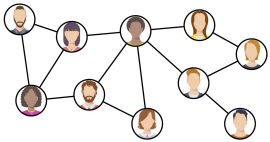
Graph Neural Networks for Data-driven Optimization

Christopher Morris @chrsmrrs

November 1, 2022

RWTH Aachen University

Graph and relational data



Social Networks

Molecules

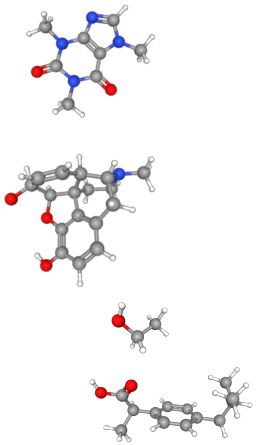
$$\operatorname{argmin}_x c^T x$$
$$Ax \geq 0$$
$$x \in \{0,1\}^n$$

Discrete Optimization

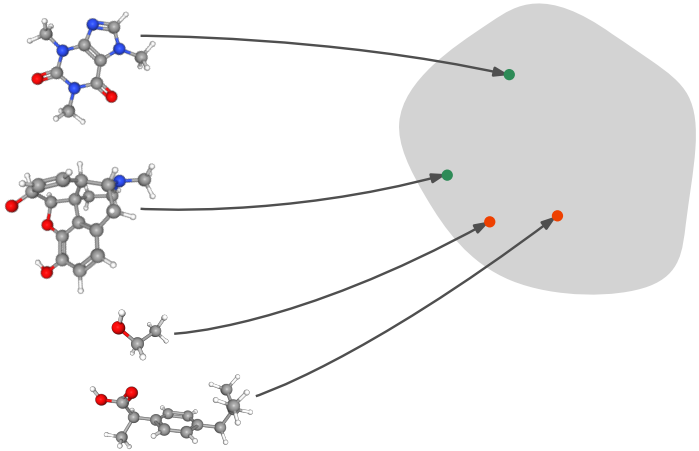
Knowledge Bases

Learning with graphs

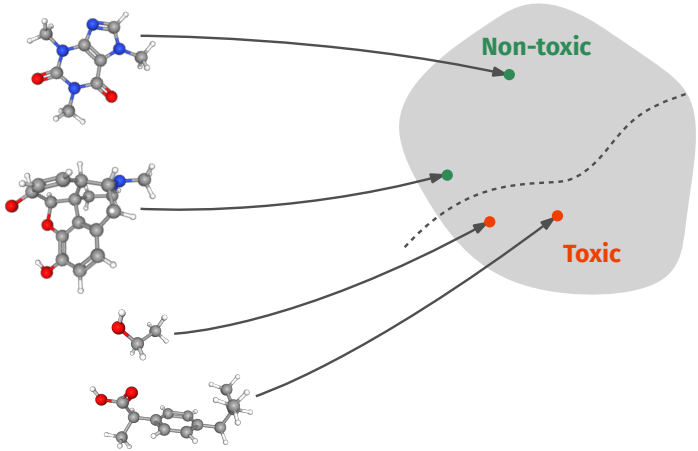
Learning with graphs



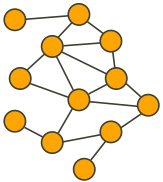
Learning with graphs



Learning with graphs

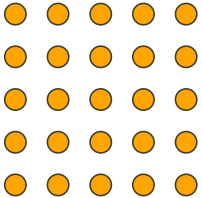


Networks are complex!



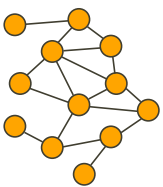
Network

vs.



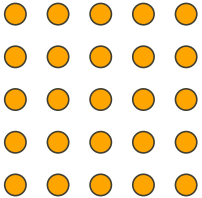
Image

Networks are complex!



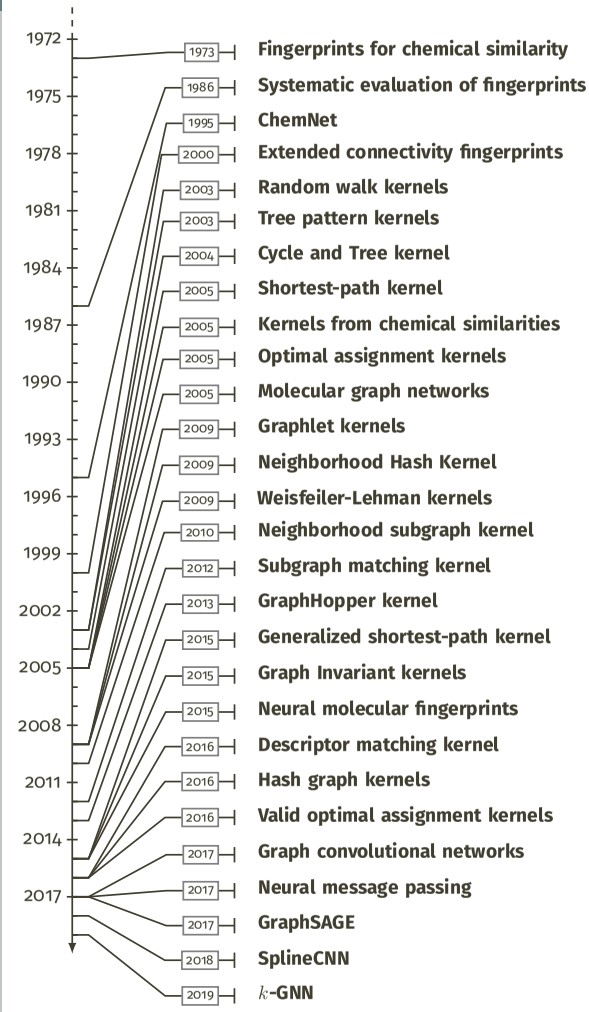
Network

vs.

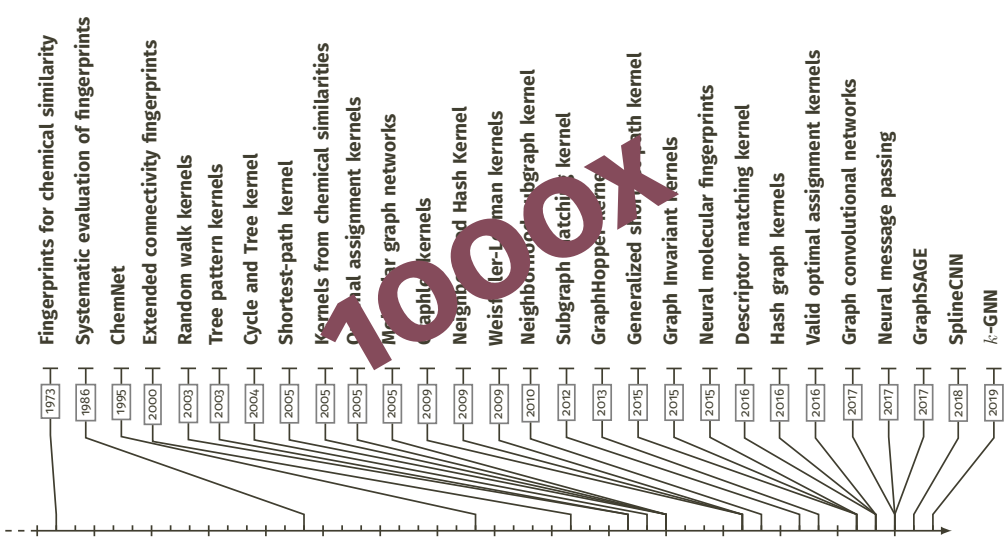


Image

- **Arbitrary size**
- **No** fixed **node ordering** (invariance to permutation)
- **Hierarchical structure**
- …


