Machine Learning for Combinatorial Optimization

Lara Scavuzzo

Outline

Part I

- A motivating example
- Why Machine Learning?
- Integration of Machine Learning and optimization

Part II

- MIPs and MIP solvers
- Enhancing MIP solvers with Machine Learning

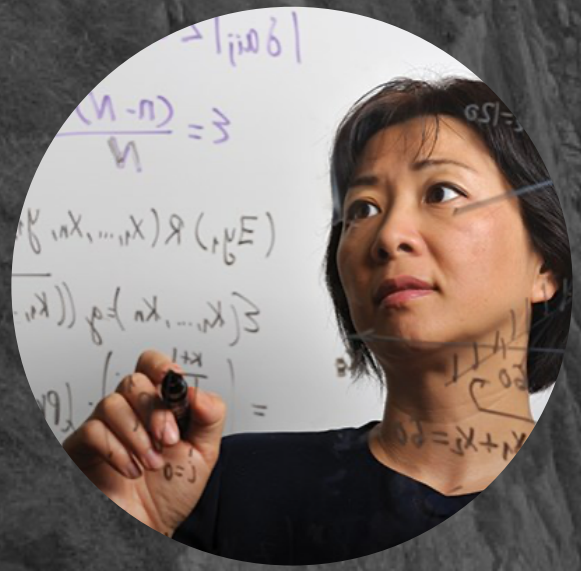
Machine learning for combinatorial optimization: A methodological tour d'horizon

Yoshua Bengio[c,b], Andrea Lodi[a,b,*], Antoine Prouvost[a,b]

[a] *Canada Excellence Research Chair in Data Science for Decision Making, École Polytechnique de Montréal, Pavillon André-Aisenstadt 2920, Chemin de la Tour Montreal, Qc, H3T 1J4 Canada*
[b] *Mila, Institut Québecois d'Intelligence Artificielle, Pavillon André-Aisenstadt 2920, Chemin de la Tour Montreal, Qc, H3T 1J4 Canada*
[c] *Université de Montréal, Département d'Informatique et de Recherche Opérationelle, Pavillon André-Aisenstadt 2920, Chemin de la Tour Montreal, Qc, H3T 1J4 Canada*

A motivating example

Textbook knowledge

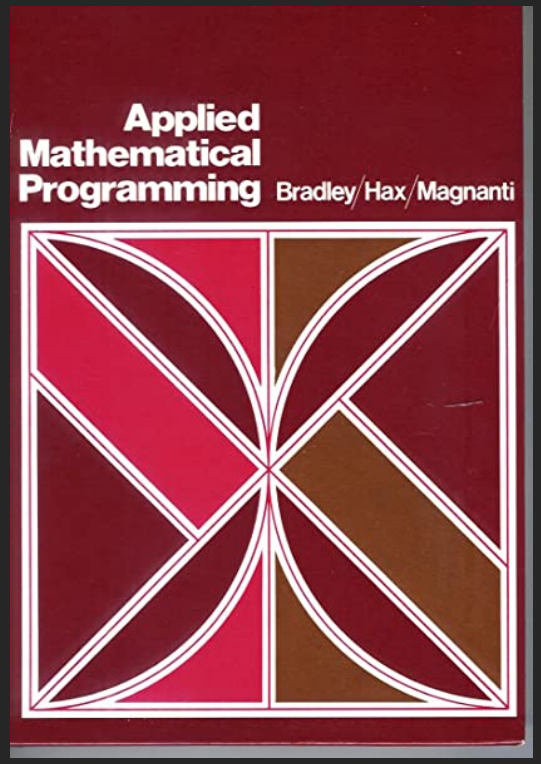


Implementation



The simplex algorithm

minimise	$c^T x$
subject to	$Ax = b$
	$x \geq 0$



Input: A feasible basis B and vectors
$X_B = A_B^{-1} b$ and $D_N = c_N - A_N^T A_N^{-T} c_B$

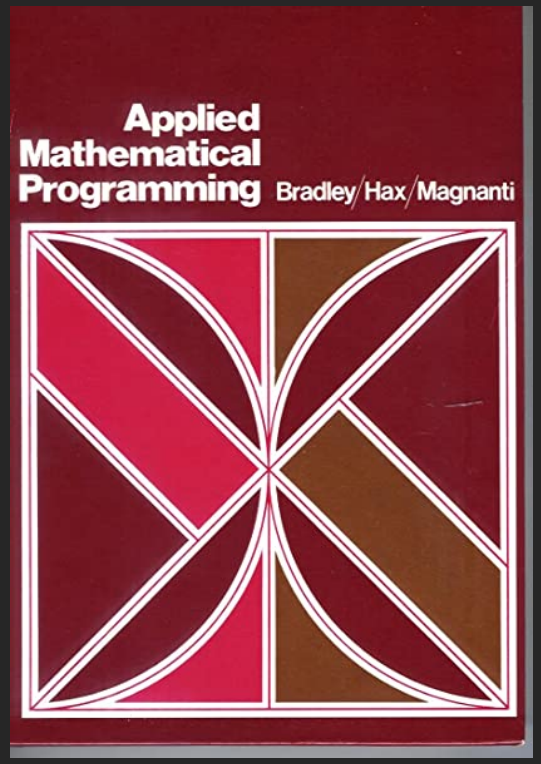
- Step 1 (pricing) — What goes in
  If $D_N \geq 0$, stop, B is optimal; else let $j = \underset{k \in N}{\operatorname{argmin}} D_k$.

- Step 2 (FTRAN) — What goes out
  Solve $A_B y = A_j$.
- Step 3 (Ratio test) if $y \leq 0$, stop, the problem is unbounded; else, let $\mathrm{i} = \underset{y_k > 0}{\operatorname{argmin}} X_{Bk}/y_k$.

- Step 4 (BTRAN) — Update
  Solve $A_B^T z = e_i$.
- Step 4 (Update)
  Compute $\alpha_N = -A_N^T$. Let $B_i = j$. Update X_B (using y) and. D_N (using α_N).

The simplex algorithm

minimise	$c^T x$
subject to	$Ax = b$
	$x \geq 0$



Input: A feasible basis B and vectors
$X_B = A_B^{-1} b$ and $D_N = c_N - A_N^T A_N^{-T} c_B$

BAD IDEA!!

