A machine learning approximation algorithm for fast prediction of solutions to discrete optimization problems

JOPT 2018, Montreal, Canada

Eric Larsen - CIRRELT and Université de Montréal
Sébastien Lachapelle - CIRRELT and Université de Montréal
Yoshua Bengio - Montreal Institute for Learning Algorithms
Emma Frejinger - CIRRELT and Université de Montréal
Simon Lacoste-Julien - Montreal Institute for Learning Algorithms
Andrea Lodi - École Polytechnique de Montréal

OVERVIEW OF THE PRESENTATION

- Introduction: motivation & methodology
- An application
- Experimental results
- Conclusion & future work

MOTIVATION

- Want to solve **discrete optimization problems** when:
  - The computational budget is restricted
  - A subset of the problem characteristics may be unknown (which renders the problem **stochastic**)
  - The application at hand may not require a fully detailed solution

THE IDEA IN BRIEF

- We use machine learning algorithms to predict solutions
- To do *supervised learning*, we need *labeled* data
  One training example: (input x, label y)
- **Input vector** x: a description of a problem instance
- **Label vector** y: its corresponding solution

THE IDEA IN BRIEF

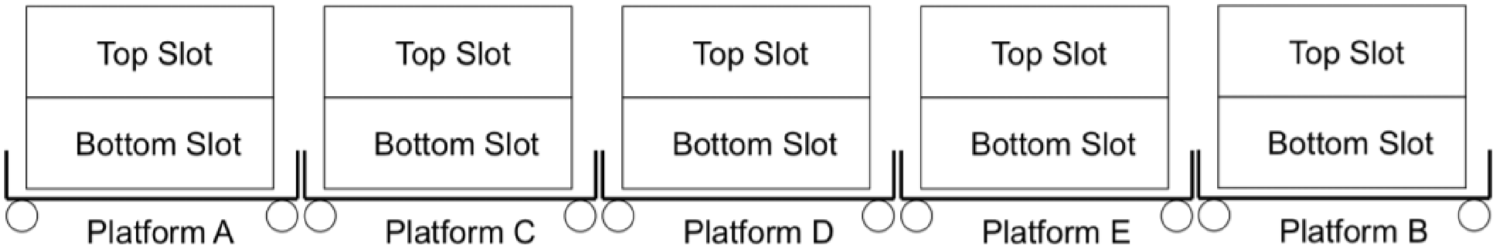
1) **Data generation:**
   - Sample many problem instances (x)
   - Use an existing solver to find their corresponding solutions (y)
2) Feed (x, y) couples to a **ML algorithm** in order to find a good mapping from x to y (*prediction function*, in our case a deep neural network)
3) Use this function for **fast prediction** in the desired application

IN THE LITERATURE

- ML as a contributor to OR, e.g. :
  - Fischetti and Fraccaro (2017): Predict objective function value at optimality in the context of offshore wind farm layout optimization problem
- ML as an alternative to OR, e.g. :
  - Vinyals et al., (2015): Supervised learning with pointer networks to solve discrete optimization problems (deterministic setting)

AN APPLICATION: LOAD PLANNING PROBLEM (LPP)

- The Load Planning Problem:
  - We have a set of containers to load on a set of railcars.
  - Each container and platform has its own characteristics (e.g. weight, size, ... etc.)
  - We must find an optimal assignment of the containers to slots on the railcars to minimize cost.



AN APPLICATION: LPP

- Many constraints related to:
  - Size of containers/railcars
  - Type of containers/railcars (e.g. some are lacking roof, some needs electricity connection)
  - **Container weights**
  - Railcars' weight capacity
  - Center of gravity

