Learning-Based Heuristic for Combinatorial Optimization of the Minimum Dominating Set Problem

Abi Kothapalli*[,1] Mudassir Shabbir[1,2] Xenofon Koutsoukos[1]

[1]Department of Computer Science, Vanderbilt University, [2]Department of Computer Science, Information Technology University, Lahore

VANDERBILT UNIVERSITY

Background and Introduction

- A **dominating set** of a graph is a subset of its vertices such that every vertex is either in the set or adjacent to a vertex in the set.
- A ***minimum* dominating set** is a dominating set of minimum cardinality. The task to find a minimum dominating set in a graph is called the **minimum dominating set (MDS) problem**.

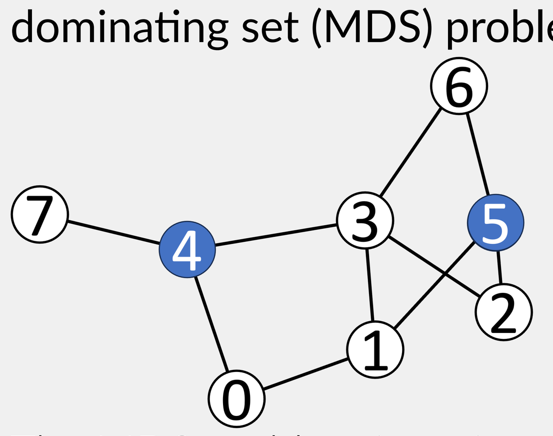


Figure 1. The set $\{4, 5\}$ comprises a dominating set since each vertex outside this set is adjacent to either vertex 4 or vertex 5. It is also a *minimum* dominating set since there is no dominating set with lower cardinality.

- The MDS problem is an important NP-hard combinatorial optimization problem [1] with a wide range of applications, including social networks [2], cybersecurity [3], and bioinformatics [4].
- There are no known algorithms that can solve the MDS problem efficiently. The purpose of this work is to propose a novel learning-based heuristic algorithm that can efficiently and accurately approximate solutions to the problem using **graph convolutional networks (GCNs)**.

Synthetic Data Generation

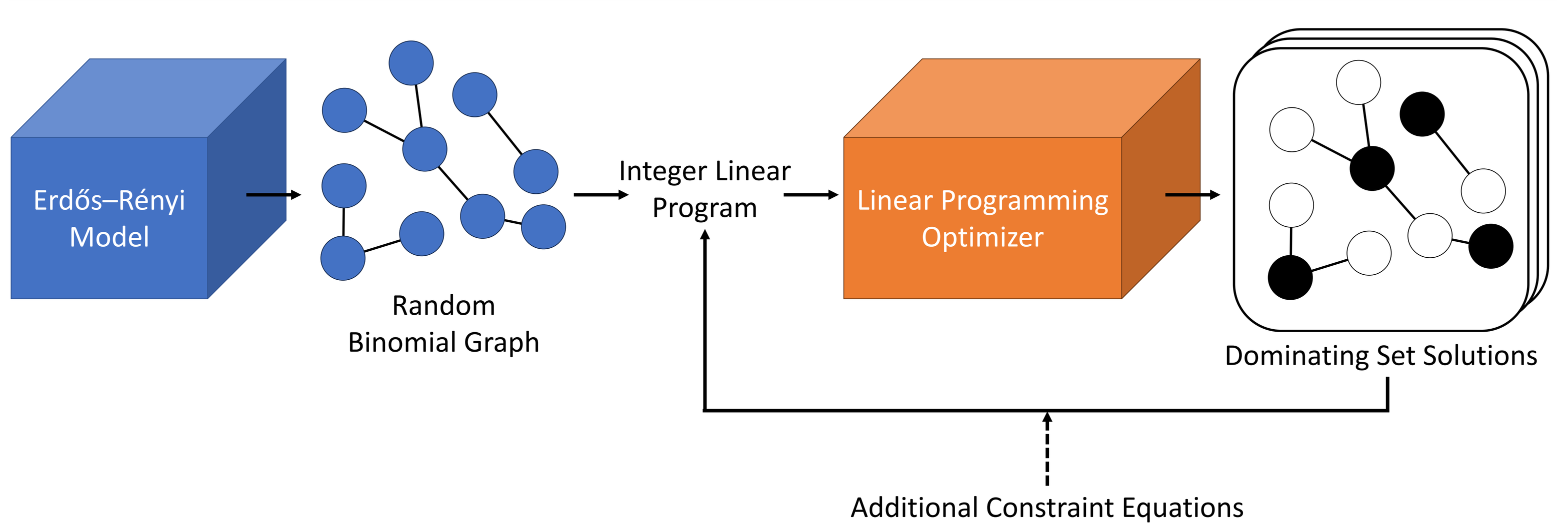


Figure 2. Synthetically generated dataset of over 1000 random binomial graphs with sizes ranging from 150-250 vertices and varying edge densities using the Erdős–Rényi model. Reduced MDS problem to integer linear program (ILP) to label each graph with solutions and iteratively increased complexity of each ILP to generate multiple unique solutions per graph.