- Step 1 (pricing)
  If $D_N \geq 0$, stop, B is optimal; else let $j = \underset{k \in N}{\operatorname{argmin}} D_k$.
- Step 2 (FTRAN)
  Solve $A_B y = A_j$.
- Step 3 (Ratio test) if $y \leq 0$, stop, the problem is unbounded; else, let $\mathrm{i} = \underset{y_k > 0}{\operatorname{argmin}} X_{Bk}/y_k$.
- Step 4 (BTRAN)
  Solve $A_B^T z = e_i$.
- Step 4 (Update)
  Compute $\alpha_N = -A_N^T$. Let $B_i = j$. Update X_B (using y) and. D_N (using α_N).

Pricing rules for simplex

- Textbook rule is terrible.
- Use **steepest-edge rule** [Goldfard and Forrest 1992] (on dual simplex).

  Example: dfl001[*]

Pricing rule	# iterations	Time (s)
Textbook	281,829	118.68
Steepest-edge	18,412	5.36

[*] Lectures by Robert Bixby at CO@Work 2015

https://www.youtube.com/watch?v=uccbVoamiUM&t=2032s

Takeaway

- Textbook definition hides great body of knowledge gained to make this one (seemingly) simple **decision within the algorithm**.
- Knowledge gained (learned?) through **computational tests**.
- **Good decision-making strategies have critical impact.**
- Optimization solvers are riddled with this type of **decision-making tasks**, e.g. in …

  … certain algorithmic steps.

  … parameter configuration.

  … tie-breaking.

Algorithmic decision-making

1

Good strategy
Efficient strategy



2

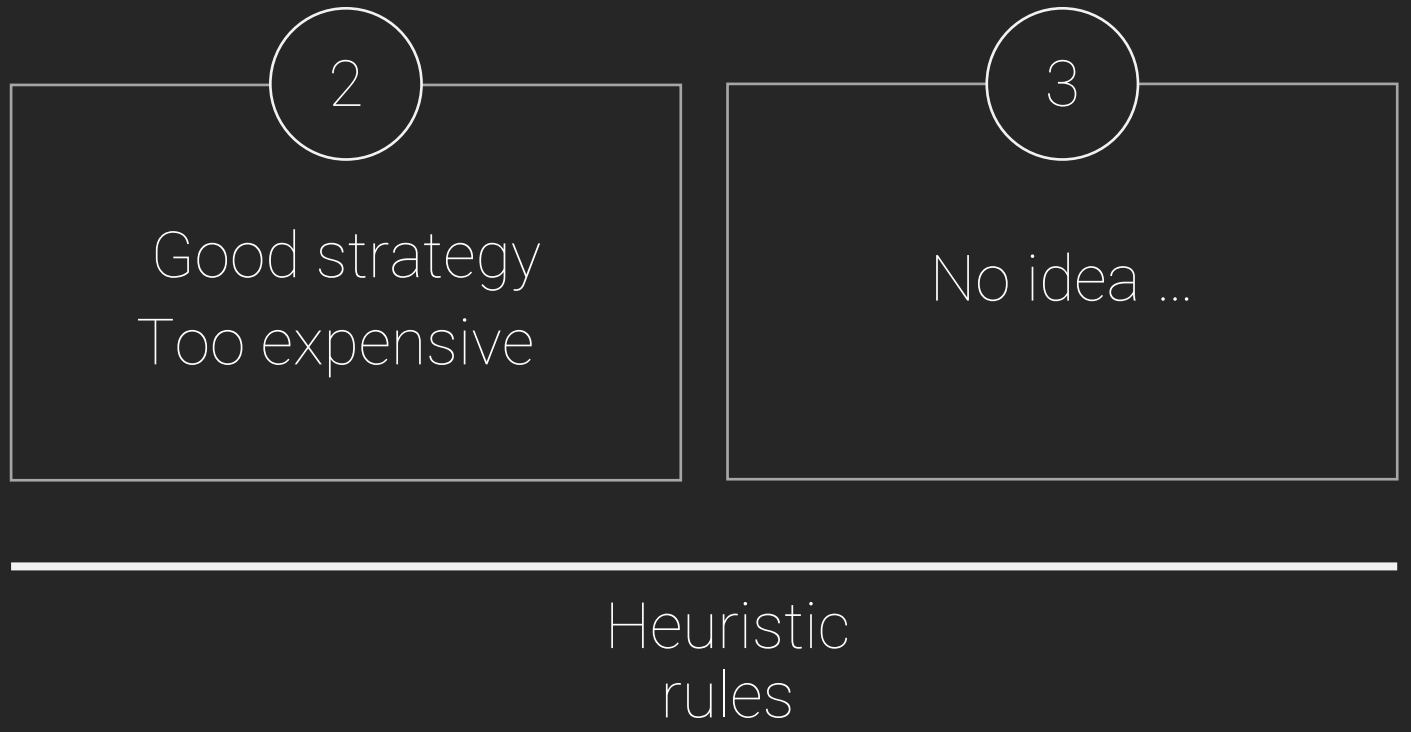
Good strategy
Too expensive

3

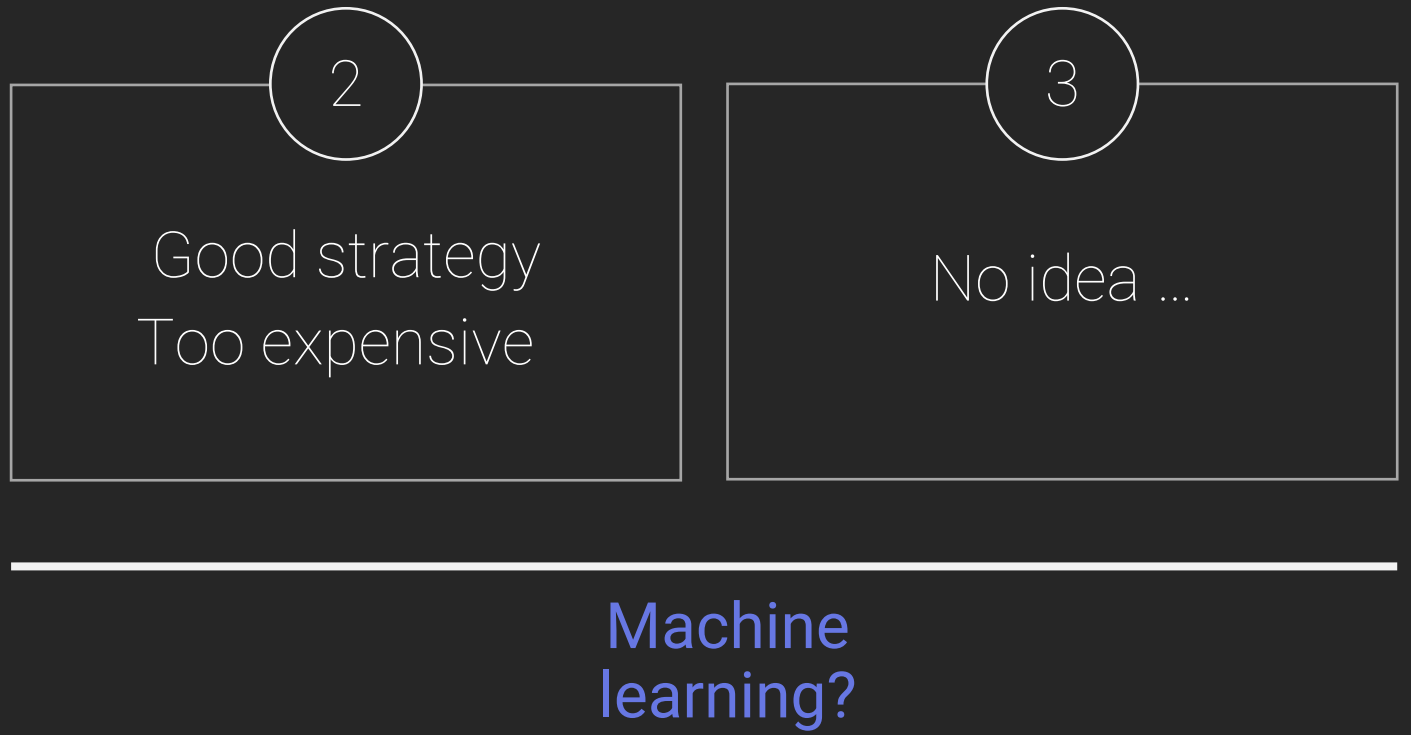
No idea ...

