Learning Pairwise Cooperation For Large-scale Vertex Subset Selection: Dominating Set Problem as Example

Zhengzhe Fu[1], Haihui Cheng[2], Danlu Shao[2], Li Ma[2], Yang Yang[1] (✉), Shiliang Pu[3], Qi Zhu[4], and Weihao Jiang[3]

[1] Zhejiang University, Hangzhou, China {fuzz,yangya}@zju.edu.cn
[2] Huayun Information Technology Co., Ltd, Hangzhou, China michael_chh@live.com, 601448191@qq.com, lima19@163.com
[3] Hikvision Research Institute, Hangzhou, China {pushiliang.hri, jiangweihao5}@hikvision.com
[4] Amazon, Seattle, WA, USA qi.Zhu.ckc@gmail.com

Abstract. The Dominating Set Problem (DSP) is a classic NP-hard vertex subset selection problem with broad significance across many real-world domains. In such applications, the value of a solution—whether measured by cost minimization or benefit maximization—is often closely tied to the scale of the underlying graph. Recent advances have introduced learning-based heuristic solvers that adapt dynamically by learning from previously solved instances. Despite their success, only a limited number of learning-based approaches explicitly address large-scale instances. A key bottleneck lies in the reliance on iterative predictions to construct a single solution, which, although beneficial for solution quality, becomes prohibitively expensive as graph size grows. A common alternative is to generate a heatmap over nodes and decode a solution from it; however, this strategy often sacrifices flexibility, as the heatmap can not be changed when the solution state changes. In this paper, we address these challenges by modeling cooperation dependency between pairs of nodes with the help of Graph Neural Network (GNN). We define cooperation dependency as the normalized frequency with which two nodes co-occur in high-quality solutions. By predicting this pairwise score, we enable rapid and flexible modification of node-level heatmaps without requiring repeated full-graph inference. This property allows our method to effectively enhance various heuristic strategies while maintaining high scalability. We evaluate our approach on datasets ranging from small graphs to large-scale instances with up to 10 million nodes. Experimental results demonstrate that our method achieves strong effectiveness and efficiency across diverse graph sizes and domains, highlighting its robustness and scalability.

Keywords: Dominating Set Problem · Large-scale · Heuristic.

1 Introduction

Large-scale combinatorial optimization problems play a crucial role in real-world applications, where solutions aim to minimize costs or maximize benefits. Among these

problems, the dominating set problem (DSP) serves as a representative example. Given a graph, a dominating set is defined as a subset of nodes such that every node in the graph either belongs to the subset or is adjacent to at least one node in it. The task of identifying a dominating set with minimum cardinality is known as the dominating set problem.

DSP has been widely studied due to its extensive applications across multiple domains, including optimal server placement in wireless networks [3,5,4,11], identification of influential individuals in social networks [15,14,35,36], analysis of critical nodes in aviation networks [23], and query selection optimization in information retrieval systems [40,32,9].

The practical value of DSP in these scenarios is closely linked to graph scale. For instance, in social networks, a company promoting a new service seeks to identify a small subset of users who can effectively influence the entire network. These users form a dominating set whose size and coverage directly reflect the potential customer base and, consequently, the business value. Unfortunately, DSP is among the most challenging combinatorial optimization problems, with computational complexity that grows exponentially with graph size. For large-scale graphs, obtaining optimal solutions is often infeasible. Exact algorithms developed in earlier studies exhibit exponential time complexity, such as $O(1.4969^n)$ [34] and $O(1.4689^n)$ [21].

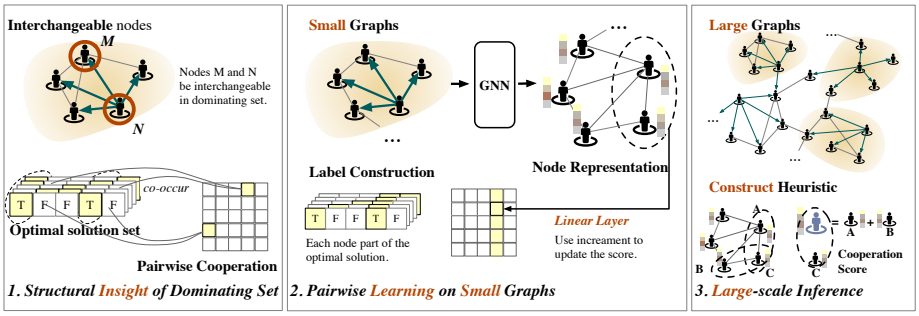

Fig. 1: Overview of our pairwise learning framework for the dominating set problem. We first identify structural insights of node replaceability and pairwise cooperation. A GNN-based network is trained on small graphs using constructed labels, and a Linear layer help to scale inference to large graphs using increment to update the score.

In recent years, learning-based heuristics have gained increasing attention in combinatorial optimization, motivated by the observation that frequently encountered problem instances often share structural patterns [7,13]. Unlike traditional handcrafted heuristics, learning-based methods can adapt dynamically by leveraging experience from previously solved instances. However, despite their success on small and medium-sized graphs, these approaches often struggle to scale to large graphs. The main challenges can be summarized as follows.

First, many learning-based heuristics rely on iterative prediction to construct high-quality solutions. For large-scale instances, repeatedly performing inference for a single solution becomes computationally prohibitive. A single-shot prediction, while efficient,

often fails to capture evolving preferences that arise when node states change during solution construction. Consequently, state-of-the-art methods [25,13,38] typically adopt iterative strategies that modify node features or graph structures after each decision. Although effective, even minor updates require recomputation over the entire graph, severely limiting scalability.

A common attempt to mitigate this issue is to generate a node-level heatmap that represents selection probabilities or priorities, and then decode a solution from this heatmap [24,29,33]. While this approach avoids explicit iteration, it significantly reduces flexibility: once the heatmap is fixed, adapting to changes in the partial solution often requires regenerating the entire heatmap, negating the intended efficiency gains.

Second, learning-based methods often suffer from weak supervision signals. In DSP and related vertex subset selection problems, multiple optimal or near-optimal solutions may exist for a single instance. Assigning labels based on a single solution can introduce bias and ambiguity, leading to unreliable node-level scores [25,20]. For example, as illustrated in Fig. 1, nodes M and N may be interchangeable within a dominating set. Training a model to favor one over the other forces it to artificially distinguish between structurally similar nodes, resulting in unstable predictions.

