Tabu Search for a Mixed Strength Sensor Location Problem

Rex K. Kincaid
Department of Mathematics
College of William & Mary
Williamsburg, VA 23187-8795
`rrkinc@wm.edu`

Robin M. Givens
Computer Science Department
Randolph Macon College
Ashland, VA 23005-5505
`robingivens@rmc.edu`

ABSTRACT

Location-detection problems are pervasive. Examples include the detection of faults in microprocessors, the identification of contaminants in ventilation systems, and the detection of illegal logging in rain forests. In each of these applications a network provides a convenient modelling paradigm. Sensors are placed at particular node locations that, by design, uniquely detect and locate issues in the network. Open locating-dominating (OLD) sets constrain a sensor's effectiveness by assuming that it is unable to detect problems originating from the sensor location. Sensor failures may be caused by extreme environmental conditions or by the act of a nefarious individual. Determining the minimum size OLD set in a network is computationally intractable, but can be modelled as an integer linear program. The focus of this work is the development of a tabu search metaheuristic for the minimum OLD set problem when sensors of varying strengths are allowed. Computational experience and solution quality is reported for geometric networks of up to 150 nodes.

ABOUT THE AUTHORS

Rex K. Kincaid is Chancellor Professor of Mathematics at the College of William & Mary where he has been a faculty member since 1984. He has served as the director for the M.S. degree program in Computational Operations Research since 1995. He completed his M.S. degree (1980) in Applied Mathematics and the Ph.D. (1984) in Operations Research from Purdue University. His research interests include discrete optimization, complex systems, network location theory and metaheuristics.

Robin M. Givens is an Assistant Professor of Computer Science at Randolph-Macon College where she has been a faculty member since 2016. She studied Mathematics and Computer Science at the University of Richmond, graduating in 2006. She completed her M.S. (2014) and Ph.D. (2018) in Computer Science at The College of William & Mary, with a focus on mixed-weight open locating-dominating sets. Her research interests span a variety of areas in computer science including graph theory, algorithms, simulation, and logic.

Tabu Search for a Mixed Strength Sensor Location Problem

Rex K. Kincaid
Department of Mathematics
College of William & Mary
Williamsburg, VA 23187-8795
`rrkinc@wm.edu`

Robin M. Givens
Computer Science Department
Randolph Macon College
Ashland, VA 23005-5505
`robingivens@rmc.edu`

INTRODUCTION

The merger of the locating and dominating requirements for a set of nodes on a network has led to a wealth of research activity in the last 20 years. The introduction of locating sets is generally attributed to (Hakimi, 1964), although both (Jordan, 1869) and (Hua, 1962) had earlier contributions. The origination of dominating sets is usually traced to (Ore, 1962). Combining the ideas of location and domination have led to a number of distinct problems including, locating-dominating sets (Slater, 1987; Colbourn et al., 1987), identifying codes (Cohen et al., 2001), open locating-dominating (OLD) sets (Seo and Slater, 2010, 2011) and metric locating-dominating (MLD) sets (Henning and Oellermann, 2004).

The detection and location of anomalies in a network is a central concern in a wide variety of application arenas. In sensor applications, sensors are placed at particular node locations designed to detect and locate anomalies that arise in the network. One way to lower initial costs and long-term maintenance costs for a collection of sensors, is to minimize the number of sensors needed to appropriately monitor the network. Unfortunately, these location–domination problems are, in general, computationally intractable. A finite, connected, simple graph $G(V,E)$ with node set V and edge set E provides a convenient model of a network or physical space. Nodes represent locations or regions and edges represent connections or communication ranges between those locations. The edges are assumed to be of unit length unless otherwise noted.

The focus of this paper is OLD sets. However, it is helpful to consider the following four related problems to understand the difference between OLD sets and other location-domination problems. Let an open neighborhood of a node v, $N(v)$, be the set of nodes adjacent to node v, excluding v. A closed neighborhood of a node v is denoted by $N[v]$ and is the union of $N(v)$ with v. Let $S \subseteq V$ denote a subset of nodes in a graph $G(V,E)$.

In (Slater, 2002) a set S is a Locating-Dominating (LD) set if every node not in S has at least one neighbor in S and there is no distinct pair of nodes not in S with the same set of neighbors in S. Hence, S is a LD set if both of the following conditions are met.

LD1 For all $v \in V - S$, $N(v) \cap S \neq \emptyset$.

LD2 For all $v, w \in V - S$ such that $v \neq w$, $(N(v) \cap S) \neq (N(w) \cap S)$.