1972
1975
1978
1981
1984
1987
1990
1993
1996
1999
2002
2005
2008
2011
2014
2017
1973 Fingerprints for chemical similarity
1986 Systematic evaluation of fingerprints
1995 ChemNet
2000 Extended connectivity fingerprints
2003 Random walk kernels
2003 Tree pattern kernels
2004 Cycle and Tree kernel
2005 Shortest-path kernel
2005 Kernels from chemical similarities
2005 Optimal assignment kernels
2005 Molecular graph networks
2009 Graphlet kernels
2009 Neighborhood Hash Kernel
2009 Weisfeiler-Lehman kernels
2010 Neighborhood subgraph kernel
2012 Subgraph matching kernel
2013 GraphHopper kernel
2015 Generalized shortest-path kernel
2015 Graph Invariant kernels
2015 Neural molecular fingerprints
2016 Descriptor matching kernel
2016 Hash graph kernels
2016 Valid optimal assignment kernels
2017 Graph convolutional networks
2017 Neural message passing
2017 GraphSAGE
2018 SplineCNN
2019 k-GNN

The state of the art?



The state of the art!

A practical theory of machine learning with graphs

Mathematical Foundations

A practical theory of machine learning with graphs

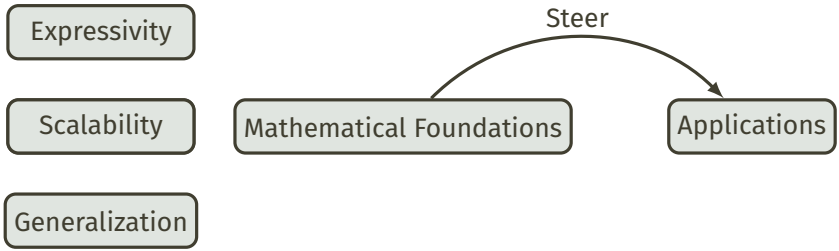
Expressivity

Scalability

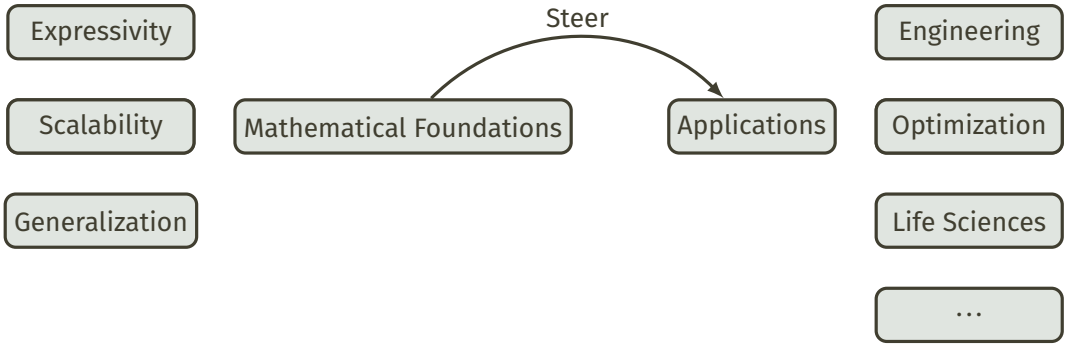
Mathematical Foundations

Generalization

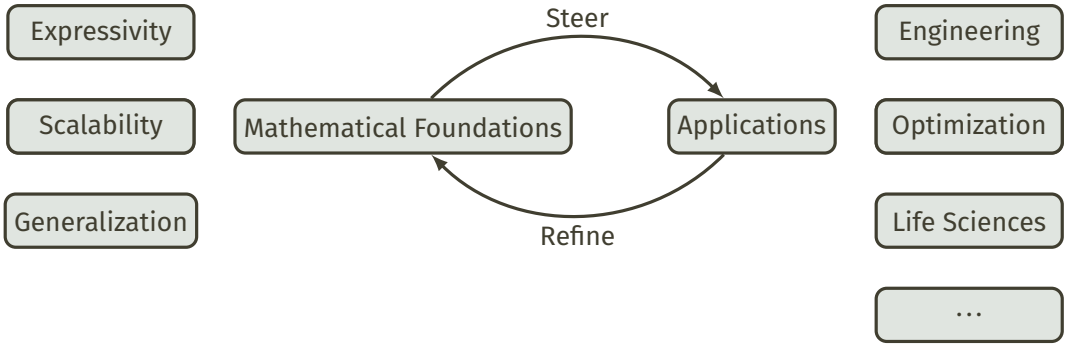
A practical theory of machine learning with graphs



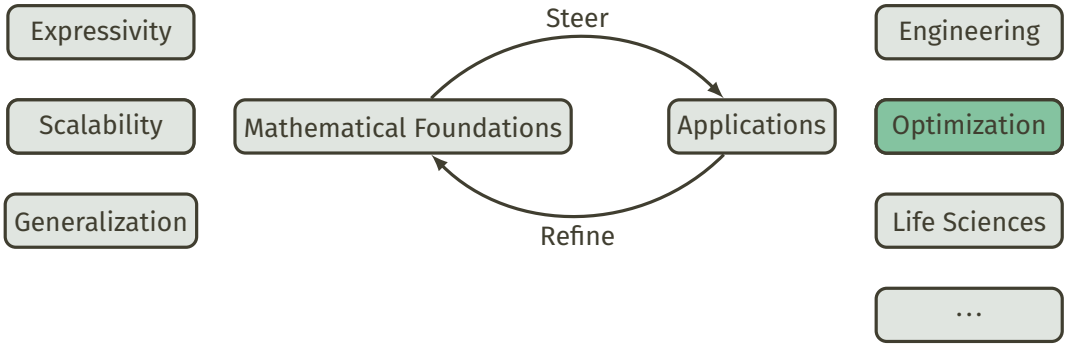
A practical theory of machine learning with graphs



