Predicting Solution Summaries to ILPs under Imperfect Information with Machine Learning

Eric Larsen[1] Sébastien Lachapelle[2] Yoshua Bengio[2] Emma Frejinger[1] Simon Lacoste-Julien[2] Andrea Lodi[3]

[1]CIRRELT, Université de Montréal [2]Mila, Université de Montréal [3]Canada Excellence Research Chair, École Polytechnique de Montréal

Overview

Summary

- **Motivation:** Solving *integer linear programs* (ILPs) when
  - The computational budget is restricted
  - A subset of the problem characteristics may be unknown
  - The application at hand may not require a fully detailed solution
- **Idea:** Using a supervised machine learning (ML) algorithm to learn a *prediction function* that maps problems to *solution summaries*.
- **Application:** Booking decisions for the load planning problem (LPP)

Related Work

- ML to predict objective value at optimality [1]
- ML to predict solutions to combinatorial optimization problems [4]

Contributions

- Use of ML to address computational budget limits
- Use of ML to predict solution summaries to ILPs
- Use of ML to address stochasticity in the context of OR

Methodology Outline

Data Generation

Problem distribution ⇓ $\{x_1, x_2, ..., x_n\}$ ⇒ Solver ⇒ $\{y_1, y_2, ..., y_n\}$

$\{x_1, x_2, ..., x_n\}$ ⇓ Remove information that will be unavailable at prediction time ⇓ $\{\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n\}$

$\{y_1, y_2, ..., y_n\}$ ⇓ Produce *solution summaries* ⇓ $\{\tilde{y}_1, \tilde{y}_2, ..., \tilde{y}_n\}$

$\{(\tilde{x}_1, \tilde{y}_1), (\tilde{x}_2, \tilde{y}_2), ..., (\tilde{x}_n, \tilde{y}_n)\}$

Machine Learning

Supervised learning ⇒ Prediction function f

Deployment

Use f in a setting where the *problem information is limited to* $\tilde{x}$ and a *fast prediction of* $\tilde{y}$ is required

An application: The Load Planning Problem (LPP)

In short

- A set of containers and a set of railcars are given
- Aim to load highest number of containers on smallest number of railcars

Solving the LPP

The problem can be cast as an *integer linear program*

The deterministic version can be solved using a commercial solver (see [3])

Constraints

Constraints are related to:

- container/railcar sizes and types
- **container weights**
- railcar weight capacities
- center of gravity

Why is the methodology useful?

- We want to get a solution summary quickly at **booking time**, i.e. when the train reservation is done for the containers
- **Container weights are unknown** at that moment

Experimentation with the LPP

Problems under Imperfect Information

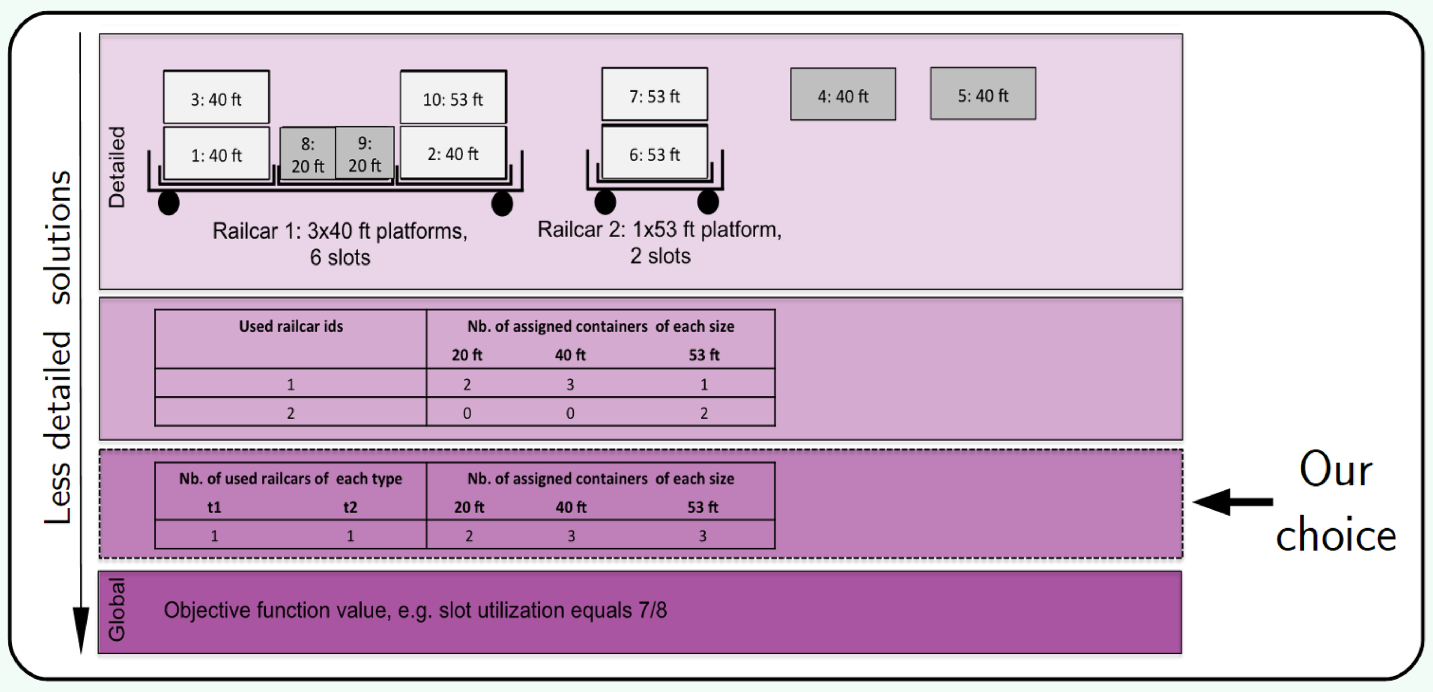
Let $\tilde{x}$ be a partially specified LPP instance.
The container weight information is not encoded.