The first requirement is a *dominating* constraint for $V - S$. while the second requirement is a *locating* constraint. The definition of Open Locating Dominating (OLD) Sets (Seo and Slater, 2010) is quite similar. A set S is an OLD set if every node in the graph has at least one neighbor in S and no two nodes in the graph have the same set of neighbors in S. That is,

OLD1 For all $v \in V$, $N(v) \cap S \neq \emptyset$.

OLD2 For all $v, u \in V$ such that $v \neq u$, $(N(v) \cap S) \neq (N(u) \cap S)$.

In both LD and OLD sets the set of neighbors is open but the domains are different, $V - S$ versus V. The open neighborhood distinguishes LD and OLD sets from node identifying codes (IC) in which a closed neighborhood is required. The set S is an identifying code if every node in the graph is dominated and no two nodes have the same set of neighbors. Specifically,

ID1 For all $v \in V$, $N[v] \cap S \neq \emptyset$.

ID2 For all $v, u \in V$ such that $v \neq u$, $(N[v] \cap S) \neq (N[u] \cap S)$.

The last related problem is the metric locating dominating set (Henning and Oellermann, 2004). The set S is called a *resolving set* if MLD2 is satisfied but MLD1 need not hold. The minimum cardnality resolving set is defined to be the *metric dimension* of G. S is a metric-locating-dominating (MLD) set if it is both a *resolving set* and a *dominating set*. That is, S is an MLD set if,

MLD1 For all $v \in V - S$, $N(v) \cap S \neq \emptyset$.

MLD2 For all $u \neq v \in V$ there is a $w \in S$ such that $d(u,w) \neq d(v,w)$.

The conditions OLD1 and ID1 are similar (closed versus open neighborhood) while LD1 and MLD1 are identical. The main difference between these two sets of definitions is the domain of the node v, either $V - S$ or V. Notice that the LD1 and MLD1 requirements do not change if $N[v]$ replaces $N(v)$ since the domain is restricted to $V - S$. It is only when the domain includes all of V that the distinction between $N(v)$ and $N[v]$ is important. It is easy to see that any set $S \subseteq V$ that satisfies the OLD1 condition also satisfies the LD1 and MLD1 conditions. When does the converse hold? That is, when does the condition $N(v) \cap S \neq \emptyset$ fail for $v \in S$. This failure can only occur if there there is a node in S that is *not* adjacent to any other node in S.

Consider the following graph in Figure 1. An MLD set is given by $S_{MLD} = \{v_1, v_2, w_1, w_2\}$ It is easy to see that S_{MLD} is a dominating set (MLD1) but tedious to check MLD2. An LD set is given by $S_{LD} = \{v_1, v_2, w_{12}, w_{22}, w_{11}, w_{21}\}$. Again, it is easy to see that S_{LD} is a dominating set. The locating constraint, LD2, only needs to be checked for nodes not in S_{LD}. Note that for any node $u \in S_{LD}$, $N(u) \cap S_{LD} = \emptyset$ due to the open neighborhood. Checking the nodes in $V - S_{LD}$ finds all of the intersections unique and non-empty. For example, $N(w_1) \cap S_{LD} = \{w_{11}, w_{12}\}$. An OLD set is given by the union of the previous two sets plus two additional nodes. That is, $S_{OLD} = \{v_1, v_2, w_{12}, w_{22}, w_{11}, w_{21}, w_1, w_2, k_1, k_4\}$. Adding $\{w_1, w_2, k_1, k_4\}$ to S_{LD} forces the intersections of elements in S_{OLD} with itself to be non-empty and unique. For example, $N(v_1) \cap S_{OLD} = \{k_1\}$. Note that S_{LD} satisfies ID1. The use of the closed neighborhood in ID1 makes each of the $N[u] \cap S_{LD} = u$ for all $u \in S_{LD}$. However, S_{LD} fails to meet the ID2 condition since $N[v_3] \cap S_{LD} = N[v_1] \cap S_{LD} = \{v_1\}$. A set that meets ID1 and ID2 is $\{v_1, v_2, w_1, w_2, k_1, k_2, k_3, k_4\}$. Thus, the four sets of interest in Figure 1 are

$$
\begin{aligned}
S_{MLD} &= \{v_1, v_2, w_1, w_2\} \\
S_{LD} &= \{v_1, v_2, w_{12}, w_{22}, w_{11}, w_{21}\} \\
S_{ID} &= \{v_1, v_2, w_1, w_2, k_1, k_2, k_3, k_4\} \\
S_{OLD} &= \{v_1, v_2, w_1, w_2, w_{12}, w_{22}, w_{11}, w_{21}, k_1, k_4\}.
\end{aligned}
$$

The paper is organized in the following way. Section 2 gives an integer linear programming formulation for the unweighted OLD set problem and notes a connection to the unicost set covering problem. Sections 3 and 4 are dedicated to mixed-weight OLD sets. Section 3 provides an example of a graph that does not admit an OLD set unless mixed-weights are allowed while section 4 summarizes the computational results for a tabu search procedure with a stingy heuristic for the neighborhood search. Concluding remarks are made in section 5.

