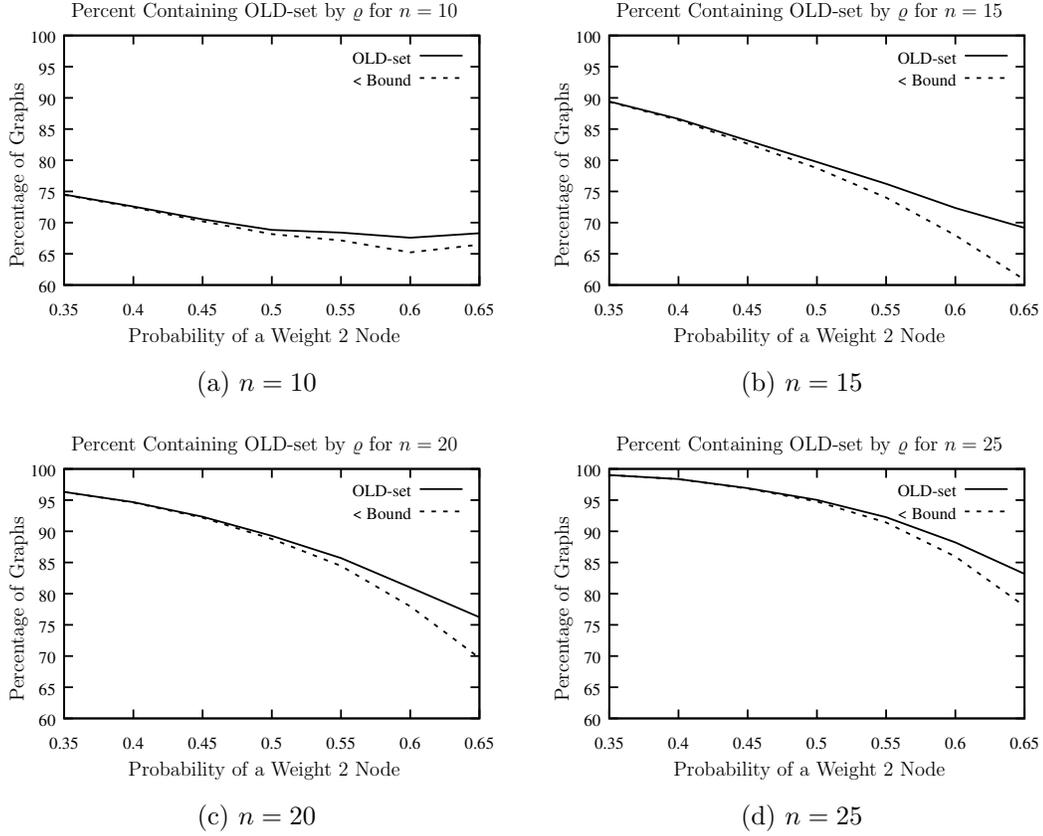
So the probability that  $z$  is in the open incoming-ball of  $x$  is  $\sum_{i=1}^k \varrho_i \sum_{j=1}^i p^j (1 - p^{j-1})$ , and the probability that  $z$  is not in the open incoming-ball of  $x$  is  $\sum_{i=1}^k \varrho_i \prod_{j=1}^i p^j (1 - p^{j-1})$ .

■

## 6.2 Simulation Results for Random Graphs

We generated random graphs and weight functions, with weights 1 and 2, and determined the actual size of their mixed-weight OLD-sets using an exhaustive search. These results were compared with bounds given in Section 6.1. Our results show that as  $n$  increases, the probability that a graph contains a mixed-weight OLD-set less than the smallest possible bound also increases.

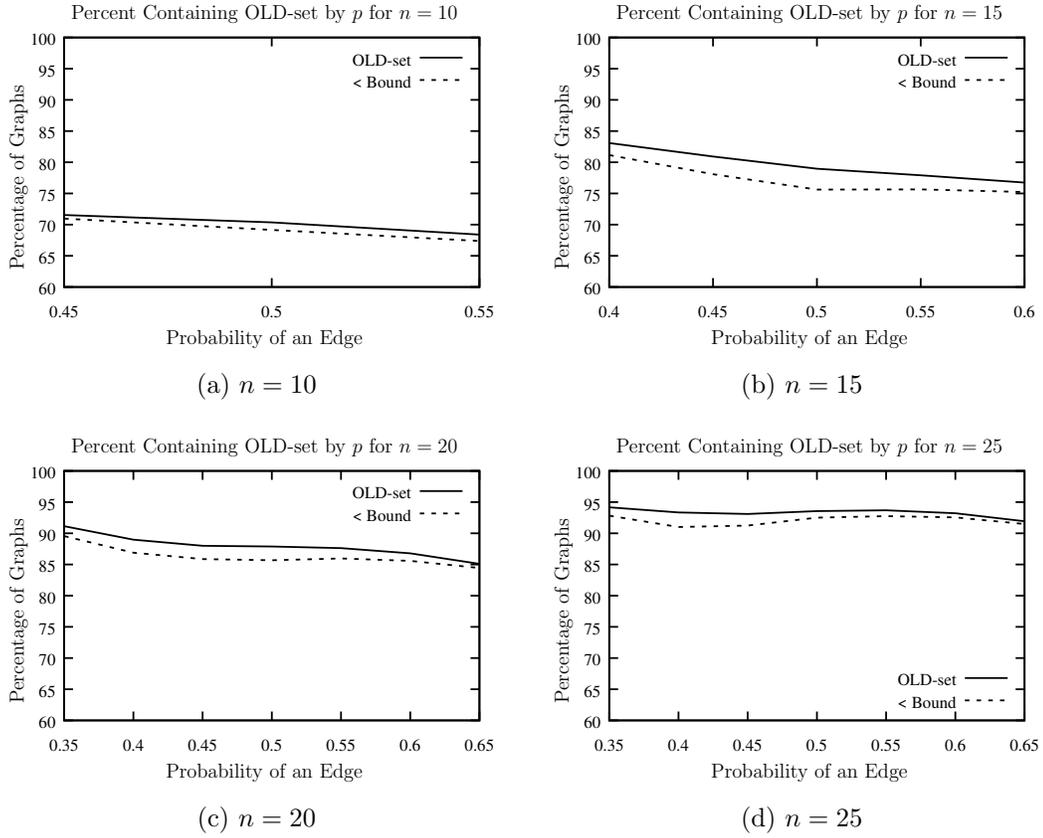
Using a Lehmer random number generator, graphs  $G(n, p)$  with corresponding weight functions,  $w(x) = 2$  with probability  $\varrho$  and  $w(x) = 1$  otherwise, were generated for each combination of  $n \in \{10, 15, 20, 25\}$ ,  $p \in [0.45, 0.55]$  for  $n = 10$ ,  $p \in [0.4, 0.6]$  for  $n = 15$ ,  $p \in [0.35, 0.65]$  for  $n = 20$  and  $n = 25$ ,  $\varrho \in [0.35, 0.65]$ . Each  $p$  and  $\varrho$  were incremented by 0.05, and both  $p$  and  $1 - p$  were  $\omega(1/\log n)$ , i.e., the values are greater than  $1/\log n$ . 10,000 graphs and corresponding weight functions were generated for each  $n$ ,  $p$ , and  $\varrho$



**Figure 6.2:** For each  $n$ , the percentage of graphs containing a mixed-weight OLD-set (solid line) and containing a mixed-weight OLD-set less than the smallest possible bound (dashed line) are graphed by  $\rho$ , the probability that a node is weight 2. Values were averaged across  $p$ . © 2017 IEEE

combination.

Consider the upper bound from Theorem 6.2,  $\frac{(2+\varepsilon)\log n}{\log 1/q}$ , on size of the mixed-weight OLD-set when  $\varepsilon$  is as small as possible, i.e.  $\varepsilon = 1/\log n$ , which we'll call the *minimum bound*. In the column titled 'OLD-set', Table 6.1 shows that as  $n$  increased the probability that a graph contained a mixed-weight OLD-set sharply increased, as expected from Lemma 6.1. Similarly the probability that a graph contains a mixed-weight OLD-set less than the minimum bound increased, as seen in the column titled '< Bound'. These trends show that the bound given in Theorem 6.2 is tight as  $n$  increases, even when  $\varepsilon$  is as small



**Figure 6.3:** For each  $n$ , the percentage of graphs containing a mixed-weight OLD-set (solid line) and containing a mixed-weight OLD-set less than the smallest possible bound (dashed line) are graphed by  $p$ , the probability that an edge exists. Values were averaged across  $\varrho$ .

as possible.

For each  $n$ , Figure 6.2 shows the percentage of graphs that had a mixed-weight OLD-set and the percentage that had a mixed-weight OLD-set less than the minimum bound given a value of  $\varrho$ . Except for  $n = 10$ , as  $\varrho$  increases, the probability of a graph containing a mixed-weight OLD-set or a mixed-weight OLD-set less than the bound decreases. One possible explanation for this pattern is, given that there are fewer possible subsets of neighbors when  $n$  is small, there may be an increased likelihood that weighted nodes cause other nodes to share open incoming-balls. This explanation is also supported by

Figure 6.2 as it appears that the effect of  $\rho$  is decreasing as  $n$  increases. It is likely  $n = 10$  is too small for a pattern to be consistent.

For comparison, Figure 6.3 shows the percentage of graphs that had a mixed-weight OLD-set and the percentage that had a mixed-weight OLD-set less than the minimum bound given a value of  $p$ . The value of  $p$  has almost no impact on the proportion of graphs that have a mixed-weight OLD-set and a mixed-weight OLD-set less than the minimum bound.

Our simulation results were constrained by the size of the graph and the subsequent time to find the smallest OLD-set. Though WSNs often contain thousands of nodes, there are many instances of smaller scale networks such as those simulated. Previously, 16 nodes monitored seismic activity on a volcano [79], 32 nodes monitored the habitat of Great Duck Island [61], and 8 nodes monitored glacial movement [63].

### 6.3 Discussion

Given that the structure of many WSNs are unknown prior to deployment, studying location-detection problems in random graphs is of particular interest. We proved that almost all random graphs have a mixed-weight OLD-set smaller than a probabilistic bound. We were able to confirm our theoretical results with simulation. These results have the potential to help determine best practices in WSN design. To our knowledge this is the first time OLD-sets have been studied in random graphs.