To address these limitations, we explicitly model pairwise cooperation, which we term *cooperation dependency* with the help of GNN, as the core learning objective. Cooperation dependency captures how frequently pairs of nodes co-occur in high-quality solutions, providing a richer and more stable supervision signal than single-node labels. Its pairwise nature enables efficient identification of subsequent nodes based on local relationships, thereby eliminating the need for costly iterative full-graph predictions. Moreover, after predicting cooperation dependency scores, we can rapidly and flexibly adjust node-level heatmaps to reflect changes in the solution state, achieving both efficiency and adaptability.

Although extracting pairwise supervision and training such models impose additional constraints—particularly for large graphs—these challenges can be effectively managed using smaller training instances and multiple constructive heuristic algorithms. Overall, our framework replaces inefficient iterative prediction steps with fast heatmap updates driven by cooperation dependency, substantially reducing computational overhead for large-scale learning-based heuristics.

The main contributions of this paper are summarized as follows:

- We present the first learning-based heuristic framework specifically designed for large-scale graphs to address vertex subset selection problems such as the dominating set problem.
- We propose explicitly modeling cooperation dependency between pairs of nodes, enabling effective utilization of diverse optimal solutions while avoiding costly iterative predictions.
- Our learning-based algorithms demonstrate strong generalization across graphs of varying sizes and diverse domain distributions. We implement our model in PyTorch[5].

[5] https://github.com/1250976476/Cooperation_DominatingSetProblem

2 Related Work

2.1 Algorithm for dominating set problem

The dominating set problem is recognized as one of the most challenging problems in combinatorial optimization. In the extensive history of its research, pioneers [34] has demonstrated that their algorithms run with a time complexity of $O(1.4969^n)$, while [21] has achieved $O(1.4689^n)$. Several state-of-the-art (SOTA) algorithms have been developed for the dominating set problem. For example, [8] proposes an Iterated Greedy algorithm, while [27] and [28] present an Ant Colony Optimization Algorithm combined with Local Search. In recent years, notable approaches have also been proposed by [9,39,6,22,41,43], further advancing the field.

Based on existing research [12,17,19], unless **P** $\approx$ **NP**, the approximation ratio of the simple greedy remains optimal in polynomial-time algorithms. In [30], they investigated the basic greedy algorithm along with several variations of this heuristic. One of the basic goals of this work is to provide a better solution than the greedy method in a similar time or guide the greedy method.

2.2 Learning-based algorithm for combinatorial optimization

As it is a recognized fact that real-world combinatorial optimization problems often share commonalities and similarities [7], recent learning-based solvers have achieved promising performance. As observed in [13,25,38,37,16], these methods typically adopt a strategy of iteratively predicting solutions through modified graphs and multiple inferences. Unfortunately, this iterative approach often results in a substantial burden of repetitive computations. Existed attempt to mitigate this issue is to generate a node-level heatmap that represents selection probabilities or priorities, and then decode a solution from this heatmap [24,29,33], which, however, ruin the flexibility. For the dominating set problem, the most recent learning-based approach is GFlowNet [42] and [10]; however, their experimentation with graph sizes remains limited. In recent years, [26] has attempted to leverage a novel probabilistic greedy mechanism and Q-learning to streamline calculations for budget-constrained problems. However, their framework fails when the budget constraint is absent. In light of this, our work introduces a effective learning-based heuristic framework to alleviate the computational demands for large-scale instances with explicit pairwise cooperation modeling.

3 Preliminaries

Consider a graph $G = (V, E)$ consisting of a node set V and an edge set $E \subseteq V \times V$, the *dominating set* and *minimum dominating set* is defined as follows.

Definition 1 (Dominating Set). *A dominating set of a graph $G = (V, E)$ is the a node subset $D \subseteq V$ where every node not in D is adjacent to at least one node in D.*

Definition 2 (Minimum Dominating Set). *The minimum dominating set of a graph is a dominating set with minimum size.*

To further demonstrate the concept of minimum dominating set, we could illustrate the problem as a boolean satisfiability problem. Given a graph G and a dominating set D, we introduce the notation $X_D = \{x_1, ..., x_{|V|}\}$ where each x_v represents a boolean value in $\{0, 1\}$, serving to indicate whether a particular node within the graph is in the dominating set D. Consequently, the arrangement of X_D must adhere to the fundamental dominating set constraint. This constraint dictates that for every node in the graph, it either belongs to the dominating set D, or it is connected to a node within D. In other words, for each node $v \in V$ and its neighbors $N(v) = \{u \in V | (u, v) \in E\}$, there must exist a node within the set D, that is:

$$\forall v \in V, x_v \vee \bigvee\nolimits_{u \in N(v)} x_u \tag{1}$$

Subsequently, to ascertain whether a given set adheres to the dominating set property ϕ, we can employ a Conjunctive Normal Form (CNF) for representation.

$$\phi = \bigwedge\nolimits_{v \in V} x_v \vee \bigvee\nolimits_{u \in N(v)} x_u \tag{2}$$

By exploring the entire solution space, we define the set of all minimum dominating sets as $\mathbf{S}_{md} = \{D \in \mathbf{S}_d \mid |D| = \min_{D' \in \mathbf{S}_d} |D'|\}$, where $\mathbf{S}_d$ represents the set of all valid dominating sets in the graph, and the size of a set is given by $|D| = \sum_{x_i \in X_D} x_i$. From $\mathbf{S}_{md}$, we derive an informative label, termed *Cooperation Dependency*, to assess the role of each node in the dominating set problem.

Intuitively, selecting a node to be part of a dominating set immediately covers its neighbors, thereby decreasing the likelihood that these immediate neighbors must be included to achieve a minimal set size. Conversely, the inclusion probabilities of distant nodes remain largely unaffected. This structural observation provides the theoretical motivation for our definition of cooperation dependency. Furthermore, this concept captures the dynamic nature of the optimization process, where a node's selection preference should adaptively respond to the changing states of other nodes.