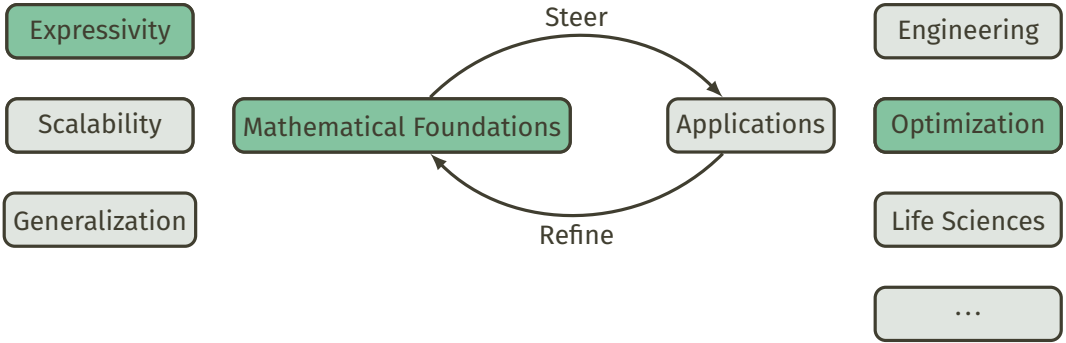
A practical theory of machine learning with graphs



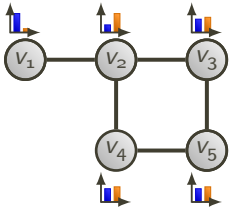
A practical theory of machine learning with graphs



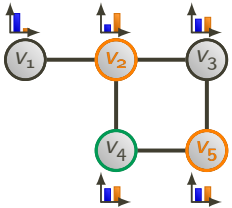
A practical theory of machine learning with graphs



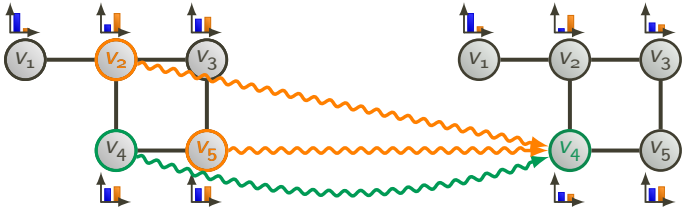
Graph Neural Networks



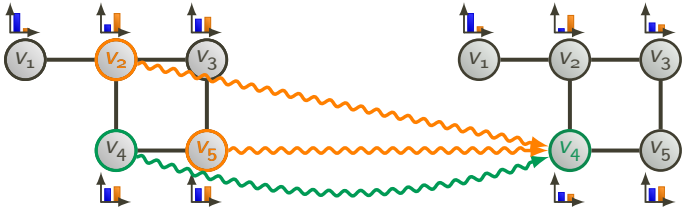
Graph Neural Networks



Graph Neural Networks

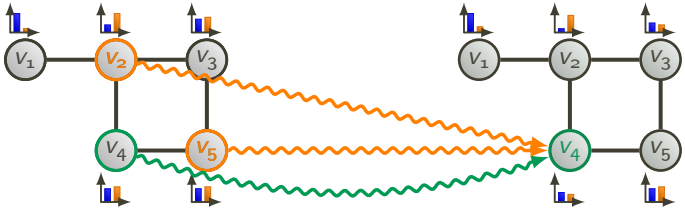


Graph Neural Networks



$$f^{(t)}(v) = \sigma\big(W_1\, f^{(t-1)}(v) + W_2 \sum_{w \in N(v)} f^{(t-1)}(w)\big)$$

Graph Neural Networks



$$f^{(t)}(v) = f_{\text{merge}}^{W_1}\left(f^{(t-1)}(v), f_{\text{aggr}}^{W_2}\left(\{\!\{f^{(t-1)}(w) \mid w \in N(v)\}\!\}\right)\right)$$

How can we use **machine learning** *to enhance* **combinatorial solvers?**

Q. Cappart, D. Chételat, E. Khalil, A. Lodi, C. Morris, and P. Velickovic. “Combinatorial optimization and reasoning with graph neural networks”. In: *International Joint Conference on Artificial Intelligence*. IJCAI, 2021

Data-driven optimization: Binary Integer Optimization

$$\arg\min_{\mathbf{x}} \mathbf{c}^T \mathbf{x}$$

$$\mathbf{A}\mathbf{x} \geq 0$$

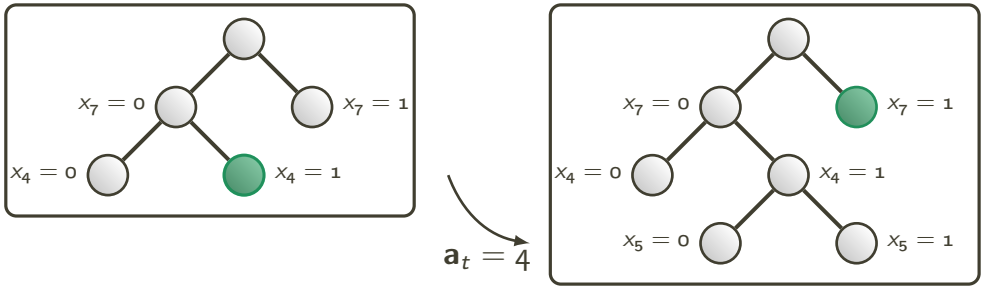
$$x_i \in \{0, 1\}$$

Data-driven optimization: Binary Integer Optimization

$$\arg\min_{x} \mathbf{c}^T \mathbf{x}$$

$$\mathbf{A}\mathbf{x} \geq 0$$

$$x_i \in \{0, 1\}$$

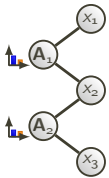


Data-driven optimization: Variable selection via GNNs

$$\arg\min_{x} \mathbf{c}^T \mathbf{x}$$

$$\mathbf{A}\mathbf{x} \geq \mathbf{o}$$

$$x_i \in \{0, 1\}$$



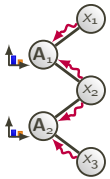
M. Gasse, D. Chételat, N. Ferroni, L. Charlin, and A. Lodi. "Exact Combinatorial Optimization with Graph Convolutional Neural Networks". In: *NeurIPS*. 2019

Data-driven optimization: Variable selection via GNNs

$$\arg\min_{x} \mathbf{c}^T \mathbf{x}$$

$$\mathbf{A}\mathbf{x} \geq \mathbf{o}$$

$$x_i \in \{0, 1\}$$



Update constraints

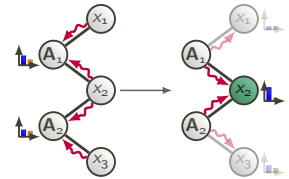
M. Gasse, D. Chételat, N. Ferroni, L. Charlin, and A. Lodi. “Exact Combinatorial Optimization with Graph Convolutional Neural Networks”. In: *NeurIPS*. 2019

