Integer Linear Programming Models and Greedy Heuristic for the Minimum Weighted Independent Dominating Set Problem

Stefan Kapunac

Faculty of Mathematics, University of Belgrade
Studentski Trg 16, 11000 Belgrade, Serbia
stefan.kapunac@matf.bg.ac.rs

Abstract. This paper explores the Minimum Weighted Independent Dominating Set Problem and proposes novel approaches to tackle it. Namely, two integer linear programming formulations and a fast greedy heuristic as an alternative approach are proposed. Extensive computational experiments are conducted to evaluate the performance of these approaches on the established set of benchmark instances for the problem. The obtained results demonstrate that the introduced integer linear programming models are able to achieve optimal solutions on all instances with 100 nodes and significantly outperform existing exact methods on numerous other instances. Additionally, the greedy heuristic exhibits superior performance compared to competing greedy heuristics, particularly on random graphs. These findings suggest promising directions for future research, including the integration of these methods into hybrid algorithms or metaheuristic frameworks.

Keywords: domination problems, integer linear programming, greedy heuristic.

1. Introduction

Domination problems in graphs have been studied extensively from both theoretical and practical perspectives due to their wide-ranging applications in various fields, including graph mining [4], routing in wireless sensor and ad-hoc networks [23], social network influence [8], bioinformatics [15], etc.

The canonical problem within this class of problems is the minimum dominating set problem (MDSP). Given an undirected graph $G = (V, E)$, a dominating set is defined as $S \subseteq V$, such that every node $v \in V$ is either element of S or is connected by an edge to node $u \in S$. The goal is to find such set S of minimum cardinality. While the literature has explored various variations of the MDSP, such as the minimum connected dominating set problem [3], minimum capacitated dominating set problem [16], and minimum total dominating set problem [13], this paper focuses on the minimum weighted independent dominating set problem (MWIDSP).

This NP-hard problem was originally introduced in [5] as a weighted generalization of the minimum independent dominating set problem (MIDSP). MIDSP has applications in feature selection [19, 1], virtual backbone network construction in mobile ad-hoc networks [22], clustering in wireless sensor networks [20]. The weighted variant of the problem is applicable in analogous scenarios, allowing for the assignment of weights to both nodes and edges of the graph. For instance, in the context of wireless network clustering, node weights may be determined by various factors including node degree, mobility and residual energy [6], while edge weights may naturally represent the distances between nodes.

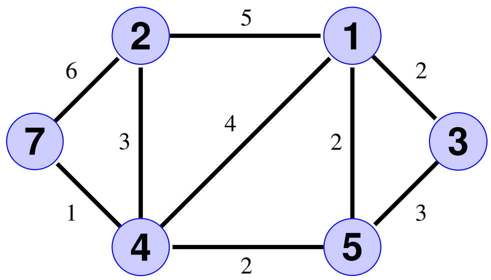
1.1. Problem definition

A set $S \subseteq V$ is called independent if for any pair of nodes $u, v \in S$ there is no edge $\{u, v\} \in E$. A set S is called independent dominating set if it is independent and dominating at the same time. The MWIDSP considers undirected graphs with node and edge weights that are positive numbers. The aim is to find an independent dominating set S that minimizes the following objective function:

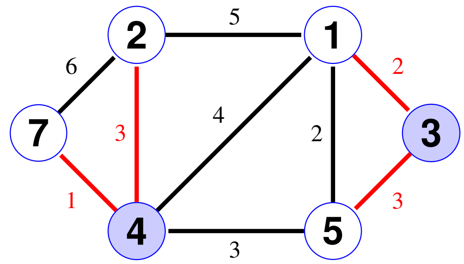
$$obj(S) = \sum_{v \in S} w_v + \sum_{v \in V \setminus S} \min\{w_e \mid e = \{v, u\}, u \in S\} \tag{1}$$

Accordingly, the objective function value of set S is calculated by adding the weights of the nodes within S to the weights of the minimum-weight edges connecting nodes outside of S to nodes within S.

1.2. Example instance



(a) Problem instance.
The numbers represent the weights of nodes and edges.



(b) The optimal solution of the problem instance (a). The set S consists of the nodes filled in blue color. Red edges contribute to the second and third sum of the objective function. The objective function value is (4 + 3) + (1 + 3 + 2 + 3) = 7 + 9 = 16.
Best viewed in color.

2. Related work

A linear time algorithm has been proposed in [5] to solve the MWIDSP in series-parallel graphs. There are two papers that handle the general graphs. In [18] multiple approaches have been proposed:

- Three integer linear programming models
  1. ILP-1 is a model based on three sets of indicator variables, that are vertex binary variables x_v denoting if node v is chosen for the solution, edge binary variables y_e indicating if edge e is choosable, and edge binary variables z_e indicating if edge e is selected for connecting a non-chosen node to a chosen one. This model has a total of $|V| + 2 \cdot |E|$ variables and $3 \cdot |V| + 5 \cdot |E|$ constraints.
  2. ILP-2 is a similar model that eliminates the set of y_e variables while utilizing additional constraints. This model comprises a total of $|V| + |E|$ variables and $3 \cdot |V| + 4 \cdot |E|$ constraints.
  3. ILP-3 is a structurally different model based on the idea of explicitly modeling edge-weight contribution of each node using integer variables q_v. This model utilizes $2 \cdot |V|$ variables and $5 \cdot |V| + 2 \cdot |E|$ constraints in total.
- Two greedy heuristics
  1. GREEDY-1 starts with an empty solution S and iteratively adds a node v with the maximum $\frac{|N_{G'}(v)|}{w_v}$ value, where G' is the remaining graph after node selection and $N_{G'}(v)$ is the neighborhood of v in G'. After adding a node v to S, v and all of its neighbor nodes and all their incident edges are removed from G'.
  2. GREEDY-2 also considers edge weights. Specifically, it uses an auxiliary objective function f^{aux} defined as the sum of contributions of nodes with respect to the partial solution S. Node $v \in S$ has contribution of w_v, and node $v \notin S$ has contribution of $\min\{w_{uv} \mid u \in S\}$ if it has neighbors in S, otherwise it is defined as the maximum edge weight in the whole graph.
- Population based iterated greedy (PBIG) metaheuristic based on the probabilistic version of the GREEDY-2 heuristic, and it is applied both in isolation and as an integral component of the construct, merge, solve and adapt (CMSA) [2] framework. This hybrid is denoted as CMSA-PBIG.

