# Learning to Control Traffic

Jeroen van Riel

November 2023

# Learning problem

- set of problem instances $\mathcal{I}$
- distribution $P$ over instances
- set of algorithms $\mathcal{A}$
- measure of optimality $m : \mathcal{I} \times \mathcal{A} \to \mathbb{R}$

based on [Bengio et al., 2020]

# Learning problem

- general learning objective

$$\min_{a \in \mathcal{A}} \mathbb{E}_{i \sim P}\ m(i, a) \tag{1}$$

- no access to $\mathcal{I}$ or $P$, so use samples

$$\min_{a \in \mathcal{A}} \sum_{i \in D_{train}} \frac{1}{|D_{train}|} m(i, a) \tag{2}$$

# Learning problem

- demonstration
- experience

# Demonstration

- parameterization of algorithms, e.g., by using neural network with weights $\theta \in \mathbb{R}^p$

$$\min_{\theta \in R^p} \mathbb{E}_{i \sim P} m(i, a(\theta)) \tag{3}$$

# Experience

- greedy TSP heuristic = picking next node

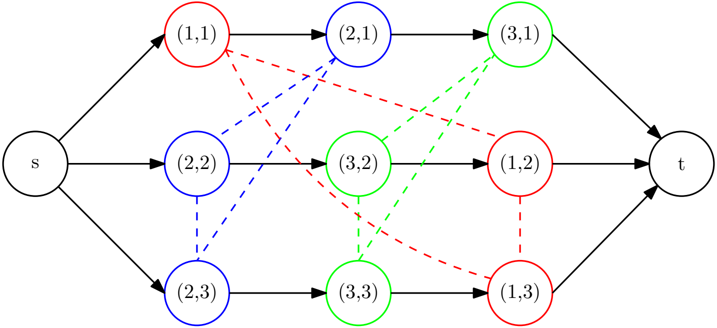
# Learning to cut (example)

# Job shop

- $m$ machines
- $n$ jobs
- fixed machine order for each job

# Disjunctive graph

- directed graph $G = (N, \mathcal{C}, \mathcal{D})$
- conjunctive arcs
- disjunctive arcs



# Job shop MILP

- makespan objective
- mixed-integer linear program

$$
\begin{aligned}
&\text{minimize } C_{\max} \\
&y_{ij} + p_{ij} \le y_{kj} && \text{for all } (i,j) \to (k,j) \in \mathcal{C} \\
&y_{il} + p_{il} \le y_{ij} \text{ or } y_{ij} + p_{ij} \le y_{il} && \text{for all } (i,l) \text{ and } (i,j), i = 1, \ldots, m \\
&y_{ij} + p_{ij} \le C_{\max} && \text{for all } (i,j) \in N \\
&y_{ij} \ge 0 && \text{for all } (i,j) \in N
\end{aligned}
$$

# Traffic scheduling problem

- total completion time
- release dates
- chains
- setup times (switch-over)

# References

Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine Learning for Combinatorial Optimization: A Methodological Tour d'Horizon, March 2020.