e.g. $\tilde{x} =$

Nb. of available railcars of each type		Nb. of containers to load of each size		
t1	t2	20 ft	40 ft	53 ft
1	1	2	5	3

Solution Summary Levels

Let $\tilde{y}$ be a summary of the solution to x.
A solution can be summarized in different ways:



Four Data Classes

Class name	Description	# of containers	# of platforms
A	Simple ILP instances	[1, 150]	[1, 50]
B	More containers than A (excess demand)	[151, 300]	[1, 50]
C	More platforms than A (excess supply)	[1, 150]	[51, 100]
D	Larger and harder instances	[151, 300]	[51, 100]

Machine Learning Details

- Prediction function f is a *multilayer perceptron* (MLP)
- ≈ 7 hidden layers of ≈ 500 units with ReLU activations
- Two approaches: regression (RegMLP) and classification (ClassMLP)
- Models were trained on GPUs for 2 to 10 hours
- Hyperparameter tuning: *early stopping* and *random search*

Performance Metric

We evaluate our models with the *mean absolute error* (MAE) measured in containers and/or railcar slots.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{12}|f(\tilde{x}_j^{(i)}) - \tilde{y}_j^{(i)}|s_j$$

Legend:

i : Training example index
j : Container/railcar index
s_j: Number of slots on railcar type j for $j = 1, ..., 10$ $(s_{11} = s_{12} = 1)$

Empirical Results

Training-validation data	A	A	ABC	A	ABC
# examples	200K	20M	600K	20M	600K
Testing data	A	A	ABC	D	D
ClassMLP	1.481	0.965	2.312	NA	14.831
LogReg	5.956	5.887	9.051	NA	29.568
RegMLP	1.304	0.985	2.109	4.412	2.372
LinReg	18.306	18.372	39.907	24.560	72.847
HeurV	14.733	14.753	27.24	33.737	33.737
HeurS	17.841	17.842	31.448	43.303	43.303

MAE for different models trained, validated and tested on different data sets

- ClassMLP and RegMLP successfully predict solution summaries
- RegMLP outperforms its competitors
- RegMLP performs well on data it was never trained-validated on (yet with wide performance ranges across hyperparameter sets)

Conclusion

- We illustrated the usefulness of our methodology on the stochastic LPP
- We showed that fast solution summaries predictors can be learned via supervised learning using deep learning methods for the LPP application

Future work:

- Comparison to a stochastic programming approach
- Active Learning (to reduce the high cost of data generation)

References

[1] Fischetti, M. & Fraccaro, M. (2017). Using OR + AI to predict the optimal production of offshore wind parks: A preliminary study. In *Optimization and Decision Science: Methodologies and Applications*, volume 217 (pp. 203–211). Springer.

[2] Larsen, E., Lachapelle, S., Bengio, Y., Frejinger, E., Lacoste-Julien, S., & Lodi, A. (2018). Predicting solution summaries to integer linear programs under imperfect information with machine learning. arXiv:1807.11876.

[3] Mantovani, S., Morganti, G., Umang, N., Crainic, T. G., Frejinger, E., & Larsen, E. (2018). The load planning problem for double-stack intermodal trains. *EJOR*, (1).

[4] Vinyals, O., Fortunato, M., & Jaitly, N. (2015). Pointer Networks. In *Advances in Neural Information Processing Systems* (pp. 2692–2700).