Definition 3 (Cooperation Dependency). *Cooperation Dependency of dominating set problem between a pair of nodes (u, v) in a network is the normalized frequency that these two nodes both get involved in a minimum dominating set. Given the set of all minimum dominating set* $\mathbf{S}_{md} = \{D : |D| = \min_{D' \in \mathbf{S}_d}(|D'|)\}$*, the cooperation dependency between node u and v is:*

$\mathbf{C}_{u,v} = \frac{\sum_{D' \in \mathbf{S}_{md}} \omega_{D'}(x_u, x_v)}{\sqrt{\sum_k \sum_{D' \in \mathbf{S}_{md}} \omega_{D'}(x_k, x_v)} \sqrt{\sum_k \sum_{D' \in \mathbf{S}_{md}} \omega_{D'}(x_u, x_k)}}$ *where the* $\omega_{D'}(x_u, x_v) = 1$ *when* $x_u = 1$ *and* $x_j = v$ *in dominating set* D'.

In cooperation dependency, we use symmetric normalization to avoid some frequent observed nodes causing mistakenly distinguishing relation between pairwise nodes. Moreover, by definition, cooperation dependency is highly specific to the given optimization problem, yet remains generalizable across vertex subset selection problems that admit multiple optimal solutions. The true cooperation probability is defined over a uniform distribution on the set of all minimum dominating sets (MDS).

Table 1: Notations

Notation	Description
G	graph $G = (V, E)$, node set V and edge set $E \subseteq V \times V$
D	a dominating set
X_D	the boolean value representation of dominating set D
x_v	boolean value denoting whether a node is part of dominating set
$\mathbf{C}_{u,v}$	the weight of the cooperation dependency graph $\mathbf{C}$ between node u and v
$P_{u,v}$	the predicted cooperation dependency of node u verses v

4 Methodology

4.1 Pipeline

In this section, we provide a comprehensive description of our framework, illustrated in Fig 1. Before the training stage, labels are constructed in small-scale graphs. During training, our model establishes the connection between graph and cooperation dependency. After the inference on the entire graph, we can get nodes representation and pairwise cooperation function. A scalable construct heuristic is adapted and in this paper we mainly use two variant of greedy methods which select nodes to be added in a number of potential nodes. The effectiveness of these algorithms is guaranteed by nature of greedy methods but strongly influenced by predicted cooperation dependency.

Heuristic bound Based on some existing research [12,17,19], unless $\mathbf{P} \approx \mathbf{NP}$, the greedy approximation ratio remains optimal in polynomial-time algorithms achieving $(\ln \Delta + 2)$. However, as demonstrated in [31], when the input feature of each node comprises solely the node degree, no GNN can achieve a $(\Delta + 1 - \epsilon)$-approximation for the minimum dominating set problem, where Δ represents the maximum degree of graphs. Even if the feature vector of a node includes both the degree and the color of a weak 2-coloring, the optimal approximation ratio of GNNs for the minimum dominating set problem is $(\frac{\Delta+1}{2})$. Consequently, our primary focus lies in optimizing effectiveness and efficiency within the target domains, while also ensuring robust transferability.

4.2 Label Calculation

Our model is trained in a supervised manner and therefore requires pairwise-level cooperation dependency as training labels. However, enumerating all minimum dominating sets is computationally infeasible, as it would require exhaustively exploring the entire solution space, which is prohibitive even for graphs with only a few thousand nodes.

To obtain a tractable approximation, we construct a solution set $\hat{\mathbf{S}}$ consisting of optimal or near-optimal dominating sets. Specifically, for each node in the graph, we solve the dominating set problem once using an exact solver (e.g., Branch-and-Bound) while forcing that node to be included in the solution. This procedure results in $|V|$ candidate solutions and is feasible for small-scale graphs.

As defined in the preliminaries, cooperation dependency measures the normalized frequency with which two nodes co-occur in an optimal solution. Based on the solution

set $\hat{\mathbf{S}}$, we compute this dependency in two steps. First, we calculate the unnormalized co-occurrence frequency matrix $\hat{\mathbf{A}}$, where each entry is given by

$$\hat{\mathbf{A}}_{i,j} = \frac{1}{|\hat{\mathbf{S}}|} \sum_{D' \in \hat{\mathbf{S}}} \omega_{D'}(x_i, x_j),$$

with $\omega_{D'}(x_i, x_j)$ indicating whether both nodes x_i and x_j appear in solution D'.

Next, to account for scale differences among nodes and ensure comparability across pairs, we apply symmetric normalization to $\hat{\mathbf{A}}$. The empirical cooperation dependency matrix is defined as

$$\hat{\mathbf{C}} = \hat{\mathbf{D}}^{-\frac{1}{2}} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-\frac{1}{2}},$$

where $\hat{\mathbf{D}}$ is the degree matrix of $\hat{\mathbf{A}}$ with diagonal entries $\hat{\mathbf{D}}_{i,i} = \sum_j \hat{\mathbf{A}}_{i,j}$.

To assess the quality of this approximation, we conduct validation experiments on small graphs ($|V| \leq 30$), where all optimal solutions can be exhaustively enumerated. The resulting empirical cooperation dependency exhibits strong agreement with the ground truth, achieving a Pearson correlation of 0.89 ± 0.03 and a Spearman correlation of 0.87 ± 0.04. These results indicate that the bias introduced by our constrained solution generation strategy is empirical minimal.

4.3 Feature

In our framework, input features are divided into two categories. The first category consists of basic structural features, namely node degrees. The second category is derived from graph reduction techniques commonly used in dominating set problems.

Graph Reduction Graph reduction is a fundamental technique in many heuristic algorithms for the dominating set problem [1,41,43]. These methods exploit local structural properties of the graph to determine whether certain nodes can be safely included in or excluded from the dominating set. Among the various reduction strategies proposed in the literature, we primarily adopt the approach introduced in [1], as it is one of the most mature and widely applied methods.

This reduction framework partitions nodes into four disjoint subsets: the fixed node set, the deleted node set, the branched node set, and the involved node set. For each node v, its neighbor set $N(v)$ is further divided into three subsets, denoted as $N_1(v)$, $N_2(v)$, and $N_3(v)$, based on their local connectivity patterns:

- $N_1(v)$ consists of neighbors of v that have at least one neighbor outside $N(v)$.
- $N_2(v)$ includes neighbors that are not in $N_1(v)$ but have at least one neighbor in $N_1(v)$.
- $N_3(v)$ contains neighbors that do not belong to $N_1(v) \cup N_2(v)$.