Data-driven optimization: Variable selection via GNNs

$$\arg\min_{x} \mathbf{c}^T \mathbf{x}$$

$$\mathbf{A}\mathbf{x} \geq \mathbf{o}$$

$$x_i \in \{0, 1\}$$



Update constraints Update variables

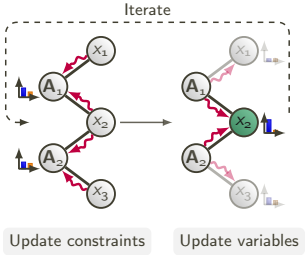
M. Gasse, D. Chételat, N. Ferroni, L. Charlin, and A. Lodi. "Exact Combinatorial Optimization with Graph Convolutional Neural Networks". In: *NeurIPS*. 2019

Data-driven optimization: Variable selection via GNNs

$$\arg\min_{x} \mathbf{c}^T \mathbf{x}$$

$$\mathbf{A}\mathbf{x} \geq \mathbf{o}$$

$$x_i \in \{0, 1\}$$



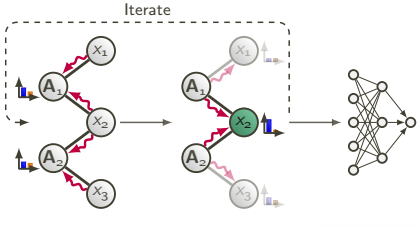
M. Gasse, D. Chételat, N. Ferroni, L. Charlin, and A. Lodi. "Exact Combinatorial Optimization with Graph Convolutional Neural Networks". In: *NeurIPS*. 2019

Data-driven optimization: Variable selection via GNNs

$$\arg\min_{x} \mathbf{c}^T \mathbf{x}$$

$$\mathbf{A}\mathbf{x} \geq \mathbf{o}$$

$$x_i \in \{0, 1\}$$



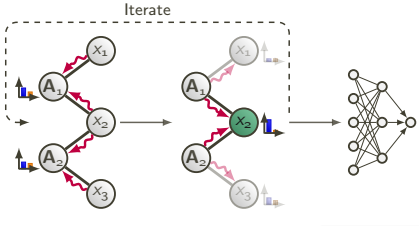
M. Gasse, D. Chételat, N. Ferroni, L. Charlin, and A. Lodi. “Exact Combinatorial Optimization with Graph Convolutional Neural Networks”. In: *NeurIPS*. 2019

Data-driven optimization: Variable selection via GNNs

$$\arg\min_{x} \mathbf{c}^T \mathbf{x}$$

$$\mathbf{A}\mathbf{x} \geq \mathbf{o}$$

$$x_i \in \{0, 1\}$$



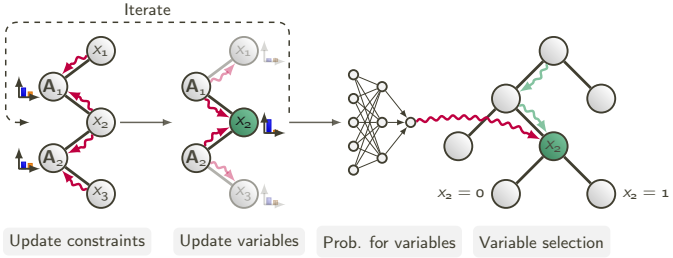
M. Gasse, D. Chételat, N. Ferroni, L. Charlin, and A. Lodi. “Exact Combinatorial Optimization with Graph Convolutional Neural Networks”. In: *NeurIPS*. 2019

Data-driven optimization: Variable selection via GNNs

$$\arg\min_{x} \mathbf{c}^T \mathbf{x}$$

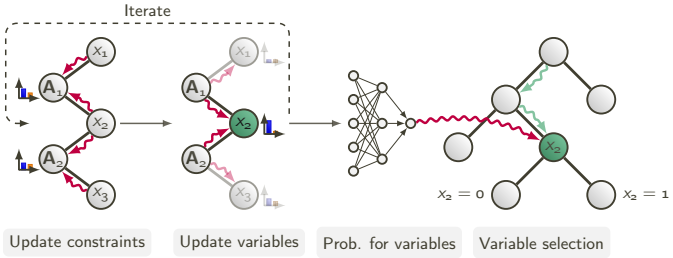
$$\mathbf{A}\mathbf{x} \geq \mathbf{0}$$

$$x_i \in \{0, 1\}$$



M. Gasse, D. Chételat, N. Ferroni, L. Charlin, and A. Lodi. "Exact Combinatorial Optimization with Graph Convolutional Neural Networks". In: *NeurIPS*. 2019

Data-driven optimization: Variable selection via GNNs



Data-driven optimization: Variable selection via GNNs

- Aim at simulate **strong branching heuristic**
- Trained in **supervised-fashion** against cross-entropy loss
- Beats default **variable-selection heuristics** of SCIP

M. Gasse, D. Chételat, N. Ferroni, L. Charlin, and A. Lodi. “Exact Combinatorial Optimization with Graph Convolutional Neural Networks”. In: *NeurIPS*. 2019

Combinatorial Optimization and Reasoning with Graph Neural Networks

Quentin Cappart
Department of Computer Engineering and Software Engineering
Polytechnique Montréal
Montréal, Canada
QUENTIN.CAPPART@POLYMTL.CA

Didier Chételat
CERC in Data Science for Real-Time Decision-Making
Polytechnique Montréal
Montréal, Canada
DIDIER.CHETELAT@POLYMTL.CA

Elias B. Khalil
Department of Mechanical & Industrial Engineering,
University of Toronto
Toronto, Canada
KHALIL@MIE.UTORONTO.CA

Andrea Lodi
Jacobs Technion-Cornell Institute
Cornell Tech and Technion - IIT
New York, USA
ANDREA.LODI@CORNELL.EDU

Christopher Morris
Department of Computer Science
RWTH Aachen University
Aachen, Germany
MORRIS@CS.RWTH-AACHEN.DE

Petar Veličković
DeepMind
London, UK

Data-driven optimization: Variable selection via GNNs

Problem

- Only works for a **single heuristic**
- **Not generic**, e.g., **different models** needed for variable and node selection

Design **generic approach** *such that* **once-trained model** *can* **replace many heuristics!**

MIP-GNN: Variable biases

Variable Biases I

- Let I be an problem instance with a corresponding **BIP formulation** $(\mathbf{A}, \mathbf{b}, \mathbf{c})$.
- Consider **feasible solutions** that are **close** to a **optimal solution** $\mathbf{x}^*$ for the instance I:

$$F_{\varepsilon}^{*}(I) = \{\mathbf{x} \in F_{\text{Int}}(I) \colon |\mathbf{c}^T\mathbf{x}^* - \mathbf{c}^T\mathbf{x}| \leq \varepsilon\}$$

- Near-optimal solutions can be enumerated with MIP solvers' "solution pool" feature.