A mixed-weight open locating-dominating set (mixed-weight OLD set), was introduced in (Givens et al., 2017; Givens, 2018) and extends the OLD set problem definition by allowing sensors of varying strengths. The varying sensor strengths are modelled through the placement of weights on the nodes of the graph. Just as an increased sensor strength expands the reach of a sensor throughout a region, an increase in weight on a node increases the reach of a node by an equal number of edges. Multiple types of sensors are often used in a wireless sensor network (WSN), such as those that monitor natural habitats (Mainwaring et al., 2002). WSNs have been used to study a variety of environmental areas such as glaciers (Martinez et al., 2005), marine pollution (Akyildiz et al., 2005), animal behavior and welfare (Mainwaring et al., 2002), and the effect of climate change on farming (Di Palma et al., 2010).

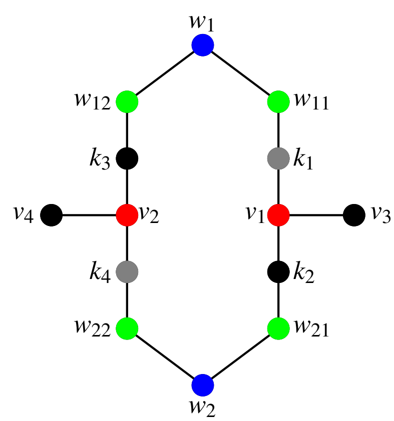


Figure 1: Compare LD, OLD, ID and MLD solutions

INTEGER LINEAR PROGRAM FOR MINIMUM OLD SETS

A natural goal, for each of the four problems identified in the previous section, is to seek a minimum number of node locations. One way to do this is to formulate the problem as an optimization model. The minimum OLD set problem was formulated as an integer linear program (ILP) for the first time in (Sweigart et al., 2014). The ILP assumes that the adjacency matrix A is available (note that $A_{i,j} = 1$ if nodes i and j are adjacenct and zero otherwise). The adjacency matrix serves to identify the open neighborhood of a node. From the adjacency matrix the shortest path distance matrix D can be computed (in polynomial time). Let $d(i,j)$ denote the shortest path distance between nodes i and j. Then, a binary matrix E is formed by letting $E_{i,j} = 1$ if $d(i,j) \leq 2$ and zero otherwise. If $d(i,j) > 2$ for any pair of nodes then the nodes cannot share any common neighbors. The resulting minimum OLD set ILP is given by

$$\text{Minimize: } \sum_{j \in V} x_j \tag{1}$$

$$\text{Subject To: } \sum_{j \in V} A_{i,j} x_j \geq 1 \qquad \forall i \in V \tag{2}$$

$$\sum_{k \in V} (A_{i,k} - A_{j,k})^2 x_k \geq E_{i,j} \qquad \forall i,j \in V \tag{3}$$

$$x_j \in \{0,1\} \qquad \forall j \in V \tag{4}$$

It is quite likely that no feasible solution may exist for the OLD set ILP. For simple, connected graphs this occurs when there is at least one pair of nodes with the same set of neighbors. Graphs with no such nodes are called *twin-free* (Foucaud et al., 2016). If the solution to the ILP for the OLD set problem has no feasible solution then it has a *twin*. Note that the OLD set ILP formulation is equivalent to the unicost set covering problem (SCP). The SCP is NP-complete but has been extensively studied (Yelbay et al., 2015). There are many heuristic search procedures that are known to generate high quality feasible solutions to the SCP (Lan et al., 2007).

In the set covering problem, as well as in the OLD set problem, all *customers*, located at the nodes, must be served. However, if the objects that are to be located have a cost associated with them, then there may be an additional *budget* constraint limiting the number of objects to be located, i.e. $\sum_{j \in V} x_j = k$ where k is a positive integer. When this constraint is added to the above formulation there will be values of k for which not all of the *customers* can be dominated and a new objective is needed. One reasonable objective, with a constraint limiting the size of the OLD set, is the one used in the maximum covering location problem. By maximizing coverage the constraint violations are minimized. In the OLD set problem there are two distinct types of constraints that may be violated. Constraint violations for the OLD1 dominating constraint are straight forward to minimize, but violations of the OLD2 constraint are less clear. One possibility is to only enforce the OLD2 constraint for those nodes that are dominated in the OLD1 constraint. In any case, when node weights other than one are added to the OLD set formulation, this version of the problem has additional interest since some nodes are now more important than others. Maximum covering location models for OLD sets have been proposed and studied in (Sweigart, 2019; Sweigart and Kincaid, 2017).