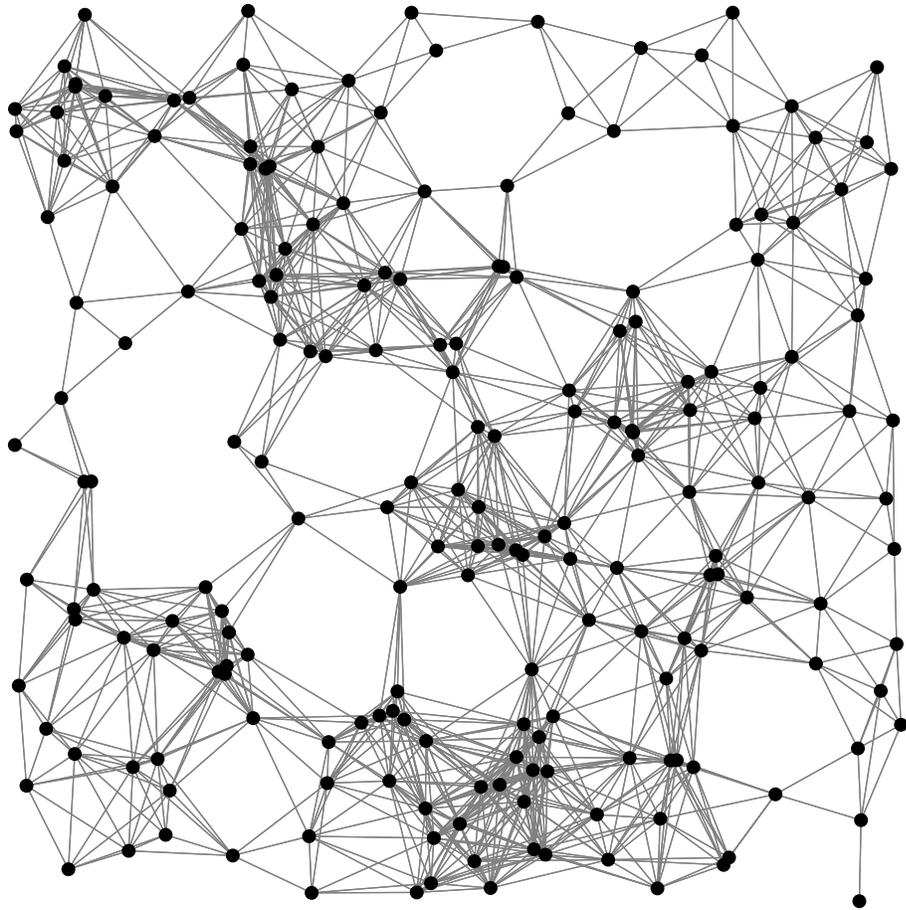
## Chapter 7

# An Integer Linear Program for Mixed-Weight OLD-sets

The mixed-weight OLD-set problem can be modeled as a integer linear program (ILP). For mixed-weight OLD-sets, the linear objective function in the ILP describes the size of the mixed-weight OLD-set to be minimized, and the linear constraints model the appropriate requirements for a set of nodes to be a mixed-weight OLD-set. Every node in the graph is either in or not in the mixed-weight OLD-set, so the decision variables in the ILP are binary (0 or 1).

Since the mixed-weight OLD-set problem is NP-complete, branch and bound is used to attempt to find optimal solutions to the accompanying ILP [31]. Branch and bound is an enumeration procedure in which feasible solutions (leading to upper bounds) coupled with LP relaxations (leading to lower bounds) are used to accelerate the enumeration process. All commercial linear programming (LP) solvers are bundled with branch and bound code for generating solutions to the ILPs. For large graphs complete enumeration of all mixed-weight OLD-sets is impractical.

An ILP formulation for the (unweighted) OLD-set problem can be found in [77]. We extend the OLD-set ILP formulation in [77] to include mixed-weight OLD-sets. We check the branch and bound results for correctness and consider the utility of the solution to



**Figure 7.1:** Example of a geometric graph  $G(n, r)$  on a  $1 \times 1$  plane with  $n = 200$  nodes and adjacencies determined by distance  $r = 0.15$

the LP-relaxation of the ILP.

We also use the ILP model to find minimum-sized mixed-weight OLD-sets in large random geometric graphs. Random geometric graphs are of particular interest as they closely resemble WSNs that are distributed at random in the field. Random geometric graphs,  $G(n, r)$ , are created by randomly generating  $n$  nodes on a surface and creating adjacencies between nodes that fall within a given physical distance  $r$ . Figure 7.1 shows a random geometric graph on a plane with 200 nodes.

In Section 7.1 we provide an ILP formulation for the mixed-weight OLD-set, and give results in Section 7.2. We explore the ILP relaxation of the integer constraint in Section

---

**Algorithm 7.1** The Floyd-Warshall Algorithm

---

```
1: procedure FLOYD-WARSHALL( $E'$ ) ▷  $E'$  is the adjacency matrix
2:   let  $D$  be a  $|V| \times |V|$  matrix
3:   for each  $i \in V$  do ▷ Algorithm set-up
4:     for each  $j \in V$  do
5:       if  $i = j$  then ▷ Shortest path is 0 when  $i = j$ 
6:          $D_{ij} \leftarrow 0$ 
7:       else if  $E'_{ij} = 1$  then ▷ Shortest path is 1 when  $(i, j)$  is an edge
8:          $D_{ij} \leftarrow 1$ 
9:       else ▷ Shortest path is unknown, otherwise
10:         $D_{ij} \leftarrow \infty$ 
11:     for each  $i \in V$  do ▷ Algorithm
12:       for each  $j \in V$  do
13:         for each  $k \in V$  do
14:           if  $D_{ij} > D_{ik} + D_{kj}$  then ▷ A shorter path is found
15:              $D_{ij} \leftarrow D_{ik} + D_{kj}$ 
```

---

7.3. In Section 7.4 we conclude our discussion of the ILP formulation. The majority of the work presented in this chapter has been published in [35] © 2018 IEEE.

The results of this chapter are used throughout the rest of this thesis through the construction of the ILP model in the modeling language AMPL [29] with solver Gurobi [40]. In Chapter 8 we compare the greedy algorithm to the ILP results and use the greedy estimates as hot starts to the ILP model. In Chapter 9 we use the results of the ILP model to test estimated bounds on the size of mixed-weight OLD-sets in random geometric graphs. We note that there is no restriction on the weight of a node, however, for our experiments we use two weights, 1 and 2.

## 7.1 ILP Formulation

Given a graph,  $G = (V, E)$ , with  $|V| = n$ , and weight function  $w(x)$  for  $x \in V$ , we want to determine the minimum-sized mixed-weight OLD-set  $S \subset V$ . To set up the ILP formulation, label the nodes  $1, 2, \dots, n$  and define  $n$  variables  $s_1, s_2, \dots, s_n$  such that  $s_i = 1$  if and only if node  $i$  is in the mixed-weight OLD-set  $S$ , and  $s_i = 0$  otherwise. To minimize the size of  $S$ , we minimize the number of  $s_i$  variables that have the value 1. Similarly, we

---

**Algorithm 7.2** Generate Matrix  $A$  for Constraint (7.2) from Matrix  $D$

---

```

1: procedure GENERATE_A( $D$ )                                ▷  $D$  is the shortest path matrix
2:   let  $A$  be a  $|V| \times |V|$  matrix
3:   for each  $i \in V$  do
4:     for each  $j \in V$  do
5:       if  $0 < D_{ij} \leq w(j)$  then                        ▷  $j \in B^+(i)$ 
6:          $A_{ij} \leftarrow 1$ 
7:       else                                                ▷  $j \notin B^+(i)$ 
8:          $A_{ij} \leftarrow 0$ 

```

---



---

**Algorithm 7.3** Generate Matrix  $C$  for Constraint (7.3) from Matrix  $A$

---

```

1: procedure GENERATE_C( $A$ )                                ▷  $A$  is the Constraint (7.2) matrix
2:   let  $C$  be a  $|V| \times |V| \times |V|$  matrix
3:   for each  $i \in V$  do
4:     for each  $j \in V$  do
5:       for each  $k \in V$  do
6:         if  $A_{ik} + A_{jk} = 1$  then                        ▷  $k \in B^+(i)$  or  $k \in B^+(j)$ , but not both
7:            $C_{ijk} \leftarrow 1$ 
8:         else
9:            $C_{ijk} \leftarrow 0$ 

```

---

can minimize the sum of the  $s_i$  variables, which gives us the objective function

$$\text{Minimize } \sum_i s_i. \tag{7.1}$$

Recall that the open incoming-ball of a node,  $B^+(x) = \{y \in V \mid 0 < d(x, y) \leq w(y)\}$ , where  $d(x, y)$  is the geodesic distance or length of the shortest path between  $x$  and  $y$ , is the set of nodes that can reach node  $x$ , also called the neighbors of  $x$ . In order for a mixed-weight OLD-set  $S$  to follow the dominating property, every node in the graph must have a neighbor in  $S$ , i.e.,  $B^+(x) \cap S \neq \emptyset \forall x \in V$ .

Define an  $n \times n$  matrix  $A$  such that  $A_{ij} = 1$  if and only if  $j \in B^+(i)$ , and  $A_{ij} = 0$  otherwise. To guarantee that  $B^+(i) \cap S$  is non-empty for each node  $i$ , there must be at least one node  $k$  such that  $k \in S$ , i.e.,  $s_k = 1$ , and  $k \in B^+(i)$ , i.e.,  $A_{ik} = 1$ . Thus if  $A_{ik}s_k = 1$ ,  $i$  has a neighbor in the mixed-weight OLD-set. Similarly, if the sum over all nodes  $j$  of  $A_{ij}s_j$  is at least 1, then  $s_i$  has a neighbor in the mixed-weight OLD-set. If we

set the constraint

$$\sum_j A_{ij}s_j \geq 1, \forall i \in V, \quad (7.2)$$

then every node in  $V$  must have a neighbor in the mixed-weight OLD-set.

In order for a mixed-weight OLD-set to follow the locating property, any two nodes in the graph must not have the same set of neighbors in the mixed-weight OLD-set, i.e.,  $B^+(i) \cap S \neq B^+(j) \cap S$  for any  $i, j \in V, i \neq j$ . Define an  $n \times n \times n$  matrix  $C$  such that  $C_{ijk} = 1$  if and only if  $k \in B^+(i)$  or  $k \in B^+(j)$ , but not both, and  $C_{ijk} = 0$  otherwise. To guarantee that  $B^+(i) \cap S$  is unique for every node in the graph, for every two nodes  $i$  and  $j$  there must be some node  $k \in S$ , i.e.  $s_k = 1$ , that is in either  $B^+(i)$  or  $B^+(j)$ , but not both, i.e.  $C_{ijk} = 1$ , otherwise  $B^+(i) = B^+(j)$ . Thus if  $C_{ijk}s_k = 1$  for some  $k$ , then  $B^+(i) \neq B^+(j)$ . Similarly, if the sum over all nodes  $k$  of  $C_{ijk}s_k$  is at least 1, then nodes  $i$  and  $j$  do not have the same set of neighbors in the mixed-weight OLD-set. If we set the constraint

$$\sum_k C_{ijk}s_k \geq 1, \forall i, j \in V, \quad (7.3)$$

then any two nodes in the graph do not share the same set of neighbors in the mixed-weight OLD-set.

We have shown that the objective function (7.1) minimizes the size of the set, and the constraints (7.2) and (7.3) guarantee the set is a mixed-weight OLD-set. In order to use this ILP formulation, we must be able to compute it in reasonable time. Assume we begin with an adjacency matrix for the graph,  $E'$  where  $E'_{ij} = E'_{ji} = 1$  if there is an edge between nodes  $i$  and  $j$ , and  $E'_{ij} = E'_{ji} = 0$ , otherwise. Define the shortest path matrix  $D$  such that  $D_{ij}$  is the length of the shortest path between nodes  $i$  and  $j$ . This matrix can be computed in  $O(n^3)$  using the Floyd-Warshall algorithm [27], as seen in Algorithm 7.1. This algorithm finds the shortest path between all nodes in a graph from the original

---

**Algorithm 7.4** Generate Array of Open Outgoing-Balls  $B^-$ 

---

```
1: procedure GENERATE_ $B^-$ ( $E'$ ) ▷  $E'$  is the adjacency matrix
2:   let  $N$  be a  $|V|$  array of empty sets ▷ adjacencies from edges in the graph
3:   let  $B^-$  be a  $|V|$  array of empty sets
4:   for each  $i \in V$  do ▷ determine adjacent nodes
5:     for each  $j \in V$  do
6:       if  $E'_{ij} = 1$  then ▷  $i$  is adjacent to  $j$ 
7:          $B_i^- \leftarrow B_i^- \cup \{j\}$ 
8:          $Ni \leftarrow Ni \cup \{j\}$ 
9:   for each  $i \in V$  do ▷ determine nodes reached by weight
10:    for each  $\ell$  from 2 to  $w(i)$  do ▷ consider weights  $> 1$ 
11:     for each  $j \in B_i^-$  do ▷ each node reached at weight  $< \ell$ 
12:       $B_i^- \leftarrow B_i^- \cup N(j) - \{i\}$ 
```

---

---

**Algorithm 7.5** Generate Array of Open Incoming-Balls  $B^+$  from  $B^-$ 

---

```
1: procedure GENERATE_ $B^+$ ( $B^-$ )
2:   let  $B^+$  be a  $|V|$  array of empty sets
3:   for each  $i \in V$  do ▷ determine adjacent nodes
4:     for each  $j \in B_i^-$  do
5:        $B_j^+ \leftarrow B_j^+ \cup \{i\}$ 
```

---

adjacency matrix using dynamic programming.