Based on this partitioning, if $N_3(v)$ is non-empty, node v can be placed into the fixed node set, indicating that v is guaranteed to appear in at least one optimal solution. Once v is fixed, all nodes in $N_2(v)$ and $N_3(v)$ can be assigned to the deleted node set, as they are already dominated by v and can therefore be removed from further consideration.

Next, nodes that are neither fixed nor deleted and belong to $N_2(v)$ are classified as involved nodes. These nodes have no connections to vertices outside the local subgraph

induced by v and its neighbors. Finally, nodes in $N_1(v)$ that are not included in the fixed, deleted, or involved sets are categorized as branched nodes. The roles of these nodes remain uncertain, as they may connect to other regions of the graph, and thus require further branching or exploration to determine their inclusion in the dominating set.

4.4 Training

We train a deep graph neural network with residual connections to effectively capture both local and global structural information in the graph.

Cooperation Dependency The training objective is designed to align the model's predictions with the cooperation dependency, which serves as the core supervisory signal in our framework. This dependency plays a critical role in the heuristic process, as it guides the selection of subsequent nodes to be added to a partial solution. By improving the model's ability to approximate cooperation dependency, we enhance its capacity to identify nodes that are most beneficial for solution construction.

To this end, we randomly sample pairs of nodes from the graph, with half of the pairs exhibiting non-zero cooperation dependency and the other half having zero dependency. This balanced sampling strategy mitigates class imbalance during training. For each sampled pair, we apply a binary cross-entropy loss to minimize the discrepancy between the model's predicted scores and the corresponding ground-truth labels.

$$\mathcal{L}_c = \sum_{(u,v)\in E} \mathbb{1}\{\mathbf{C}_{u,v} > 0\} P_{u,v} + \mathbb{1}\{\mathbf{C}_{u,v} = 0\}(1 - P_{u,v}) \tag{3}$$

4.5 Fast Inference

Directly predicting cooperation dependency for all node pairs incurs a prohibitive $O(n^2)$ computational cost. To address this issue, we further accelerate the selection of the next node by replacing explicit pairwise prediction with a lightweight linear scoring mechanism. Specifically, we employ a linear layer without an activation function to efficiently score candidate nodes using an aggregated representation of the current partial dominating set. We refer to this score as the *Cooperation Score*.

Formally, the pairwise cooperation score between nodes u and v is computed as

$$\log P_{u,v} = \textit{Linear}(\mathbf{u}, \mathbf{v}), \tag{4}$$

where $\mathbf{u}$ and $\mathbf{v}$ denote the learned representations of nodes u and v, respectively. For a partial dominating set D^-, the aggregated cooperation score of a candidate node u can be efficiently computed as

$$\sum_{v\in D^-} \log P_{u,v} = \textit{Linear}\left(\mathbf{u}, \sum_{v\in D^-} \mathbf{v}\right). \tag{5}$$

Since the representation of the partial set $\sum_{v\in D^-} \mathbf{v}$ can be updated incrementally as nodes are added, this formulation enables linear-time scoring with respect to the size of the partial solution, significantly reducing inference overhead.

4.6 Heuristic with Cooperation Dependency

As learning-based heuristics generally lack theoretical guarantees on approximation ratios, we design alternative strategies that preserve the strong approximation properties of classical greedy methods while incorporating richer structural information from cooperation dependency. Specifically, we propose a modified greedy heuristic that selects nodes from a dynamically maintained candidate set informed by cooperation scores, as well as a variant of the greedy_Rev algorithm [30]. These hybrid heuristics combine the robustness of traditional greedy selection with the global awareness provided by learned cooperation dependency.

The traditional greedy method always selects the node with the largest gain for the partial solution. However, its main limitation lies in its susceptibility to local optima due to its myopic decision-making process. In our construct heuristic, we define a candidate set size K based on the size of the graph. During each selection step, the K most beneficial nodes are added to the candidate set, and the cooperation dependency is used to determine the final node to be selected. Each node in the candidate set is scored based on the cooperation score $\sum_{v \in D^-} \log P_{u,v}$. Algorithm 1 is the pseudo code of our Greedy Construct Heuristic algorithm with Cooperation Dependency.

Algorithm 1 Greedy Construct Heuristic Algorithm with Cooperation Dependency

Input: Graph $G = (V, E)$, Cooperation Dependency Prediction P
Output: Dominating set D
  Fixed $\leftarrow$ *Graph_Reduction*();
  $D^- \leftarrow$ *Fixed*;
  for $v \in V$ **do**
    $score(v) = \sum_{u \in N(v)} \mathbb{1}\{\forall k \notin D^- | k \in N(u)\}$
    $+\mathbb{1}\{\forall u \notin D^- | u \in N(v)\}$;
  end for
  while G not dominated by D **do**
    Candidate Set $\leftarrow$ *TopK*(*score*);
    $v_0 \leftarrow$ node with largest cooperation score in *Candidate Set*;
    $D^- \leftarrow D^- \cup \{v_0\}$;
    update the *score*;
    Calculate the marginal gain for node v_i, and update its cooperation score: c_i $+=$ *Linear* $(\mathbf{u}, \mathbf{v_0})$ (Refer to Equation. 5)
  end while
  $D \leftarrow D^-$;
  return D;

The variant of the Greedy_Rev also needs the predicted cooperation dependency as score (heatmap). At first, every node will be in a set (a trivial dominating set); this algorithm will examine the nodes following a node sequence sorted by the cooperation score and try to delete them from the trivial dominating set as soon as possible. When removing a node from the dominating set would lead to a situation where no node in its neighborhood is part of the dominating set, such a node cannot be removed.

In contrast, if this condition is not met, the node is promptly eliminated from the dominating set. The pseudo code of these two construct heuristic algorithms is provided in the Appendix.

Table 2: Average Performance of Experiments in ER Random Graphs (smaller is better).