Heuristic
rules

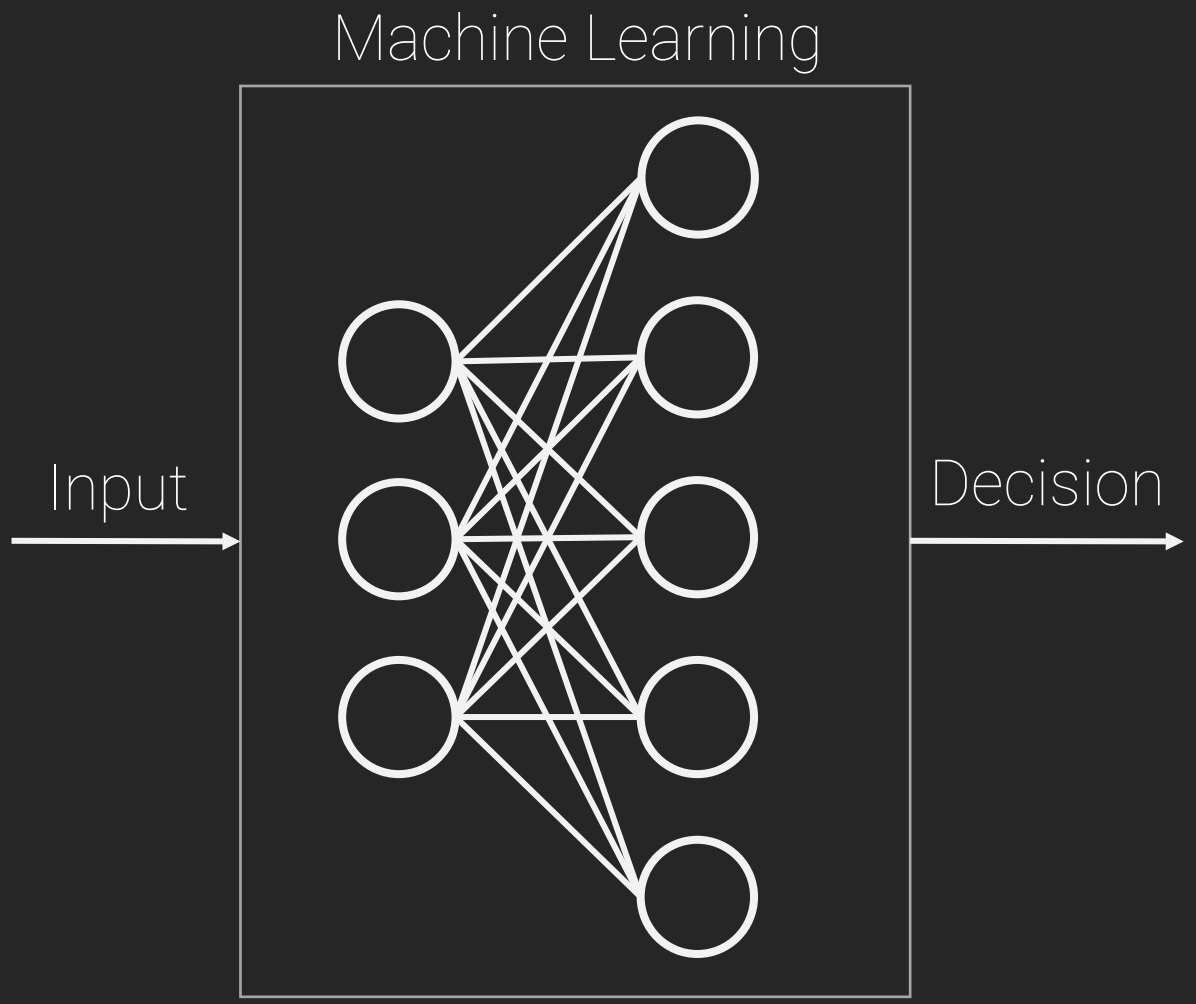
Algorithmic decision-making



Algorithmic decision-making



Machine learning for decision-making



- Fast approximations
- Automated rule learning.
- Adaptiveness
  - Instance (through more data)
  - Instance distribution

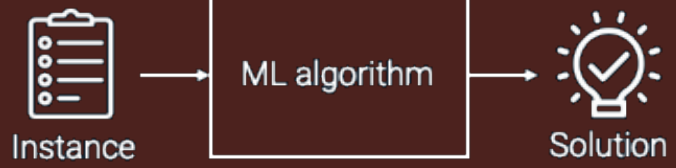
Industry setting: repeatedly solving similar instances