Matrix  $A$  can be generated in  $O(n^2)$  time from  $D$  by checking every pair of nodes, as seen in Algorithm 7.2,  $A_{ij} = 1$ , i.e.  $j \in B^+(i)$ , if and only if  $0 < D_{ij} \leq w(j)$ . Matrix  $C$  can be generated in  $O(n^3)$  time from  $A$  by checking every set of three nodes, as seen in Algorithm 7.3,  $C_{ijk} = 1$  if and only if  $A_{ik} + A_{jk} = 1$ , i.e.,  $k \in B^+(i)$  or  $k \in B^+(j)$ , but not both. Overall, the calculation of  $D$ ,  $A$ , and  $C$  will require  $O(n^3)$  time.

When weights are small, i.e.,  $w(x) \ll n \forall x \in V$ ,  $A$  can be generated more quickly from the open outgoing-ball,  $B^-(x) = \{y \in V \mid 0 < d(x, y) \leq w(x)\}$ , and the open incoming ball,  $B^+$ . The open outgoing ball  $B^-$  can be generated by starting with  $E'$ , the adjacency matrix, then looping on the weight of each node to add the neighbors of each neighbor to  $B^-$ . The open incoming ball  $B^+$  can be generated inversely from  $B^-$ , and  $A$  can be generated directly from  $B^+$ , as seen in Algorithms 7.4, 7.5, and 7.6. Generating  $B^-$  is  $O(mn^3)$  where  $m$  is the maximum of all the weights in the graph, though processing time

---

**Algorithm 7.6** Generate Matrix  $A$  for Constraint (7.2) from  $B^+$

---

```

1: procedure GENERATE_A( $B^+$ )                                ▷  $B^+$  an array of open incoming-balls
2:   let  $A$  be a  $|V| \times |V|$  matrix
3:   for each  $i \in V$  do
4:     for each  $j \in V$  do
5:       if  $j \in B_i^+$  then
6:          $A_{ij} \leftarrow 1$ 
7:       else
8:          $A_{ij} \leftarrow 0$ 

```

---

may be reduced by small  $m$ ,  $|B^-|$  and  $|N|$ . Generating  $B^+$  is  $O(n^3)$ . However, time is reduced when these sets and  $m$  are small, but further reduced by small  $|B^-|$  and  $|B^+|$ . On the other hand, when generating  $D$ , the Floyd-Warshall algorithm is  $\Omega(n^3)$ . With both methods, generating  $A$  after initial setup is  $O(n^2)$ , if determining a member of a set is constant time.

## 7.2 ILP Results

We generated 500 graphs for each combination of and  $r, \varrho \in \{0.25, 0.5, 0.75\}$ , where  $\varrho$  is the probability that a node is weight 2, for each  $n \in \{10, 15, 20, 25\}$ , for a total of 4500 graphs at each size. To validate our model, for each graph with a mixed-weight OLD-set, we solved for the minimum-sized mixed-weight OLD-set by brute force and by the ILP model in AMPL with Gurobi via branch and bound. Whether a graph contains a mixed-weight OLD-set can be quickly determined by testing if the set of all nodes is a mixed-weight OLD-set.

For graphs with a mixed-weight OLD-set, we would expect the ILP model to return a minimum-sized mixed-weight OLD-set. As expected, the size of the mixed-weight OLD-set found by the brute force method and by solving the ILP model with Gurobi were equivalent. Table 7.1 shows the results of our tests including the percentage of graphs that had mixed-weight OLD-sets and the average size of the minimum mixed-weight OLD-set. For graphs of size 10, 15, 20, and 25 the percentage of graphs with a mixed-weight OLD-set

$n$	$r, \varrho$	MW-OLD-set	Avg. Min.
10	{0.25, 0.5, 0.75}	56.9%	6.6
15	{0.25, 0.5, 0.75}	64.2%	8.9
20	{0.25, 0.5, 0.75}	70.6%	11.1
25	{0.25, 0.5, 0.75}	76.8%	13.1

**Table 7.1:** Percentage of graphs with a mixed-weight OLD-set and the average minimum size of the set in random geometric graphs of size  $n$  with distance  $r$  and probability of weight 2 node  $\varrho$ . ‘MW-OLD-set’ is the percentage of graphs that had a mixed-weight OLD-set. ‘Avg. Min.’ is the average minimum mixed-weight OLD-set size.

was 56.9%, 64.2%, 70.%, and 76.8%, and the average size of the minimum mixed-weight OLD-set was 6.6, 8.9, 11.1, and 13.1, respectively.

### 7.3 ILP Relaxation

Although ILP solutions are often found quickly, because the problem of finding a mixed-weight OLD-set is NP-complete, a quick solution is not guaranteed. Relaxation of the integer constraint converts the problem into a linear programming (LP) model, which is polynomial-time solvable. We allow  $0 \leq s_i \leq 1$ , and, once a solution has been found, we say  $i$  is in the mixed-weight OLD-set if  $s_i$  is close to 1, and not in the OLD-set if  $s_i$  is close to 0.

We generated 500 random geometric graphs  $G(n, r)$  for each combination of  $n \in \{50, 100, 150\}$ , which are similar to the size of several WSNs discussed in [43], and  $r, \varrho \in \{0.25, 0.5, 0.75\}$ , where  $\varrho$  is the probability that a node is weight 2, for a total of 4500 graphs and mixed-weight functions for each  $n$ . Of these graphs, 190 at size  $n = 50$  and 6 at size  $n = 100$  did not contain mixed-weight OLD-sets. All graphs at size  $n = 150$  contained a mixed-weight OLD-set. All results are for graphs with mixed-weight OLD-sets.

We generated solutions to the LP relaxation using AMPL with Gurobi for each graph that had a mixed-weight OLD-set. For the results of the LP model, we would want to

LP Integer Results			LP Fractional Results			LP Bound Results	
$n$	$s_i = 0$	$s_i = 1$	$s_i = 0.5$	$s_i < 0.5$	$s_i > 0.5$	LP Obj. Value	Greedy Size
50	16.3	1.5	22.9	23.8	3.3	16.0	27.3
100	39.8	2.3	33.6	59.5	6.9	27.0	46.5
150	65.6	3.0	40.5	98.7	10.9	36.5	63.0

**Table 7.2:** LP (relaxed ILP) results in random geometric graphs of size  $n$  with  $r, \varrho \in \{0.25, 0.5, 0.75\}$ . The number of nodes with decision variables equal to 0 and 1 ( $s_i = 0$  and  $s_i = 1$ ), and the number of nodes with decision variables equal to, less than, and greater than 0.5 ( $s_i = 0.5$ ,  $s_i < 0.5$  and  $s_i > 0.5$ ). ‘LP Obj. Value’ is the average objective value from the LP result, and ‘Greedy Size’ is the average size of the mixed-weight OLD-set generated by greedy search through the LP result. © 2018 IEEE

see values close to 1 or close to 0. However, as seen in Figure 7.2, very few values were greater than 0.5 and a large portion were equal to 0.5. A majority of graphs did not have a mixed-weight OLD-set when only considering nodes  $i$  such that  $s_i \geq 0.5$ , and the percentage that had a mixed-weight OLD-set decreased as  $n$  increased (51.5%, 28.1%, and 18.1% for  $n = 50, 100$ , and  $150$ , respectively).

We were able to estimate small mixed-weight OLD-sets by ordering the nodes by largest  $s_i$  and selecting nodes in a greedy manner until an OLD-set is formed, thus nodes with values less than 0.5 would be selected for most graphs. Table 7.2 shows the objective value returned by the LP relaxation which provides a lower bound for the optimal objective value of the ILP. The feasible greedy result provides an upper bound.

## 7.4 Discussion

In this chapter, we introduced an integer linear programming model for mixed-weight OLD-sets and studied their use in random geometric graphs. Finding optimal mixed-weight OLD-sets in large graphs by enumerating all possible solutions is not feasible due to the NP-complete nature of the problem. ILP provides a way to solve for optimal mixed-

weight OLD-sets in reasonable time in most cases by using branch and bound. Random geometric graphs are of particular interest due to their similarities to randomly distributed wireless sensor networks.

This is the first time, to our knowledge, that (mixed-weight) OLD-sets have been studied in random geometric graphs. Solutions to the Linear Program (LP), resulting from relaxing the integer constraint in the ILP, were shown to be highly fractional but do provide a lower bound on the optimal objective value of the ILP. A feasible solution to the ILP was greedily constructed from the fractional LP solution which provides an upper bound.

## Chapter 8

# Greedy Algorithms for Mixed-Weight OLD-sets

Greedy algorithms take a simple and intuitive approach to constructing feasible solutions to optimization problems. During each iteration of a greedy algorithm, the option that appears best or optimal is chosen, and this process continues until a feasible solution is constructed. Greedy algorithms have been used to solve several problems in graph theory, such as finding the shortest path between two nodes [23] and finding a minimum spanning tree [23, 55, 67]. Although greedy algorithms can provide a quick and understandable solution, they do not typically provide an optimal solution. Even so, greedy algorithms can often provide a reasonable, and sometimes even good, estimate of the optimal solution. Several NP-complete problems in graph theory have been estimated with greedy algorithms such as the maximal independent set problem [42] and graph coloring [48].

In this chapter we introduce several greedy algorithms for mixed-weight OLD-sets and study their use in random geometric graphs. Our algorithms include a naïve approach, a method inspired by the maximal independent set problem, and a stingy method. We compare these greedy algorithms to the greedy solution derived from the relaxed ILP results from Section 7.3. We also use the greedy results as a hot start, or initial feasible

---

**Algorithm 8.1** Naïve Greedy Method for finding a mixed-weight OLD-set

---

```
1: procedure NAIVE_GREEDY( $B^-, B^+$ )           ▷ open outgoing- and incoming-balls
2:   Let  $R$  be  $V$  sorted by chosen method ▷ largest, smallest, alternating, median  $B^-$ 
3:   Let  $S$  be an empty set                       ▷ mixed-weight OLD-set
4:   while  $S$  is not a mixed-weight OLD-set do           ▷ use  $B^+$  to check
5:      $v \leftarrow$  next node in  $R$ 
6:      $S = S \cup \{v\}$                                ▷ add next node to mixed-weight OLD-set
7:   end while
```

---

solution, to the ILP model. A hot start can often shorten the time needed to solve a large ILP problem.