E. Khalil, C. Morris, and A. Lodi. "MIP-GNN: A data-driven framework for guiding combinatorial solvers". In: *AAAI*. 2022

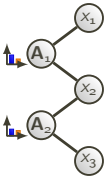
Variable Biases II

The vector of **variable biases** $\bar{\mathbf{b}}(I) \in \mathbb{R}^n$ of I w.r.t. to $F_\varepsilon^*(I)$ is the **component-wise average** over all elements in $F_\varepsilon^*(I)$, namely

$$\bar{\mathbf{b}}(I) = {}^1\!/\!_{|F_\varepsilon^*|} \sum_{\mathbf{x} \in F_\varepsilon^*(I)} \mathbf{x}.$$

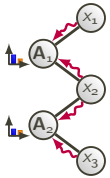
Probability of a **variable** *being* **0 or 1 in an optimal solution**

MIP-GNN: Node selection



E. Khalil, C. Morris, and A. Lodi. “MIP-GNN: A data-driven framework for guiding combinatorial solvers”. In: *AAAI*. 2022

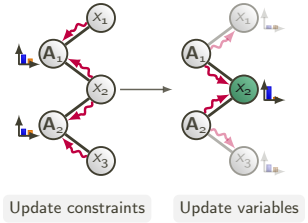
MIP-GNN: Node selection



Update constraints

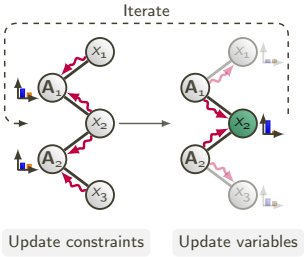
E. Khalil, C. Morris, and A. Lodi. “MIP-GNN: A data-driven framework for guiding combinatorial solvers”. In: *AAAI*. 2022

MIP-GNN: Node selection



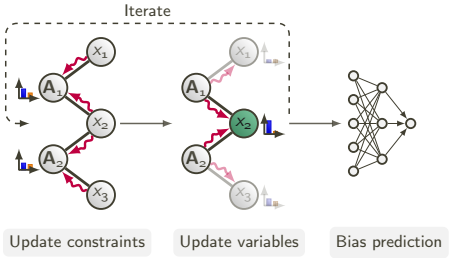
E. Khalil, C. Morris, and A. Lodi. “MIP-GNN: A data-driven framework for guiding combinatorial solvers”. In: *AAAI*. 2022

MIP-GNN: Node selection



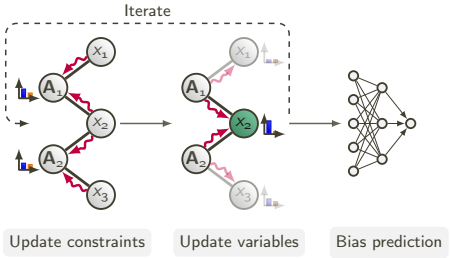
E. Khalil, C. Morris, and A. Lodi. “MIP-GNN: A data-driven framework for guiding combinatorial solvers”. In: *AAAI*. 2022

MIP-GNN: Node selection



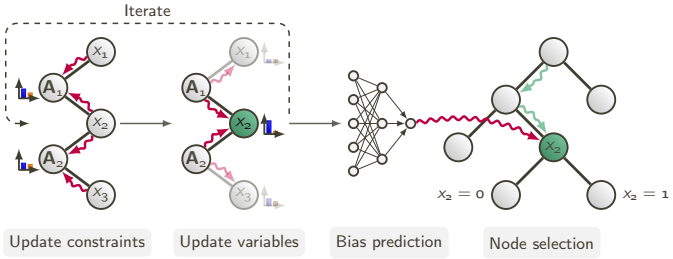
E. Khalil, C. Morris, and A. Lodi. “MIP-GNN: A data-driven framework for guiding combinatorial solvers”. In: *AAAI*. 2022

MIP-GNN: Node selection



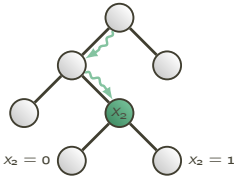
E. Khalil, C. Morris, and A. Lodi. “MIP-GNN: A data-driven framework for guiding combinatorial solvers”. In: *AAAI*. 2022

MIP-GNN: Node selection



E. Khalil, C. Morris, and A. Lodi. “MIP-GNN: A data-driven framework for guiding combinatorial solvers”. In: *AAAI*. 2022

MIP-GNN: Node selection



Node selection via bias prediction

- **Confidence score** $\texttt{score}(\widehat{\mathbf{p}}_i) = 1 - \left|\widehat{\mathbf{p}}_i - \lfloor \widehat{\mathbf{p}}_i \rceil\right|$
- **Node score**

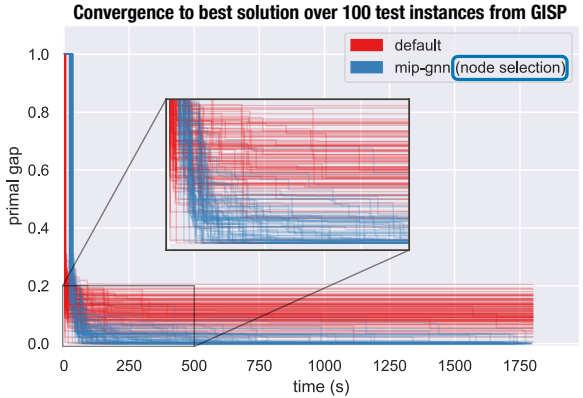
$$\texttt{node-score}(N; \widehat{\mathbf{p}}) = \sum_{i \in \text{ fixed-vars}(N)} \begin{cases} \texttt{score}(\widehat{\mathbf{p}}_i), & \text{if } x_i^N = \lfloor \widehat{\mathbf{p}}_i \rceil, \\ 1 - \texttt{score}(\widehat{\mathbf{p}}_i), & \text{otherwise}, \end{cases}$$

MIP-GNN: Experimental results

Experimental setup

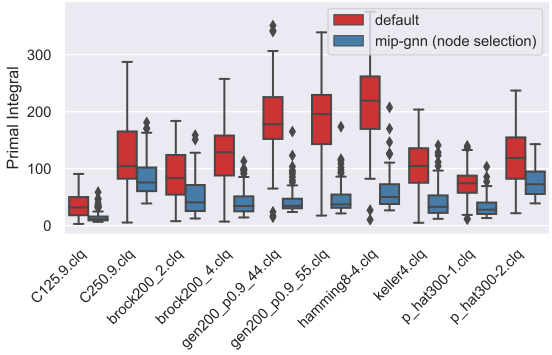
- **Benchmark problems:**
  - Generalized Independent Set Problem (GISP)
- **Solver details**:
  - IBM CPLEX 12.10.0
  - *Full-fledged setup:* Presolve, cuts, and primal heuristics enabled

MIP-GNN: Experimental results



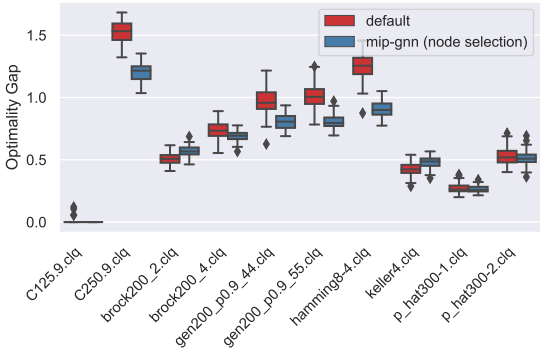
Distribution of **Primal Integrals** for ten problem sets of the Generalized Independent Set Problem, each with 100 unseen test instances.