The current state-of-the-art method is a local search algorithm with reinforcement learning inspired repair procedure (LSRR) proposed in [21]. Three different scoring functions are introduced that are specifically suited for different weight properties of nodes and edges, and are used within a greedy heuristic and local search.

2.1. Main contribution

Building upon established methodologies used in solving combinatorial optimization problems such as the uncapacitated facility location problem (UFLP)[11] and the weighted total domination problem (WTDP)[14], two novel ILP models are introduced tailored to address the MWIDSP. These models offer significant advancements over existing exact methods, showcasing superior performance in terms of solution quality and computational efficiency.

In addition to the ILP models, a novel greedy algorithm is specifically designed for MWIDSP. This algorithm surpasses the performance of two competing greedy algorithms from the literature, demonstrating its effectiveness in producing high-quality solutions efficiently.

3. ILP models

In this section two alternative ILP models are presented that, as is shown in Section 5, outperform competing models from [3].

3.1. NEW-1 ILP model

The first of the two newly proposed ILP models, referred to as NEW-1, employs two sets of binary variables. Specifically, x_u for each node $u \in V$ indicates whether or not u is selected for inclusion in the solution. Additionally, y_{uv} is assigned a value of 1 for each node pair (u, v) if node $v \in V$ is adjacent to the node $u \in S$ through the arc $(u, v) \in A$. Here, $A = \{(u, v) \cup (v, u) \mid \forall e = \{u, v\} \in E\}$ represents the set of bidirected arcs associated with E. Before presenting the formulation, let us define the set of incoming arcs to node u as $\delta^-(u)$, and, analogously, the set of outgoing arcs from node u as $\delta^+(u)$. Using this notation, the MWIDSP can be formulated as:

$$\min \sum_{u \in V} w_u x_u + \sum_{(u,v) \in A} w_{uv} y_{uv} \tag{2}$$

$$s.t.$$

$$x_u + x_v \leq 1, \quad \forall e = \{u, v\} \in E \tag{3}$$

$$x_u + \sum_{(v,u) \in \delta^-(u)} y_{vu} = 1, \quad \forall u \in V \tag{4}$$

$$y_{uv} \leq x_u, \quad \forall (u, v) \in A \tag{5}$$

$$x_u \in \{0, 1\}, \quad \forall u \in V \tag{6}$$

$$y_{uv} \in \{0, 1\}, \quad \forall (u, v) \in A \tag{7}$$

Constraints (3) ensure no two neighboring nodes are elements of the solution, threreby enforcing the solution to be an independent set. Additionally, constraints (4) guarantee that every node u is either inside the solution, or is covered by exactly one neighbor v from the solution. Moreover, constraints (5) verify that node u must be included in the solution if it is to cover its neighbor v. Coupled with the second part of the objective (2), they ensure the accurate computation of edge weights connecting nodes outside the solution to those within. Final two sets of constraints (6) and (7) affirm that, as previously stated, variables x and y are binary.

The total number of variables in this formulation amounts to $|V| + 2 \cdot |E|$, while the number of constraints is $2 \cdot |V| + 5 \cdot |E|$.

3.2. NEW-2 ILP model

Drawing inspiration from previous work on similar problems [11, 14], in the second ILP model named NEW-2, Benders decomposition is employed to project out the y variables. Continuous variables q are introduced to account for the edge weights connecting nodes outside the solution to those within. Let us define $N'(u)$ as the list of neighboring nodes of node u arranged in ascending order on the edge weights connecting them to u. Utilizing this notation, MWIDSP can be expressed as:

$$\min \sum_{u \in V} w_u x_u + q_u \tag{8}$$

$$s.t.$$

$$x_u + x_v \leq 1, \quad \forall e = \{u, v\} \in E \tag{9}$$

$$x_u + \sum_{v \in N(u)} x_v \geq 1, \quad \forall u \in V \tag{10}$$

$$q_u \geq w_{su} - \sum_{t=1}^{s-1} (w_{su} - w_{tu}) x_t - w_{su} x_u, \quad \forall u \in V, \forall s \in \{1, \ldots, |N'(u)|\} \tag{11}$$

$$x_u \in \{0, 1\}, \quad \forall u \in V \tag{12}$$

$$q_u \geq 0, \quad \forall u \in V \tag{13}$$

Constraints (9), as before in NEW-1, represent the independent set condition. Constraints (10) make sure the solution is a dominating set. Let us break the third set of constraints (11) in two cases for easier exposition. First, when $x_i = 0$, indicating node i is not in the solution, the constraints become similar to Benders optimality cuts for UFLP and WTDP, and the variable q_i will have the value of minimum weight edge connecting node i to some node in the solution. On the other hand, when $x_i = 1$, the right hand side is at most zero, so the value of q_i will be zero for every node within the solution. Constraints (12) and (13) state that variables x and q are binary and positive continuous, respectively.

The total number of variables in this formulation is $2 \cdot |V|$, while the number of constraints is $3 \cdot |V| + 3 \cdot |E|$. It is worth noting that the number of constraints (11) equals the sum of all node degrees in the graph, i.e., $2 \cdot |E|$. The remaining calculations follow straightforwardly.

4. Greedy algorithm

The newly constructed greedy algorithm starts from an empty solution S. Then, at each iteration it adds to S the highest score node v^* from the set of all nodes not currently covered V'. A node is considered covered if it adheres to the independent dominating set constraint, meaning it either belongs to the solution or neighbors a node within the solution. The score value of a node v is calculated as:

$$score(v) = \frac{\sum_{u \in N(v)} (w_u + \sum_{t \in N(u) \setminus \{v\}} w_{ut})}{w_v + \sum_{u \in N(v)} w_{uv}} \tag{14}$$

As the aim is to maximize the score value, a large numerator and a small denominator are desirable. The denominator indicates a preference for the total weight of node v and its incident edges to be relatively small, which aligns well with the objective function (1). Conversely, the numerator denotes that neighboring nodes of v and their incident edges (except for those connecting them to v) are preferably of relatively large weights. This represents the gain achieved after covering those nodes with v – their large weight and large edge weights to potential alternative nodes from the solution need not be accounted for.