We present our greedy algorithms in Section 8.1 and study the mixed-weight OLD-sets they produce in Section 8.2. We use the mixed-weight OLD-set results as hot starts to the ILP model in Section 8.3. We conclude our study of greedy algorithms for mixed-weight OLD-sets in Section 8.4. Most of the work presented in this chapter has been published in [35] © 2018 IEEE.

## 8.1 Greedy Algorithms

We tested two greedy methods and one stingy method, each with four different selection methods: largest, smallest, alternating the largest and smallest (alternating), and median open outgoing-ball size, to find initial solutions to the mixed-weight OLD-set problem. Each of the algorithms begins by sorting the nodes according to the choice option. In particular if  $\{x_1, x_2, \dots, x_n\}$  is the set of nodes ordered from smallest to largest open outgoing-ball, and the middle node can be found at  $m = \lfloor \frac{n+1}{2} \rfloor$ , then the largest choice option is sorted as  $\{x_n, x_{n-1}, \dots, x_1\}$ , smallest is sorted as  $\{x_1, x_2, \dots, x_n\}$ , alternating is sorted as  $\{x_n, x_1, x_{n-1}, x_2, \dots, x_m\}$ , and, median is sorted as  $\{x_m, x_{m+1}, x_{m-1}, x_{m+2}, \dots, x_n\}$ . The alternating and median choice methods were considered since, in a (mixed-weight) OLD-set, a node that reaches a lot of nodes does not help locate in the graph and a node that does not reach many nodes does not help dominate.

The naïve greedy algorithm and the stingy algorithm choose each node in turn from

---

**Algorithm 8.2** Independent Set Greedy Method for finding a mixed-weight OLD-set

---

```
1: procedure IND_SET_GREEDY( $B^-, B^+$ )           ▷ open outgoing- and incoming-balls
2:   Let  $R$  be  $V$  sorted by chosen method ▷ largest, smallest, alternating, median  $B^-$ 
3:   Let  $S$  be an empty set                       ▷ mixed-weight OLD-set
4:   Let  $T$  be an empty set                       ▷ current unavailable nodes
5:   Let  $U$  be  $V$                                 ▷ current available nodes
6:   while  $S$  is not a mixed-weight OLD-set do           ▷ use  $B^+$  to check
7:     if  $|U| = 0$  then                                   ▷ no nodes are available
8:        $U \leftarrow T$ 
9:        $T \leftarrow \{\}$ 
10:     $v \leftarrow$  next node in  $R$  that is in  $U$            ▷ next in order that is available
11:     $S \leftarrow S \cup \{v\}$                              ▷ add next node to mixed-weight OLD-set
12:     $U \leftarrow U - \{v\}$                                ▷ node is no longer available
13:    for each  $x \in B^-(v)$  do                             ▷ make nodes in  $B^-$  unavailable
14:      if  $x \in U$  then
15:         $U \leftarrow U - \{x\}$ 
16:         $T \leftarrow T \cup \{x\}$ 
17:  end while
```

---

---

**Algorithm 8.3** Stingy Method for finding a mixed-weight OLD-set

---

```
1: procedure NAIVE_GREEDY( $B^-, B^+$ )           ▷ open outgoing- and incoming-balls
2:   Let  $R$  be  $V$  sorted by chosen method ▷ largest, smallest, alternating, median  $B^-$ 
3:   Let  $S$  be  $V$                                     ▷ mixed-weight OLD-set
4:   for each  $v \in R$  do
5:     if  $S - \{v\}$  is a mixed-weight OLD-set then           ▷ use  $B^+$  to check
6:        $S \leftarrow S - \{v\}$ 
```

---

the sorted nodes. The independent set algorithm chooses the first node from the sorted nodes as long as that node is currently available.

The first greedy method is a simple naïve method (Naïve) that chooses the next node by one of the four choice options forming a set of nodes. The method stops when the set is a mixed-weight open locating-dominating set, and can be seen in Algorithm 8.1.

The second greedy algorithm was inspired by the Maximal Independent Set problem (independent set) which attempts to find the maximum set of nodes in the graph that are not adjacent. Random geometric graphs are generated on a surface, with edges formed locally, so it is possible that choosing nodes from different areas of the graph may help estimate mixed-weight OLD-sets. The independent set greedy algorithm chooses the next

node  $x$  by one of the sorting methods and then removes all nodes in the open outgoing-ball of  $x$  from the set of available choices, making them unavailable for selection. The algorithm continues to choose the first node from the sorted nodes that is available. Once there are no available choices, the removed nodes are returned to the set of available choices. This algorithm continues until a mixed-weight open locating-dominating set is formed, similar to the naïve method, as seen in Algorithm 8.2.

The stingy algorithm begins with all nodes in the graph in the set, which is guaranteed to be a mixed-weight OLD-set, and one node is removed in turn by one of the four choice options. If the set is still a mixed-weight OLD-set, the algorithm continues. Otherwise the node is added back into the set before the algorithm continues. The algorithm finishes when all nodes have been checked, as seen in Algorithm 8.3.

Determining if a set is a mixed-weight OLD-set in the graph is an  $O(n^3 \log(n))$  operation if intersecting sets and checking sets for equality requires sorting: for every pair of nodes, the intersection of the set with both open incoming-balls must be checked for equality. However, some implementations, such as using a hashed set, may intersect and check for equality in linear time making the mixed-weight OLD-set check  $O(n^3)$ . Thus each greedy method is  $O(n^4 \log(n))$  or  $O(n^4)$  depending on implementation.

## 8.2 Greedy Algorithm Results

To test the greedy algorithms, we used the same graphs as those in Section 7.3. As seen in Figure 8.1, the stingy method is the clear winner for generating the smallest mixed-weight OLD-set overall, and choosing the nodes by largest open outgoing-ball gave the best result for this method, followed closely by choosing the median. The naïve and independent set inspired methods performed similarly to each other, and choosing the smallest open outgoing-ball gave better results for both of these methods, with the choice of median performing almost the same as the smallest for the independent set inspired algorithm. We note that the stingy method performed better than the LP relaxation of the

<b><math>n = 50</math></b>	<b>Naïve</b>	<b>Ind. Set</b>	<b>Stingy</b>
largest	44.0	44.6	23.4
smallest	41.3	42.2	27.6
alternating	43.9	44.2	25.2
median	41.7	42.2	24.1
<b><math>n = 100</math></b>	<b>Naïve</b>	<b>Ind. Set</b>	<b>Stingy</b>
largest	89.8	90.4	39.4
smallest	83.7	85.8	48.6
alternating	89.8	90.0	44.1
median	86.2	85.8	40.5
<b><math>n = 150</math></b>	<b>Naïve</b>	<b>Ind. Set</b>	<b>Stingy</b>
largest	136.3	136.6	53.5
smallest	126.7	129.7	68.2
alternating	136.0	136.1	61.1
median	131.9	130.0	55.0

**Table 8.1:** Average mixed-weight OLD-set size generated from each greedy algorithm by sort method. © 2018 IEEE

ILP from Figure 7.2 in all cases except when ordered by smallest outgoing-ball. Overall the stingy method and the LP relaxation of the ILP provide similar quality bounds on the optimal value to the ILP. It is important to note that the stingy method constructs a feasible solution to the ILP model while the LP relaxation does not.

Our results also showed that the naïve and independent set methods had similar running time, and the stingy method was less than 50% slower. Although determining if a set is a mixed-weight OLD-set is  $O(n^3 \log(n))$ , the processing time is greatly affected by the size of the set being tested. The stingy method starts with all nodes in the OLD-set, and removes them one by one, if at all. Thus, on average, the size of the set being tested will be much larger for the stingy method, resulting in increased processing time. The stingy method also checks every node in the graph, whereas the other methods stop once a mixed-weight OLD-set is found. Table 8.2 shows that the naïve and independent set

<b><math>n = 50</math></b>	<b>Naïve</b>	<b>Ind. Set</b>	<b>Stingy</b>
largest	2.4	2.7	3.4
smallest	2.1	2.3	3.5
alternating	2.5	2.6	3.5
median	2.4	2.5	3.3
<b><math>n = 100</math></b>	<b>Naïve</b>	<b>Ind. Set</b>	<b>Stingy</b>
largest	20.6	20.3	25.2
smallest	17.7	18.8	26.3
alternating	20.0	20.2	26.2
median	18.7	19.3	24.7
<b><math>n = 150</math></b>	<b>Naïve</b>	<b>Ind. Set</b>	<b>Stingy</b>
largest	73.5	73.9	91.6
smallest	64.3	70.1	97.9
alternating	72.7	74.0	96.5
median	69.6	71.3	90.4

**Table 8.2:** Average time in milliseconds to generate OLD-set from greedy algorithm by sort method in Java using an Intel® Core™ i5-6500 CPU @3.2GHz with 6MB L3 cache and 16GB DDR3 RAM.

methods also had similar running time. The time to move nodes between sets and test for existence in a set, for the independent set method, increased processing time by a very small amount over the naïve method. The stingy method, however, required more time than the other two methods.

Figure 8.3 shows the average size of the mixed-weight OLD-set generated by the ILP model, which provides an optimal solution, by the stingy algorithm, and by greedily choosing from the fractional values found by the LP relaxation of the ILP from Section 7.3. The stingy algorithm provides a good estimate for the size of mixed-weight OLD-set. The stingy estimate gets worse as the size of the graph increases (also true for the LP relaxation), but for each graph size tested, the difference from the optimal result for the stingy method is less than 20%. As was previously noted, the stingy method provides a feasible mixed-weight OLD-set solution, whereas the LP relaxation does not.

$n$	Optimal	Stingy	Relaxed	Stingy-Diff	Relaxed-Diff
50	21.4	23.4	27.3	9.6%	27.8%
100	34.6	39.4	46.5	13.9%	34.3%
150	45.7	53.5	63.0	17.2%	37.7%

**Table 8.3:** Average size of the mixed-weight OLD-set returned by the ILP model (optimal), by the stingy method choosing by largest outgoing ball, and by greedily selecting nodes from the results of the ILP relaxation. ‘Stingy-Diff’ and ‘Relaxed-Diff’ are the percentage differences between the stingy method and the greedy relaxed ILP method, respectively, with the optimal ILP results. © 2018 IEEE

### 8.3 Greedy Hot Starts for the ILP Model

We used the stingy algorithm, sorting by largest open outgoing-ball, to generate hot starts for the graphs in Section 7.3. We tested the ILP formulation with the stingy hot starts, with no hot start, and with all nodes in the graph as the hot start. If a graph contains a mixed-weight OLD-set, then the set of all nodes is a mixed-weight OLD-set, providing a trivial hot start. We found that using a hot start with the ILP model did not significantly reduce the time to find the minimum mixed-weight OLD-set, and that there was not a significant time difference between using all nodes in the graph or just those returned by the stingy method as the hot start. However, finding optimal solutions to the ILP model was an order of magnitude slower than finding a feasible solution with the stingy heuristic.

Table 8.4 shows the average ILP solve time for graphs with no hot start, all nodes in the graph as the hot start, and the stingy-generated hot start, as well as the time to find the stingy hot start in Java. In each case the average time to find the stingy hot start was slightly greater than the average time saved by having the stingy hot start. The hot start with all nodes in the graph also performed similarly to the hot start from the stingy method.