MIP-GNN: Experimental results



Distribution of **Primal Integrals** for ten problem sets of the Generalized Independent Set Problem, each with 100 unseen test instances.

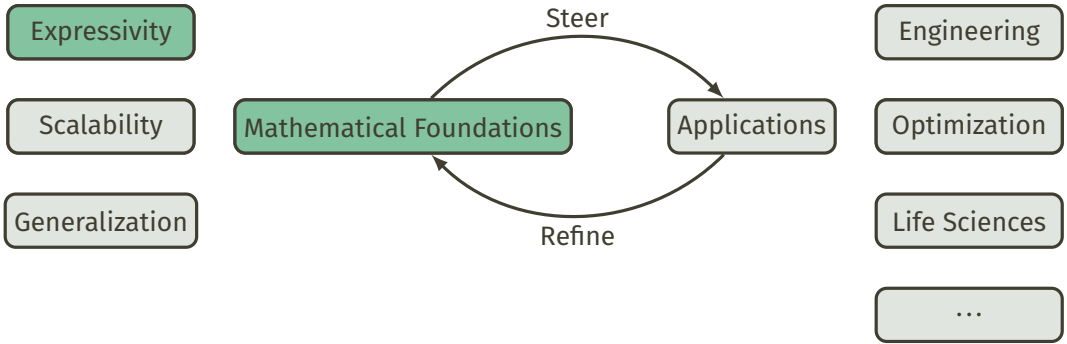
MIP-GNN: Experimental results: Limitations



Distribution of **Optimality Gaps** for ten problem sets of the Generalized Independent Set Problem, each with 100 unseen test instances.

GNNs *have severe* **limitations** *in distinguishing graphs with* **different structures!**

A practical theory of machine learning with graphs

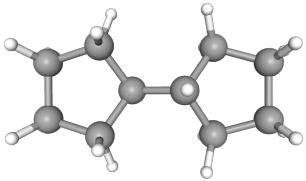


Expressivity of Graph Neural Networks

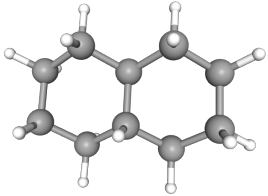
Insight

Any possible **GNN** architecture **misses crucial patterns** in the data!

Expressivity of Graph Neural Networks



(a) Bicyclopentyl



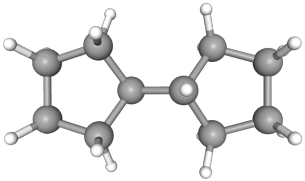
(b) Decalin

Insight

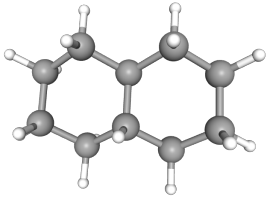
Any possible **GNN** architecture **misses crucial patterns** in the data!

C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. Eric Lenssen, G. Rattan, and M. Grohe. “Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks”. In: *AAAI*. 2019
C. Morris, G. Rattan, and P. Mutzel. “Weisfeiler and Leman Go Sparse: Higher-order Graph Embeddings”. In: *NeurIPS*. 2020

A hierarchy of more powerful models

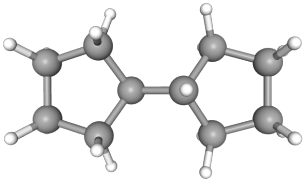


(a) Bicyclopentyl

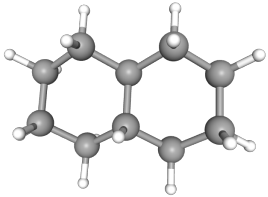


(b) Decalin

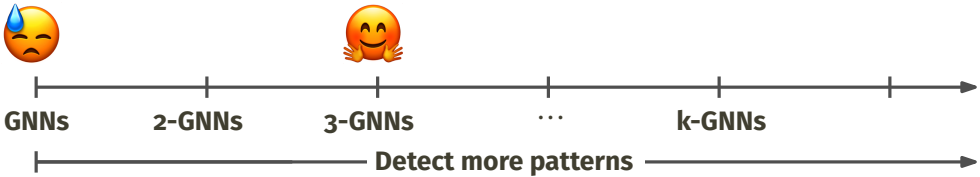
A hierarchy of more powerful models



(a) Bicyclopentyl



(b) Decalin



1-dim. Weisfeiler-Leman algorithm

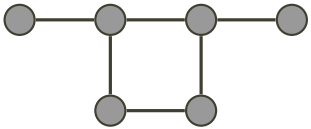
Heuristic for **graph isomorphism testing**

Iteration: Two vertices get **identical colors** iff their **colored neighborhoods are identical**

1-dim. Weisfeiler-Leman algorithm

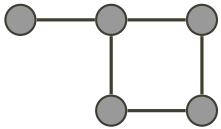
Heuristic for **graph isomorphism testing**

Iteration: Two vertices get **identical colors** iff their **colored neighborhoods are identical**



$\phi(G_1) = ($ $)$

(a) G_1



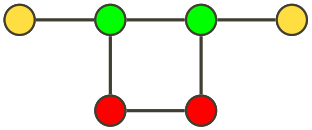
$\phi(G_2) = ($ $)$

(b) G_2

1-dim. Weisfeiler-Leman algorithm

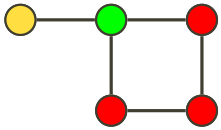
Heuristic for **graph isomorphism testing**

Iteration: Two vertices get **identical colors** iff their **colored neighborhoods are identical**



$\phi(G_1) = (2, 2, 2, \quad)$

(a) G_1



$\phi(G_2) = (1, 1, 3, \quad)$

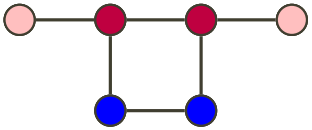
(b) G_2

B. Weisfeiler. *On Construction and Identification of Graphs.* Lecture Notes in Mathematics, Vol. 558. Springer, 1976

1-dim. Weisfeiler-Leman algorithm

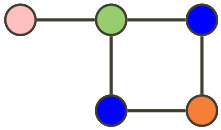
Heuristic for **graph isomorphism testing**

