On Representing (Mixed-Integer) Linear Programs by Graph Neural Networks

Chaolong Ying

School of Data Science
The Chinese University of Hong Kong, Shenzhen

May, 2023

Table of Contents

Linear Programming

Linear Programming (LP)

$$\min_{x\in\mathbb{R}^n} c^T x, \quad s.t. \; Ax \circ b, l \le x \le u \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $l \in (\mathbb{R} \cup \{-\infty\})^n$, $u \in (\mathbb{R} \cup \{-\infty\})^n$, and $\circ \in \{\le, =, \ge\}^m$. Any LP problems must follow one of the following three cases:

- *Infeasible*
- *Unbounded*
- *Feasible and bounded*

Question: Are there GNNs that can predict the *feasibility, boundedness and an optimal solution* of LP?

LP represented as weighted bipartite graph

- A weighted bipartite graph $G = (V \cup W, E)$ consists of a vertex set $V \cup W$ that are divided into two groups V and W with $V \cap W = \emptyset$.
- Each vertex in W represents a variable in LP and each vertex in V represents a constraint.

Bipartite Graph Representation

$$\min_{x \in \mathbb{R}^n} c^T x, \quad s.t.\ Ax \circ b, l \leq x \leq u \tag{2}$$

- Vertex v_i represents the i-th constraint in $Ax \circ b$, and vertex w_j represents the j-th variable x_j.
- Constraints is involved in the feature of v_i: $h_i^V = (b_i, \circ_i)$.
- Variables is involved in the feature of w_i: $h_j^W = (c_j, l_j, u_j)$.
- The edge connecting v_i and w_j has weight $E_{i,j} = A_{i,j}$.

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & x_1 + 2x_2, \\ \text{s.t.} \quad & x_1 + 2x_2 \geq 1, \\ & 2x_1 + x_2 = 2, \\ & x_1 \geq 0,\ x_2 \geq -1. \end{aligned}$$

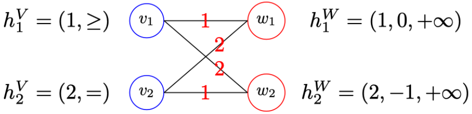


Figure 1: An example of LP-graph

Graph Neural Networks

Graph Neural Networks for LP

$$
\begin{aligned}
h_i^{l,V} &= g_l^V \left(h_i^{l-1,V}, \sum_{j=1}^{n} E_{i,j} f_l^W \left(h_j^{l-1,W} \right) \right), \quad i = 1, 2, \ldots, m \\
h_j^{l,W} &= g_l^W \left(h_j^{l-1,W}, \sum_{i=1}^{m} E_{i,j} f_l^V \left(h_i^{l-1,V} \right) \right), \quad j = 1, 2, \ldots, n
\end{aligned} \tag{3}
$$

Graph-level output:

$$
y_{\text{out}} = f_{\text{out}} \left(\sum_{i=1}^{m} h_i^{L,V}, \sum_{j=1}^{n} h_j^{L,W} \right) \tag{4}
$$

Node-level output:

$$
y_{\mathsf{out}} \left(w_j \right) = f_{\mathsf{out}}^W \left(\sum_{i=1}^{m} h_i^{L,V}, \sum_{j=1}^{n} h_j^{L,W}, h_j^{L,W} \right), \quad j = 1, 2, \cdots, n. \tag{5}
$$

Revisiting Question

Question: Are there GNNs that can predict the *feasibility, boundedness and an optimal solution* of LP?

- **Feasibility mapping** The feasibility mapping is a classification function

$$\Phi_{\mathsf{feas}} \ : G_{m,n} \times H_m^V \times H_n^W \to \{0, 1\} \tag{6}$$

- **Optimal objective value mapping**

$$\Phi_{\mathsf{obj}} \ : G_{m,n} \times H_m^V \times H_n^W \to \mathbb{R} \cup \{\infty, -\infty\} \tag{7}$$

Revisiting Question

Remark. In the case that a LP problem has finite optimal objective value, it is possible that the problem admits multiple optimal solutions. However, the optimal solution with the smallest l_2-norm must be unique. If $x \neq x'$ are two different solutions with $||x|| = ||x'||$

$$\left\| \frac{1}{2} \left(x + x' \right) \right\|^2 < \frac{1}{2} \|x\|^2 + \frac{1}{2} \left\| x' \right\|^2 = \|x\|^2 = \left\| x' \right\|^2 \tag{8}$$

- **Optimal solution mapping** For any $(G, H) \in \Phi_{obj}^{-1}(\mathbb{R})$, we have remarked before that the LP problem associated with (G, H) has a unique optimal solution with the smallest l_2-norm. Let

$$\Phi_{solu} : \Phi_{obj}^{-1}(\mathbb{R}) \to \mathbb{R}^n \tag{9}$$

  be the mapping that maps $(G, H) \in \Phi_{obj}^{-1}(\mathbb{R})$ to the optimal solution with the smallest l_2-norm.