An important note is that after adding the best node with respect to the score (14), its closed neighborhood $N[v^*]$ is removed from the graph, as all those nodes are now covered. Specifically, all the neighbors of v^* and their incident edges are removed. Therefore, they do not affect the summations of edge weights, and transitively, the score, in the following iterations of the algorithm.

The pseudocode of the greedy algorithm is given in Algorithm 1.

Algorithm 1 GREEDY-NEW

```
Input: graph G = (V, E)
Output: solution S
S ← ∅
G' = (V', E') ← G = (V, E)
edgeWeights ← [∑_{v∈N(u)} w_uv | u ∈ V]
while size(V') > 0 do
    v* ← −1
    scoreMax ← −1
    for v ∈ V' do
        nodeWeights ← 0
        uvEdgeWeights ← 0
        otherEdgeWeights ← 0
        for u ∈ N_G'(v) do
            nodeWeights ← nodeWeights + w_u
            uvEdgeWeights ← uvEdgeWeights + w_uv
            otherEdgeWeights ← otherEdgeWeights + edgeWeights_u − w_uv
        end for
        score ← (nodeWeights + otherEdgeWeights)/(w_v + uvEdgeWeights)
        if score > scoreMax then
            scoreMax ← score
            v* ← v
        end if
    end for
    S.add(v*)
    for v ∈ N_G'[v*] do
        for u ∈ N_G'(v) do
            edgeWeights_u ← edgeWeights_u − w_uv
        end for
        remove node v from G'
    end for
end while
return S
```

The worst-case time complexity of the algorithm is $O(|V| \cdot |E|)$.

5. Experimental evaluation

The two newly introduced ILP models and the greedy algorithm are evaluated on a comprehensive set of benchmark instances previously established in the literature. Their results are compared to 8 competing approaches mentioned in the Section 2. All methods are implemented in PYTHON. The experiments were conducted on a PC with Intel Core i9-11900 @ 2.5GHz CPU with a memory limit of 4GB RAM per execution, under Ubuntu 22.04 OS. The ILP models were solved using the CPLEX 20.1 solver.

5.1. Benchmark instances

The same set of benchmark instances as the competition [18, 21] was utilized. There are two types of graphs within the benchmark set: random graphs and random geometric graphs. For both of these types graphs of varying size were generated, specifically $|V| \in \{100, 500, 1000\}$.

The random graphs were generated adding edges between nodes at random, using the Erdos-Renyi model [9], with a probability ep for each edge. Thus, the parameter ep controls the density of the graph, with $ep \in \{0.05, 0.15, 0.25\}$ representing graphs of different densities.

On the other hand, the random geometric graphs were generated assigning random coordinates from the unit square to nodes and then connecting any two nodes at a distance smaller than r by an edge. This graph generation method is also known as the Unit disk model [7] and is often used to simulate ad-hoc wireless networks [10]. In this case the graph density is controlled by the radius r. To match the random graphs densities the values of $r \in \{0.14, 0.24, 0.34\}$ were used.

The node and edge weights were generated using three different schemes:

1. Both node and edge weights were drawn uniformly at random from the set $\{0, \ldots, 100\}$. This group is referred to as the neutral graphs (NG).
2. Node weights were drawn uniformly at random from the set $\{0, \ldots, 1000\}$, and edge weights uniformly at random from the set $\{0, \ldots, 10\}$. Consequently, in these node-oriented graphs (NG), the choice of nodes is crucial.
3. Node weights were drawn uniformly at random from the set $\{0, \ldots, 10\}$, and edge weights uniformly at random from the set $\{0, \ldots, 1000\}$. Therefore, in this group, called edge-oriented (EG) graphs, the selection of edges is key.

For each combination of the graph type, number of nodes, edge probability ep (or radius r), and weight generation scheme, 10 instances were generated. Hence, the benchmark set consists of a total of 540 graphs, comprising 270 random graphs and 270 random geometric graphs.

The problem instances, along with the implementation of the proposed methods, are publicly available at `https://github.com/StefanKapunac/mwids_public`.

5.2. Experimental results

The time limit for each algorithm is $3{\cdot}|V|$ seconds per graph. The results are presented in two pairs of tables, each corresponding to one of the two graph types (random or random geometric) and algorithm types (exact or heuristic). Tables 1 and 2 display the results of exact methods for all random and random geometric graphs, respectively. Similarly, Tables 3 and 4 showcase the results of heuristic methods for all random and random geometric graphs, respectively. All tables are structured as follows. The first three columns represent the number of nodes in the graph ($|V|$), the weight scheme, and the graph density parameter, i.e. edge probability ep for random graphs and radius r for random geometric graphs. The subsequent columns contain the results of the considered algorithms. Namely, the column named *result* is the average result obtained across the corresponding 10 problem instances. Similarly, *time* column indicates the average time (in seconds) taken to find the solution, across the 10 corresponding problem instances. Besides the aforementioned *result* and *time*[1], the results of all ILP models include a *gap* column, representing the average optimality gap (in percentage). The overall best result is highlighted in bold, while the best result among the greedy heuristics is underscored.

Based on the experimental results of exact methods, the following observations can be made:

– NEW-1 and NEW-2 obtain higher quality results on the vast majority of random instances, in comparison to the other three ILP models. Additionally, both models successfully solve all instances with 100 nodes to optimality, in contrast to models from [18] that achieve optimality in only 4 or 5 groups out of the total 9 instance groups of that size. It is worth noting that the average execution time of NEW-1 and NEW-2 for instances with 100 nodes is relatively short, with around half completing in less than 1 second. The longest execution time observed is 35.3 seconds for the densest edge-oriented graphs.
– NEW-1 and NEW-2 consistently outperform the three competing ILP models across the majority of random geometric instances. Both new models successfully solve all instances with 100 nodes to optimality. Additionally, they achieve optimality for all node-oriented instances with 500 nodes and some instances with 1000 nodes. Specifically, NEW-1 solves the node-oriented instance group with the lowest density (radius $r = 0.14$), while NEW-2 solves nearly all node-oriented instances except for three instances from the densest group (radius $r = 0.34$). It is worth mentioning that the execution times of NEW-1 and NEW-2 are significantly lower than those of the metaheuristics in some cases, especially in lower density instances.

