COMPUTER & INFORMATION SCIENCES

VN-Solver: Vision-based Neural Solver for Combinatorial Optimization over Graphs

Dhana Lakshmi Kankanala, Mina Samizadeh, Guangmo Tong
Computational Data Science Lab, 220 Smith Hall.

Introduction

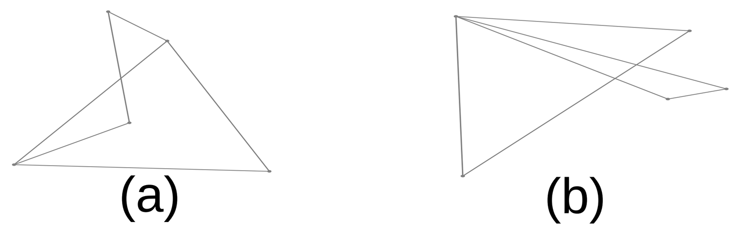
Data-driven approaches have been proven effective in solving combinatorial optimization problems over graphs such as the traveling salesman problems and the vehicle routing problem. The rationale behind such methods is that the input instances may follow distributions with salient patterns that can be leveraged to overcome the worst-case computational hardness. For optimization problems over graphs, the common practice of neural combinatorial solvers consumes the inputs in the form of adjacency matrices. In this paper, we explore a vision-based method that is conceptually novel: can neural models solve graph optimization problems by visualizing the graph pattern? Our results suggest that the performance of such vision-based methods is not only non-trivial but also comparable to the state-of-the-art matrix-based methods, which opens a new avenue for developing data-driven optimization solvers.

Motivation

- The state-of-the-art neural solvers are largely matrix-based, i.e., reasoning over the adjacency matrix such as and by using deep neural networks.

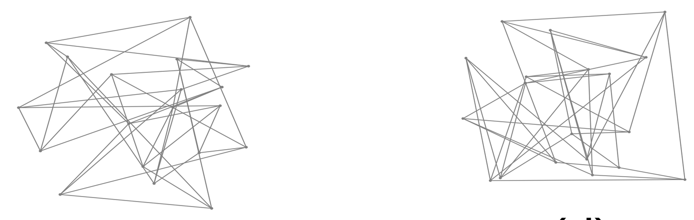
$$A_a = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad A_b = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

- For humans, visualizations can be much more intuitive than the adjacency matrix for certain instances.
- For example, Figures (a) and (b) associated with the and matrix adjacencies.
- By looking at the figures we can discern graph in Figure (a) is Hamiltonian and the one in Figure (b) is non-Hamiltonian, which is hard to understand by looking the matrix-adjacencies.



(a) (b)

- When the graph becomes larger, i.e., Figures (c) and (d) it is harder to decide about such property, i.e., Hamiltonicity, of the graph.