Iteration: Two vertices get **identical colors** iff their **colored neighborhoods are identical**



$\phi(G_1) = (2, 2, 2, 2, 2, 2, 0, 0)$

(a) G_1



$\phi(G_2) = (1, 1, 3, 2, 0, 1, 1, 1)$

(b) G_2

B. Weisfeiler. *On Construction and Identification of Graphs.* Lecture Notes in Mathematics, Vol. 558. Springer, 1976

Relationship between 1-WL and GNN

1-WL coloring

$$c^{(t)}(v) = \text{recolor}\left(c^{(t-1)}(v), \{\!\{c^{(t-1)}(w) \mid w \in N(v)\}\!\}\right)$$

Relationship between 1-WL and GNN

1-WL coloring

$$c^{(t)}(v) = \text{recolor}\Big(c^{(t-1)}(v), \{\!\{c^{(t-1)}(w) \mid w \in N(v)\}\!\}\Big)$$

General form of GNNs

$$h^{(t)}(v) = f_{\text{merge}}^{W_1^{(t)}}\Big(h^{(t-1)}(v),\ f_{\text{aggr}}^{W_2^{(t)}}\left(\{\!\{h^{(t-1)}(w) \mid w \in N(v)\}\!\}\right)\Big)$$

Relationship between 1-WL and GNN

1-WL coloring

$$c^{(t)}(v) = \text{recolor}\Big(c^{(t-1)}(v), \{\!\{c^{(t-1)}(w) \mid w \in N(v)\}\!\}\Big)$$

General form of GNNs

$$h^{(t)}(v) = f_{\text{merge}}^{W_1^{(t)}}\Big(h^{(t-1)}(v),\ f_{\text{aggr}}^{W_2^{(t)}}\left(\{\!\{h^{(t-1)}(w) \mid w \in N(v)\}\!\}\right)\Big)$$

Theorem (Informal)

GNNs cannot *be more* ***expressive*** *than* ***1-WL*** *in terms of distinguishing non-isomorphic graphs.*

C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. Eric Lenssen, G. Rattan, and M. Grohe. “Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks”. In: *AAAI*. 2019

K. Xu, W. Hu, J. Leskovec, and S. Jegelka. “How Powerful are Graph Neural Networks?” In: *ICLR*. 2019

Relationship between 1-WL and GNNs

1-WL coloring

$$c^{(t)}(v) = \text{hash}\left(c^{(t-1)}(v), \{\!\{c^{(t-1)}(w) \mid w \in N(v)\}\!\}\right)$$

General form of GNNs

$$h^{(t)}(v) = f_{\text{merge}}^{W_1^{(t)}}\left(h^{(t-1)}(v),\ f_{\text{aggr}}^{W_2^{(t)}}\left(\{\!\{h^{(t-1)}(w) \mid w \in N(v)\}\!\}\right)\right)$$

Relationship between 1-WL and GNNs

1-WL coloring

$$c^{(t)}(v) = \text{hash}\left(c^{(t-1)}(v), \{\!\{c^{(t-1)}(w) \mid w \in N(v)\}\!\}\right)$$

General form of GNNs

$$h^{(t)}(v) = f_{\text{merge}}^{W_1^{(t)}}\left(h^{(t-1)}(v),\ f_{\text{aggr}}^{W_2^{(t)}}\left(\{\!\{h^{(t-1)}(w) \mid w \in N(v)\}\!\}\right)\right)$$

Theorem (Informal)

*There exists a **GNN architecture** and corresponding **weights** such that it has the **same power** as the **1-WL**.*

C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. Eric Lenssen, G. Rattan, and M. Grohe. “Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks”. In: *AAAI*. 2019

Relationship between 1-WL and GNNs

Theorem (Informal)

*There exists a **GNN architecture** and corresponding **weights** such that it has the **same power** as the **1-WL**.*

Relationship between 1-WL and GNNs

Theorem (Informal)

*There exists a **GNN architecture** and corresponding **weights** such that it has the **same power** as the **1-WL**.*

Relationship between 1-WL and GNNs

Theorem (Informal)

*There exists a **GNN architecture** and corresponding **weights** such that it has the **same power** as the **1-WL**.*

Take Away

GNNs have the **same power** as **1-WL** in distinguishing **non-isomorphic graphs**.

Relationship between 1-WL and GNNs

Theorem (Informal)

*There exists a **GNN architecture** and corresponding **weights** such that it has the **same power** as the **1-WL**.*

Take Away

GNNs have the **same power** as **1-WL** in distinguishing **non-isomorphic graphs**. **Limits** of 1-WL are **well understood**.

V. Arvind, J. Köbler, G. Rattan, and O. Verbitsky. “On the Power of Color Refinement”. In: *International Symposium on Fundamentals of Computation Theory*. 2015, pp. 339–350

Limits of 1-WL and GNNs

Observation

GNNs cannot distinguish very **basic graph properties**, e.g.,

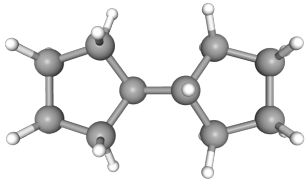
- Cycles of different lengths
- Triangle counts
- Bipartite graphs

Limits of 1-WL and GNNs

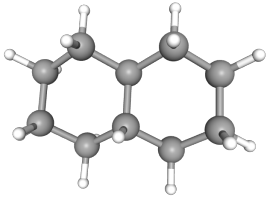
Observation

GNNs cannot distinguish very **basic graph properties**, e.g.,

- Cycles of different lengths
- Triangle counts
- Bipartite graphs



(a) Bicyclopentyl



(b) Decalin

Weisfeiler and Leman go Machine Learning: The Story so far

Christopher Morris[1], Yaron Lipman[2], Haggai Maron[3], Bastian Rieck[4], Nils M. Kriege[5], Martin Grohe[6], Matthias Fey[7], and Karsten Borgwardt[8]

[1]McGill University and Mila – Quebec AI Institute
[2]Department of Computer Science and Applied Mathematics, Weizmann Institute of Science
[3]NVIDIA Research
[4]AIDOS Lab, Institute of AI for Health, Helmholtz Zentrum München
[5]University of Vienna, Faculty of Computer Science, Vienna
[6]Department of Computer Science, RWTH Aachen University
[7]Department of Computer Science, TU Dortmund University
[8]Machine Learning & Computational Biology Lab, Department of Biosyste
Engineering, ETH Zürich, and Swiss Institute of Bioinf

Is the **Weisfeiler-Leman algorithm** *the* **right yard stick** *to study* **GNNs for data-driven optimization?**

Are **GNNs powerful** *enough* **to solve LPs?**

Is

$$Ax \geq b$$

feasible?

S. Arora, E. Hazan, and S. Kale. “The Multiplicative Weights Update Method: a Meta-Algorithm and Applications”. In: *Theory of Computing* 8.1 (2012), pp. 121–164

Multiplicatice weight update algorithm

Is

$$p^T Ax \geq p^T b$$

feasible?