The experimental results of heuristic methods yield the following observations:

[1] Note that information regarding the average time for the ILP models from [18] was not available.

Table 1: Numerical results of exact methods on random graphs

$\lvert V\rvert$	weight scheme	ep	ILP-1 result	ILP-1 gap	ILP-2 result	ILP-2 gap	ILP-3 result	ILP-3 gap	NEW-1 result	NEW-1 gap	NEW-1 time	NEW-2 result	NEW-2 gap	NEW-2 time
100	NG	0.05	**3049.8**	0.7	**3049.8**	0	3051.9	0.5	**3049.8**	0	0.2	**3049.8**	0	0.1
		0.15	2445.2	30.4	2396.3	28.3	2398.4	44.6	**2330.2**	0	2.2	**2330.2**	0	2.5
		0.25	2161.1	31.3	2114.3	24.3	2167.6	58.8	**2069.3**	0	8.3	**2069.3**	0	8.8
	VG	0.05	**7715.4**	0	**7715.4**	0	**7715.4**	0	**7715.4**	0	0.2	**7715.4**	0	0.1
		0.15	**3046.6**	0	**3046.6**	0	**3046.6**	0	**3046.6**	0	0.8	**3046.6**	0	0.5
		0.25	**1808.4**	0	**1808.4**	0	**1808.4**	0	**1808.4**	0	1.1	**1808.4**	0	0.5
	EG	0.05	**14378.7**	0	**14378.7**	0	**14378.7**	0	**14378.7**	0	0.4	**14378.7**	0	0.1
		0.15	15473	42.4	15198.9	39.6	15098.3	57.7	**14563.3**	0	17.7	**14563.3**	0	11.1
		0.25	16001.7	61.5	14557.7	28.5	15346.3	69.3	**14382.2**	0	35.3	**14382.2**	0	28.5
500	NG	0.05	11857.9	51.9	11809.3	50.5	12882.6	91.7	**10869.1**	30.7	TLE	11163.6	32.5	TLE
		0.15	10050.1	69.4	9928.8	68.1	13196.7	98.1	9568.7	55.1	TLE	**9348.5**	54.4	TLE
		0.25	11341.1	78	9465.7	73.9	12722.4	99	**9078.9**	62.5	TLE	9099.1	62.5	TLE
	VG	0.05	12557.6	52.6	11403	47.1	12059.3	58	10529.8	34.7	TLE	**10048.3**	29.5	TLE
		0.15	5940.1	61.3	6122.4	59.3	5474.6	80.5	4815.5	48.7	TLE	**4107.2**	31.3	TLE
		0.25	8166.2	79.4	10145.8	82.9	3628.7	85.6	3514.6	45.1	TLE	**2749.4**	14.3	TLE
	EG	0.05	97915.3	82.9	89580.7	81.1	107463.5	98.7	87091.4	70.5	TLE	**83820.5**	69.6	TLE
		0.15	107834.7	93.4	91564.1	91.7	117036.1	99.9	**84871.4**	84	TLE	86965	84.5	TLE
		0.25	100750.3	95	88687.7	94	113798	99.9	**87735.9**	87.6	TLE	100113	89	TLE
1000	NG	0.05	25156.1	69.5	25986	68.7	27158.7	97	**21151.2**	51.9	TLE	21308.9	52.5	TLE
		0.15	21117.8	81.1	**18282.9**	76.5	22984.5	99.1	18554.6	68.1	TLE	21105.9	71.9	TLE
		0.25	158097.1	93.2	88053.8	88.7	21821.5	99.5	**17600.2**	73.1	TLE	19726.8	75.9	TLE
	VG	0.05	35766.3	83.3	39356.1	83.3	15464.1	77.4	14648.2	56.7	TLE	**13056.9**	50.4	TLE
		0.15	35678.2	91.3	35904.3	97	11586.3	92.3	6596.4	51.8	TLE	**5733.7**	42.4	TLE
		0.25	35838.9	96.5	22569.5	100	11674.7	96.7	7311	60.7	TLE	**4242.6**	38.2	TLE
	EG	0.05	209391.8	91.1	198540.7	90.1	229778.7	99.7	**190225.4**	84.7	TLE	197814.4	85.7	TLE
		0.15	832437.1	97.7	**191911.2**	96.2	229888.1	100	304074.6	94.7	TLE	194334.6	91.8	TLE
		0.25	1128240.1	98.6	2041102.4	99.7	221492	100	363299.1	97.1	TLE	**186606.1**	93.2	TLE

- GREEDY-NEW consistently yields higher quality solutions across nearly all random graphs, compared to the other two greedy heuristics. The only two exceptions are observed in the group of node-oriented instances with 100 nodes and 0.05 or 0.15 edge probability, where GREEDY-1 achieves slightly better average result.
- The comparison between greedy heuristics for random geometric graphs is not as definitive as it is for random graphs. However, on average, GREEDY-NEW algorithm achieves higher-quality solutions compared to the competing GREEDY-1 and GREEDY-2.
- Metaheuristic approaches outperform greedy heuristics, particularly on larger graphs, prioritizing higher quality at the expense of longer execution times. Notably, LSRR stands out as a state-of-the-art algorithm, achieving optimal solutions for all instances where the optimal solution is known, matching the results of the two proposed ILP models.

5.3. Statistical analysis

The statistical significance of the observed differences among the competing approaches was assessed using the following statistical methodology. Initially, all algorithms were collectively analyzed using the Friedman's test [12]. Subsequently, in cases where the null hypothesis was rejected (H_0 indicating that competitor approaches are statistically equal), pairwise comparisons were conducted using the Nemenyi's post-hoc test [17].