AN APPLICATION: LPP

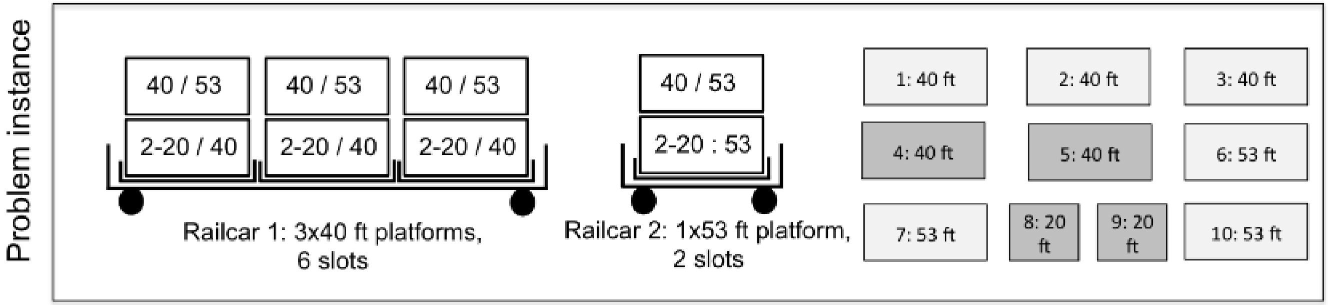
- The problem can be cast as an Integer Linear Program (ILP)
- Deterministic version can be solved using a commercial solver (see Mantovani et al. 2017)

AN APPLICATION: LPP

- We want to solve the LPP at **booking time** (containers need train reservations)
- Which means:
  - We want the computation to be quick (for real-time application)
  - We do not have all information (container weights are unknown)
  - We do not need a fully detailed description of the solution
- The methodology presented can deal with all those requirements

PROBLEM REPRESENTATION

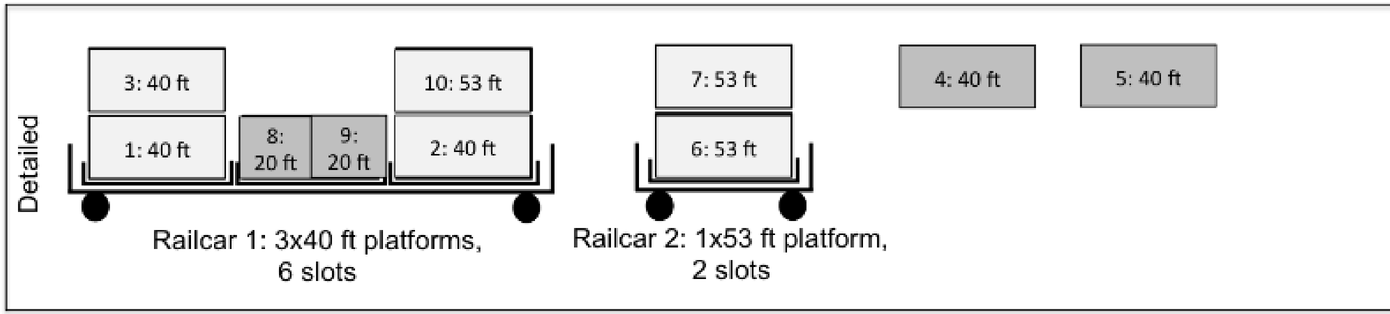
- The **problem instances** are encoded as vectors: $x \in \mathbb{N}^{12}$
- Each component corresponds to the number of railcars of each type and containers of each length available in the problem
- The container weights are not encoded



CN CHAIR IN OPTIMIZATION OF RAILWAY OPERATIONS

SOLUTION DESCRIPTION

Possible solution descriptions



Used railcar ids	Nb. of assigned containers of each size		
	20 ft	40 ft	53 ft
1	2	3	1
2	0	0	2

Nb. of used railcars of each type		Nb. of assigned containers of each size		
t1	t2	20 ft	40 ft	53 ft
1	1	2	3	3