Pure Machine Learning

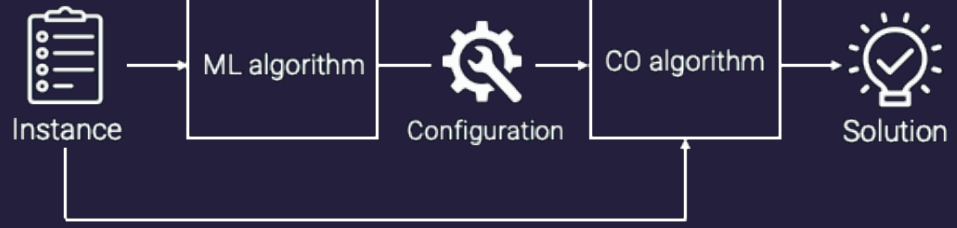
Synergy

Classic optimization

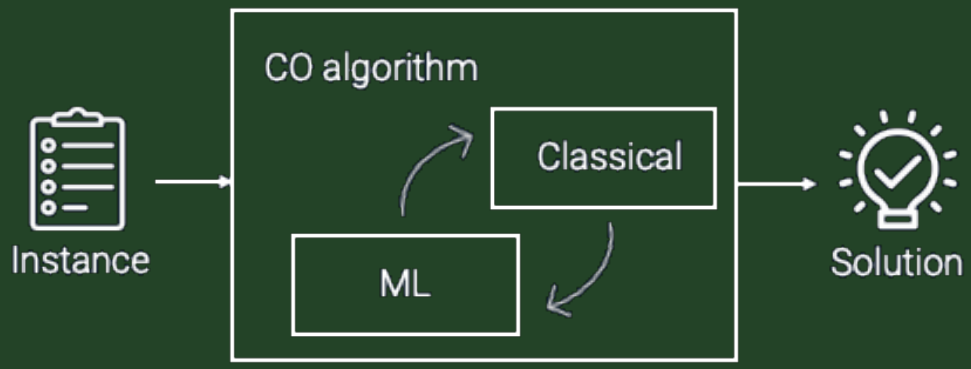
End-to-end



Tuning



Sequential



Guaranteed:

- Feasibility
- Optimality

Outline

Part I

- A motivating example
- Why Machine Learning?
- Integration of Machine Learning and optimization

Part II

- MIPs and MIP solvers
- Enhancing MIP solvers with Machine Learning

Machine Learning for Mixed Integer Programming

Mixed Integer Programming

minimise $c^T x$

subject to $Ax \leq b$

Linear

$x \in \mathbb{R}^n$

$x_i \in \mathbb{Z}, \forall i \in I \subseteq \{1, \ldots, n\}$

Mixed Integer Programming

Mixed Integer Quadratic Programming

$$\frac{1}{2}x^T Q x + c^T x$$

minimise ~~$c^T x$~~

subject to $\mathbf{A}x \leq b$

$x \in \mathbb{R}^n$

$x_i \in \mathbb{Z}, \forall i \in I \subseteq \{1, \ldots, n\}$

Mixed Integer Programming

Mixed Integer Non-Linear Programming

$$g_k(x) \leq 0, \quad k = 1, \ldots, m$$

minimise $c^T x$

subject to ~~$Ax \leq b$~~

$x \in \mathbb{R}^n$

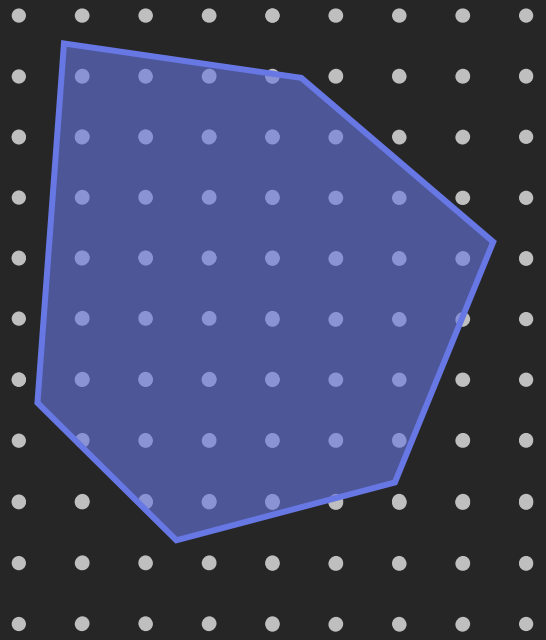
$x_i \in \mathbb{Z}, \forall i \in I \subseteq \{1, \ldots, n\}$

Mixed Integer Programming

$$
\begin{aligned}
&\textit{minimise} && \boldsymbol{c}^T\boldsymbol{x} \\
&\textit{subject to} && \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b} \\
& && \boldsymbol{x} \in \mathbb{R}^n \\
& && x_i \in \mathbb{Z}, \forall i \in I \subseteq \{1, \ldots, n\}
\end{aligned}
$$