GCN Design: Non-Uniqueness of MDS Solutions

- A major challenge to designing a machine learning model to predict solutions to the MDS problem is the non-uniqueness of solutions:

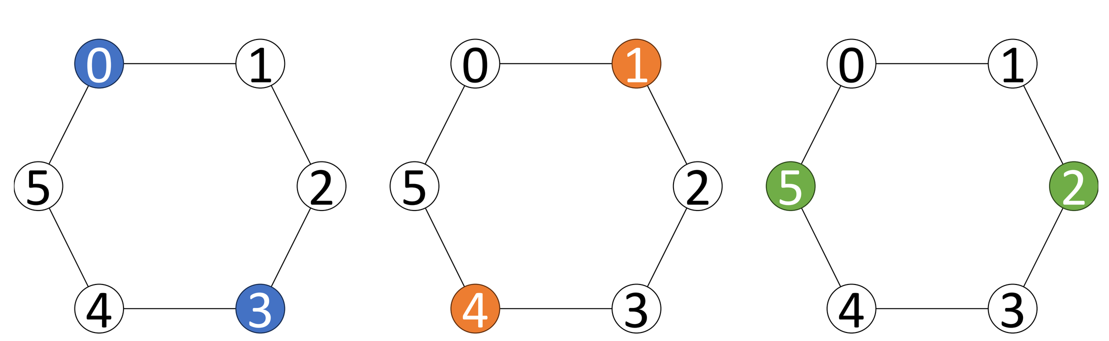


Figure 3. This figure illustrates the following three unique MDS solutions for the same graph: $\{0, 3\}$, $\{1, 4\}$, $\{2, 5\}$. A poorly designed model might produce predictions that assigns equal likelihood to each vertex being in a solution, which is not useful.

- Renders traditional single-output models ineffective for this task.

GCN Architecture + Training

- Designed GCN architecture that outputs m distinct probability maps that each encode likelihood of each vertex belonging to MDS solution.
- Allows the model to capture the diversity of MDS solutions on each graph. Chose $m = 32$ after hyperparameter tuning, as this was the point at which performance saturated.
- Trained the GCN model using approximately 80% of the generated dataset, leaving the remaining 20% as hold-out for evaluation.

Dominating Set Construction

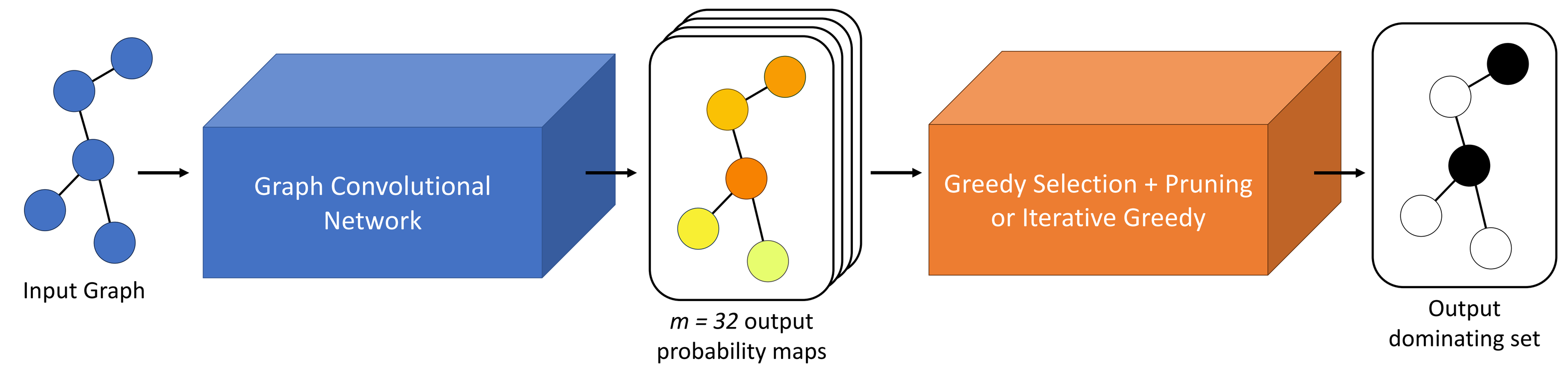


Figure 4. Computational pipeline to construct dominating sets using GCN.