DEFINING MIXED-WEIGHT OLD SETS

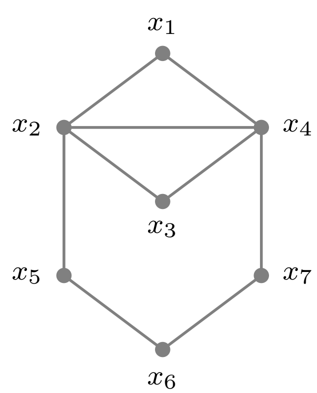
Allowing OLD set nodes to have non-uniform weights results in a mixed-weight OLD set problem. Without loss of generality, assume that OLD set locations are for sensors and that the graph represents a sensor network. In (Givens et al., 2017; Givens, 2018) a mixed-weight OLD set problem is defined in which each sensor has weight 1 or 2 Moreover, the potential weights for each node location is known apriori. If a weight 2 sensor is located at a node then a signal may be sensed up to 2 edges away. That is, the range of the antenna associated with the sensor is doubled. For the *unweighted* OLD set problem the OLD1 and OLD2 requirements rely solely on the neighborhood of a node and its interaction with the OLD set S. For weight 2 nodes the antenna strength allows not only the neighborhood to be served but also the neighbors of the neighbors. To capture this effect define, for each node $v \in V$, $B^{out}(v)$, the set of nodes that can be reached from v, as well $B^{in}(v)$, the set of nodes that can reach v. Let the *outgoing* ball $B^{out}(v) = \{u | u \in V$ and $d(v,u) \leq w(v)\}$ and the *incoming* ball $B^{in}(v) = \{u | u \in V$ and $d(u,v) \leq w(u)\}$. When all of the weights are 1, $N(v) = B^{out}(v) = B^{in}(v)$ for all $v \in V$. For the sensor location problem, $S \subseteq V$ is a mixed-weight OLD set if

MW-OLD1 for all $v \in V$, $B^{in}(v) \cap S \neq \emptyset$ and

MW-OLD2 for all $v, u \in V$ such that $v \neq u$, $(B^{in}(v) \cap S) \neq (B^{in}(u) \cap S)$.

Note that an equivalent mixed-weight OLD set problem can be defined by replacing the role of $B^{in}(v)$ with $B^{out}(v)$. In this version of the problem the locations *send* out a signal rather then *receive* a signal.

Not all connected, simple, undirected graphs admit an OLD set. Consider the following example. In the graph in Figure 2 all the node weights are 1. No OLD set solution exists since both x_1 and x_3 have the same incoming ball $B^{in}(x_1) = B^{in}(x_3) = \{x_2, x_4\}$. That is, x_1 and x_3 are *twins* with $N(x_1) = N(x_3)$. However in Figure 3, by allowing the weight at node x_3 to be 2, a mixed-weight OLD set solution can be found since all of the incoming balls, $B^{in}(v)$, are unique for all $v \in V$. For convenience, directed edges are added from the weight 2 node to all nodes two edges away, as is pictured in Figure 3.

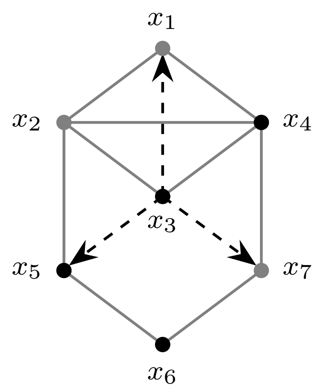


x	$N(x) = B^{in}(x)$
x_1	$\{\boldsymbol{x_2}, \boldsymbol{x_4}\}$
x_2	$\{x_1, x_3, x_4, x_5\}$
x_3	$\{\boldsymbol{x_2}, \boldsymbol{x_4}\}$
x_4	$\{x_1, x_2, x_3, x_7\}$
x_5	$\{x_2, x_6\}$
x_6	$\{x_5, x_7\}$
x_7	$\{x_4, x_6\}$

Figure 2: All node weights are 1; x_1 and x_3 have same incoming-ball $\{x_2, x_4\}$.