The corresponding outcomes are depicted in Figure 1 through critical difference plots. Essentially, each approach is positioned within the segment based on its average ranking across the considered subset of instances. Subsequently, the critical difference (CD) is calculated for a significance level of 0.05, and the

Table 2: Numerical results of exact methods on random geometric graphs

$\|V\|$	weight scheme	ep	ILP-1 result	ILP-1 gap	ILP-2 result	ILP-2 gap	ILP-3 result	ILP-3 gap	NEW-1 result	NEW-1 gap	NEW-1 time	NEW-2 result	NEW-2 gap	NEW-2 time
100	NG	0.14	**3261.1**	0	**3261.1**	0	**3261.1**	0	**3261.1**	0	<0.1	**3261.1**	0	<0.1
		0.24	2942.9	21.7	2917.5	15.6	2884.5	3	**2882.5**	0	0.6	**2882.5**	0	0.4
		0.34	2878.5	26.5	2841.8	8.6	2876.7	17.6	**2828**	0	1.3	**2828**	0	1.5
	VG	0.14	**5731.8**	0	**5731.8**	0	**5731.8**	0	**5731.8**	0	<0.1	**5731.8**	0	<0.1
		0.24	**1981.8**	0	**1981.8**	0	**1981.8**	0	**1981.8**	0	<0.1	**1981.8**	0	<0.1
		0.34	**940.5**	0.3	**940.5**	0	**940.5**	0	**940.5**	0	0.2	**940.5**	0	<0.1
	EG	0.14	**19179.4**	0	**19179.4**	0	**19179.4**	0	**19179.4**	0	<0.1	**19179.4**	0	<0.1
		0.24	22519.7	20.5	22325.5	16.5	**22065.9**	8.2	**22065.9**	0	0.6	**22065.9**	0	0.4
		0.34	24295.9	31.1	**23717.6**	7	24009.2	19	**23717.6**	0	1.4	**23717.6**	0	1.7
500	NG	0.14	14942.7	56	15016.4	54.9	15457.2	85.1	13726.8	15.7	TLE	**13565**	12.7	TLE
		0.24	19028.3	72.7	15668.7	66.5	16788.4	95.5	13375.1	27.8	TLE	**13352.3**	28.3	TLE
		0.34	18602.1	75.8	15468.3	67.8	18948.1	97.6	13379.3	29.6	TLE	**13253.1**	29.5	TLE
	VG	0.14	**4377**	0.4	**4377**	0	4381.6	1.4	**4377**	0	4.9	**4377**	0	0.6
		0.24	2619.3	7.6	2596.4	5.8	2665.1	39	**2573.7**	0	49.3	**2573.7**	0	5.9
		0.34	3933.7	51.2	2205.3	17	2305.3	63.8	**2181.6**	0	294	**2181.6**	0	12.4
	EG	0.14	128902.8	66.2	128784.8	65.7	129521.2	91.8	116047.1	21.1	TLE	**115232**	19.7	TLE
		0.24	175979.6	76.6	147009	71.5	162967.5	98.3	124047.4	31.1	TLE	**122879.2**	31.5	TLE
		0.34	180632.5	78.4	149305.9	71.2	188131.4	99	**125499.8**	31.2	TLE	126997.4	33	TLE
1000	NG	0.14	38289.8	73.1	33123.7	68.5	38904.3	95.8	27649.2	30.7	TLE	**27066.6**	29.6	TLE
		0.24	61533.9	86.9	**33744.7**	79.7	38610.1	98.3	33758.1	48	TLE	35032.5	50	TLE
		0.34	121566.1	100	48329.9	95.9	38549.2	98.9	50289.4	100	TLE	**34725.5**	49.6	TLE
	VG	0.14	6071.9	11.6	6614.5	14.1	6144.7	43.9	**5830.7**	0	335.9	**5830.7**	0	21.7
		0.24	11493	64.6	12485.5	92.3	4528.1	74.7	4356.6	10	TLE	**4279.4**	0	669.7
		0.34	17662.5	100	9742.4	100	6214.7	88.7	4211.3	20.8	TLE	**3975.1**	2	2085.2
	EG	0.14	362985.2	78.8	311510.5	75.5	350946.2	98.4	**252929.5**	35.5	TLE	255236.5	36.6	TLE
		0.24	347041.7	88.6	**326670.5**	83.2	375956.1	99.4	375018.2	62.4	TLE	362021.8	56.7	TLE
		0.34	1232171.4	99.7	476998.4	97.3	370791.1	99.6	498328.3	100	TLE	**360711.3**	56.2	TLE

performance of algorithms with a difference lower than CD is considered statistically equivalent – denoting no significant difference. This equivalence is illustrated in the graphic by black horizontal bars connecting the respective algorithms.

Analyzing the CD plots the following conclusions are obtained:

- Metaheuristic approaches have the lowest average rank, indicating higher quality solutions, with LSRR particularly standing out for achieving the lowest average rank. However, when considering all instances together, there is no statistical difference among the results delivered by the three metaheuristic approaches (LSRR, CMSA-PBIG and PBIG) and New-2 ILP model. As for the NEW-1, only LSRR is statistically significantly better. Among the greedy heuristics, GREEDY-NEW has the lowest average rank and is significantly better than GREEDY-2, but there is no statistical difference between GREEDY-NEW and GREEDY-1.
- Among exact methods, both NEW-1 and NEW-2 demonstrate statistically significant superiority over the three competing approaches. Notably, there is no statistical difference between NEW-1 and NEW-2. Additionally, there is no statistical difference among ILP-1, ILP-2, and ILP-3.
- In the realm of greedy heuristics, GREEDY-NEW significantly outperforms both GREEDY-1 and GREEDY-2. While GREEDY-2 achieves a lower average rank than GREEDY-1, the distinction lacks statistical significance.