$n$	ILP-None	ILP-All	ILP-Stingy	Stingy
50	78.9	78.8	77.6	3.4
100	537.6	518.2	514.4	25.2
150	2254.9	2196.6	2172.1	91.6

**Table 8.4:** Average solve time, in milliseconds, for the ILP in AMPL with Gurobi with no hot start (‘ILP-None’), all nodes in the graph as the hot start (‘ILP-All’), and using the result of the stingy method as the hot start (‘ILP-Stingy’), and the average time to find the stingy hot start in Java using an Intel® Core™ i5-6500 CPU @3.2GHz with 6MB L3 cache and 16GB DDR3 RAM.

## 8.4 Discussion

We introduced three greedy algorithms for the mixed-weight OLD-set: the naïve, the independent set, and the stingy. We also used several selection methods for the “best” choice during an iteration of each greedy algorithm: smallest, largest, alternating smallest and largest, and median size of a node’s open outgoing-ball. For random geometric graphs, the stingy algorithm outperformed the two other greedy algorithms. The best selection method for the stingy method was largest open outgoing-ball.

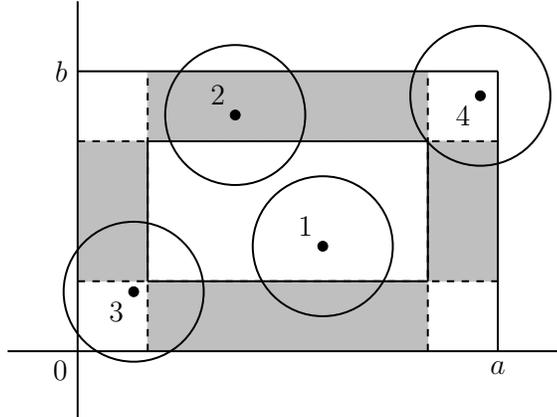
The stingy heuristic was also superior to the greedy estimate of the optimal mixed-weight OLD-set using the results from the relaxation of the ILP model discussed in Section 7.3. The stingy algorithm performed an order of magnitude quicker than the ILP model, and provided small estimates within 20% of the optimal solution for large random geometric graphs. These results show that the stingy heuristic has the potential to aid in the design of wireless sensor networks.

## Chapter 9

# Mixed-Weight OLD-sets in Random Geometric Graphs

Random geometric graphs  $G(n, r)$  are graphs on a surface where the set of nodes is  $n$  points chosen at random from the surface, and  $r$  is the threshold in which two nodes are adjacent given their Euclidean distance. Typical random graphs, like those considered in Chapter 6, often miss key characteristics of the of the networks they are meant to represent, such as WSNs, as in the field these networks often fall somewhere between completely random and highly structured [9]. However, random geometric graphs closely resemble how WSNs are distributed through a physical area. Random geometric graphs prove to be more theoretically difficult because their construction is limited by physical boundaries and relies on actual distances, making edge probabilities dependent. Random geometric graphs have been studied for identifying codes and locating-dominating sets [65], but have not been studied in (mixed-weight) OLD-sets.

We determine the average probability two nodes are adjacent in a plane, and, with bounds proved in [34] for random graphs, estimate upper bounds on the size of mixed-weight OLD-sets in random geometric graphs. We generate minimum-sized mixed-weight OLD-sets using the ILP model in Section 7.1 and compare these results to the estimated bounds.



**Figure 9.1:** Four cases for the location of a node in a random geometric graph on a plane. © 2018 IEEE

In Section 9.1 we find the average probability of an edge in a random geometric graph, and in Section 9.2 we use the average probability to estimate bounds on the size of a mixed-weight OLD-set in random geometric graphs. We conclude our study of random geometric graphs in Section 9.3. The majority of the work presented in this chapter has been published in [35] © 2018 IEEE.

## 9.1 Average Probability of an Edge in Random Geometric Graphs

Consider a plane of size  $a \times b$  starting at the origin and a graph  $G(n, r)$  where  $a, b \geq 2r$ . For  $G(n, r)$ , the nodes,  $V$ , are  $n$  random points on the plane,  $u = (x, y)$ , and the edges are  $E = \{(u, v) \mid d(u, v) \leq r\}$ , where  $d(u, v)$  is the Euclidean distance. For each node  $u$  in the graph, the probability that another node  $v$  is adjacent to  $u$  is equivalent to the probability that  $v$  falls within a circle of radius  $r$  around  $u$ . If  $a, b \gg 2r$ , this probability is  $\frac{\pi r^2}{ab}$  in most cases. Define the *range* of a node  $u$ ,  $R(u)$  to be the area of the circle of radius  $r$  around  $u$  that falls within the plane. Thus for any node  $u$ , the probability that  $v$  is adjacent to  $u$  is  $\frac{R(u)}{ab}$ .

Figure 9.1 shows the four locations that a node can fall in a plane with respect to the shape of the node's range. The following lemmas determine the (average) range of a node given its location in the plane. We then provide a proposition using the average ranges to find the average probability that two nodes are adjacent in a random geometric graph generated on the plane.

**Lemma 9.1** *The range of a node  $u = (x, y)$  given that  $r \leq x \leq a - r$  and  $r \leq y \leq b - r$  is  $R_1(u) = \pi r^2$ .*

**Proof:** When  $r \leq x \leq a - r$  and  $r \leq y \leq b - r$ , as in case 1 in Figure 9.1, the node is in the middle of the plane, and the entire range falls within the plane. ■

**Lemma 9.2** *The average value of the range of  $u = (x, y)$  given that  $x < r$  or  $x > a - r$ , and  $r \leq y \leq b - r$  or  $y < r$  or  $y > b - r$ , and  $r \leq x \leq a - r$  is  $R_2(u) = \frac{3\pi r^2 - 2r^2}{3}$ .*

**Proof:** When  $x < r$  or  $x > a - r$ , and  $r \leq y \leq b - r$  or  $y < r$  or  $y > b - r$ , and  $r \leq x \leq a - r$ , as in case 2 in Figure 9.1, the node is near and overlaps the side of the plane. WLOG, suppose  $x < r$  and  $r \leq y \leq b - r$ . The angle from the node to the points of overlap is  $\theta = 2 \cos^{-1}\left(\frac{x}{r}\right)$ , therefore the area of the segment is  $\frac{\theta}{2\pi} \pi r^2 - x \cdot \sqrt{r^2 - x^2}$ . Thus the average value of the range of  $u$  is

$$\begin{aligned}
R_2(u) &= \frac{1}{r(b-2r)} \int_0^r \int_r^{b-r} \left( \pi r^2 - \left( \frac{\theta}{2\pi} \pi r^2 - x \cdot \sqrt{r^2 - x^2} \right) \right) dy dx \\
&= \frac{1}{r} \int_0^r \left( \pi r^2 - \left( \frac{\theta}{2\pi} \pi r^2 - x \cdot \sqrt{r^2 - x^2} \right) \right) dx \\
&= \pi r^2 - \left( \frac{1}{r} \int_0^r \frac{\theta}{2\pi} \pi r^2 dx \right) + \frac{r^2}{3} \\
&= \pi r^2 - \left( r \int_0^r \cos^{-1}\left(\frac{x}{r}\right) dx \right) + \frac{r^2}{3} \\
&= \pi r^2 - r^2 + \frac{r^2}{3} \\
&= \pi r^2 - \frac{2r^2}{3}.
\end{aligned}$$

■

**Lemma 9.3** *The average value of the range of  $u = (x, y)$  given that given that  $x < r$  and  $\sqrt{r^2 - x^2} \leq y < r$ , etc., is  $R_3(u) = \frac{(24\pi - 3\pi^2 - 32)r^2}{6(4 - \pi)}$ .*

**Proof:** When  $x < r$  and  $\sqrt{r^2 - x^2} \leq y < r$ , etc., as in case 3 in Figure 9.1, the node is close to and overlaps two sides of the plane. WLOG, suppose  $x < r$  and  $\sqrt{r^2 - x^2} \leq y < r$ . The sum of the angles that overlap the  $x$ - and  $y$ - axes is  $\theta = 2 \cos^{-1} \left( \frac{x}{r} \right) + 2 \cos^{-1} \left( \frac{y}{r} \right)$ , therefore the area of the segments is  $\frac{\theta}{2\pi} \pi r^2 - x \cdot \sqrt{r^2 - x^2} - y \cdot \sqrt{r^2 - y^2}$ . Thus the average value of the range of  $u$  is

$$\begin{aligned} R_3(u) &= \frac{1}{r^2(1 - \pi/4)} \int_0^r \int_{\sqrt{r^2 - x^2}}^r \left( \pi r^2 - \left( \frac{\theta}{2\pi} \pi r^2 - x \cdot \sqrt{r^2 - x^2} - y \cdot \sqrt{r^2 - y^2} \right) \right) dy dx \\ &= \frac{1}{r^2(1 - \pi/4)} \int_0^r \left( \left( \pi r^2 - \frac{\theta_x}{2\pi} \pi r^2 + x \cdot z \right) (r - z) + \frac{x^3}{3} - r^2 \left( z \cos^{-1} \left( \frac{z}{r} \right) - x \right) \right) dx \\ &= \frac{(24\pi - 3\pi^2 - 32)r^2}{6(4 - \pi)}. \end{aligned}$$

■

**Lemma 9.4** *The average value of the range of  $u = (x, y)$  given that given that  $x < r$  and  $0 \leq y < \sqrt{r^2 - x^2}$ , etc., is  $R_4(u) = \frac{(1 + \pi^2)r^2}{2\pi}$ .*

**Proof:** When  $x < r$  and  $0 \leq y < \sqrt{r^2 - x^2}$ , etc., the node is in the corner of the plane and overlaps the corner. WLOG, suppose  $x < r$  and  $y < \sqrt{r^2 - x^2}$ . The angle of the portion that overlaps the corner is  $\theta = \cos^{-1} \left( \frac{x}{r} \right) + \cos^{-1} \left( \frac{y}{r} \right) + \frac{\pi}{2}$ , therefore the area of the overlap is  $\frac{\theta}{2\pi} \pi r^2 - \frac{x}{2} \cdot \sqrt{r^2 - x^2} - \frac{y}{2} \cdot \sqrt{r^2 - y^2} - xy$ . Thus the average value of the range of  $u$  is

$$\begin{aligned} R_4(u) &= \frac{1}{r^2(\pi/4)} \int_0^r \int_0^{\sqrt{r^2 - x^2}} \left( \pi r^2 - \left( \frac{\theta}{2\pi} \pi r^2 - \frac{x}{2} \cdot \sqrt{r^2 - x^2} - \frac{y}{2} \cdot \sqrt{r^2 - y^2} - xy \right) \right) dy dx \\ &= \frac{(1 + \pi^2)r^2}{2\pi}. \end{aligned}$$

■

**Proposition 9.5** *The average probability that there is an edge between any two nodes in  $G(n, r)$ , if  $a, b \geq 2r$ , is*

$$\phi = \frac{\pi r^2}{ab} - \frac{4r^3}{3ab^2} - \frac{4r^3}{3a^2b} + \frac{r^4}{2(ab)^2}.$$

**Proof:** Figure 9.1 shows the four cases for the location of a node  $u = (x, y)$ . In cases 2-4, the range of the node is not the entire circle around it, affecting the probability that it will be adjacent to another node.