Branch and bound

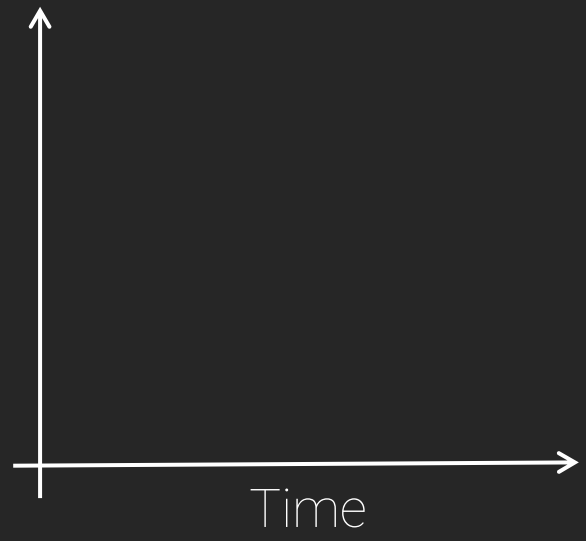
Polyhedral view



Tree view

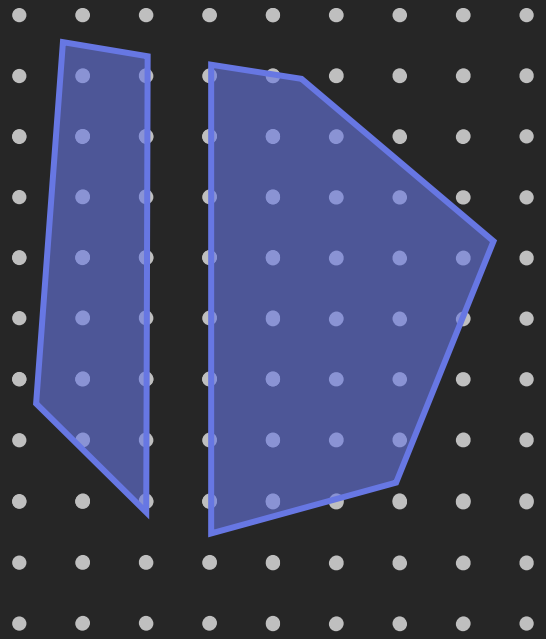


Bound view

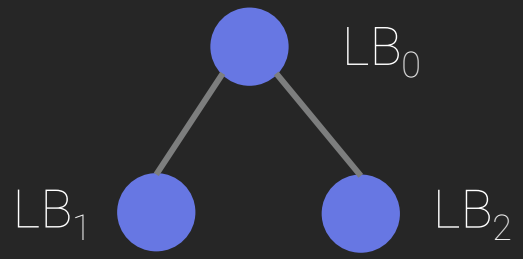


Branch and bound

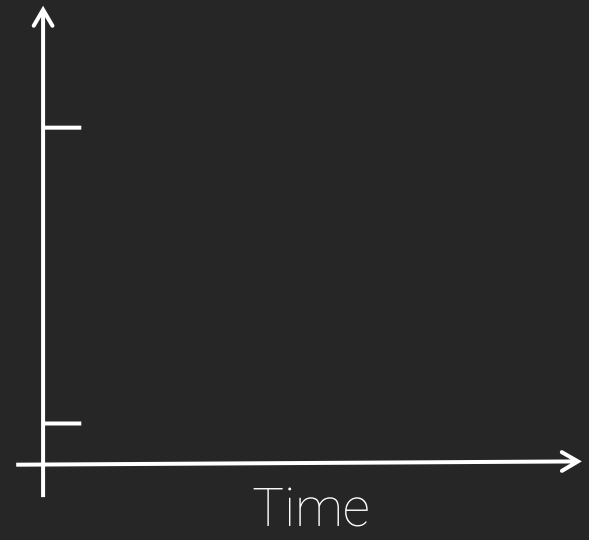
Polyhedral view



Tree view

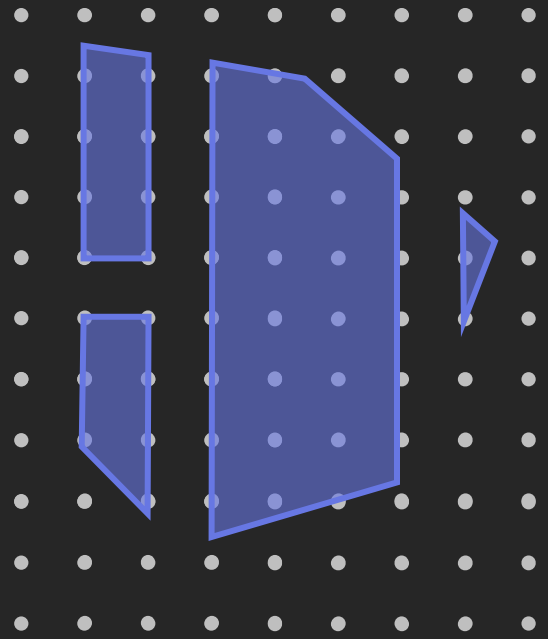


Bound view

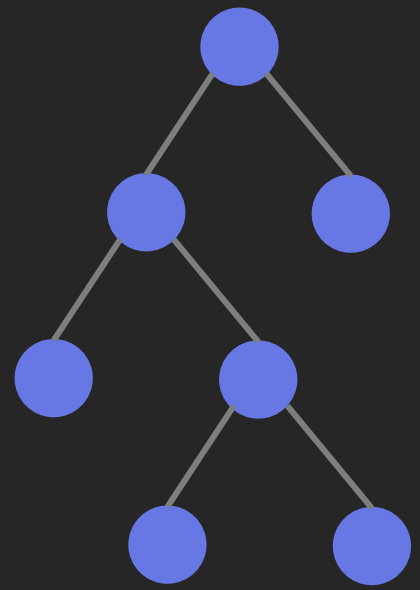


Branch and bound

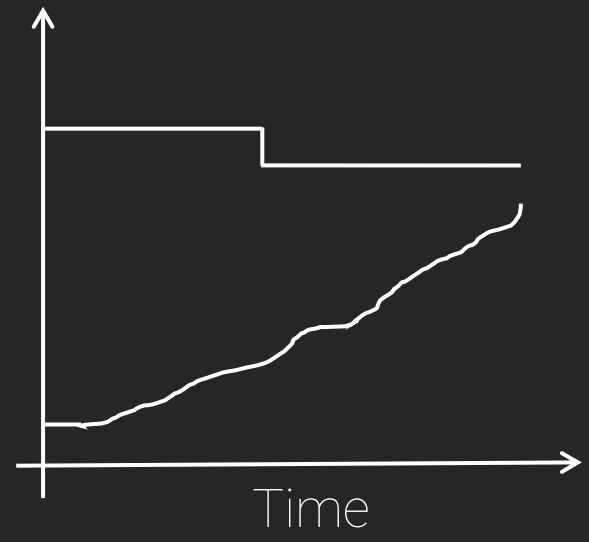
Polyhedral view



Tree view



Bound view



The MIP solver

Pre-processing
[Bonami et al. 2018]
[Berthold et al. 2020]

CPLEX ≥12.10

Fico Xpress ≥ 8.9

Primal side

Dual side

Configuration
[Kruber et al. 2017]
[Hendel et al. 2020]

SCIP ≥ 7.0

Primal heuristics
[Khalil et al. 2017]
[Ding et al. 2020]
[Nair et al. 2020]

Node selector
[He et al. 2014]

Brancher
[Khalil et al. 2014]
[Gasse et al. 2019]
[Etheve et al. 2020]
[Nair et al. 2020]

Cutting planes
[Tang et al. 2019]

The Machine Learning recipe

1. How do we formulate this as a learning task?
2. How do we represent the problem?
3. How do we learn?

The Machine Learning recipe

1. How do we formulate this as a learning task?
2. How do we represent the problem?
3. How do we learn?

The MIP solver

Pre-processing
[Bonami et al. 2018]
[Berthold et al. 2020]

Configuration
[Kruber et al. 2017]
[Hendel et al. 2020]

Primal side

Dual side

Primal heuristics
[Khalil et al. 2017]
[Ding et al. 2020]
[Nair et al. 2020]

Node selector
[He et al. 2014]

Brancher
[Khalil et al. 2014]
[Gasse et al. 2019]
[Etheve et al. 2020]
[Nair et al. 2020]

Cutting planes
[Tang et al. 2019]

How to formulate this as a learning task?