6. Conclusions and future work

This study delved into the minimum weighted independent dominating set problem, presenting two distinct integer linear programming models (NEW-1 and NEW-2) alongside a novel greedy heuristic (GREEDY-

Table 3: Numerical results of heuristic methods on random graphs

$\|V\|$	weight scheme	ep	GREEDY-1 result	GREEDY-1 time	GREEDY-2 result	GREEDY-2 time	GREEDY-NEW result	GREEDY-NEW time	PBIG result	PBIG time	CMSA-PBIG result	CMSA-PBIG time	LSRR result	LSRR time
100	NG	0.05	3589.1	<0.1	3519.1	<0.1	3340.2	<0.1	**3049.8**	0.54	**3049.8**	8	**3049.8**	<0.1
		0.15	3014.4	<0.1	2981.3	<0.1	2768.3	<0.1	2330.9	28.31	2338.1	48.3	**2230.2**[*]	0.15
		0.25	2883.5	<0.1	2796.1	<0.1	2577.4	<0.1	2070.9	0.16	2093.9	54.1	**2069.3**	0.66
	VG	0.05	10465.6	<0.1	11756.6	<0.1	10785.4	<0.1	7747	76.47	7860	59.4	**7715.4**	0.13
		0.15	4891.6	<0.1	5845.4	<0.1	4941.6	<0.1	3050.3	16.9	3070.3	40.6	**3046.6**	0.41
		0.25	3297.5	<0.1	3488.9	<0.1	2962.6	<0.1	**1808.4**	3.5	**1808.4**	15.5	**1808.4**	0.79
	EG	0.05	25698.7	<0.1	22269.3	<0.1	19976.7	<0.1	**14378.7**	0.78	**14378.7**	37.9	**14378.7**	<0.1
		0.15	27528.4	<0.1	23404.5	<0.1	20973	<0.1	14687.8	0.19	**14563.3**	17.2	**14563.3**	<0.1
		0.25	25451.4	<0.1	21770	<0.1	21302.8	<0.1	14506.6	<0.1	**14382.2**	37.7	**14382.2**	<0.1
500	NG	0.05	14143.1	<0.1	13535.1	<0.1	12272	0.4	10327.3	127.37	10140.6	667.3	**10041.5**	619.14
		0.15	12268.5	<0.1	11558	<0.1	10793.8	0.4	8297.5	47.33	8046.2	688.2	**7989**	869.88
		0.25	11630.3	<0.1	10429.5	0.1	10038.2	0.5	7633.7	10.16	7443	538.6	**7374.3**	841.53
	VG	0.05	15501.5	<0.1	18298.1	<0.1	14745.2	0.3	9822.6	924.21	9588.2	912.1	**8421.1**	848.96
		0.15	6496.3	<0.1	7300.1	<0.1	6021.8	0.4	3581.7	219.52	3557.8	599.8	**3497**	815.76
		0.25	4212.4	<0.1	4463.7	0.1	4112.9	0.6	2590.6	52.82	2586.5	521.1	**2572.5**	725.55
	EG	0.05	125357.6	<0.1	108178	<0.1	100325.8	0.4	70028.6	1008.89	67528.9	713.6	**65441.3**	599.88
		0.15	114951	<0.1	102365.1	<0.1	90468.8	0.6	64673.8	109.71	62950.1	798.1	**61456.5**	930.79
		0.25	111012.3	<0.1	99018.2	0.1	90066.8	0.5	64112.7	33.26	61411.1	294.9	**60637.4**	769.72
1000	NG	0.05	25569.6	<0.1	23489.7	0.1	22880	2.4	17723.7	1750.12	17819.4	1262.1	**17478**	1985.83
		0.15	20827.1	<0.1	20689.1	0.2	19301.5	2	14731.3	260.52	14461.9	813.3	**14340.7**	1762.62
		0.25	20858.8	<0.1	19280.5	0.4	18828.8	2.4	13968.5	340.14	13685.6	1480.6	**13397.9**	1688.65
	VG	0.05	18048.6	<0.1	20142.3	0.1	17464	1.4	11301.7	2018.01	11034.6	1363.5	**10392.2**	1690.35
		0.15	7408.3	<0.1	7987.4	0.2	6790.6	2.7	4540.3	695.71	4456.4	1362.7	**4440.7**	1638.47
		0.25	4941.9	<0.1	5566.6	0.4	4802.3	2.4	3519	306.67	3460.4	1049.3	**3454**	1610.84
	EG	0.05	238600	<0.1	202992	0.1	192329.5	1.7	133667.8	2121.82	130889.2	1786.2	**127796**	1906.56
		0.15	209709.3	<0.1	182726.6	0.2	176920.4	2	123760.9	99.67	120997.2	1478.6	**119250.3**	1781.76
		0.25	198537	<0.1	181150	0.4	170618.2	2.4	124193.3	447.95	120288.7	1359.1	**114124.2**	1793.35

[*] It seems likely that there was a typographical error in [21], considering that the average of optimal solutions found by the new ILP models is larger but shares similar digits (2330.2).

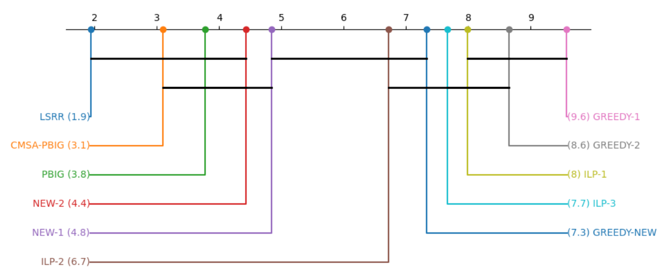
NEW). Extensive computational experiments were conducted to assess the efficacy of the proposed methods. The introduced ILP models, NEW-1 and NEW-2, attained optimal solutions across all instances with 100 nodes and outperformed competing ILP models on the majority of other cases. While metaheuristic approaches remain preferable for larger instances, the proposed ILP models offer a compelling alternative for specific instance classes, particularly node-oriented random geometric graphs. Notably, within the lower density graphs of this type, NEW-2 demonstrated significantly reduced execution times compared to metaheuristics.

The novel greedy algorithm GREEDY-NEW outperformed the competing greedy heuristics from the litreature across the majority of instances. Notably, it also demonstrated superior performance compared to some of the competing ILP models, particularly on random graphs with larger number of nodes.

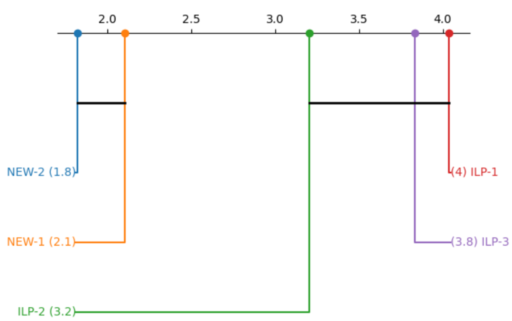
This study has demonstrated the high potential of the proposed ILP models and/or greedy heuristics to be integrated as core components within the design of new hybrid algorithms or to improve the performance of existing metaheuristics. In light of these findings, this represents a promising direction for future research.