**Case 1:** The node is in the middle rectangle of the plane, i.e.,  $r \leq x \leq a - r$  and  $r \leq y \leq b - r$ , with probability  $(a - 2r)(b - 2r)/(ab)$ . From Lemma 9.1, the probability that a node  $v$  is adjacent to  $u$  given its location, is  $\frac{\pi r^2}{ab}$ . Thus  $u$  and  $v$  are connected and  $u$  is in the middle of the plane with probability

$$\begin{aligned} \phi_1 &= \frac{(a - 2r)(b - 2r)}{ab} \cdot \frac{\pi r^2}{ab} \\ &= \frac{\pi r^2}{ab} - \frac{2\pi r^3}{ab^2} - \frac{2\pi r^3}{a^2b} + \frac{4\pi r^4}{a^2b^2} \end{aligned}$$

**Case 2:** The node is closest to one side of the plane, with probability  $2r(a + b - 4r)/ab$ , when  $x < r$  or  $x > a - r$ , and  $r \leq y \leq b - r$  or when  $y < r$  or  $y > b - r$ , and  $r \leq x \leq a - r$ . From Lemma 9.2, the average probability that a node  $v$  is adjacent to  $u$  given its location, is  $\frac{3\pi r^2 - 2r^2}{3ab}$ . Thus  $u$  and  $v$  are connected and  $u$  is near the side of the plane with average probability

$$\begin{aligned} \phi_2 &= \frac{2r(a + b - 4r)}{ab} \cdot \frac{3\pi r^2 - 2r^2}{3ab} \\ &= \frac{2\pi r^3}{ab^2} - \frac{4r^3}{ab^2} + \frac{2\pi r^3}{a^2b} - \frac{4r^3}{a^2b} + \frac{16r^4}{3a^2b^2} - \frac{8\pi r^4}{a^2b^2} \end{aligned}$$

**Case 3:** The node is closest to two sides of the plane, i.e.,  $x < r$  and  $\sqrt{r^2 - x^2} \leq y < r$ , etc., with probability  $\frac{4r^2 - \pi r^2}{ab}$ . From Lemma 9.3, the average probability that a node  $v$  is adjacent to  $u$  given its location, is  $\frac{(24\pi - 3\pi^2 - 32)r^2}{6ab(4 - \pi)}$ . Thus  $u$  and  $v$  are connected and  $u$  is near the two sides of the plane with average probability

$$\begin{aligned}\phi_3 &= \frac{4r^2 - \pi r^2}{ab} \cdot \frac{(24\pi - 3\pi^2 - 32)r^2}{6ab(4 - \pi)} \\ &= \frac{4\pi r^4}{a^2 b^2} - \frac{16r^4}{3a^2 b^2} - \frac{\pi^2 r^4}{2a^2 b^2}\end{aligned}$$

**Case 4:** The node is closest to a corner, i.e.  $x < r$  and  $0 \leq y < \sqrt{r^2 - x^2}$ , etc., with probability  $\frac{\pi r^2}{ab}$ . From Lemma 9.4, the average probability that a node  $v$  is adjacent to  $u$  given its location, is  $\frac{(1+\pi^2)r^2}{2\pi ab}$ . Thus  $u$  and  $v$  are connected and  $u$  is near the corner of the plane with average probability

$$\begin{aligned}\phi_4 &= \frac{\pi r^2}{ab} \cdot \frac{(1 + \pi^2)r^2}{2\pi ab} \\ &= \frac{\pi^2 r^4}{2a^2 b^2} + \frac{r^4}{2a^2 b^2}\end{aligned}$$

Summing the probabilities we find  $\phi = \phi_1 + \phi_2 + \phi_3 + \phi_4$  and thus  $\phi = \frac{\pi r^2}{ab} - \frac{4r^3}{3ab^2} - \frac{4r^3}{3a^2 b} + \frac{r^4}{2(ab)^2}$ . ■

## 9.2 Estimated Random Geometric Graph Bounds

In Chapter 6, we showed that almost every random graph  $G(n, p)$ , where  $n$  is the number of nodes in the graph and  $p$  is the probability that two nodes are adjacent, has a (mixed-weight) OLD-set  $S$  with  $|S| \leq \frac{(2+\varepsilon)\log n}{\log 1/q}$  for  $\varepsilon > 1/\log(n)$ . We also showed that the probability that a node is either in or not in both open incoming-balls of two other nodes is

$$q = \left( \sum_{i=1}^d \varrho_i \sum_{j=1}^i p^j (1 - p^{j-1}) \right)^2 + \left( \sum_{i=1}^d \varrho_i \prod_{j=1}^i (1 - p^j) \right)^2,$$

where  $\varrho_i$  is the probability that a node is weight  $i$  for  $1 \leq i \leq d$ .

Similarly, we can estimate the  $q$  for random geometric graphs by using the average probability that two nodes are adjacent. In this case we find

$$q = \left( \sum_{i=1}^d \varrho_i \sum_{j=1}^i \phi^j (1 - \phi^{j-1}) \right)^2 + \left( \sum_{i=1}^d \varrho_i \prod_{j=1}^i (1 - \phi^j) \right)^2,$$

$n$	$r$	OLD-set	ILP Size	Min. Bound	< Bound	< 2×Bound
50	0.25	88.7%	18.2	27.1	99.8%	100.0%
100		99.6%	27.5	31.3	90.3%	100.0%
150		100.0%	35.6	33.8	26.9%	100.0%
50	0.50	99.9%	18.3	12.9	0.9%	96.7%
100		100.0%	29.7	14.9	0.0%	60.9%
150		100.0%	38.8	16.1	0.0%	15.7%

**Table 9.1:** Simulation trends for graphs  $G(n, r)$  with the probability of a weight 2 node  $\varrho \in \{0.25, 0.5, 0.75\}$ . Results are averaged across  $\varrho$ . ‘OLD-set’ is the percentage of graphs that contained a mixed-weight OLD-set. ‘Size’ is the average size of the mixed-weight OLD-set returned by the ILP. ‘Min. Bound’ is the average estimated minimum upper bound of the mixed-weight OLD-set size. ‘< Bound’ is the percentage of graphs that had a mixed-weight OLD-set with size less than the minimum bound. ‘< 2× Bound’ is the same for twice the minimum bound. © 2018 IEEE

and with the estimated  $q$ , we can estimate the upper bound on the size of the mixed-weight OLD-set. To test these estimated bounds with large random geometric graphs, we used the ILP model from Chapter 7 to find mixed-weight OLD-sets and compared these results with the estimated upper bound.

To test the estimated bounds for random geometric graphs, we used the same graphs as those in Section 7.3 restricted to  $r \in \{0.25, 0.5\}$ , so that  $2r \leq a, b$ , and each value of  $\varrho \in \{0.25, 0.5, 0.75\}$ , for a total of 3000 graphs at each  $n$ . For  $r = 0.25$ , the average probability that two nodes are connected is  $\phi = 0.157$ , and for  $r = 0.5$ ,  $\phi = 0.483$ . The minimum possible estimated upper bound can be generated when  $\varepsilon = 1/\log(n)$ , which we call the *minimum bound*. For each graph, we compared the mixed-weight OLD-set size generated by the ILP model to the minimum bound and to twice the minimum bound.

As seen in Figure 9.1, as either  $n$  or  $r$  increase, the likelihood that the ILP result is less than the minimum bound decreases. As  $n$  increases, the random geometric graph will have more localized groups of almost complete subgraphs, which can increase the required

size of the mixed-weight OLD-set in order to satisfy the locating property.

For all graphs with a mixed-weight OLD-set with  $r = 0.25$ , the ILP model provided a size that was less than twice the minimum bound. However, when  $r = 0.5$ , the ILP results were less likely to be less than twice the minimum bound. When  $r = 0.5$ , all of  $\phi$  is calculated from edge cases, but when  $r = 0.25$ , fewer nodes are likely to fall on an edge case. Thus the probability of an adjacency at any location is more variable when  $r = 0.5$ , likely causing the bound to be less precise. However, when  $r = 0.5$ , most ILP results were less than quadruple the minimum bound (100.0%, 99.9%, and 99.7% for  $n = 50, 100$ , and 150, respectively).

### 9.3 Discussion

Due to the theoretical difficulty of their construction, estimation is a useful tool in the study of random geometric graphs. In this chapter we estimated mixed-weight OLD-set bounds for random geometric graphs using the average probability two nodes are adjacent. We applied the average probability to the probabilistic bounds we found in Chapter 6 for random graphs. The estimation performed better for random geometric graphs with smaller size and smaller radius. These results reinforce the difficult nature of working with random geometric graphs.

# Chapter 10

## Conclusion

In this thesis we addressed the problem of finding an optimal placement of networked sensors with varying strength by proposing the mixed-weight open locating-dominating set. The mixed-weight OLD-set provides a framework to study a wider range of location-detection problems than ever before, and mixed-weight OLD-sets are particularly useful in the study and design of wireless sensor networks. For mixed-weight OLD-sets, sensor strength is represented in a graph by placing weights on the nodes relative to the sensor strength. Nodes with a weight of one behave like unweighted nodes where neighbors are determined by edges. Nodes with higher weights are able to reach further in the graph, with the reach corresponding to the node's weight, becoming neighbors of other nodes in the graph. This simulates the ability of a higher strength sensor to monitor a larger physical area and receive communications at longer distances. In this chapter we discuss our contributions in Section 10.1, and in Section 10.2 we examine several areas of future work.

### 10.1 Contributions

The mixed-weight OLD-set provides the foundation for the theoretical study of networked sensors of different strengths. Related problems have been studied for systems with stronger than normal sensors [8, 15, 25, 26, 39, 45, 47, 50, 51, 66, 68, 70, 76], but this

is the first time strong sensors have been studied in OLD-sets and the first time varied strength sensors have been considered for any location-detection problems. Our contributions include a theoretical introduction to mixed-weight OLD-sets and their properties, results in a variety of graphs, establishing an integer linear program to solve for the sets, and examining greedy algorithms to estimate the sets in large graphs.

Basic properties of the mixed-weight OLD-set and a proof of the NP-completeness of its decision problem were provided in Chapter 3 and Chapter 4, respectively. In general we are interested in minimizing the size of the mixed-weight OLD-set, or, potentially, the total weight of the mixed-weight OLD-set, in order to reduce the cost of setup and maintenance of a related sensor network. However, finding the minimum mixed-weight OLD-set is an intractable problem.

Many intractable problems have polynomial-time solutions in special cases, and this is true for mixed-weight OLD-sets in paths and cycles where the weights are not preset. We provided a linear time solution for finding minimum-sized mixed-weight OLD-sets in paths and cycles with weights  $\leq 2$  in Chapter 5. Adding weights to nodes in a cycle results in the graph behaving like a directed subgraph of a circulant graph. We studied the non-weighted OLD-set in circulant graphs in Chapter 2, providing two proof techniques to find the lower bound on the size of the OLD-set. Although these results may be limiting for some applications, solutions in paths and cycles can provide mixed-weight OLD-set estimates for portions of larger graphs that are shaped like paths and cycles.

We established probabilistic upper bounds for the size of minimum mixed-weight OLD-sets in random graphs in Chapter 6 and used the results to estimate upper bounds for minimum mixed-weight OLD-sets in random geometric graphs in Chapter 9. Random graphs and random geometric graphs simulate the effect of distributing sensors at random in the field. Random graphs are probability-based, which often corresponds to simpler solutions in problems such as location-detection, but do not fully resemble randomly distributed sensor networks. Random geometric graphs are generated on a surface, similar to the placement of sensor networks in a physical area, and better correlated to randomly

distributed networks. However, the same properties that make random geometric graphs closely related to randomly distributed sensors also make them theoretically difficult to study.