Size Range	information		Greedy		Cooperation(Greedy)		Cooperation_Iter	Greedy_Iter
	Size	Time(s)	Size	Time(s)	Size	Time(s)	Size	Size
50-100	58.70	17.95	18.80	0.08	**18.25**	0.20	**18.25**	18.70
1 000-1 000	1000.00	249.15	274.60	0.42	264.90	0.76	**259.80**	270.10
2 000-2 500	2228.60	547.60	604.13	1.18	589.33	1.24	**575.73**	591.40
10 000-50 000	28191.80	-	7691.40	13.00	7634.40	14.71	7389.20	7541.00
100 000-500 000	202407.00	-	55087.00	151.24	54919.75	107.18	53106.25	54011.25
1 000 000-5 000 000	2522474.3	-	686429.3	1117.19	685823.5	1456.51	660589.3	671993.3
10 000 000	10000000	-	2721476	4097.47	2720294	5851.69	2617302	2664054
Size Range	Random		Greedy_Rev		Cooperation(Greedy_Rev)		Greedy_Vote	
	Size	Time(s)	Size	Time(s)	Size	Time(s)	Size	Time(s)
50-100	22.90	0.08	19.30	0.09	**18.25**	0.18	18.65	0.01
1 000-1 000	283.40	0.44	289.50	0.43	260.80	0.42	267.65	0.12
2 000-2 500	907.26	1.46	641.13	1.21	592.07	0.42	595.53	0.29
10 000-50 000	11278.00	40.84	8145.60	13.05	**7326.80**	3.04	7564.00	4.14
100 000-500 000	55116.00	156.42	58464.75	148.79	**52518.25**	17.04	54360.75	40.76
1 000 000-5 000 000	686433.3	1160.50	728640.8	1103.45	**654287.5**	204.35	677581.3	3878.63
10 000 000	2721672	4286.09	2889054	4031.95	**2594060**	830.06	2685965	51645.48

Table 3: Average Performance of Experiments in BA Random Graphs (smaller is better).

Size Range	information		Greedy		Cooperation(Greedy)		Cooperation_Iter	Greedy_Iter
	Node	Optimal	Size	Time(s)	Size	Time(s)	Size	Size
50-100	77.65	11.20	11.80	0.07	13.00	0.21	12.75	**11.60**
1 000-1 000	1000.00	132.95	144.80	0.72	139.40	0.94	137.30	142.60
2 000-2 500	2222.55	287.22	316.11	1.49	306.06	1.48	**301.89**	310.33
10 000-50 000	27647.60	-	3851.40	20.71	3811.20	17.49	3747.40	3804.40
100 000-500 000	261013.75	-	36305.25	183.70	36193.00	178.39	35482.00	35899.75
1 000 000-5 000 000	2850365.00	-	397142.7	2099.82	396855.0	2025.61	387242.3	392363.7
10 000 000	10000000	-	1392966	6234.62	1392498	7448.83	1358219	1374431
Size Range	Random		Greedy_Rev		Cooperation(Greedy_Rev)		Greedy_Vote	
	Size	Time(s)	Size	Time(s)	Size	Time(s)	Size	Time(s)
50-100	13.20	0.08	12.30	0.07	11.65	0.17	11.85	0.01
1 000-1 000	150.90	0.76	150.20	0.66	**137.00**	0.43	141.45	0.15
2 000-2 500	321.83	1.68	329.50	1.38	314.28	0.35	310.94	0.29
10 000-50 000	3864.20	47.90	4034.80	18.74	**3707.60**	2.80	3783.40	3.82
100 000-500 000	36285.25	189.20	38137.50	159.83	**35137.00**	21.96	35788.50	48.38
1 000 000-5 000 000	397131.0	2100.02	416906.3	1747.96	**384797.0**	237.28	391253.7	2981.40
10 000 000	1393041	6367.40	1462114	5168.50	**1351641**	837.61	1372249	27003.31

5 Experiments

To conduct our methods in large-scale graphs, we must face the problem of training and inference with the size discrepancy. The dependency to the greedy heuristic unfolds a

Table 4: Solution Quality on Zero-shot Real-world Datasets.

dataset	Random	Random_Iter	Greedt_Rev	Greedy	Greedy_Iter	Cooperation_Iter	Cooperation (Greedy_Rev)	Greedy_Vote	S2V-DQN	GFlowNet
DBLP	4331	*4324*	4433	4335	4326	**4311**	4421	4351	17677	4696
Full Cora	3769	*3757*	3876	3773	3764	**3751**	3853	3776	19772	4112
Cora	637	636	655	636	*635*	**632**	675	643	2543	676
Photo	650	639	669	642	635	*633*	708	**627**	7354	1212
Physics	4441	*4337*	4649	4431	4348	**4290**	4569	4354	34472	-
CS	3018	*2967*	3149	3019	2981	**2951**	3109	3003	18305	3342
GitHub	4568	4564	4660	4567	4565	**4556**	4652	*4558*	36531	-
FacebookPagePage	3331	3296	3390	3323	3294	**3283**	3401	*3291*	22196	4552
Computers	1199	*1173*	1233	1188	1177	1178	1296	**1170**	13115	2701
ENZYMES	8.00	7.50	7.85	7.65	7.40	7.40	7.5	*7.35*	12.05	**7.05**
PROTEINS	16.80	16.00	16.95	16.25	15.95	**15.60**	16	15.85	27.25	*15.65*

Table 5: Performance on Large-scale Real-world Datasets.

dataset	information		Greedy		Greedy_Vote		Cooperation_Iter		Cooperation(Greedy_Rev)	
	#nodes	#edges	Size	Time(s)	Size	Time(s)	Size	Time(s)	Size	Time(s)
youtube	1.13M	2.99M	*213131*	818.88	213271	386.87	**213126**	969.98	271457	198.62
soc-pokec	1.63M	30.62M	221667	932.63	**210931**	664.56	*211618*	2951.84	220281	295.30
gowallah	196.5K	950.3K	*41745*	649.50	41826	28.90	**41669**	783.53	45306	35.44

Table 6: The graphon distances between ER random graphs and real-world datasets.