HEURISTIC SEARCH FOR MIXED-WEIGHT OLD SET

The computational experiments in this section make use of the geometric graph testbed generated in (Givens, 2018). Several heuristics were proposed and tested for constructing solutions to the mixed-weight OLD set problem (weight



x	$B^{in}(x)$
x_1	$\{x_2, x_3, x_4\}$
x_2	$\{x_1, x_3, x_4, x_5\}$
x_3	$\{x_2, x_4\}$
x_4	$\{x_1, x_2, x_3, x_7\}$
x_5	$\{x_2, x_3, x_6\}$
x_6	$\{x_5, x_7\}$
x_7	$\{x_3, x_4, x_6\}$

Figure 3: When $w(x_3) = 2$, $B^{in}(x_i)$ is unique for all $x_i \in V$, allowing for a mixed-weight OLD-set.

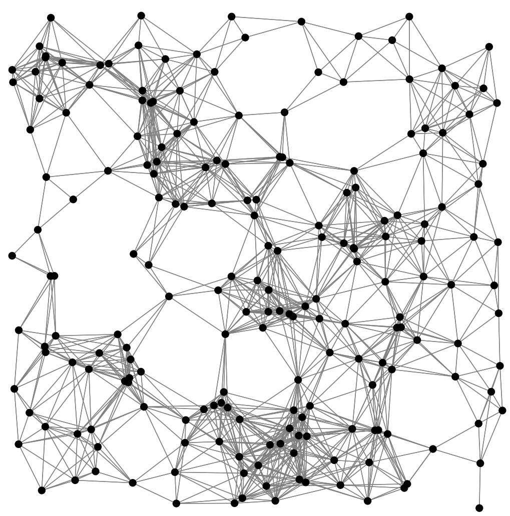


Figure 4: Random Geometric Graph on Unit Square, $r = 0.15$

1 and 2 node locations given) on randomly generated geometric graphs with 50, 100 and 150 nodes in (Givens, 2018). Geometric graphs (see Figure 4) are constructed by randomly generating an ordered pair of coordinates, (x, y), on the unit square. That is, both x and y are uniformly generated on the interval $(0, 1)$. Then, an edge connects any pair of coordinates (x_i, y_i) and (x_j, y_j) if the distance between the points is less than or equal to a pre-determined distance threshhold r. The node weights (1 or 2) are also randomly assigned with the probability of a weight 2 node, ρ, chosen from $\{0.25, 0.50, 0.75\}$. The distance r which governs the assignment of edges is also chosen from $\{0.25, 0.50, 0.75\}$. 500 graphs are generated for each combination of r and ρ, for a total of 4500 graphs. Of the 4500 graphs 4310 admitted OLD set for 50 nodes, 4444 admitted OLD sets for 100 nodes, and all 4500 admitted OLD sets for 150 nodes. The random geometric graphs may be a single connected component or multiple connected components. The heuristic which performed the best in (Givens, 2018) is a stingy heuristic. If a graph admits an OLD set then $S = V$ is an OLD set. The stingy heuristic deletes a node from the current OLD set, checks to see if the reduced set of nodes satisfies MW-OLD1 and MW-OLD2, and continues until no further reduction can be made. The most successful order in which to delete the nodes from the OLD set is based on the size of the out going ball for each node, $B^{out}(v)$ (Givens, 2018).

The stingy heuristic for the mixed-weight OLD set (Givens, 2018) is extended in two ways—a diversification scheme and a tabu search. The diversification approach takes as input an OLD set constructed by the stingy heuristic. Next, all pairwise swaps between nodes in S and $V - S$ are examined. After each swap, the resulting set is checked to see if

MW-OLD1 and MW-OLD2 are satisfied. If true, then the swap is accepted. Otherwise the swap is rejected. After all potential swaps have been examined, the resulting mixed-weight OLD set will be the same size as the one constructed by the stingy heuristic. Lastly, the stingy heuristic is invoked with the diversifed solution as the initial solution.