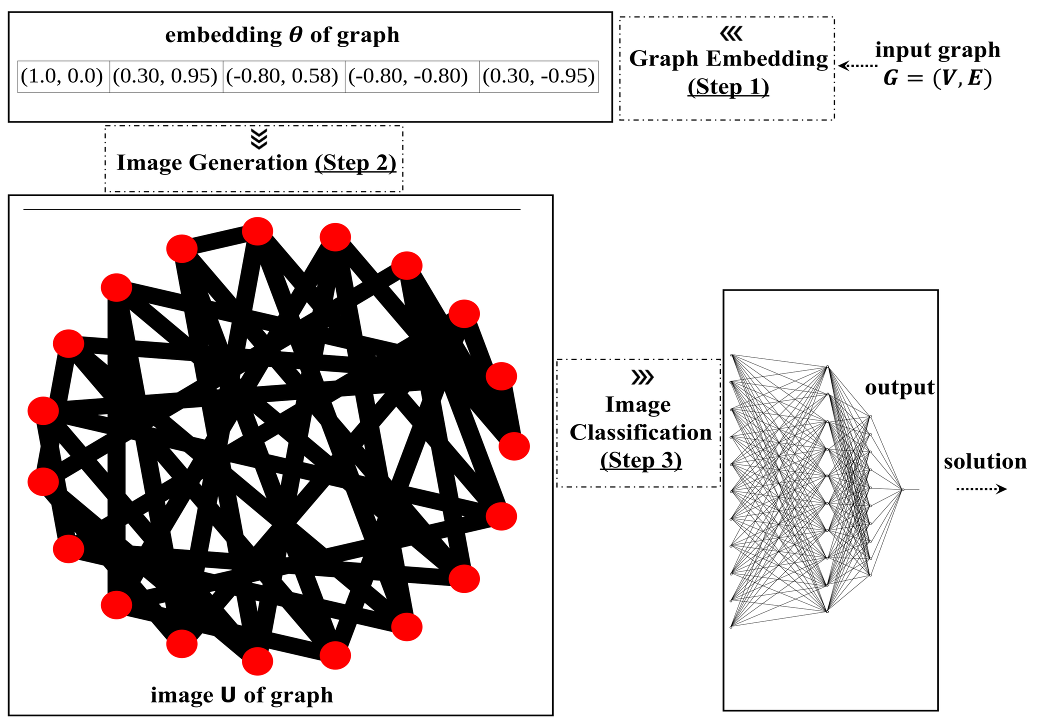


(c) (d)

- We utilize computer vision methods to solve such decision problems where we take Hamiltonian Cycle Problem as an example.

VN-Solver Framework

graph embedding: $F_{em} : V \rightarrow R^2$
image generation: $F_{graph} : R^{2\times|V|} \rightarrow R^{3\times dim_x}$
image classification: $F_{class} : R^{3\times dim_x \times dim_y} \rightarrow \{0,1\}$



Graph Embedding

Different graph embedding layouts produce different positions for the nodes in the 2D Euclidean space. For example, Figures are 2D embeddings of a same graph in Random, Ellipse, and Spiral layouts with different parameters.

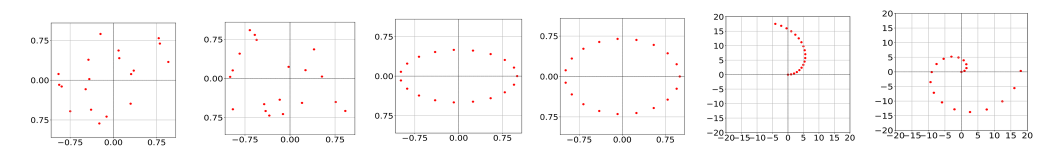
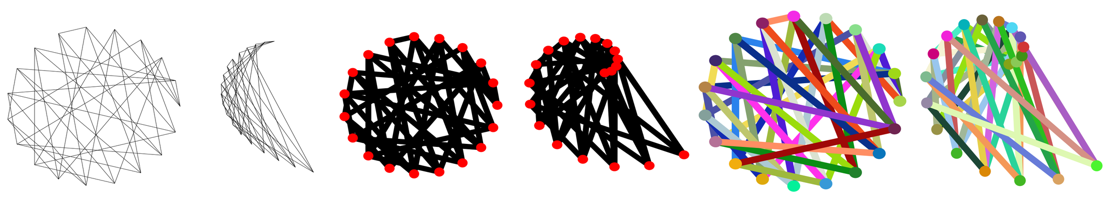


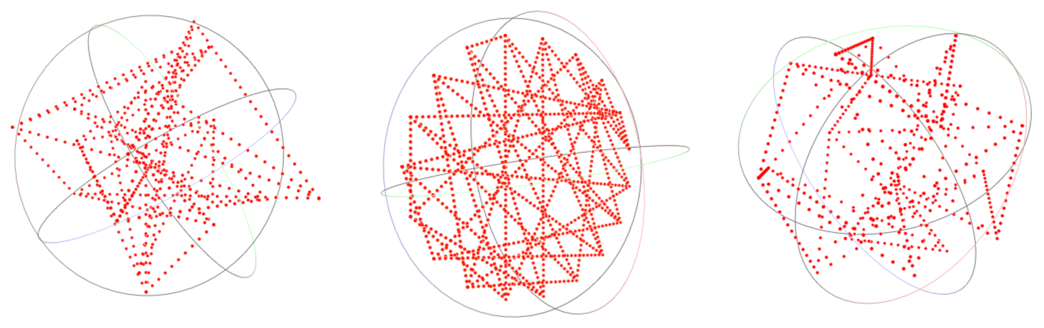
Image Generation

Different settings for Image generation produces different visualizations, For example, the following images are visualizations of a same graph in Circular and Spiral embedding with Gray, Uniform, and Random color Schemes.