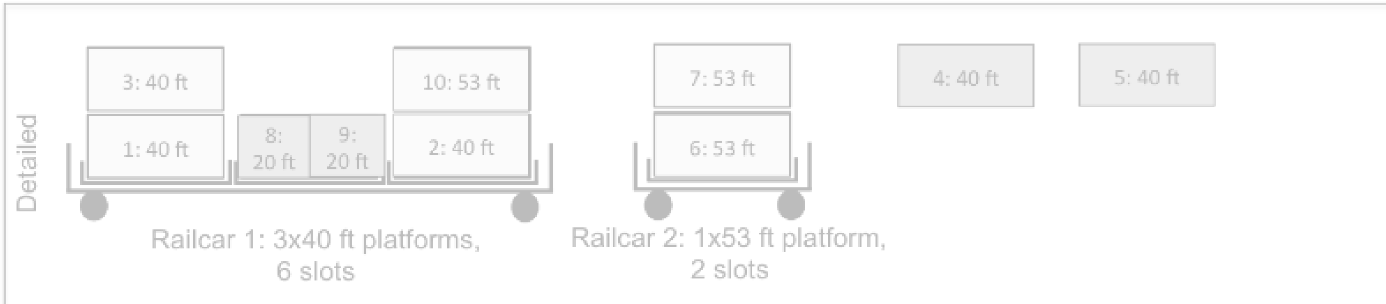
Global

Objective function value, e.g. slot utilization equals 7/8

CN CHAIR IN OPTIMIZATION OF RAILWAY OPERATIONS

SOLUTION DESCRIPTION

Possible solution descriptions



Used railcar ids	Nb. of assigned containers of each size		
	20 ft	40 ft	53 ft
1	2	3	1
2	0	0	2

Nb. of used railcars of each type		Nb. of assigned containers of each size		
t1	**t2**	**20 ft**	**40 ft**	**53 ft**
1	1	2	3	3

Global

Objective function value, e.g. slot utilization equals 7/8

SOLUTION DESCRIPTIONS

- The **solution descriptions** are encoded as vectors: $y \in \mathbb{N}^{12}$
- Each component corresponds to the number of railcars and containers used in the solution
- The precise assignation is not encoded

Nb. of used railcars of each type		Nb. of assigned containers of each size		
t1	t2	20 ft	40 ft	53 ft
1	1	2	3	3

CN CHAIR IN OPTIMIZATION OF RAILWAY OPERATIONS

DATA GENERATION & AGGREGATION

- The ML predictor must work with **unknown input characteristics**
- Aggregate over output
- Two approaches reported among five possible:
  - Aggregate **through training**: Model is trained to predict a <u>solution description</u>
  - Aggregate **before training**: Model is trained to predict a <u>representative solution description</u>

AGGREGATE THROUGH TRAINING

- Notation:
  x : Problem instance (without container weights)
  w : Container weights
  y : Detailed solution
  $\tilde{y}$: Solution description
- To sample one training example:

Sample one problem instance

$$\downarrow$$

$$(x, w) \xrightarrow{\text{Solve}} (x, w, y) \xrightarrow{\text{Disregard weights}} (x, y) \xrightarrow{\text{Use solution description}} (x, \tilde{y})$$

$$\downarrow$$

Add to your training set

AGGREGATE BEFORE TRAINING

- To sample one training example (Two-stage sampling):

x : Problem instance (without container weights)
w : Container weights
y : Detailed solution
$\tilde{y}$: Solution description

Sample one x
Sample n weights

$$\begin{bmatrix} (x, w_1) \\ \vdots \\ (x, w_n) \end{bmatrix} \xrightarrow{\text{Solve}} \begin{bmatrix} (x, w_1, y_1) \\ \vdots \\ (x, w_n, y_n) \end{bmatrix} \xrightarrow{\text{Choose representative solution}} (x, w_{j'}, y_{j'}) \xrightarrow{\text{Disregard weights}} (x, y_{j'}) \xrightarrow{\text{Use solution description}} (x, \tilde{y}_{j'})$$