Revisiting Question

- **Invariance**: we say a function $F : G_{m,n} \times H_m^V \times H_n^W \to \mathbb{R}$ is invariant if it satisfies

$$F(G, H) = F\left((\sigma_V, \sigma_W) * (G, H)\right), \forall \sigma_V \in S_m, \sigma_W \in S_n \tag{10}$$

- **Equivariance**: function F is equivariant if it satisfies

$$\sigma_W\left(F_W(G, H)\right) = F_W\left((\sigma_V, \sigma_W) * (G, H)\right), \quad \forall \sigma_V \in S_m, \sigma_W \in S_n \tag{11}$$

Theorem

Given any two LP instances $(G, H), (\hat{G}, \hat{H}) \in G_{m,n} \times H_m^V \times H_n^W$, as long as their feasibility or boundedness are different, there must exist $F \in F_{GNN}$ that can distinguish them: $F(G, H) \neq F(\hat{G}, \hat{H})$. Moreover, as long as their optimal solutions with the smallest l_2-norm are different, there must exist $F_W \in F_{GNN}^W$ that can distinguish them: $F_W(G, H) \neq F_W(\hat{G}, \hat{H})$.

Theorem

Algorithm 1 The WL test for LP-Graphs[2] (denoted by $\mathrm{WL}_{\mathrm{LP}}$)

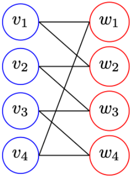
Require: A graph instance $(G, H) \in \mathcal{G}_{m,n} \times \mathcal{H}_m^V \times \mathcal{H}_n^W$ and iteration limit $L > 0$.

1: Initialize with $C_i^{0,V} = \mathrm{HASH}_{0,V}(h_i^V)$, $C_j^{0,W} = \mathrm{HASH}_{0,W}(h_j^W)$.
2: **for** $l = 1, 2, \cdots, L$ **do**
3: $\quad C_i^{l,V} = \mathrm{HASH}_{l,V}\left(C_i^{l-1,V}, \sum_{j=1}^n E_{i,j}\mathrm{HASH}'_{l,W}\left(C_j^{l-1,W}\right)\right)$.
4: $\quad C_j^{l,W} = \mathrm{HASH}_{l,W}\left(C_j^{l-1,W}, \sum_{i=1}^m E_{i,j}\mathrm{HASH}'_{l,V}\left(C_i^{l-1,V}\right)\right)$.
5: **end for**
6: **return** The multisets containing all colors $\{\{C_i^{L,V}\}\}_{i=0}^m, \{\{C_j^{L,W}\}\}_{j=0}^n$.

Theorem. For any $(G, H), (\hat{G}, \hat{H})$, the followings are equivalent:

- (G, H) and $(\hat{G}, \hat{H})$ are not distinguishable by **Algorithm 1**.
- $F(G, H) = F(\hat{G}, \hat{H}), \forall F \in F_{GNN}$.
- For $\forall F_W \in F_{GNN}^W$, there exist $\sigma_W \in S_n$ such that $F_W(G, H) = \sigma_W(F_W(\hat{G}, \hat{H}))$.

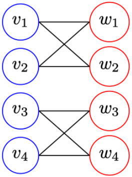
Examples



$$\begin{aligned} \min\ & x_1+x_2+x_3+x_4, \\ \text{s.t. } & x_1+x_2=1, \\ & x_2+x_3=1, \\ & x_3+x_4=1, \\ & x_4+x_1=1, \\ & x_j \geq 1,\ 1\leq j\leq 4. \end{aligned}$$

$$\begin{aligned} \min\ & x_1+x_2+x_3+x_4, \\ \text{s.t. } & x_1+x_2\leq 1, \\ & x_2+x_3\leq 1, \\ & x_3+x_4\leq 1, \\ & x_4+x_1\leq 1, \\ & x_j \leq 1,\ 1\leq j\leq 4. \end{aligned}$$

$$\begin{aligned} \min\ & x_1+x_2+x_3+x_4, \\ \text{s.t. } & x_1+x_2=1, \\ & x_2+x_3=1, \\ & x_3+x_4=1, \\ & x_4+x_1=1, \\ & x_j \leq 1,\ 1\leq j\leq 4. \end{aligned}$$



$$\begin{aligned} \min\ & x_1+x_2+x_3+x_4, \\ \text{s.t. } & x_1+x_2=1, \\ & x_2+x_1=1, \\ & x_3+x_4=1, \\ & x_4+x_3=1, \\ & x_j \geq 1,\ 1\leq j\leq 4. \end{aligned}$$