PointCloud Visualization

Diverge from visualization in 2D space, we can adopt 3D representation of graphs for our solver. Particularly, we utilize Point Cloud. Figures illustrate Random, Sphere, and Spiral visualizations of a same graph in 3D space.



Experiments

Data Statistics

Small Dataset (4-20 nodes)		Large Dataset (21-50 nodes)	
Hamiltonian	Non-hamiltonian	Hamiltonian	Non-hamiltonian
2,277	1,838	7,453	5,739

- **Baselines:** Graphormer (Graph transformer) and Naïve Bayesian
- **VN-Solver**: ResNet50 is adopted for image classification step, fine-tuning it with Adam.
- Experiments have been done in three different settings with 100, 200, and 1000 number of samples where 20 percent used for validation.
- 500 number of samples used for testing.
- Experiments run 5 times with different seeds.
- The average and standard deviation on 5 experiments is reported.

Results

Dataset: Small			100 F1	200 F1	1000 F1
VN-Solver	Gray	Circular	0.62 ± 0.03	0.61 ± 0.02	**0.78** ± 0.08
		Spiral	0.63 ± 0.04	0.65 ± 0.06	0.76 ± 0.02
		Random	0.50 ± 0.28	0.61 ± 0.02	0.37 ± 0.34
	Uniform color	Circular	**0.63** ± 0.09	**0.69** ± 0.04	**0.83** ± 0.03
		Spiral	**0.65** ± 0.05	**0.72** ± 0.05	0.76 ± 0.07
		Random	0.62 ± 0.00	0.60 ± 0.00	0.65 ± 0.00
	Random color	Circular	0.61 ± 0.03	0.64 ± 0.04	**0.81** ± 0.02
		Spiral	**0.64** ± 0.04	**0.65** ± 0.03	0.74 ± 0.04
Graphormer			0.60 ± 0.14	0.64 ± 0.11	0.65 ± 0.18
Naive-Bayesian			0.54 ± 0.03	0.55 ± 0.02	0.55 ± 0.01
Dataset: Large			100 F1	200 F1	1000 F1
VN-Solver	Gray	Circular	0.62 ± 0.02	0.61 ± 0.02	0.92 ± 0.04
		Spiral	0.5 ± 0.28	0.72 ± 0.15	**0.94** ± 0.02
		Random	0.37 ± 0.34	0.26 ± 0.36	0.72 ± 0.03
	Uniform color	Circular	**0.74** ± 0.10	**0.90** ± 0.08	**0.94** ± 0.03
		Spiral	**0.75** ± 0.07	**0.83** ± 0.12	**0.95** ± 0.02
		Random	0.51 ± 0.29	0.64 ± 0.00	0.58 ± 0.01
	Random color	Circular	0.63 ± 0.03	0.80 ± 0.10	0.91 ± 0.03
		Spiral	0.64 ± 0.09	0.81 ± 0.10	0.93 ± 0.03
Graphormer			**0.74** ± 0.12	**0.83** ± 0.02	0.92 ± 0.01
Naive-Bayesian			0.51 ± 0.03	0.54 ± 0.04	0.51 ± 0.03

- VN-Solver demonstrates improved performance with more training data, indicating its ability to learn from data for solving the Hamiltonian cycle problem.
- Comparing to Graphormer, VN-Solver performs comparably with gray visualizations and surpasses it with uniform-color schemes.
- Results from VN-Solver and Graphormer are statistically significant, as they outperform Naive-Bayesian.

Acknowledgement

This project is supported in part by National Science Foundation under Career Award IIS-2144285.

COLLEGE OF ENGINEERING

Take a picture to **download** the **full paper**