- Used the m probability maps as m heuristic functions where the heuristic value of each vertex was the likelihood given by the probability map.
- Implemented two metaheuristic frameworks to build dominating sets from heuristics:
  - **Greedy selection (GS) scheme**: iteratively adds vertices with highest heuristic value to set until solution is reached, and then prunes out redundancies.
  - **Iterative greedy (IG) scheme**: constructs solution similar to GS and then iteratively destroys and reconstructs solution using one or more supplied heuristics.

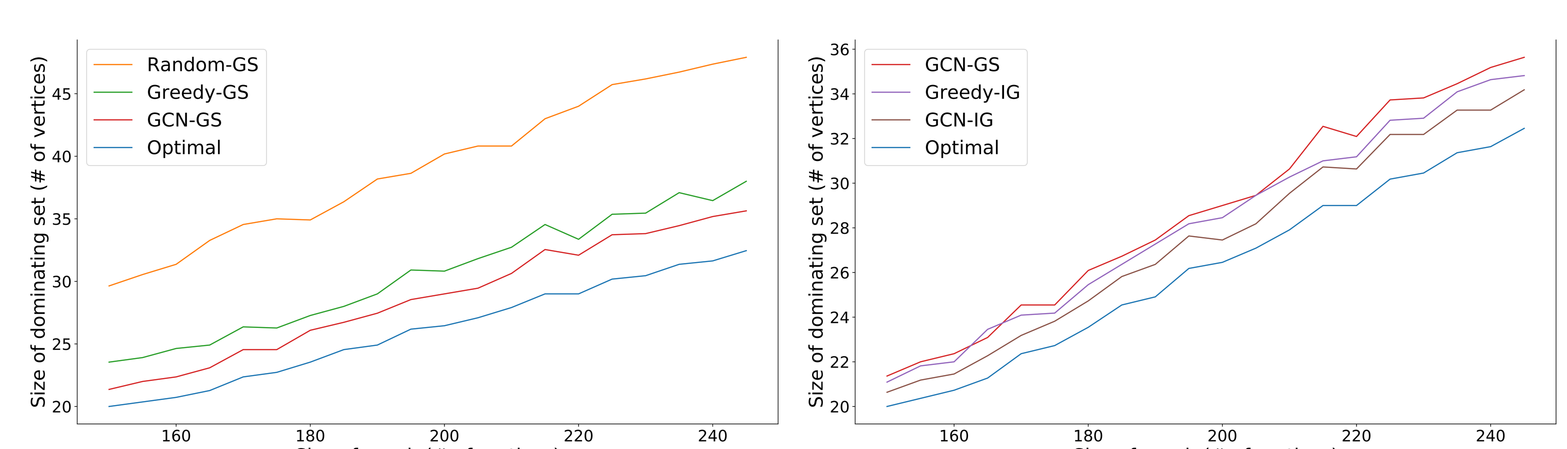
Experimental Evaluation: Hold-Out Synthetic Data

Our approaches:

- GCN-based heuristic using GS scheme (GCN-GS)
- GCN-based heuristic using IG scheme (GCN-IG)

Comparison approaches:

- Random (uniformed) heuristic in GS scheme (Random-GS)
- Traditional greedy heuristic in GS scheme (Greedy-GS)
- Traditional greedy heuristic in IG scheme (Greedy-IG)