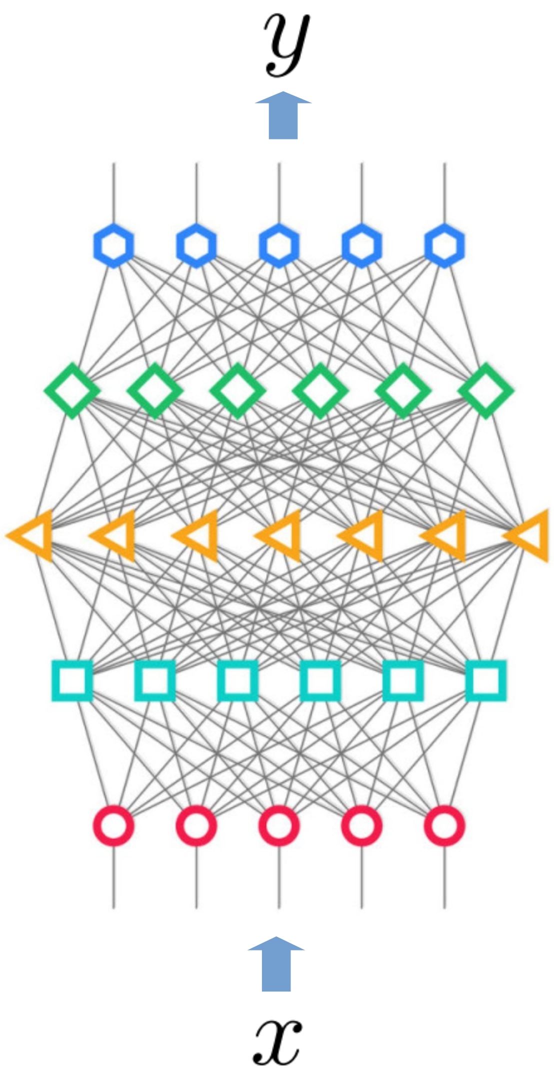
Add to your training set

e.g. representative solution = the solution featuring median objective function value at optimality

CN CHAIR IN OPTIMIZATION OF RAILWAY OPERATIONS

MACHINE LEARNING DETAILS

- **Multilayer perceptron**
  - $\approx$ 7 hidden layers
  - $\approx$ 500 units per layer
- **Training:**
  - GPU
  - Duration: between 2 to 10 hours
- **Hyperparameter selection**:
  - Early stopping
  - Random Search



FOUR DATA CLASSES

- We considered datasets with **varying difficulty**

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]

We never train on D ...

Computation Time

Data class	# instances	Percentiles time (s) P_5	P_{50}	P_{95}
A	20M	0.011	0.64	2.87
B	20M	0.02	1.26	3.43
C	20M	0.72	2.59	6.03
D	10M	2.64	5.44	20.89

PREDICTING SOLUTION DESCRIPTION IS FAST

Approximates **solution description** in **stochastic** setting

Computation time (s)

Data Percentiles	A P_5	A P_{50}	A P_{95}	D P_5	D P_{50}	D P_{95}
RegMLP	7.1×10^{-4}	8.3×10^{-4}	1.0×10^{-3}	7.4×10^{-4}	1.5×10^{-3}	2.3×10^{-3}
Commercial solver	0.011	0.64	2.87	2.64	5.44	20.89

Computes **detailed solution** in **deterministic** setting

PERFORMANCE EVALUATION

- Mean Absolute Error (MAE)
- Measured in containers and slots

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

$\tilde{y}$: Solution description (ground truth)
$\hat{y}$: Predicted solution description
i : Training example index
j : Container/railcar index

EXPERIMENTAL RESULTS

- MAE on "testing data"
- Aggregation "before training"
- Heuristics are simple and don't have access to weights
- High capacity models perform well

Data # examples	Abef 200K	ABCbef 600K
ClassMLP	1.181 (0.016)	1.807 (0.012)
LogReg	5.911 (0.029)	8.944 (0.027)
RegMLP	0.967 (0.016)	1.597 (0.012)
LinReg	18.301 (0.094)	40.014 (0.084)
HeurV	14.744 (0.074)	27.269 (0.083)
HeurS	17.808 (0.083)	31.465 (0.089)

TESTING ON HARDER PROBLEMS

- MAE on dataset D
- The models continue to perform well on **harder problems they have never seen**
- High variance between different hyperparameters (range in bracket)

Training-validation data # examples	2S-Abef 200K	2S-ABCbef 600K
ClassMLP	NA	14.823 [9.532, 23.782] (0.061)
LogReg	NA	28.171 (0.048)
RegMLP	2.852 [0.741, 9.052] (0.011)	0.323 [0.323, 1.109] (0.052)
LinReg	22.94 (0.047)	71.322 (0.054)
HeurV	32.098 (0.069)	32.098 (0.069)
HeurS	41.792 (0.077)	41.792 (0.077)

→ We probably got lucky...
Extrapolation seems risky

CN CHAIR IN OPTIMIZATION OF RAILWAY OPERATIONS

CONCLUSION

- We presented a ML-based methodology that:
  - is useful to predict **solution descriptions**
  - is useful to deal with **stochasticity** (through sampling & proper aggregation)
  - shows good results on the LPP
  - has **low average cost** when the predictor is used a lot of times

FUTURE WORK

- Consider different levels of detail in the solution
  - Implies variable input/output lengths
- Experiment with different ways of dealing with missing inputs
- Data generation is costly: explore active learning