Datasets	**DBLP**	**Full Cora**	**Cora**	**Photo**	**Physics**	**CS**	**GitHub**	**FacebookPagePage**	**Computers**
Graphon Distance	1.77E-04	8.25E-05	6.89E-03	1.31E-04	6.86E-05	2.35E-05	7.72E-03	2.32E-03	8.71E-03
Ours/Greedy	4311/4335	3751/3773	632/636	633/642	4290/4431	2951/3019	4556/4567	3283/3323	1178/1188

possibility of dealing instances encountering domain discrepancy. This guess is examined by experiments training on random graphs, but inference on zero-shot real-world datasets. For random graph dataset, we generate Erdos-Renyi (ER) [18] and Barabasi-Albert (BA) [2] graphs which function widely in modeling many real-world networks. For the training phase, our dataset consists of random graphs containing exactly 1000 vertices. In our experiments, cooperation trained on ER graphs use 18 layer GCN and cooperation trained on BA graphs use 16 layer GCN. All GCN is with residual connection. In addition to the dominating set problem, we provide experimental results for the Minimum Vertex Cover (MVC) problem using the same training configuration.

Baseline We conduct comprehensive comparisons of our methods against two distinct categories of approaches. First, we evaluate them against traditional heuristic methods, including Greedy, Greedy_Rev and Greedy_Vote [30] all with graph reduction. Second, we assess their performance in comparison to a learning-based approach, exemplified by S2V-DQN [13], GFlowNet [42].

Variant methods Based on variations in training data and the choice of heuristics, we designate our variant methods as follows:

– **Cooperation(Greedy)**: Our main approach utilizes a variant of the greedy method as construct heuristic, scoring candidate nodes based on cooperation dependency.

– **Cooperation(Greedy_Rev)**: The fastest version of our framework. No graph reduction and only use degree as feature. Random sample a subset of nodes and use them

to calculate the cooperation dependency. Score all nodes with the sum of cooperation dependency and use the variant of the Greedy_Rev method as construct heuristic.

To further evaluate solution quality, we present the results from four iterations involving regret and reconstruction (methods end with “iter").

5.1 Random graph experiments

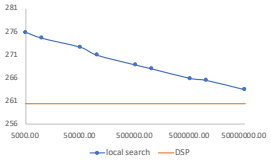

Fig. 2: The performance curve of local search with different number of iterations.

Addressing the dominating set problem within random graphs presents a non-trivial challenge. As shown in Figure 2, our methods can exhibit performance comparable to executing over 50 million iterations of some simple local search methods, even when tackling graphs with as few as 1,000 nodes.

In Table 2 and Table 3, our report encompasses the average size of the obtained dominating sets and the average running times across multiple instances within corresponding size ranges. Given the substantial computational demands posed by large graphs, it is often infeasible for algorithms like “branch and cut” to ascertain optimal solutions. Consequently, we do not provide optimal solutions for such instances. To clarify the settings in these tables, the ‘information’ column provides the foundational profile and characteristics of the datasets used in our evaluation. Meanwhile, the ‘random’ column refers to a specific strategy employed during the Greedy Construct phase; instead of strictly selecting the single optimal node, this baseline randomly samples a node from the top-k candidates.

Within the area of traditional algorithms, the Greedy_Vote heuristic outperform almost all the other methods. When considering learning-based methods, our simplified framework, Cooperation (Greedy), consistently outperforms all greedy methods in finding smaller solutions, except in tiny graphs. Furthermore, the Cooperation_Iter, when iterated a limited number of times, further enhances problem-solving capabilities. In terms of efficiency, all these methods, with the exception of Greedy_Vote and methods with iterations, demonstrate near-linear time complexity. This characteristic is also reflected in the actual running times observed during our experiments. In addition, the graph reduction phase accounts for the majority of the computational cost in these methods. Among the implementations, the iterative algorithms — Greedy_Iter and Cooperation with Iteration — exhibit similar magnitudes of time cost. The fastest version of our framework, Cooperation(Greedy_Rev), show excellent performance with smallest time cost in middle to large graphs, however, its performance is less guaranteed and easy affected by distribution changes.

5.2 Zero-shot real-world datasets experiments

To thoroughly assess the versatility of our framework, we extend our experimentation to encompass various commonly used real-world datasets with zero-shot setting. For consistency, we treat all edges as undirected and concentrate solely on the underlying structure of the graphs. All methods with cooperation are trained in ER random graphs.

Table 4 illustrates that our framework maintains competitive performance when applied to instances from various distributions. To enhance clarity, we have highlighted the **best results** in bold, while the *second-best results* are denoted in italics. To further validate our framework, experiments on large-scale public graph datasets are also presented in Table 5. While our methods may not offer a universal solution for all real-world datasets, it is noteworthy that Cooperation_Iter exhibits superior performance in various data domains, such as purchase networks and coauthor networks. This may be attributed to the fact that ER random graphs are general and informative. However the Cooperation(Greedy_Rev) excels in random graphs and faces challenges with distribution shift. These observations underscore the adaptability and potential of our approach, particularly when dealing with datasets that share structural similarities with the training random graph.

5.3 Transferability Analysis

We analyze the transferability of our method through the lens of graphon theory, which is widely used to characterize structural similarities between graph datasets and to assess the effectiveness of pre-trained models across domains. Specifically, we quantify dataset similarity by computing the Gromov–Wasserstein (GW) distances between the graphons of Erdős–Rényi (ER) random graphs and those of real-world datasets. The resulting distances provide a principled measure of how closely real-world graphs resemble the ER model. The outcomes are summarized in Table 6.

We compute the Pearson correlation coefficient between solution quality and the corresponding GW distances. The resulting value of 0.514 indicates a moderate positive correlation, suggesting that our method tends to achieve better performance on datasets that are structurally closer to ER random graphs, even without fine-tuning. Although the correlation is not strong, it reveals a clear trend: as the structural distance between datasets increases, the performance of our method gradually degrades. This degradation highlights both the strengths and limitations of transferability in learning-based approaches.

5.4 Generality on Vertex-Subset Selection Tasks

Table 7: Performance comparison on the Minimum Vertex Cover problem.

Methods	50–100	1000	2000–2500	10000–50000
Cooperation (Greedy_Rev)	28.8	266.8	1182.53	14930.2
Greedy_Rev	29.65	280.8	1227.13	15536.0
Greedy	29.75	279.2	1223.13	15469.6

To demonstrate the generality of the proposed framework beyond the dominating set problem, we further evaluate it on another classical vertex-subset selection task, namely the Minimum Vertex Cover problem (MVC). Importantly, this evaluation is conducted under the same training setup.