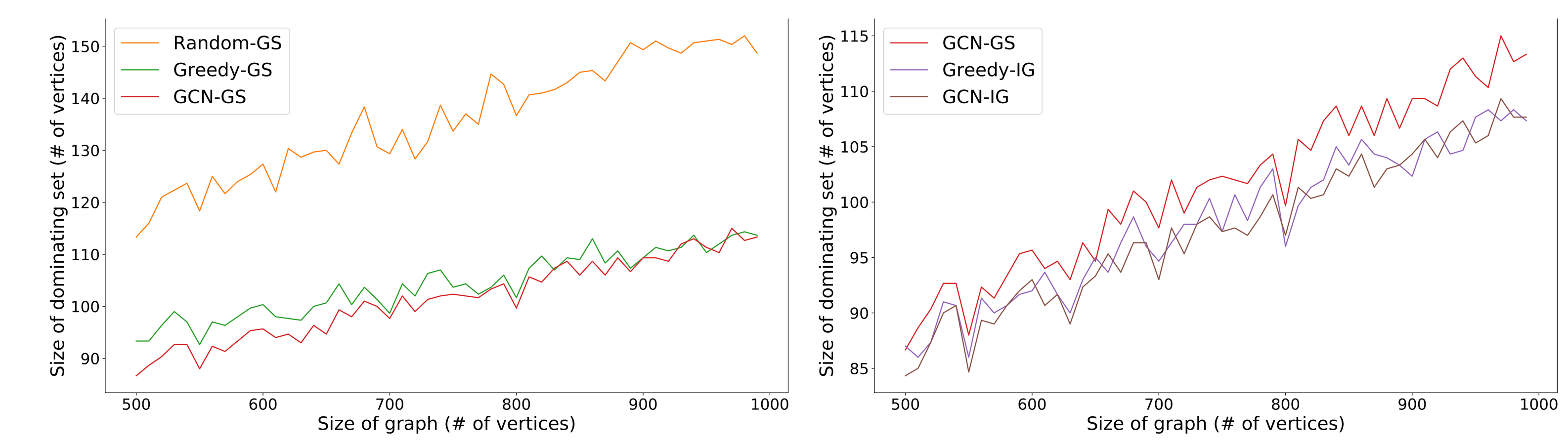


(a) Comparison of different heuristics with GS + Pruning scheme.

(b) Comparison of different heuristics with IG scheme. We reproduce the GCN-GS curve for comparison.

Figure 5. Comparison of all procedures on hold-out graphs from generated dataset. Optimal MDS sizes are displayed as ground truth. Note that the traditional greedy heuristic uses vertex degree as heuristic value. Furthermore, the Greedy-IG is the previous state-of-the-art heuristic algorithm for the MDS problem, while the Greedy-GS is the most commonly used.

Experimental Evaluation: Higher-Order Synthetic Graphs



(a) Comparison of different heuristics with GS + Pruning scheme.

(b) Comparison of different heuristics with IG scheme. We reproduce the GCN-GS curve for comparison.

Figure 6. Comparison of all procedures on higher-order (number of vertices) random binomial graphs. Optimal MDS sizes are unavailable in this setting as they are computationally intractable.

Experimental Evaluation: Real-World Graphs

Dataset	Greedy-GS	GCN-GS	Greedy-IG	GCN-IG	Optimal
BZR	13.13	**13.11**	**13.11**	**13.11**	13.11
dblp_ct1	8.33	**8.32**	**8.32**	**8.32**	8.32
DD	57.7	53.56	53.75	**52.48**	52.48
DHFR	13.91	**13.90**	**13.90**	**13.90**	13.90
facebook_ct2	19.99	**19.98**	**19.98**	**19.98**	19.98
FIRSTMM_DB	334.32	311.00	304.00	**302.10**	302.10
github_stargazers	17.43	17.49	**17.40**	**17.40**	17.40
MSRC_21	15.21	14.19	14.25	**13.80**	13.80
NCI1	10.21	9.80	9.81	**9.78**	9.78
OHSU	21.34	20.67	20.65	**20.51**	20.51
REDDIT-MULTI-5K	98.13	98.09	**97.94**	**97.94**	97.94

Table 1. Comparison of performance of testing procedures on real-world graph datasets (our approaches underlined). Reported figures are mean dominating set size, with best performing procedure(s) per dataset in bold. Mean optimal MDS size is given as ground truth.

Conclusions

- The GCN-based heuristic algorithms consistently outperform the equivalent comparison approaches, obtaining near-optimal performance.
- The GCN heuristics generalize to higher-order graphs than those on which it was trained. It is also able to generalize across datasets, despite being trained exclusively on synthetic data, underscoring the robustness of the proposed approaches.
- The GCN-IG algorithm outperforms the Greedy-IG algorithm, which was previously considered the state-of-the-art for this problem. Hence, our approach sets a new state-of-the-art in computing dominating sets.

References + Author Contributions

[1] T. Haynes, *Domination in graphs: volume 2: advanced topics*. Routledge, 2017.
[2] F. Wang, H. Du, E. Camacho, *et al.*, "On positive influence dominating sets in social networks.", 2011.
[3] D. Angel, "Application of graph domination to defend medical info networks against cyberthreats.", 2022.
[4] J. Nacher and T. Akutsu, "Minimum dominating set-based methods for analyzing biological networks.", 2016.

Author Contributions: AK generated data, implemented all methodology/software, performed all experiments, completed all data analysis/visualization, and designed poster. AK, MS conceptualized methodology. MS, XK provided supervision, funding, and resources for the project.

Contact: abi.kothapalli@vanderbilt.edu