PACE Solver Description: LZJ

Zhaojie Liu
Huazhong University of Science and Technology, Wuhan, China

Qingyun Zhang
Huazhong University of Science and Technology, Wuhan, China

Zhouxing Su
Huazhong University of Science and Technology, Wuhan, China

Zhipeng Lü
Huazhong University of Science and Technology, Wuhan, China

Abstract

We propose a heuristic solver for the dominating set problem in the PACE 2025 Challenge. Our approach employs three primary reduction rules to simplify the problem. An initial solution is generated through an iterative greedy algorithm. Subsequently, a weighted local search strategy is applied to optimize the solution. Additionally, we design compact data structures to enhance the effectiveness of the local search.

2012 ACM Subject Classification Mathematics of computing → Combinatorial optimization; Computing methodologies → Search methodologies

Keywords and phrases PACE 2025, Dominating Set, Heuristic Optimization, Weighted 19 Local Search

Digital Object Identifier 10.4230/LIPIcs...XX

Supplementary Material *Source Code*: `https://github.com/G2LiuZhaoJie/pace2025-DS`

1 Challenge Problem and Transformation

The PACE 2025 Challenge focuses on the dominating set problem, a fundamental NP-hard problem. A dominating set for a graph G is a set $D \subseteq V(G)$ such that every vertex or one of its neighbors is contained in D, that is, for all $u \in V(G)$, we have $u \in D$ or there is $v \in D$ with $uv \in E(G)$. The objective is to find the smallest such set D, i.e., the minimum dominating set.

2 Solver Methodology

2.1 Reduction Rules

We iteratively apply three classical reduction rules to simplify the dominating set problem while preserving optimality. For large-scale special instances, we employ a specialized pruning method.

- **Rule 1 (Single Dominator):** If an undominated vertex v has only one dominator u, we include u in the mandatory selection.
- **Rule 2 (Subset Coverage):** If there exist two distinct undetermined vertices u and v such that the coverage of u is a subset of $N[v]$, we include u in the mandatory exclusion.
- **Rule 3 (Dominated Neighbor Subset):** If there exist two distinct undominated vertices u and v, where the set of all undominated vertices dominated by u is a subset of the neighbors of v, then the domination status of v can be ignored, but v itself may still be selected as a dominating vertex in the solution.

- **Specialized Pruning Method:** If a vertex v has exactly two neighbors and there exists an edge between these two neighbors, implying they can dominate each other, then v is included in the mandatory exclusion. If both neighbors of v are mandatorily excluded, then v is included in the mandatory selection.

2.2 Vertex Importance-Based Greedy Initialization

We generate an initial solution using an iterative greedy algorithm based on the concept of "vertex importance." For each vertex in the graph, we aim to quantify its domination capability through a numerical metric. Inspired by PageRank and the concept of "propagation" in neural networks, we define the importance of a vertex to reflect its ability to dominate others. Specifically, the importance score of a vertex v_i, denoted $\mathrm{IS}(v_i)$, is calculated as the sum of $\frac{1}{\deg(v_j)}$ over all undominated neighbors v_j of v_i, where $\deg(v_j)$ is the degree of v_j. Formally:

$$\mathrm{IS}(v_i) = \sum_{v_j \in N(v_i) \cap U} \frac{1}{\deg(v_j)}$$

where $N(v_i)$ is the set of neighbors of v_i, and U is the set of currently undominated vertices. The greedy algorithm iteratively selects the vertex with the highest importance score, updates the set of dominated vertices, and recalculates scores for affected vertices. To accelerate computation and updates, we employ a heap data structure and a lazy update strategy, enabling efficient incremental updates of the importance scores for the remaining vertices.

2.3 Adaptive Weighted Local Search

After obtaining a feasible dominating set, we perform a local search to iteratively seek smaller dominating sets by transforming the optimization problem into a series of decision problems. Specifically, upon finding a dominating set of size k, we attempt to identify a dominating set of size $k-1$. To this end, we design two distinct weighting strategies tailored to the characteristics of the reduced graph. The primary criterion for selecting a strategy is the number of undetermined vertices remaining after reduction. If the number of undetermined vertices is large, we employ a dual-objective reduction strategy that simultaneously explores regions for dominating sets of sizes $k-1$ and $k-2$. Conversely, if the number of undetermined vertices is small, we adopt a more intensive search strategy focused exclusively on the $k-1$ region.

2.3.1 Dual-Objective Iterative Weighted Search

This strategy comprises two main phases: the removal phase and the insertion phase.

Removal Phase:

- After finding a feasible solution in the current iteration, randomly remove one dominating vertex.
- Sample a subset of currently selected dominating vertices and remove the vertex with the minimum weight.
- If the current number of dominating vertices equals the size of the historical best solution minus 2, randomly remove an additional dominating vertex.

This process ensures that, at the end of the removal phase, the number of dominating vertices is at most the size of the best-known solution minus 3.

Insertion Phase:

- Insert the vertex with the maximum domination weight.
- With a certain probability, repeat the insertion of the vertex with the maximum domination weight.

This process ensures that, at the end of the insertion phase, the number of dominating vertices is restored to either the size of the best-known solution minus 1 or minus 2.

To accelerate the identification of vertices with maximum and minimum weights for insertion and removal, we design a dynamically maintained heap to facilitate efficient updates.

2.3.2 Single-Objective Iterative Weighted Search

This strategy is more aggressive and is designed for scenarios with few undetermined vertices. It determines moves by simultaneously evaluating the global weight cost of inserting one vertex and removing another. Since the state must be restored after evaluating a move, the computational cost is higher, but the results are more reliable.

Vertex Pair Evaluation:

- Select a neighbor of a currently undominated vertex to insert into the dominating set, ensuring that at least the undominated vertex becomes dominated.
- Select a vertex from the current dominating set to remove. Due to the reduction in the number of undetermined vertices after Section 2.1, the current dominating set is small.

Tabu Strategy: In each iteration, when selecting a vertex to remove, we prohibit choosing the vertex that was inserted in the previous iteration, implementing a single-step tabu mechanism.

References

1 Zhouxing Su, Qingyun Zhang, Zhipeng Lü, Chu-Min Li, Weibo Lin, and Fuda Ma. Weighting-based variable neighborhood search for optimal camera placement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12400–12408, 2021.

2 Ziliang Xiong and Mingyu Xiao. Exactly solving minimum dominating set and its generalization. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 7056–7064, 2024.

3 Qingyun Zhang, Zhipeng Lü, Zhouxing Su, and Chumin Li. A vertex weighting-based double-tabu search algorithm for the classical p-center problem. *Computers & Operations Research*, 160:106373, 2023.