Tabu search and, more generally, adaptive memory programming continues to have startling success in efficiently generating high-quality solutions to difficult practical optimization problems. (Glover, 1996) provides forty-two vignettes each of which describes a different application of tabu search by researchers and practitioners. Constructing a simple tabu search heuristic requires defining a *move set* or *neighborhood*, the *tabu tenure* of an accepted move, an *aspiration criterion* and the maximum number of neighborhoods to be examined. One iteration of a simple tabu search examines the neighborhood of the current solution and either selects the first-improving neighbor or the least non-improving (non-tabu) neighbor. For the OLD set problem a *move* is the exchange of any node currently in the OLD set with a node that is not in the OLD set. Each move generates a *neighbor* of the current OLD set solution. A move is tabu if the exchange has previously been accepted and fewer than *tabu tenure* iterations have transpired. The search proceeds by first checking if the neighbor is an OLD set. If so, then the stingy heuristic is applied to the neighbor to see if the OLD set size can be decreased. If the move is not tabu and the OLD set size is decreased then the move is accepted and the OLD set is updated. A tabu move may be accepted if the OLD set size is smaller than any previously discovered OLD set (e.g meets the aspiration criterion). If a non-tabu move does not result in an OLD set then the node that was removed from the OLD set is appended. If the resulting (larger) set of nodes is an OLD set then the solution is considered as a possible least non-improving OLD set. At any iteration the first-improving move (decrease in OLD set size) is accepted and the iteration terminates. If no improving move is found among the neighbors, then the least non-improving (uphill) non-tabu solution is accepted.

The next four tables catalog the performance of the stand alone stingy heuristic, the previously described diversification scheme followed by the stingy heuristic, and a tabu search on the $4,500$ geometric graph instances. Optimal solutions are found by solving the integer linear programming (ILP) formulation via branch and bound. The ILP linear programming relaxation objective value is also recorded for comparison purposes as it provides an upper bound on the optimal OLD set size. For tabu search, the tabu tenure is set to one half the number of nodes and the maximum number of iterations (neighborhoods examined) is 50 for both the 50, 100 and 150 node cases in Table 1 and Table 2. The 150 node case requires about 50 minutes per geometric graph for tabu search on an iMac with a 3.2 GHz Intel Core i5 processor and 16 GB of 1867 MHz DDR3 memory. (In comparison, the computation time for the 100 node cases is about 6.3 minutes for each geometric graph and 15 seconds for each 50 node geometric graph.) As a result, tabu search is not tested on all $4,500$ of the 150 node geometric graphs. Instead, the 500 geometric graphs with $r = 0.25$ and $\rho = 0.75$ are used. Tabu search results for these experiments also set the tabu tenure to one half the number of nodes, but set the maximum number of iterations to the number of nodes (see Table 5).

Table 1 records the average optimal OLD set sizes (column 2) as well as the average OLD set sizes for each of the heuristics (columns 4-6) for the $4,500$ geometric graphs. Column 3 lists the average ILP relaxation objective values which provides an upper bound on the optimal value, but does not provide an OLD set (feasible solution). Table 2 records the average relative error, (heuristic value - optimal value)/(optimal value), for the same set of graphs. All three heuristics generate OLD sets with significantly better values than the ILP relaxation bound. Moreover, tabu search, which requires the most computational effort, provides the highest quality solutions.

Table 1: Average OLD set sizes for 4500 geometric graphs

n	Optimal	ILP-Relax	Stingy	Diverse+Stingy	Tabu
50	21.4	27.3	24.1	23.1	22.2
100	34.6	46.5	39.5	37.1	34.5
150	45.7	63.0	53.7	49.75	—

Results for the $4,500$ geometric graphs are shown for each grouping of 500 in Table 3 and Table 4. When $r = 0.75$ the resulting geometric graphs have the most edges of any of the graphs generated. The results show that these cases are the easiest ones to find a high quality OLD set. In fact, when $r = 0.75$ and $\rho = 0.75$ the optimal solution is uncovered by the stingy heuristic for all 500 graphs. The most challenging cases are when $r = 0.25$. These graphs have the

Table 2: Average Relative Error, (Heuristic-Optimal)/Optimal for 4500 geometric graphs

n	Optimal	ILP-Relax	Stingy	Diverse+Stingy	Tabu
50	21.4	27.6%	12.6%	8.1%	3.7%
100	34.6	34.4%	14.2%	7.2%	2.5%
150	45.7	37.9%	17.4%	8.9%	—

fewest edges. For the 3,000 geometric graphs with $r = 0.25$ or $r = 0.50$ the trend is the same, tabu search outperforms diversification followed by stingy which outperforms stingy. For example, the relative errors for the 500 geometric graphs generated with $r = 0.25$ and $\rho = 0.75$ were 31.8% for the stingy heuristic, 21.2% for diversification followed by stingy, and 13.6% for tabu search.