To further explore random geometric graphs and mixed-weight OLD-sets, we developed an integer linear program (ILP) to find minimum-sized mixed-weight OLD-sets in Chapter 7. For the mixed-weight OLD-sets, the objective function of the ILP ensures that the mixed-weight OLD-set is minimum-sized, and the constraints of the ILP guarantee that the set has the properties of a mixed-weight OLD-set. Commercial ILP solvers use branch and bound to find a solution in significantly reduced time in most cases. We were able to find minimum mixed-weight OLD-sets using the modeling language AMPL and solver Gurobi in reasonable time for large geometric graphs in which the size would have been prohibitive using brute force search methods. Our results also showed that relaxing the integer constraint, in an effort to estimate small mixed-weight OLD-sets, provided low quality results, underscoring the difficulty of finding mixed-weight OLD-sets in random geometric graphs.

We also constructed several greedy algorithms to estimate small mixed-weight OLD-sets and tested their use in random geometric graphs in Chapter 8. Although greedy methods are not guaranteed to find optimal solutions for NP-complete problems, they have the potential to find reasonable estimates very quickly. We considered several greedy methods with different selection techniques, and found that the stingy algorithm, which starts with all nodes in the graph and removes unnecessary nodes from the mixed-weight OLD-set, provided the best estimation and was significantly faster than the ILP model.

## 10.2 Future Work

Our work with the mixed-weight OLD-set could be extended and expanded in several ways. Extensions from our current work include the use of edge weights to represent actual distances in the field, further exploration in paths and cycles, and improved estimation

algorithms for large random and random geometric graphs. We could broaden our research by studying mixed-weight identifying codes and constructing networks in such a way that they produce the smallest possible (mixed-weight) OLD-set. These ideas are discussed in greater detail below.

### Edge and Non-Integer Weights

One limitation of the mixed-weight OLD-set presented here is that the weight placed on a node represents exactly how many edges a node can reach into the graph. For applications, a variation of the mixed-weight OLD-set where edges are weighted with real values, and node weight is representative of a physical distance a sensor can reach, would be a more representative problem. In a random geometric graphs in particular, creating neighbor relationships at the point in which edges are created would be more practical. This setup would allow sensor strength and the ability to reach other locations to rely on actual distances instead of theoretical ones.

### More Paths and Cycles

We considered paths and cycles with all weight 2 and mixed-weight 1 and 2 mixed-weight OLD-sets, and there are many avenues for further exploration into paths and cycles. It would be interesting to answer questions such as *what is the maximum number of weight 1 nodes for minimum-sized mixed-weight OLD-sets in paths and cycles?* If weight 1 nodes represent weaker, but cheaper, sensors, maximizing their use could be essential to establishing affordable sensor networks. Similarly, we could ask the question, *what is the minimum total weight of the mixed-weight OLD-set?* We could also consider the question *what do mixed-weight OLD-sets look like in paths and cycles that have preset weights, in general and in structured cases?* For example, if every other node or every third is weight 2, the minimum-mixed-weight OLD-set may also have a polynomial-time solution. Finally, we could explore mixed-weight OLD-sets in paths and cycles with weights larger than 2.

## Improved Estimation Algorithms

Although our greedy algorithms provided good results, they relied on very basic search methods. We may be able to improve the results further by using more advanced optimization heuristics. One improvement could be Tabu search which uses local searches to make incremental improvements to a solution. In order to avoid becoming stuck in local optimum, Tabu search explores neighboring solutions and uses a memory mechanism that attempts to keep the search from revisiting the same “tabu” solutions over and over. Tabu search has been successful in scheduling and space planning, as well as classical graph problems such as graph coloring and the traveling salesman [36, 37, 38], making it an attractive option for mixed-weight OLD-sets.

## Identifying Codes

We briefly defined mixed-weight identifying codes in Section 3.2. Identifying codes were introduced before OLD-sets, and, although OLD-sets have better applications in volatile locations, identifying codes are more practical for situations where losing a sensor is unexpected. Identifying codes have been studied using stronger sensors of all the same weight [8, 15, 25, 26, 39, 45, 47, 50, 51, 66, 68, 70, 76], but never for sensors of varying weights. Thus, expanding into mixed-weight identifying codes would be straightforward and productive.

## Minimum Cardinality (Mixed-Weight) OLD-sets

We studied the minimum OLD-set size for networks that have predetermined connections, but it is also of interest to construct graphs that contain (mixed-weight) OLD-sets with minimum possible cardinality given the number of nodes in the graph. Graphs that contain the minimum possible identifying code have previously been studied [64]. In [14], graphs that have OLD-sets of size 2, 3, and  $n$  were explored without focusing on providing the minimum possible OLD-set for graphs at a particular size.

For  $k$ -sized OLD-sets in a graph with  $n$  nodes, the OLD-set can produce up to

$$\binom{k}{1} + \binom{k}{2} + \binom{k}{3} + \cdots + \binom{k}{k} = \sum_{i=1}^k \binom{k}{i} = 2^k - 1$$

unique and nonempty subsets. Thus the minimum cardinality of an OLD-set in a graph with  $n$  nodes would be  $k = \lceil \log_2(n+1) \rceil$ . Determining which graphs yield (mixed-weight) OLD-sets of minimum possible cardinality could aid in the design and cost reduction of structured sensor networks.

# Appendix A

## Symbol Table

### Graphs

$V$	the set of nodes in a graph
$E$	the set of edges in a graph
$C_n(1, t)$	a circulant graph with $n$ nodes and adjacencies 1 and $t$
$C_n$	a cycle with $n$ nodes
$P_n$	a path with $n$ nodes
$G(n, p)$	a random graph with $n$ nodes and probability $p$ for an edge
$G(n, r)$	a random geometric graphs with $n$ nodes and adjacencies within distance $r$
$d(x, y)$	the geodesic distance between two nodes in a graph
$d(u, v)$	the Euclidean distance between two points on a plane
$\varepsilon(x)$	the eccentricity of a node
$R(u)$	the range of a node in a random geometric graph

### (Mixed-weight) OLD-sets

$N(x)$	the open neighborhood, or neighborhood
$N(A)$	the open neighborhood of a set of nodes, $A$ , in a graph

$N[x]$	the closed neighborhood
$S$	a (potential) OLD-set
$N(x) \cap S$	the (potential) OLD-set neighborhood induces by (potential) OLD-set $S$
$\delta$	the optimal OLD-set density
$m$	the size of a cluster of nodes in an OLD-set
$w(x)$	the weight function for each node in the graph
$w(A)$	the combined weight of a set of nodes, $A$ , in a graph
$B^-(x)$	the open outgoing-ball of a node
$B^+(x)$	the open incoming-ball of a node

### Complexity

$\Delta$	a subgraph for the MW-OLD decision problem
$G_i$	the literal component for the MW-OLD decision problem
$H_j$	the clause component for the MW-OLD decision problem

### Probability

$p$	the probability of a edge in a random graph
$q$	the probability that a node is either in or not in both open incoming-balls of two other nodes
$p'$	the probability that two nodes are not in the open incoming ball of each other
$\varrho$	the probability of a weight 2 node in a random or random geometric graph
$\phi$	the average probability of an edge in a random geometric graph

## Integer Linear Programming

- $s_i$  variable that is 1 if node  $i$  is in the (mixed-weight) OLD-set, and 0 otherwise
- $A$  matrix defining the open incoming-ball of a graph with weighted nodes
- $C$  matrix defining the unique nodes in the open incoming-ball of two nodes
- $D$  matrix providing the length of shortest path between two nodes
- $E'$  the adjacency matrix of a graph

# Bibliography

- [1] E. ACKERMAN. Topher white: Repurposing cellphones to defend the rain forest, March 2015. [Online; posted 2-March-2015].
- [2] I.F. AKYILDIZ, D. POMPILI, AND T. MELODIA. Underwater acoustic sensor networks: Research challenges. *Ad Hoc Networks*, 3:257–279, 2005.
- [3] J.N. AL-KARAKI AND A.E. KAMAL. Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 11(6):6–28, December 2004.
- [4] K. APPEL AND W. HAKEN. Every planar map is four colorable. part i: Discharging. *Illinois Journal of Mathematics*, 21(3):429–490, September 1977.
- [5] N.A.A. AZIZ AND K.A. AZIZ. Managing disaster with wireless sensor networks. In *13th International Conference on Advanced Communication Technology (ICACT2011)*, pages 202–207, February 2011.
- [6] T.Y. BERGER-WOLF, W.E. HART, AND J. SAIA. Discrete sensor placement problems in distribution networks. *Mathematical and Computer Modelling*, 42(13):1385–1396, 2005.
- [7] J.W. BERRY, L. FLEISCHER, W.E. HART, C.A. PHILLIPS, AND J. WATSON. Sensor placement in municipal water networks. *Journal of Water Resources Planning and Management*, 131(3):237–243, 2005.

- [8] N. BERTRAND, I. CHARON, O. HUDRY, AND A. LOBSTEIN. Identifying and locating-dominating codes on chains and cycles. *European Journal of Combinatorics*, 25(7):969–987, October 2004.
- [9] S. BOCCALETTI, V. LATORA, Y. MORENO, M. CHAVEZ, AND D.U. HWANG. Complex networks: Structure and dynamics. *Physics reports*, 424(4-5):175–308, 2006.
- [10] B. BOLLOBÁS. *Random graphs*. Cambridge University Press, 2001.
- [11] K. CHAKRABARTY, M.G. KARPOVSKY, AND L.B. LEVITIN. Fault isolation and diagnosis in multiprocessor systems with point-to-point communication links. In *Fault-Tolerant Parallel and Distributed Systems*, pages 285–300. Springer, 1998.
- [12] I. CHARON, S. GRAVIER, O. HUDRY, A. LOBSTEIN, M. MOLLARD, AND J. MONCEL. A linear algorithm for minimum 1-identifying codes in oriented trees. *Discrete Applied Mathematics*, 154(8):1246–1253, 2006.
- [13] I. CHARON, O. HUDRY, AND A. LOBSTEIN. Minimizing the size of an identifying or locating-dominating code in a graph is NP-hard. *Theoretical Computer Science*, 290(3):2109–2120, 2003.
- [14] M. CHELLALI, N.J. RAD, S.J. SEO, AND P.J. SLATER. On open neighborhood locating-dominating in graphs. *Electronic Journal of Graph Theory and Applications (EJGTA)*, 2(2):87–98, 2014.
- [15] C. CHEN, C. LU, AND Z. MIAO. Identifying codes and locatingdominating sets on paths and cycles. *Discrete Applied Mathematics*, 159(15):1540–1547, 2011.
- [16] G. COHEN, I. HONKALA, A. LOBSTEIN, AND G. ZÉMOR. New bounds for codes identifying vertices in graphs. *Electronic Journal of Combinatorics*, 6(1):R19, 1999.
- [17] G. COHEN, I. HONKALA, A. LOBSTEIN, AND G. ZÉMOR. On identifying codes. In *Proceedings of the DIMACS Workshop on Codes and Association Schemes*, volume 56

- of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 97–110, 2001.
- [18] C. COLBOURN, P. SLATER, AND L. STEWART. Locating dominating sets in series parallel networks. *Congressus Numerantium*, 56:135–162, 1987.
- [19] D. CRANSTON AND G. YU. A new lower bound on the density of vertex identifying codes for the infinite hexagonal grid. *Electronic Journal of Combinatorics*, 16(1):R113, 2009.
- [20] A. CUKIERMAN AND G. YU. New bounds on the minimum density of an identifying code for the infinite hexagonal grid. *Discrete Applied Mathematics*, 161(18):2910–2924, December 2013.
- [21] Z. CVETANOVIC. Performance analysis of the alpha 21364-based hp gs1280 multi-processor. In *Proceedings of the 30th Annual International Symposium on Computer Architecture (ISCA '03)*, pages 218–229, New York, NY, USA, 2003. ACM.
- [22] D. DI PALMA, L. BENCINI, G. COLLODI, G. MANES, F. CHITI, AND R. FANTACCI. Distributed monitoring systems for agriculture based on wireless sensor network technology. *International Journal on Advances in Networks and Services*, 3(1):18–28, 2010.
- [23] E.W. DIJKSTRA. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.
- [24] W. DU, J. DENG, Y.S. HAN, P.K. VARSHNEY, J. KATZ, AND A. KHALILI. A pairwise key pre-distribution scheme for wireless sensor networks. *ACM Transactions on Information and System Security (TISSEC)*, 8(2):228–258, May 2005.
- [25] G. EXOO, V. JUNNILA, AND T. LAIHONEN. Locating-dominating codes in cycles. *Australasian Journal of Combinatorics*, 49:177–194, 2011.