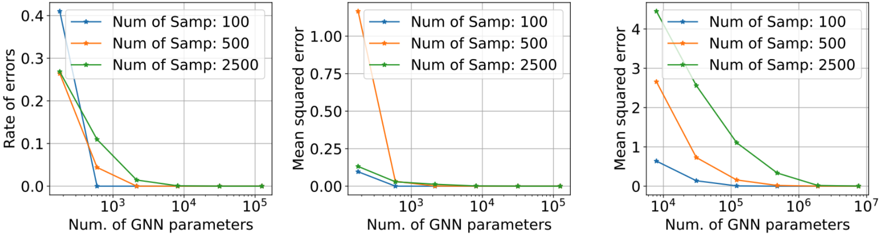
$$\begin{aligned} \min\ & x_1+x_2+x_3+x_4, \\ \text{s.t. } & x_1+x_2\leq 1, \\ & x_2+x_1\leq 1, \\ & x_3+x_4\leq 1, \\ & x_4+x_3\leq 1, \\ & x_j \leq 1,\ 1\leq j\leq 4. \end{aligned}$$

$$\begin{aligned} \min\ & x_1+x_2+x_3+x_4, \\ \text{s.t. } & x_1+x_2=1, \\ & x_2+x_1=1, \\ & x_3+x_4=1, \\ & x_4+x_3=1, \\ & x_j \leq 1,\ 1\leq j\leq 4. \end{aligned}$$

Figure 2: LP-graphs that cannot be distinguished by the WL test. Since the features and neighbor information of $\{v_i\}$ and $\{w_j\}$ in the two graphs are equal, it holds for both graphs that $C_1^{l,V} = \cdots C_4^{l,V}$ and $C_1^{l,W} = \cdots C_4^{l,W}$ for all $l \geq 0$, whatever the hash functions are chosen. Based on this graph pair, we construct three pairs of LPs that are both infeasible, both unbounded, both feasible bounded with the same optimal solution, respectively.

Numerial Experiments

$m = 10, n = 50$



(a) Feasibility (b) Optimal objective value (c) Optimal solution

Figure 3: GNN can approximate Φ_{feas}, Φ_{obj}, and Φ_{solu}

Table of Contents

Linear Programming

Mixed-Integer Linear Programming (MILP)

$$\min_{x \in \mathbb{R}^n} c^T x, \quad s.t. \ Ax \circ b, l \leq x \leq u, \quad x_j \in \mathbb{Z}, \forall j \in I \tag{12}$$

where $A \in \mathbb{R}^{m \times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $l \in (\mathbb{R} \cup \{-\infty\})^n$, $u \in (\mathbb{R} \cup \{-\infty\})^n$, and $\circ \in \{\leq, =, \geq\}^m$. The index set $I \in \{1, 2, ..., n\}$ includes those indices j where x_j are constrained to be an integer.

Question: Are there GNNs that can predict the *feasibility, boundedness and an optimal solution* of MILP?

Bipartite Graph Representation

$$\min_{x \in \mathbb{R}^n} c^T x, \quad s.t.\ Ax \circ b, l \leq x \leq u, \quad x_j \in \mathbb{Z}, \forall j \in I \tag{13}$$

- Vertex v_i represents the i-th constraint in $Ax \circ b$, and vertex w_j represents the j-th variable x_j.
- Constraints is involved in the feature of v_i: $h_i^V = (b_i, \circ_i)$.
- Variables is involved in the feature of w_i: $h_j^W = (c_j, l_j, u_j, \tau_j)$.
- The edge connecting v_i and w_j has weight $E_{i,j} = A_{i,j}$.