Table 3: Average OLD set sizes for 9 sets of 500 geometric graphs with 50 nodes

r, ρ	Optimal	Stingy	Diverse+Stingy	Tabu
0.25, 0.25	18.45	24.78	22.62	21.15
0.25, 0.50	17.74	24.47	22.29	20.77
0.25, 0.75	18.40	24.28	22.29	20.90
0.50, 0.25	15.50	19.23	17.58	15.62
0.50, 0.50	17.24	19.01	18.29	17.29
0.50, 0.75	22.25	22.70	22.52	22.26
0.75, 0.25	23.10	23.82	23.61	23.12
0.75, 0.50	25.72	26.00	25.89	25.73
0.75, 0.75	33.13	33.13	33.13	33.13

Table 4: Average OLD set sizes for 9 sets of 500 geometric graphs with 100 nodes

r, ρ	Optimal	Stingy	Diverse+Stingy	Tabu
0.25, 0.25	27.69	36.43	32.04	28.45
0.25, 0.50	27.11	35.99	31.40	28.09
0.25, 0.75	27.61	36.48	31.24	28.34
0.50, 0.25	24.73	33.62	29.65	25.34
0.50, 0.50	27.81	32.45	30.65	28.18
0.50, 0.75	36.48	37.66	37.23	36.55
0.75, 0.25	38.48	40.77	40.21	38.71
0.75, 0.50	43.51	44.32	44.07	43.56
0.75, 0.75	57.74	57.78	57.76	57.74

Table 5 catalogs the performance of tabu search for the OLD set problem on 500 geometric graphs. In Table 1 all nine families of 500 geometric graphs are tested. The results for the family of 500 geometric graphs recorded in Table 5 have an edge between two points on the unit square if the distance between the points was less than 0.25 and the probability of a weight 2 vertex was 0.75. Tabu tenure is set to one half the number of nodes and the maximum number of iterations (neighborhoods examined) is set to the number of nodes. Notice that the solution quality improved as the number of nodes increased and the number of instances in which the aspiration criterion is invoked also increased with the number of nodes as well. For the 150 node graphs tabu search uncovered a solution that satisfied the aspiration criterion in 310 out of the 500 instances. No instance had more than one solution that met the aspiration criterion.

Table 5: Tabu Search (500 geometric graphs with $r = 0.25$ and $\rho = 0.75$)

n	**Optimal**	**Tabu/%Opt**	**Tabu Tenure**	**Itrs**	**Aspiration**
50	18.4	20.9/13.6	25	6.0	0.05
100	27.6	28.2/2.2	50	23.7	0.34
150	35.8	36.4/1.7	75	50.5	0.62

CONCLUDING REMARKS

The main contribution in this paper is the development and testing of two heuristcs for the mixed-weight OLD set problem. The first imposes a diversification scheme upon the solution generated by a stingy heuristic and then re-applies the stingy heuristic to the diversified solution. The second is a tabu search heuristic which makes use of a short term memory function (tabu tenure) in an attempt to drive the solution away from locally optimal solutions. In addition, an aspiration criterion is included to allow tabu moves to be accepted if they lead to solutions that have not yet been uncovered. Both heuristics led to better OLD sets than those found by the stand alone stingy heuristic for the $4,500$ geometric graphs tested. It is expected that the same performance would be observed if all the weights on the nodes are uniform.

For the $4,500$ geometric graphs tested, the weight associated with each node is fixed, apriori. Deciding which weight, 1 or 2, to assign to each node doubles the number of decision variables in the ILP. Future efforts will be directed to this case. In (Sweigart, 2019; Sweigart and Kincaid, 2017) a mixed-weight OLD set ILP is formulated that allows sensors of any integer strength from $1, 2, \ldots R$ to be considered.

As was noted, the computational time for the tabu search when 150 nodes are considered is roughly 50 minutes per 50 iterations and does not scale well for larger graphs. Pursuit of strategies to address the excessive computing time are of interest. For example, a candidate list strategy may implemented in which a small sample of the neighborhood is examined at each iteration. Such candidate list strategies have proven successful in other application venues. Identifying good canditate lists for OLD set problems is a topic for future research.

As was noted in figure 1, MLD sets contain many fewer nodes than OLD sets. In addition, MLD sets arise in a variety of applications. Three applications of MLD sets and resolving sets are provided in (N. Mladenovic and Cangalovic, 2012): network discovery (Beerliova et al., 2006), comparing chemical compounds (G. Chartrand and Oellermann, 2000) and robot navigation (S. Khuller and Rosenfield, 1996). Consider a robot navigating a space described by a graph. The robot seeks information on its current position in the graph by sending out a signal to a set of landmarks and determines its distance from them. The problem is to compute the minimum number of landmarks as well as their locations so that the robot is always able to uniquely determine its current position. As was the case for the OLD set problem, it is straight forward to extend an optimization model (see (N. Mladenovic and Cangalovic, 2012)) for the MLD set problem to a maximum covering location problem when a budget constraint forces MLD1 or MLD2 to be violated. To the best of our knowledge this problem has not been studied.