Table 7 reports the performance comparison among different methods on graphs of varying sizes. The proposed Cooperation-enhanced Greedy_Rev consistently outperforms the standard Greedy_Rev and Greedy baselines across all graph scales. This performance gain is particularly evident on larger graphs, indicating that incorporating cooperation dependency enables the heuristic to better capture global structural information even when transferred to a different combinatorial objective.

These results confirm that the core idea of modeling cooperation among selected vertices is not limited to a single task. Instead, it can be effectively generalized to other vertex-subset selection problems, highlighting the robustness and adaptability of our framework.

6 Conclusion

In this paper, we propose the concept of cooperation dependency and demonstrate how it can be leveraged to construct fast and scalable learning-based heuristics. By modeling pairwise relationships, our framework enables more effective utilization of multiple optimal solutions while avoiding the computational overhead caused by repeated inference under minor graph changes.

Extensive experiments on graphs containing up to 10 million nodes show that our approach remains efficient and effective at large scales. Moreover, evaluations on real-world datasets confirm that our framework consistently delivers competitive performance, demonstrating its practical applicability and robustness across diverse graph domains.

Acknowledge

This work was supported by National Natural Science Foundation of China (No. 62322606, No. 62441605).

References

1. Alber, J., Fellows, M.R., Niedermeier, R.: Polynomial-time data reduction for dominating set. J. ACM **51**(3), 363–384 (May 2004). https://doi.org/10.1145/990308.990309, https://doi.org/10.1145/990308.990309
2. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Rev. Mod. Phys. **74**, 47–97 (Jan 2002). https://doi.org/10.1103/RevModPhys.74.47, https://link.aps.org/doi/10.1103/RevModPhys.74.47
3. Aoun, B., Boutaba, R., Iraqi, Y., Kenward, G.: Gateway placement optimization in wireless mesh networks with qos constraints. Selected Areas in Communications, IEEE Journal on **24**, 2127 – 2136 (12 2006). https://doi.org/10.1109/JSAC.2006.881606

4. Bai, X., Zhao, D., Bai, S., Wang, Q., Li, W., Mu, D.: Minimum connected dominating sets in heterogeneous 3d wireless ad hoc networks. Ad Hoc Networks **97** (2020)
5. Boria, N., Murat, C., Paschos, V.T.: The probabilistic minimum dominating set problem. Discrete Applied Mathematics **234**, 93–113 (2018)
6. Cai, S., Hou, W., Wang, Y., Luo, C., Lin, Q.: Two-goal local search and inference rules for minimum dominating set. In: Bessiere, C. (ed.) Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20. pp. 1467–1473. International Joint Conferences on Artificial Intelligence Organization (7 2020). https://doi.org/10.24963/ijcai.2020/204, https://doi.org/10.24963/ijcai.2020/204, main track
7. Cappart, Q., Chételat, D., Khalil, E.B., Lodi, A., Morris, C., Veličković, P.: Combinatorial optimization and reasoning with graph neural networks. In: Zhou, Z.H. (ed.) Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21. pp. 4348–4355. International Joint Conferences on Artificial Intelligence Organization (8 2021). https://doi.org/10.24963/ijcai.2021/595, https://doi.org/10.24963/ijcai.2021/595, survey Track
8. Casado, A., Bermudo, S., López-Sánchez, A., Sánchez-Oro, J.: An iterated greedy algorithm for finding the minimum dominating set in graphs. Mathematics and Computers in Simulation **207**, 41–58 (2023). https://doi.org/https://doi.org/10.1016/j.matcom.2022.12.018, https://www.sciencedirect.com/science/article/pii/S0378475422005055
9. Chalupa, D.: An order-based algorithm for minimum dominating set with application in graph mining. Information Sciences **426**, 101–116 (feb 2018). https://doi.org/10.1016/j.ins.2017.10.033, https://doi.org/10.1016%2Fj.ins.2017.10.033
10. Chen, M., Liu, S., He, W.: Learn to solve dominating set problem with gnn and reinforcement learning. Applied Mathematics and Computation **474**, 128717 (2024). https://doi.org/https://doi.org/10.1016/j.amc.2024.128717, https://www.sciencedirect.com/science/article/pii/S0096300324001899
11. Chinnasamy, A., Sivakumar, B., Selvakumari, P., Suresh, A.: Minimum connected dominating set based rsu allocation for smartcloud vehicles in vanet. Cluster Computing **22**, 12795–12804 (2018)
12. Chlebík, M., Chlebíková, J.: Approximation hardness of dominating set problems in bounded degree graphs. Information and Computation **206**(11), 1264–1275 (nov 2008). https://doi.org/10.1016/j.ic.2008.07.003
13. Dai, H., Khalil, E.B., Zhang, Y., Dilkina, B., Song, L.: Learning combinatorial optimization algorithms over graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 6351–6361. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
14. 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**, 570–593 (2018)
15. Dinh, T.N., Shen, Y., Nguyen, D.T., Thai, M.T.: On the approximability of positive influence dominating set in social networks. Journal of Combinatorial Optimization **27**, 487–503 (2014)
16. Drakulic, D., Michel, S., Andreoli, J.M.: GOAL: A generalist combinatorial optimization agent learner. In: The Thirteenth International Conference on Learning Representations (2025), https://openreview.net/forum?id=z2z9suDRjw
17. Duh, R.c., Fürer, M.: Approximation of k-set cover by semi-local optimization. In: Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing. p. 256–264.