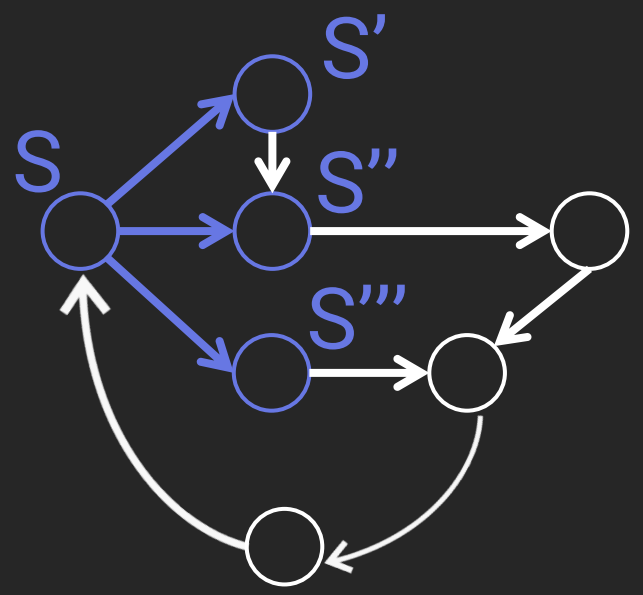
Markov Decision Process

- Agent
- Environment

How to formulate this as a learning task?

Markov Decision Process

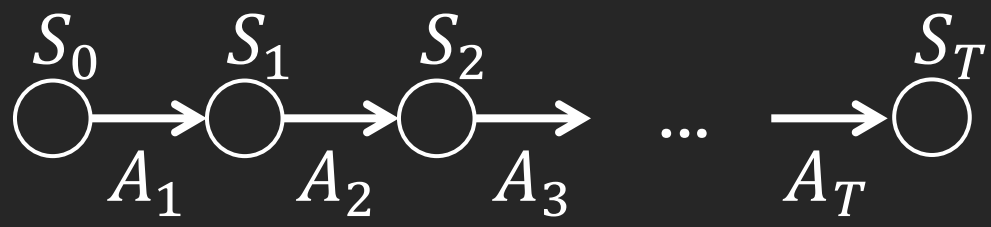
- Agent
- Environment



How to formulate this as a learning task?

Markov Decision Process

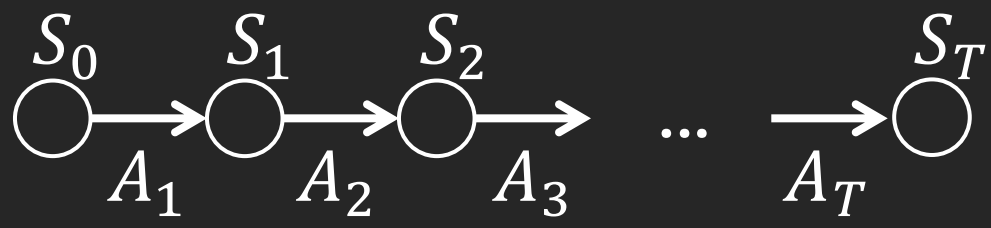
- Agent
- Environment
- Trajectory



How to formulate this as a learning task?

Markov Decision Process

- Agent
- Environment
- Trajectory
- Markov property



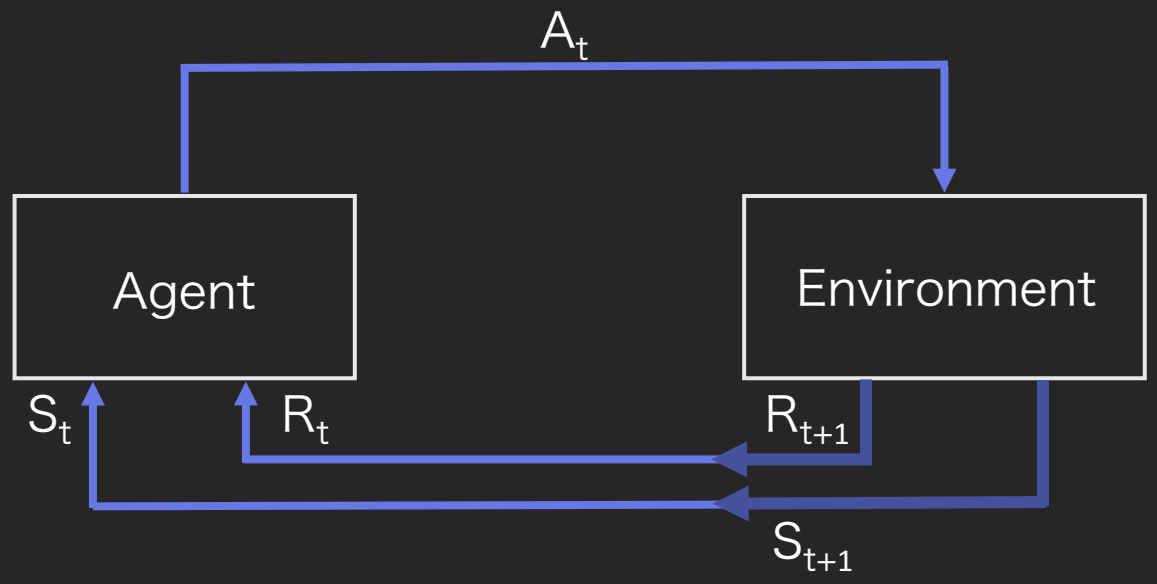
$$P[S_{t+1}|S_t] = P[S_{t+1}|S_t, S_{t-1}, \dots, S_0]$$

How to formulate this as a learning task?