Table 4: Numerical results of heuristic methods on random geometric graphs

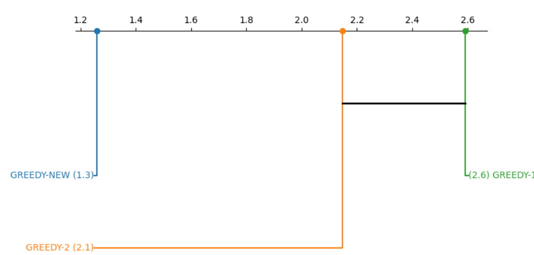
$\|V\|$	weight scheme	ep	GREEDY-1 result	GREEDY-1 time	GREEDY-2 result	GREEDY-2 time	GREEDY-NEW result	GREEDY-NEW time	PBIG result	PBIG time	CMSA-PBIG result	CMSA-PBIG time	LSRR result	LSRR time
100	NG	0.14	3870.6	<0.1	3646.4	<0.1	3506.2	<0.1	**3261.1**	11.9	**3261.1**	3.3	**3261.1**	0.1
		0.24	3798.8	<0.1	3378.1	<0.1	3248.7	<0.1	**2882.5**	3	**2882.5**	2.7	**2882.5**	<0.1
		0.34	3766.6	<0.1	3388.1	<0.1	3175.6	<0.1	**2828**	0.7	**2828**	27.9	**2828**	<0.1
	VG	0.14	7364.9	<0.1	7514.3	<0.1	7471.2	<0.1	**5731.8**	12.4	5740	2	**5731.8**	<0.1
		0.24	2880.1	<0.1	2724.2	<0.1	2808.3	<0.1	**1981.8**	<0.1	**1981.8**	2.2	**1981.8**	<0.1
		0.34	1741	<0.1	1832.3	<0.1	1718.6	<0.1	968.8	<0.1	**940.5**	1.2	**940.5**	<0.1
	EG	0.14	29011.6	<0.1	24998.3	<0.1	21463.6	<0.1	19313.2	117.6	**19179.4**	10.1	**19179.4**	<0.1
		0.24	35312.1	<0.1	28647.1	<0.1	26572.3	<0.1	22108.3	6.4	**22065.9**	5.5	**22065.9**	<0.1
		0.34	37929.4	<0.1	30503.4	<0.1	27020.8	<0.1	23900	1.1	**23717.6**	76.7	**23717.6**	<0.1
500	NG	0.14	18408.3	<0.1	16208.4	<0.1	15574.6	0.2	13341.5	835.4	13301.2	548.6	**13269.4**	551.1
		0.24	18548.8	<0.1	15882.9	<0.1	15613.7	0.2	12943.1	195.7	12783.3	129.6	**12764**	180.1
		0.34	18311.4	<0.1	15497.8	0.1	16497.8	0.3	13065.5	6.2	12954.8	327.5	**12887.5**	86.1
	VG	0.14	6807.8	<0.1	7087.8	<0.1	6974.8	0.2	4400.9	513.7	4389.7	165.8	**4377**	554.1
		0.24	3526.6	<0.1	3121.1	<0.1	3553.5	0.2	**2573.7**	9.7	**2573.7**	17.8	**2573.7**	3.6
		0.34	2632.2	<0.1	2830.7	0.1	2438	0.3	**2181.6**	7.2	**2181.6**	13.1	**2181.6**	2
	EG	0.14	177816.7	<0.1	148267	<0.1	136384.9	0.2	112404.8	731	111638.2	680.5	**110988.7**	190.3
		0.24	181739.2	<0.1	149980.2	<0.1	148916.9	0.2	118765.6	16.3	117439.8	405.9	**117020.2**	23.5
		0.34	190996.4	<0.1	155128.6	0.1	159464	0.3	122977.3	5.2	120883.7	346.4	**120381.3**	31.4
1000	NG	0.14	36214.6	<0.1	32393.4	0.1	31882	0.6	25892.9	776.7	25719	1695.2	**25563.3**	1792.2
		0.24	36750.4	<0.1	31462.5	0.2	31698.7	0.9	25570.6	524.8	25138.9	1081.1	**25111.7**	1388.1
		0.34	36913.2	<0.1	30752.1	0.3	33603.5	1.2	25714.2	17.2	25345	1698	**25114.3**	704.2
	VG	0.14	8552.3	<0.1	8613.7	0.1	8193.4	0.7	5869	1778.5	5890.4	950.4	**5830.7**	1244.3
		0.24	4977.4	<0.1	5146.2	0.2	4901.6	0.9	4281.2	109.1	4283.6	90.1	**4279.4**	177.8
		0.34	4656.5	<0.1	4693.1	0.4	4417.7	1.2	3974.5	63.4	3974.3	264.2	**3970.8**	44.2
	EG	0.14	358550.4	<0.1	305034.7	0.1	289269.1	0.7	236226.6	2582.6	233127.8	1711.2	**230030.4**	1327
		0.24	357735.8	<0.1	295099.4	0.2	307039.9	0.9	239560.7	451.8	238364.5	1277.3	**236439.8**	907
		0.34	369051.9	<0.1	303712.9	0.3	323414.4	1.2	244685.7	24.3	246171.8	1195.2	**243178.8**	793.7



(a) All approaches



(b) Exact methods



(c) Greedy heuristics

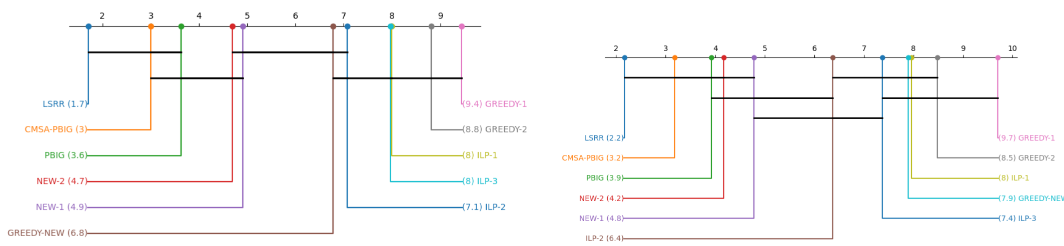
Fig. 1: Critical difference plots for all instances of the two benchmark sets