STOC '97, Association for Computing Machinery, New York, NY, USA (1997). https://doi.org/10.1145/258533.258599, https://doi.org/10.1145/258533.258599
18. Erdos, P., Renyi, A.: On the evolution of random graphs. Publ. Math. Inst. Hungary. Acad. Sci. **5**, 17–61 (1960)
19. Feige, U.: A threshold of ln n for approximating set cover. J. ACM **45**(4), 634–652 (jul 1998). https://doi.org/10.1145/285055.285059, https://doi.org/10.1145/285055.285059
20. Hudson, B., Li, Q., Malencia, M., Prorok, A.: Graph neural network guided local search for the traveling salesperson problem. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=ar92oEosBIg
21. Iwata, Y.: A faster algorithm for dominating set analyzed by the potential method. In: Marx, D., Rossmanith, P. (eds.) Parameterized and Exact Computation. pp. 41–54. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
22. Jiang, H., Zheng, Z.: An exact algorithm for the minimum dominating set problem. In: Elkind, E. (ed.) Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23. pp. 5604–5612. International Joint Conferences on Artificial Intelligence Organization (8 2023). https://doi.org/10.24963/ijcai.2023/622, https://doi.org/10.24963/ijcai.2023/622, main Track
23. Li, J., Wen, X., Wu, M., Liu, F., Li, S.: Identification of key nodes and vital edges in aviation network based on minimum connected dominating set. Physica A: Statistical Mechanics and its Applications **541**(C) (2020). https://doi.org/10.1016/j.physa.2019.1233, https://ideas.repec.org/a/eee/phsmap/v541y2020ics0378437119318692.html
24. Li, Y., Guo, J., Wang, R., Zha, H., Yan, J.: Fast t2t: Optimization consistency speeds up diffusion-based training-to-testing solving for combinatorial optimization. Advances in Neural Information Processing Systems **37**, 30179–30206 (2024)
25. Li, Z., Chen, Q., Koltun, V.: Combinatorial optimization with graph convolutional networks and guided tree search. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. p. 537–546. NIPS'18, Curran Associates Inc., Red Hook, NY, USA (2018)
26. Manchanda, S., MITTAL, A., Dhawan, A., Medya, S., Ranu, S., Singh, A.: Gcomb: Learning budget-constrained combinatorial algorithms over billion-sized graphs. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 20000–20011. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper_files/paper/2020/file/e7532dbeff7ef901f2e70daacb3f452d-Paper.pdf
27. Potluri, A., Singh, A.: Two hybrid meta-heuristic approaches for minimum dominating set problem. In: Panigrahi, B.K., Suganthan, P.N., Das, S., Satapathy, S.C. (eds.) Swarm, Evolutionary, and Memetic Computing. pp. 97–104. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
28. Potluri, A., Singh, A.: Hybrid metaheuristic algorithms for minimum weight dominating set. Applied Soft Computing **13**(1), 76–88 (2013). https://doi.org/https://doi.org/10.1016/j.asoc.2012.07.009, https://www.sciencedirect.com/science/article/pii/S1568494612003092
29. Qiu, R., Sun, Z., Yang, Y.: Dimes: A differentiable meta solver for combinatorial optimization problems. Advances in Neural Information Processing Systems **35**, 25531–25546 (2022)
30. Sanchis, L.A.: Experimental analysis of heuristic algorithms for the dominating set problem. Algorithmica **33**(1), 3–18 (2002). https://doi.org/10.1007/s00453-001-0101-z, https://doi.org/10.1007/s00453-001-0101-z

31. Sato, R., Yamada, M., Kashima, H.: Approximation ratios of graph neural networks for combinatorial problems. Curran Associates Inc., Red Hook, NY, USA (2019)
32. Shen, C., Li, T.: Multi-document summarization via the minimum dominating set. In: Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010). pp. 984–992. Coling 2010 Organizing Committee, Beijing, China (Aug 2010), https://aclanthology.org/C10-1111
33. Sun, Z., Yang, Y.: Difusco: Graph-based diffusion solvers for combinatorial optimization. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 3706–3731. Curran Associates, Inc. (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/0ba520d93c3df592c83a611961314c98-Paper-Conference.pdf
34. van Rooij, J.M., Bodlaender, H.L.: Exact algorithms for dominating set. Discrete Applied Mathematics **159**(17), 2147–2164 (2011). https://doi.org/https://doi.org/10.1016/j.dam.2011.07.001, https://www.sciencedirect.com/science/article/pii/S0166218X11002393
35. Wang, F., Du, H., Camacho, E., Xu, K., Lee, W., Shi, Y., Shan, S.: On positive influence dominating sets in social networks. Theoretical Computer Science **412**(3), 265–269 (2011)
36. Wang, G., Wang, H., Tao, X., Zhang, J.: Finding weighted positive influence dominating set to make impact to negatives: a study on online social networks in the new millennium. ICTs and the Millennium Development Goals: A United Nations Perspective pp. 67–80 (2014)
37. Wang, H.P., Li, P.: Unsupervised learning for combinatorial optimization needs meta learning. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=-ENYHCE8zBp
38. Wang, R., Hua, Z., Liu, G., Zhang, J., Yan, J., Qi, F., Yang, S., ZHOU, J., Yang, X.: A bi-level framework for learning to solve combinatorial optimization on graphs. In: Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems (2021), https://openreview.net/forum?id=nwWLJsTJfv
39. Wang, Y., Cai, S., Chen, J., Yin, M.: A fast local search algorithm for minimum weight dominating set problem on massive graphs. IJCAI International Joint Conference on Artificial Intelligence **2018-July**, 1514–1522 (2018). https://doi.org/10.24963/ijcai.2018/210
40. Wu, P., Wen, J.R., Liu, H., Ma, W.Y.: Query selection techniques for efficient crawling of structured web sources. In: 22nd International Conference on Data Engineering (ICDE'06). pp. 47–47. IEEE (2006)
41. Xiong, Z., Xiao, M.: Exactly solving minimum dominating set and its generalization. In: Larson, K. (ed.) Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24. pp. 7056–7064. International Joint Conferences on Artificial Intelligence Organization (8 2024). https://doi.org/10.24963/ijcai.2024/780, https://doi.org/10.24963/ijcai.2024/780, main Track
42. Zhang, D., Dai, H., Malkin, N., Courville, A., Bengio, Y., Pan, L.: Let the flows tell: Solving graph combinatorial problems with GFlownets. In: Thirty-seventh Conference on Neural Information Processing Systems (2023), https://openreview.net/forum?id=sTjW3JHs2V
43. Zhu, E., Zhang, Y., Wang, S., Strash, D., Liu, C.: A dual-mode local search algorithm for solving the minimum dominating set problem. Knowledge-Based Systems **298**, 111950 (2024). https://doi.org/https://doi.org/10.1016/j.knosys.2024.111950, https://www.sciencedirect.com/science/article/pii/S0950705124005847