S. Arora, E. Hazan, and S. Kale. “The Multiplicative Weights Update Method: a Meta-Algorithm and Applications”. In: *Theory of Computing* 8.1 (2012), pp. 121–164

Multiplicatice weight update algorithm

1. **Guess initial weighting**, e.g., $p_i = 1/m$

S. Arora, E. Hazan, and S. Kale. “The Multiplicative Weights Update Method: a Meta-Algorithm and Applications”. In: *Theory of Computing* 8.1 (2012), pp. 121–164

Multiplicatice weight update algorithm

1. **Guess initial weighting**, e.g., $p_i = 1/m$
2. **Solve** $p^T Ax \geq p^T b$

S. Arora, E. Hazan, and S. Kale. “The Multiplicative Weights Update Method: a Meta-Algorithm and Applications”. In: *Theory of Computing* 8.1 (2012), pp. 121–164

Multiplicatice weight update algorithm

1. **Guess initial weighting**, e.g., $p_i = 1/m$
2. **Solve** $p^T Ax \geq p^T b$
3. Compute **error signal**

$$e_j \leftarrow 1/\rho \left(\sum_{i \in N(j)} A_{ji} \mathbf{x}_i \right) - \mathbf{b}_j$$

S. Arora, E. Hazan, and S. Kale. “The Multiplicative Weights Update Method: a Meta-Algorithm and Applications”. In: *Theory of Computing* 8.1 (2012), pp. 121–164

Multiplicatice weight update algorithm

1. **Guess initial weighting**, e.g., $p_i = 1/m$
2. **Solve** $p^T Ax \geq p^T b$
3. Compute **error signal**

$$e_j \leftarrow 1/\rho \left(\sum_{i \in N(j)} A_{ji} \mathbf{x}_i \right) - \mathbf{b}_j$$

4. **Update** $w_i \leftarrow (1 - \eta e_j) w_i$

S. Arora, E. Hazan, and S. Kale. “The Multiplicative Weights Update Method: a Meta-Algorithm and Applications”. In: *Theory of Computing* 8.1 (2012), pp. 121–164

Multiplicatice weight update algorithm

1. **Guess initial weighting**, e.g., $p_i = 1/m$
2. **Solve** $p^T Ax \geq p^T b$
3. Compute **error signal**

$$e_j \leftarrow 1/\rho \left(\sum_{i \in N(j)} A_{ji} \mathbf{x}_i \right) - \mathbf{b}_j$$

4. **Update** $w_i \leftarrow (1 - \eta e_j) w_i$
5. **Normalize weights** $p_j \leftarrow w_j / \Gamma(t)$ for i in $[m]$, where $\Gamma(t) = \sum_{i \in [m]} \mathbf{w}_i$

S. Arora, E. Hazan, and S. Kale. “The Multiplicative Weights Update Method: a Meta-Algorithm and Applications”. In: *Theory of Computing* 8.1 (2012), pp. 121–164

Multiplicatice weight update algorithm

1. **Guess initial weighting**, e.g., $p_i = 1/m$
2. **Solve** $p^T Ax \geq p^T b$
3. Compute **error signal**
$$e_j \leftarrow 1/\rho \left(\sum_{i \in N(j)} A_{ji} \mathbf{x}_i \right) - \mathbf{b}_j$$
4. **Update** $w_i \leftarrow (1 - \eta e_j) w_i$
5. **Normalize weights** $p_j \leftarrow w_j / \Gamma(t)$ for i in $[m]$, where $\Gamma(t) = \sum_{i \in [m]} \mathbf{w}_i$
6. **Update solution** $\bar{\mathbf{x}} \leftarrow \bar{\mathbf{x}} + \mathbf{x}$

S. Arora, E. Hazan, and S. Kale. “The Multiplicative Weights Update Method: a Meta-Algorithm and Applications”. In: *Theory of Computing* 8.1 (2012), pp. 121–164

Multiplicatice weight update algorithm

Theorem

Given a BIP $I = (\mathbf{A}, \mathbf{b}, \mathbf{c})$, its corresponding relaxation $\widehat{I}$, and $\varepsilon > 0$, the MWU with

$$T = \left\lceil \frac{4\rho \ln(m)}{\varepsilon^2} \right\rceil,$$

outputs an ε-feasible solution for $\widehat{I}$ or determines that $\widehat{I}$ is not feasible.

S. Arora, E. Hazan, and S. Kale. "The Multiplicative Weights Update Method: a Meta-Algorithm and Applications". In: *Theory of Computing* 8.1 (2012), pp. 121–164

GNNs and the multiplicatice weight update algorithm

Theorem (Informal)

There exists a GNN architecture and corresponding weight assignments such it can simulate the MWU algorithm.

E. Khalil, C. Morris, and A. Lodi. “MIP-GNN: A data-driven framework for guiding combinatorial solvers”. In: *AAAI*. 2022

Can **GNNs simulate** *more nature* **LP solver?**

Conclusion

1. **MIP-GNN framework for data-driven optimization**
2. **Limitations of GNNs**
3. **Preliminary results to better understand capabilites of GNNs for data-driven optimization**

References i

S. Arora, E. Hazan, and S. Kale. “The Multiplicative Weights Update Method: a Meta-Algorithm and Applications”. In: *Theory of Computing* 8.1 (2012), pp. 121–164.

V. Arvind, J. Köbler, G. Rattan, and O. Verbitsky. “On the Power of Color Refinement”. In: *International Symposium on Fundamentals of Computation Theory*. 2015, pp. 339–350.

Q. Cappart, D. Chételat, E. Khalil, A. Lodi, C. Morris, and P. Velickovic. “Combinatorial optimization and reasoning with graph neural networks”. In: *International Joint Conference on Artificial Intelligence*. IJCAI, 2021.

M. Gasse, D. Chételat, N. Ferroni, L. Charlin, and A. Lodi. “Exact Combinatorial Optimization with Graph Convolutional Neural Networks”. In: *NeurIPS*. 2019.

References ii

E. Khalil, C. Morris, and A. Lodi. “MIP-GNN: A data-driven framework for guiding combinatorial solvers”. In: *AAAI*. 2022.

C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. Eric Lenssen, G. Rattan, and M. Grohe. “Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks”. In: *AAAI*. 2019.

C. Morris, G. Rattan, and P. Mutzel. “Weisfeiler and Leman Go Sparse: Higher-order Graph Embeddings”. In: *NeurIPS*. 2020.

B. Weisfeiler. *On Construction and Identification of Graphs*. Lecture Notes in Mathematics, Vol. 558. Springer, 1976.

K. Xu, W. Hu, J. Leskovec, and S. Jegelka. “How Powerful are Graph Neural Networks?” In: *ICLR*. 2019.