Markov Decision Process

- Agent $a \sim \pi_\theta(s)$
- Environment
- Trajectory τ
- Markov property

BASIC INGREDIENTS



The Machine Learning recipe

1. How do we formulate this as a learning task?
2. How do we represent the problem?
3. How do we learn?

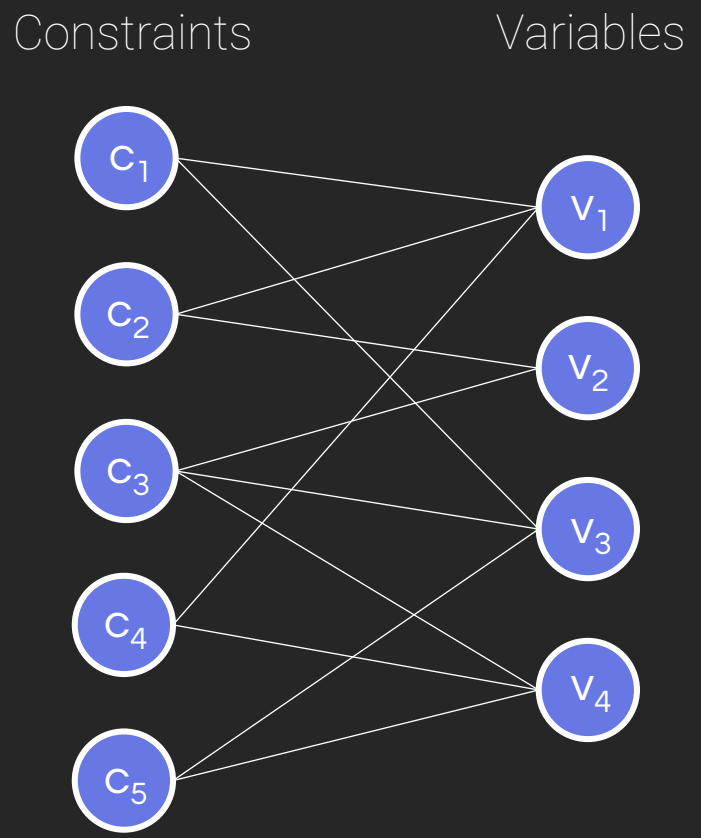
Problem representation

Features

- Scale independent.
- Instance size independent.

Bipartite graph representation of MIPs*

- Use Graph Convolutional Neural Network.
- Works on graphs of any size.
- Automatically encodes constraint-variable interactions.



*[Gasse et al. 2019]

The Machine Learning recipe

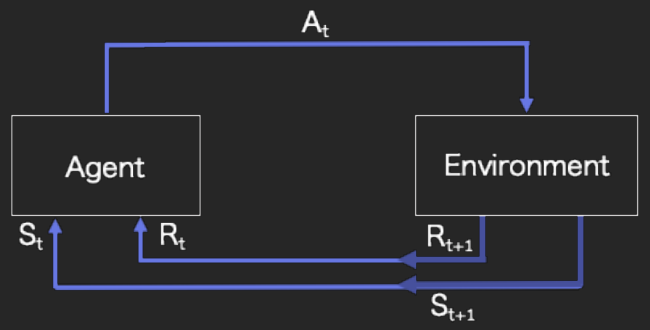
1. How do we formulate this as a learning task?
2. How do we represent the problem?
3. How do we learn?

The Machine Learning recipe

1. How do we formulate this as a learning task?
2. How do we represent the problem?
3. How do we learn?

How to learn?

Supervision vs. exploration



SUPERVISION

- Imitation learning
- Expert (oracle)
- Learning samples:

$$\{ (S_i, A_i^*) \}_{i=1}^N$$

- Objective:

$$\theta = \underset{\theta}{\operatorname{argmin}} \, \mathbb{E}_{(s,a^*)} [\, l(a^*, \pi_\theta(s)) \,]$$

EXPLORATION

- Reinforcement learning
- Return (long-term reward)

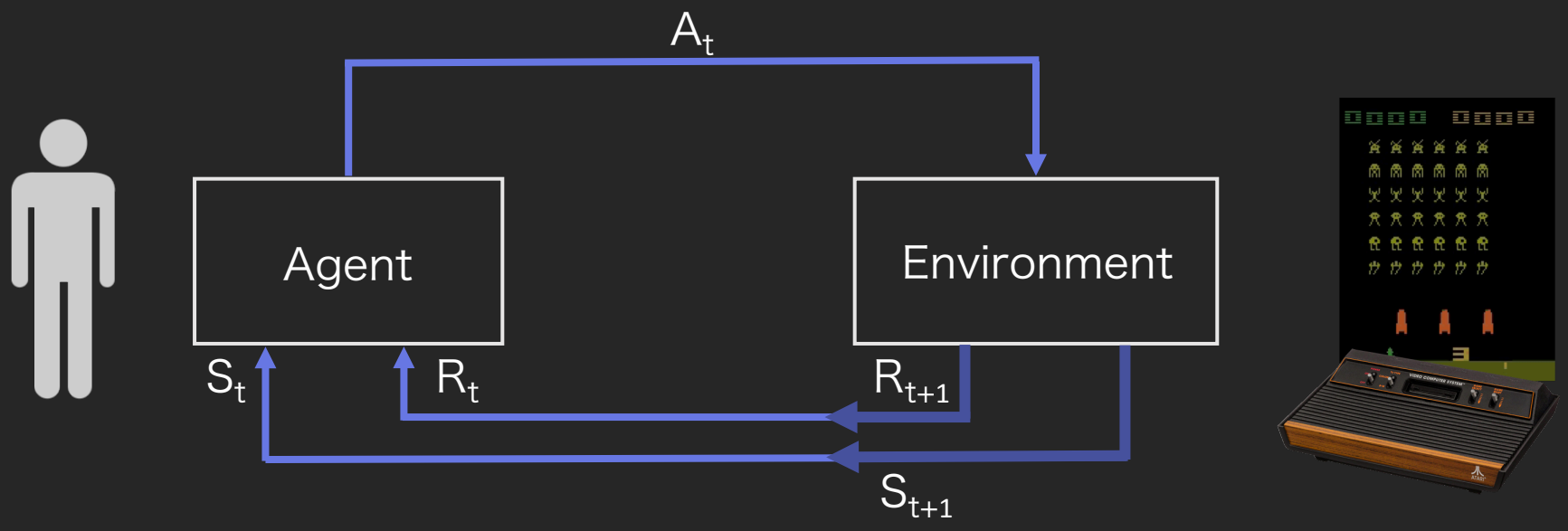
$$G_t = \sum_k \gamma^k R_{t+1+k} \,, \qquad \gamma \in [0,1]$$

- Learning samples: τ
- Objective:

$$\theta = \underset{\theta}{\operatorname{argmax}} \, \mathbb{E}_\tau [G_t]$$

Learning through exploration

An example: Atari games



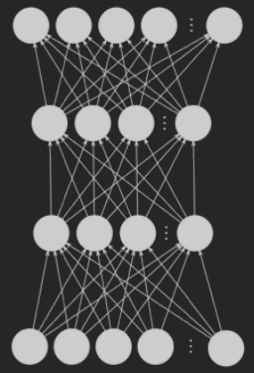
A_t :