References

1. Barr, J.R., Abu-Khzam, F.N., Shaw, P.: Feature selection via independent domination. In: 2023 Fifth International Conference on Transdisciplinary AI (TransAI). pp. 197–200 (2023)
2. Blum, C., Pinacho, P., López-Ibáñez, M., Lozano, J.A.: Construct, merge, solve adapt a new general algorithm for combinatorial optimization. Computers Operations Research 68, 75–88 (2016), https://www.sciencedirect.com/science/article/pii/S0305054815002452
3. Bouamama, S., Blum, C., Fages, J.G.: An algorithm based on ant colony optimization for the minimum connected dominating set problem. Applied Soft Computing 80, 672–686 (2019), https://www.sciencedirect.com/science/article/pii/S1568494619302200
4. Chalupa, D.: An order-based algorithm for minimum dominating set with application in graph mining. Information Sciences 426, 101–116 (Feb 2018), http://dx.doi.org/10.1016/j.ins.2017.10.033
5. Chang, S.C., Liu, J.J., Wang, Y.L.: The weighted independent domination problem in series-parallel graphs. Frontiers in Artificial Intelligence and Applications 274, 77–84 (01 2015)
6. Chatterjee, M., Das, S.K., Turgut, D.: Wca: A weighted clustering algorithm for mobile ad hoc networks. Cluster computing 5, 193–204 (2002)
7. Clark, B.N., Colbourn, C.J., Johnson, D.S.: Unit disk graphs. Discrete mathematics 86(1-3), 165–177 (1990)
8. Daliri Khomami, M.M., Rezvanian, A., Bagherpour, N., Meybodi, M.R.: Minimum positive influence dominating set and its application in influence maximization: a learning automata approach. Applied Intelligence 48(3), 570–593 (Jul 2017), http://dx.doi.org/10.1007/s10489-017-0987-z
9. Erdös, P., Rényi, A.: On random graphs i. Publicationes Mathematicae Debrecen 6, 290–297 (1959)
10. Fernandess, Y., Malkhi, D.: K-clustering in wireless ad hoc networks. In: Proceedings of the second ACM international workshop on Principles of mobile computing. pp. 31–37 (2002)
11. Fischetti, M., Ljubić, I., Sinnl, M.: Redesigning benders decomposition for large-scale facility location. Management Science 63(7), 2146–2162 (Jul 2017), http://dx.doi.org/10.1287/mnsc.2016.2461
12. Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the american statistical association 32(200), 675–701 (1937)
13. Hu, S., Liu, H., Wang, Y., Li, R., Yin, M., Yang, N.: Towards efficient local search for the minimum total dominating set problem. Applied Intelligence 51(12), 8753–8767 (Apr 2021), http://dx.doi.org/10.1007/s10489-021-02305-6
14. Álvarez Miranda, E., Sinnl, M.: Exact and heuristic algorithms for the weighted total domination problem. Computers Operations Research 127, 105157 (2021), https://www.sciencedirect.com/science/article/pii/S0305054820302744
15. Nacher, J.C., Akutsu, T.: Minimum dominating set-based methods for analyzing biological networks. Methods 102, 57–63 (2016), https://www.sciencedirect.com/science/article/pii/S1046202315300967, pan-omics analysis of biological data
16. Nakkala, M.R., Singh, A., Rossi, A.: Multi-start iterated local search, exact and matheuristic approaches for minimum capacitated dominating set problem. Applied Soft Computing 108, 107437 (2021), https://www.sciencedirect.com/science/article/pii/S1568494621003604
17. Nemenyi, P.B.: Distribution-free multiple comparisons. Princeton University (1963)
18. Pinacho Davidson, P., Blum, C., Lozano, J.A.: The weighted independent domination problem: Integer linear programming models and metaheuristic approaches. European Journal of Operational Research 265(3), 860–871 (2018), https://www.sciencedirect.com/science/article/pii/S0377221717307750
19. Shu, L., Ma, T., Latecki, L.J.: Stable feature selection with minimal independent dominating sets. In: Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics. p. 450–457. BCB'13, Association for Computing Machinery, New York, NY, USA (2013), https://doi.org/10.1145/2506583.2506600
20. Wang, Y., Li, R., Zhou, Y., Yin, M.: A path cost-based grasp for minimum independent dominating set problem. Neural Computing and Applications 28(S1), 143–151 (Apr 2016), http://dx.doi.org/10.1007/s00521-016-2324-6
21. Wang, Y., Pan, S., Li, C., Yin, M.: A local search algorithm with reinforcement learning based repair procedure for minimum weight independent dominating set. Information Sciences 512, 533–548 (2020), https://www.sciencedirect.com/science/article/pii/S0020025519309107
22. Yan, X.F., Liu, A.Q., Yang, T.: A virtual backbone network algorithm based on a minimal independent dominating set for manets 35, 1134–1138 (06 2007)
23. Yu, J., Wang, N., Wang, G., Yu, D.: Connected dominating sets in wireless ad hoc and sensor networks – a comprehensive survey. Computer Communications 36(2), 121–134 (2013), https://www.sciencedirect.com/science/article/pii/S014036641200374X

A. Additional statistical analysis

- When examining the group of random graphs, LSRR emerges as the only method statistically superior to both proposed ILP models, NEW-1 and NEW-2. As before, there is no statistical difference among the three metaheuristic approaches. While GREEDY-NEW exhibits a lower average rank compared to all ILP-1, ILP-2, GREEDY-1, and GREEDY-2, the disparity is not statistically significant.
- In the case of random geometric graphs, there exists no statistical difference between the introduced ILP models (NEW-1 and NEW-2) and the metaheuristics (LSRR, CMSA-PBIG and PBIG), despite the metaheuristic approaches achieving a lower average rank. GREEDY-NEW once again obtains lower average rank compared to the competing GREEDY-1 and GREEDY-2, yet the difference is not significant.



(a) Random graph instances

(b) Random geometric graph instances

Fig. 2: Critical difference plots for two groups of instances