REFERENCES

Akyildiz, I., Pompili, D., and Melodia, T. (2005). Underwater acoustic sensor networks: Research challenges. *Ad Hoc Networks*, 3:257–279.

Beerliova, Z., Eberhard, F., Erlebach, T., Hall, A., Hoffman, M., Mihalak, M., and Ram, L. (2006). Network discovery and verification. *IEEE Journal on Selected Areas in Communication*, 24:2168–2181.

Cohen, G., Honkala, I., Lobstein, A., and Zémor, G. (2001). On identifying codes. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 56:97 – 109.

Colbourn, C., Slater, P., and Stewart, L. (1987). Locating dominating sets in series parallel networks. *Congressus Numerantium*, 56:135 – 162.

Di Palma, D., Bencini, L., Collodi, G., Manes, G., Chiti, F., and Fantacci, R. (2010). Distributed monitoring systems for agriculture based on wireless sensor network technology. *International Journal on Advances in Networks and Services*, 3(1–2):18–28.

Foucaud, F., Henning, M., Lowenstein, C., and Sasse, T. (2016). Locating-dominating sets in twin-free graphs. *Discrete Applied Mathematics*, 200:52–58.

G. Chartrand, L. Eroha, M. J. and Oellermann, O. (2000). Resolvability in graphs and the metric dimension of a graph. *Discrete Applied Mathematics*, 105:99–113.

Givens, R. M. (2018). *Mixed-Weight Open Locating Dominating Sets*. PhD thesis, College of William and Mary.

Givens, R. M., Kincaid, R. K., Mao, W., and Yu, G. (2017). Mixed-weight open locating-dominating sets. In *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6.

Glover, F. (1996). Tabu search and adaptive memory programming–advances, applications, and challenges. In Barr, H. and Kennington, editors, *Interfaces in Computer Science and Operations Research*. Kluwer Academic Publishers.

Hakimi, S. (1964). Optimal locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12:450–459.

Henning, M. A. and Oellermann, O. R. (2004). Metric-locating-dominating sets in graphs. *Ars Combinatorica*, 73:129141.

Hua, L.-K. (1962). Application of mathematical methods to wheat harvesting. *Chinese Mathematics*, 2:77–91.

Jordan, C. (1869). Sur les assemblages de lignes. *J. Reine Angew. Math.*, 70:185–190.

Lan, G., Depuy, G., and Whitehouse, G. (2007). An effective and simple heuristic for the set covering problem. *European Journal of Operational Research*, 176:1387–1403.

Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., and Anderson, J. (2002). Wireless sensor networks for habitat monitoring. In *Proceedings of WSNA '02*, pages 88–97, New York, NY, USA. ACM.

Martinez, K., Padhy, P., Riddoch, A., Ong, H., and Hart, J. (2005). Glacial environment monitoring using sensor networks. In *In Proceedings of Real-World Wireless Sensor Networks*. ACM Press.

N. Mladenovic, J. Kratica, V. K.-V. and Cangalovic, M. (2012). Variable neighborhood search for metric dimension and minimal doubly resovlving set problems. *European Journal of Operational Research*, 220:328–337.

Ore, O. (1962). *Theory of Graphs*, volume 38. American Mathematical Society Colloquium Publications.

S. Khuller, B. R. and Rosenfield, A. (1996). Landmarks in graphs. *Discrete Applied Mathematics*, 70:217–229.

Seo, S. and Slater, P. (2010). Open neighborhood locating-dominating sets. *Australasian Journal of Combinatorics*, 46:109–120.

Seo, S. and Slater, P. (2011). Open neighborhood locating-dominating in trees. *Discrete Applied Mathematics*, 159(6):484–489.

Slater, P. J. (1987). Domination and location in acyclic graphs. *Networks*, 17:5564.

Slater, P. J. (2002). Fault-tolerant locating-dominating sets. *Discrete Mathematics*, 249:179–189.

Sweigart, D. (expected 2019). *Mixed Weight Open Locating Dominating Sets and Maximal Covering Models*. PhD thesis, College of William and Mary. unpublished thesis.

Sweigart, D. and Kincaid, R. (2017). Homogeneous and mixed weight open locating-dominating sets. In *International Symposium on Locational Decisions, ISOLDE XIV*, pages 210–212.

Sweigart, D., Presnell, J., and Kincaid, R. (2014). 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.

Yelbay, B., Birbil, S. I., and Bulbul, K. (2015). The set covering problem revisited: An empirical study of the value of dual information. *Journal of Industrial and Management Optimization*, 11(2):575–594.