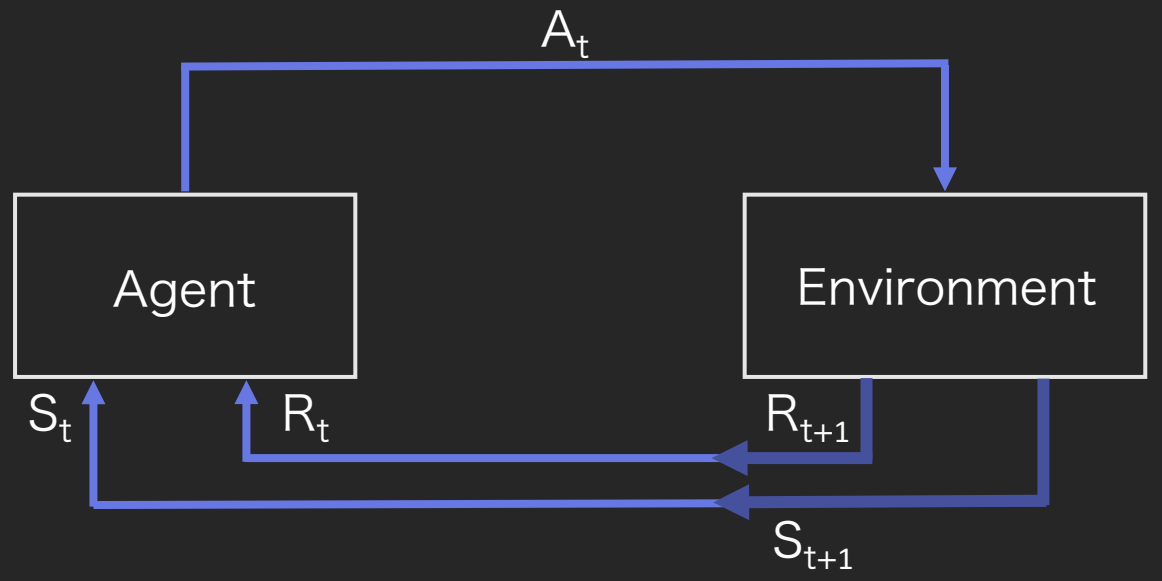
S_t :

R_t :

Learning through exploration

An example: branching





A_t : Branching variable

S_t : Solver statistics

R_t : Reduction in bound gap

How to learn?

Supervision vs. exploration

SUPERVISION

- Imitation learning
- Expert (oracle)
- Learning samples:

$$\{ (S_i, A_i^*) \}_{i=1}^N$$

- Objective:

$$\theta = \underset{\theta}{\operatorname{argmin}}\, \mathbb{E}_{(s,a^*)}\left[l\left(a^*, \pi_\theta(s)\right) \right]$$

EXPLORATION

- Reinforcement learning
- Return (long-term reward)

$$G_t = \sum_k \gamma^k R_{t+1+k} \,, \qquad \gamma \in [0,1]$$

- Learning samples: τ
- Objective:

$$\theta = \underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_\tau[G_t]$$

The Machine Learning recipe

1. How do we formulate this as a learning task?
2. How do we represent the problem?
3. How do we learn?

Takeaways

- There is great potential in the **collaboration** between Machine Learning and Optimization algorithms.
- Specially in terms of **learning decision-making strategies**.
- In the case of **MIPs** this potential is starting to manifest.
- We are **gathering general know-how** on how to foster these opportunities.

Machine Learning for Combinatorial Optimization

Lara Scavuzzo

References

[Bello et al. 2016]
Bello, I., Pham, H., Le, Q. V., Norouzi, M., & Bengio, S. (2016). Neural combinatorial optimization with reinforcement learning. arXiv preprint arXiv:1611.09940.

[Bonami et al. 2018]
Bonami, P., Lodi, A., & Zarpellon, G. (2018, June). Learning a classification of mixed-integer quadratic programming problems. In International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research(pp. 595-604). Springer, Cham.

[Berthold et al. 2020]
Berthold, T., & Hendel, G. (2020). Learning To Scale Mixed-Integer Programs.

[Gasse et al. 2019]
Gasse, M., Chételat, D., Ferroni, N., Charlin, L., & Lodi, A. (2019). Exact combinatorial optimization with graph convolutional neural networks. In Advances in Neural Information Processing Systems (pp. 15580-15592).

[He et al. 2014]
He, H., Daume III, H., & Eisner, J. M. (2014). Learning to search in branch and bound algorithms. In Advances in neural information processing systems (pp. 3293-3301).

[Hendel et al. 2020]
Hendel, G., Anderson, D., Le Bodic, P., & Pfetsch, M. E. (2020). Estimating the Size of Branch-and-Bound Trees.

References

[Khalil et al. 2016]
Khalil, E. B., Le Bodic, P., Song, L., Nemhauser, G., & Dilkina, B. (2016, February). Learning to branch in mixed integer programming. In Thirtieth AAAI Conference on Artificial Intelligence.

[Khalil et al. 2017]
Khalil, E. B., Dilkina, B., Nemhauser, G. L., Ahmed, S., & Shao, Y. (2017, August). Learning to Run Heuristics in Tree Search. In IJCAI (pp. 659-666).

[Kool et al. 2018]
Kool, W., Van Hoof, H., & Welling, M. (2018). Attention, learn to solve routing problems!. arXiv preprint arXiv:1803.08475.

[Kruber et al. 2017]
Kruber, M., Lübbecke, M. E., & Parmentier, A. (2017). Learning When to Use a Decomposition. In Integration of AI and OR techniques in constraint program- ming . In Lecture Notes in Computer Science (pp. 202–210). Springer, Cham.

[Larsen et al. 2018]
Larsen, E., Lachapelle, S., Bengio, Y., Frejinger, E., Lacoste-Julien, S., Lodi, A. (2018). arXiv preprint, arXiv:1807.11876.

[Nair et al. 2020]
Nair, V., Bartunov, S., Gimeno, F., von Glehn, I., Lichocki, P., Lobov, I., ... & Zwols, Y. (2020). Solving Mixed Integer Programs Using Neural Networks. arXiv preprint arXiv:2012.13349.

References

[Nazari et al. 2018]
Nazari, M., Oroojlooy, A., Snyder, L., & Takác, M. (2018). Reinforcement learning for solving the vehicle routing problem. In Advances in Neural Information Processing Systems (pp. 9839-9849).

[Tang et al. 2019]
Tang, Y., Agrawal, S., & Faenza, Y. (2019). Reinforcement learning for integer programming: Learning to cut. arXiv preprint arXiv:1906.04859.

[Vinyals et al. 2015]
Vinyals, O., Fortunato, M., & Jaitly, N. (2015). Pointer networks. In Advances in neural information processing systems (pp. 2692-2700).

[Vinyals et al. 2019]
Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., ... & Oh, J. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature, 575(7782), 350-354.