- [26] G. EXOO, V. JUNNILA, AND T. LAIHONEN. Locatingdominating codes in paths. *Discrete Mathematics*, 311(17):1863 – 1873, 2011.
- [27] R.W. FLOYD. Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):345, June 1962.
- [28] F. FOUCAUD AND M.A. HENNING. Location-domination and matching in cubic graphs. *Discrete Mathematics*, 339(4):1221–1231, April 2016.
- [29] R. FOURER, D.M. GAY, AND B.W. KERNIGHAN. *AMPL: A Modeling Language for Mathematical Programming, 2nd Ed.* Cengage Learning, 2002.
- [30] A. FRIEZE, R. MARTIN, J. MONCEL, M. RUSZINKÓ, AND C. SMYTH. Codes identifying sets of vertices in random networks. *Discrete Mathematics*, 307(9-10):1094–1107, May 2007.
- [31] K. GENOVA AND V. GULIASHKI. Linear integer programming methods and approaches—a survey. *Journal of Cybernetics and Information Technologies*, 11(1), 2011.
- [32] M. GHEBLEH AND L. NIEPEL. Locating and identifying codes in circulant networks. *Discrete Applied Mathematics*, 161(13-14):2001–2007, 2013.
- [33] A. GHOBAKHLOUR, S. SHANMUGANATHAN, AND P. SALLIS. Wireless sensor networks for climate data management systems. In *18th IMACS/MODSIM World Congress*, pages 959–965, 2009.
- [34] R.M. GIVENS, R.K. KINCAID, W. MAO, AND G. YU. Mixed-weight open locating-dominating sets. In *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6, March 2017.
- [35] R.M. GIVENS, R.K. KINCAID, W. MAO, AND G. YU. An integer linear program for mixed-weight open locating-dominating sets. In *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6, March 2018.

- [36] F. GLOVER. Tabu searchpart i. *ORSA Journal on computing*, 1(3):190–206, 1989.
- [37] F. GLOVER. Tabu searchpart ii. *ORSA Journal on computing*, 2(1):4–32, 1990.
- [38] F. GLOVER AND M. LAGUNA. *Tabu search*. John Wiley & Sons, Inc., 1993.
- [39] S. GRAVIER, J. MONCEL, AND A. SEMRI. Identifying codes of cycles. *European Journal of Combinatorics*, 27(5):767–776, 2006.
- [40] GUROBI OPTIMIZATION, INC. Gurobi optimizer reference manual 7.5, 2017. [Online; accessed 9-May-2018].
- [41] P. HALL. On representatives of subsets. *Journal of the London Mathematical Society*, 10(1):26–30, 1935.
- [42] M.M. HALLDÓRSSON AND J. RADHAKRISHNAN. Greed is good: Approximating independent sets in sparse and bounded-degree graphs. *Algorithmica*, 18(1):145–163, May 1997.
- [43] J.K. HART AND K. MARTINEZ. Environmental sensor networks: A revolution in the earth system science? *Earth-Science Reviews*, 78(3):177–191, 2006.
- [44] I. HONKALA. On  $r$ -locating–dominating sets in paths. *European Journal of Combinatorics*, 30(4):1022–1025, 2009.
- [45] I. HONKALA, T. LAIHONEN, AND S. RANTO. On strongly identifying codes. *Discrete Mathematics*, 254(1-3):191–205, 2002.
- [46] I. HONKALA, T. LAIHONEN, AND S. RANTO. On locating-dominating codes in binary hamming spaces. *Discrete Mathematics and Theoretical Computer Science*, 6(2), 2004.
- [47] S. JANSON AND T. LAIHONEN. On the size of identifying codes in binary hypercubes. *Journal of Combinatorial Theory, Series A*, 116(5):1087–1096, 2009.

- [48] D.S. JOHNSON. Worst case behavior of graph coloring algorithms. In *Proceedings of the Fifth Southeastern Conference on Combinatorics, Graph Theory and Computing*, pages 513–527, 1974.
- [49] V. JUNNILA AND T. LAIHONEN. Optimal identifying codes in cycles and paths. *Graphs and Combinatorics*, 28(4):469–481, 2012.
- [50] V. JUNNILA AND T. LAIHONEN. Optimal lower bound for 2-identifying codes in the hexagonal grid. *Electronic Journal of Combinatorics*, 19(2):R38, 2012.
- [51] M.G. KARPOVSKY, K. CHAKRABARTY, AND L.B. LEVITIN. On a new class of codes for identifying vertices in graphs. *IEEE Transactions on Information Theory*, 44(2):599–611, 1998.
- [52] K. KHEDO, P. RAJIV, AND M. AVINASH. A wireless sensor network air pollution monitoring system. *International Journal of Wireless & Mobile Networks (IJWMN)*, 2:31–45, 2010.
- [53] R.K. KINCAID, A. OLDHAM, AND G. YU. Optimal open-locating-dominating sets in infinite triangular grids. *Discrete Applied Mathematics*, 193:139–144, 2015.
- [54] A.V. KOSTOCHKA AND B.Y. STODOLSKY. An upper bound on the domination number of  $n$ -vertex connected cubic graphs. *Discrete Mathematics*, 309(5):1142–1162, 2009.
- [55] J.B. KRUSKAL. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956.
- [56] M. LAIFENFELD AND A. TRACHTENBERG. Disjoint identifying-codes for arbitrary graphs. In *Proceedings of the International Symposium on Information Theory (ISIT)*, pages 244–248, 2005.
- [57] M. LAIFENFELD AND A. TRACHTENBERG. Identifying codes and covering problems. *IEEE Transactions on Information Theory*, 54(9):3929–3950, September 2008.

- [58] M. LAIFENFELD, A. TRACHTENBERG, R. COHEN, AND D. STAROBINSKI. Joint monitoring and routing in wireless sensor networks using robust identifying codes. In *Broadband Communications, Networks and Systems (BROADNETS 2007)*, pages 197–206, September 2007.
- [59] S. LIAW, G. CHANG, F. CAO, AND D.F. HSU. Fault-tolerant routing in circulant networks and cycle prefix networks. *Annals of Combinatorics*, 2(2):165–172, 1998.
- [60] A. LOBSTEIN. Watching systems, identifying, locating-dominating and discriminating codes in graphs: a bibliography, 2018. [Online; accessed 1-February-2018].
- [61] A. MAINWARING, D. CULLER, J. POLASTRE, R. SZEWCZYK, AND J. ANDERSON. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications (WSNA '02)*, pages 88–97. ACM, 2002.
- [62] P.D. MANUEL. Locating and liar domination of circulant networks. *Ars Combinatoria*, 101:309–320, 2011.
- [63] K. MARTINEZ, P. PADHY, A. RIDDOCH, H.L.R. ONG, AND J.K. HART. Glacial environment monitoring using sensor networks. In *Proceedings of Real-World Wireless Sensor Networks*. ACM Press, 2005.
- [64] J. MONCEL. On graphs on  $n$  vertices having an identifying code of cardinality  $\lceil \log 2(n+1) \rceil$ . *Discrete Applied Mathematics*, 154(14):2032–2039, 2006.
- [65] T. MÜLLER AND J. SERENI. Identifying and locating–dominating codes in (random) geometric networks. *Combinatorics, Probability and Computing*, 18(6):925–952, 2009.
- [66] M. PELTO. New bounds for  $(r \leq 2)$ -identifying codes in the infinite king grid. *Cryptography and Communications*, 2(1):41–47, 2010.
- [67] R.C. PRIM. Shortest connection networks and some generalizations. *Bell Labs Technical Journal*, 36(6):1389–1401, 1957.

- [68] S.M. RANTO. On binary linear  $r$ -identifying codes. *Designs, Codes, and Cryptography*, 60(1):81–89, 2011.
- [69] S. RAY, D. STAROBINSKI, A. TRACHTENBERG, AND R. UNGRANGSI. Robust location detection with sensor networks. *IEEE Journal on Selected Areas in Communications*, 22(6):1016–1025, 2004.
- [70] D.L. ROBERTS AND F.S. ROBERTS. Locating sensors in paths and cycles: The case of 2-identifying codes. *European Journal of Combinatorics*, 29(1):72–82, January 2008.
- [71] K. ROMER AND F. MATTERN. The design space of wireless sensor networks. *IEEE Wireless Communications*, 11(6):54–61, December 2004.
- [72] S.J. SEO AND P.J. SLATER. Open neighborhood locating-dominating sets. *Australasian Journal of Combinatorics*, 46:109–120, 2010.
- [73] S.J. SEO AND P.J. SLATER. Open neighborhood locating-dominating in trees. *Discrete Applied Mathematics*, 159(6):484–489, 2011.
- [74] S.J. SEO AND P.J. SLATER. Open neighborhood locating-domination for infinite cylinders. In *Proceedings of the 49th Annual Southeast Regional Conference*, pages 334–335. ACM, 2011.
- [75] S.J. SEO AND P.J. SLATER. Open neighborhood locating-domination for grid-like graphs. *Bulletin of the Institute of Combinatorics and its Applications*, 65:89–100, 2012.
- [76] B. STANTON. Improved bounds for  $r$ -identifying codes of the hex grid. *SIAM Journal on Discrete Mathematics*, 25(1):159–169, January 2011.
- [77] D.B. SWEIGART, J. PRESNELL, AND R. KINCAID. An integer program for open locating dominating sets and its results on the hexagon-triangle infinite grid and other graphs. In *Systems and Information Engineering Design Symposium (SIEDS)*, pages 29–32, April 2014.

- [78] R. UNGRANGSI, A. TRACHTENBERG, AND D. STAROBINSKI. An implementation of indoor location detection systems based on identifying codes. In *Intelligence in Communication Systems*, pages 175–189. Springer, 2004.
- [79] G. WERNER-ALLEN, K. LORINCZ, M. WELSH, O. MARCILLO, J. JOHNSON, M. RUIZ, AND J. LEES. Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing*, 10(2):18–25, March 2006.
- [80] Y. YANG, A. FUNAHASHI, A. JOURAKU, H. NISHI, H. AMANO, AND T. SUEYOSHI. Recursive diagonal torus: an interconnection network for massively parallel computers. *IEEE Transactions on Parallel and Distributed Systems*, 12(7):701–715, 2001.