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & x_1 + x_2, \\ \text{s.t.} \quad & x_1 + 3x_2 \geq 1,\ x_1 + x_2 \geq 1, \\ & x_1 \leq 3,\ x_2 \leq 5,\ x_2 \in \mathbb{Z}. \end{aligned}$$

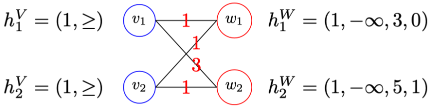


Figure 1: An example of MILP-graph

Examples

$$\min_{x \in \mathbb{R}^6} \; x_1 + x_2 + x_3 + x_4 + x_5 + x_6,$$
$$\text{s.t. } x_1 + x_2 = 1,\; x_2 + x_3 = 1,\; x_3 + x_4 = 1,$$
$$x_4 + x_5 = 1,\; x_5 + x_6 = 1,\; x_6 + x_1 = 1,$$
$$0 \le x_j \le 1,\; x_j \in \mathbb{Z},\; \forall\, j \in \{1, 2, \dots, 6\}.$$

$$\min_{x \in \mathbb{R}^6} \; x_1 + x_2 + x_3 + x_4 + x_5 + x_6,$$
$$\text{s.t. } x_1 + x_2 = 1,\; x_2 + x_3 = 1,\; x_3 + x_1 = 1,$$
$$x_4 + x_5 = 1,\; x_5 + x_6 = 1,\; x_6 + x_4 = 1,$$
$$0 \le x_j \le 1,\; x_j \in \mathbb{Z},\; \forall\, j \in \{1, 2, \dots, 6\}.$$

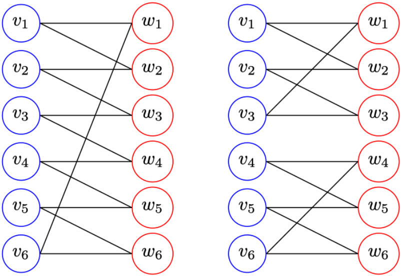


Figure 2: Two non-isomorphic MILP graphs that cannot be distinguished by WL test

Unfoldable MILP Problems

Foldable MILP. Given any MILP instance, one would obtain vertex colors $C_i^{l,V}$ by running **Algorithm 1**. We say that an MILP instance can be folded, or is foldable, if there exist $1 \leq i, i' \leq m$ or $1 \leq j, j\text{fl} \leq\leq n$ such that

$$C_i^{l,V} = C_{i'}^{l,V}, \quad i \neq i', \quad \text{or} \quad C_j^{l,W} = C_{j'}^{l,W}, j \neq j'$$

for any $l \in \mathbb{N}$ and any hash functions. In another word, at least one color in the multisets generated by the WL test always has a multiplicity greater than 1.

Theorem. As long as those foldable MILPs are removed, GNN is able to accurately predict the feasibility of all MILP instances in a dataset with finite samples.

Numerial Experiments

$m = 6, n = 20$

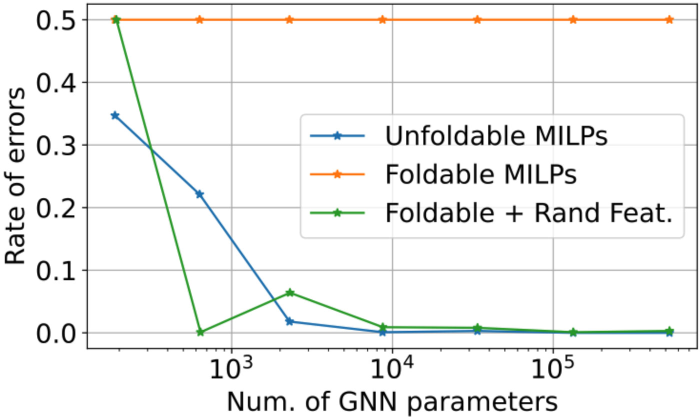
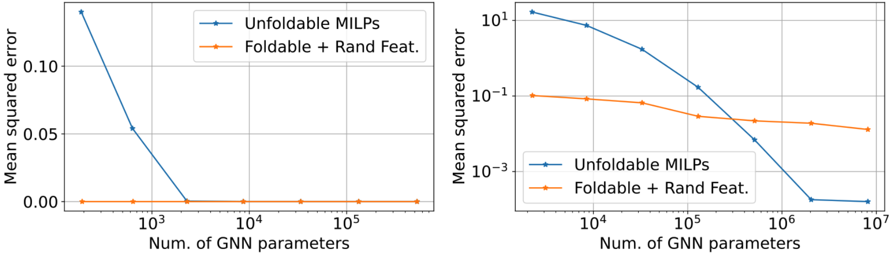


Figure 3: Feasibility

Numerial Experiments



(a) Optimal objective value

(b) Optimal solution

Figure 4: GNN can approximate Φ_{obj}, and Φ_{solu}

Thanks!