Fast Prediction of Solutions to an Integer Linear Program with Machine Learning

CERMICS 2018 – June 29, 2018, Fréjus

ERIC LARSEN, CIRRELT and Université de Montréal (UdeM)
SÉBASTIEN LACHAPELLE, CIRRELT and UdeM
YOSHUA BENGIO, Montreal Institute for Learning Algorithms
EMMA FREJINGER, CIRRELT and UdeM
SIMON LACOSTE JULIEN, Montreal Institute for Learning Algorithms
ANDREA LODI, CERC & École Polytechnique de Montréal

CN CHAIR IN OPTIMIZATION OF RAILWAY OPERATIONS

CANADA EXCELLENCE RESEARCH CHAIR DATA SCIENCE FOR REAL-TIME DECISION-MAKING

CIRRELT

OUTLINE

- Introduction
  - Background and motivating application
  - Brief literature review
- Methodology
- Numerical results
- Conclusions and future work

We gratefully acknowledge the close collaboration with the **Canadian National Railway Company (CN)** *and the funding through the CN Chair in Optimization of Railway Operations.*

INTRODUCTION

- We consider applications for which a deterministic optimization model is available
  - Cannot be used due to a limit on computing time and imperfect knowledge about problem instances
  - Detailed solutions are not required

INTRODUCTION

- We propose a methodology to compute descriptions of solutions to discrete stochastic optimization problems in very short computing time
- Descriptions of solutions
  - Global: value of the objective function
  - Detailed: values of all decision variables
  - We concentrate on solution descriptions that lie between these two extremes

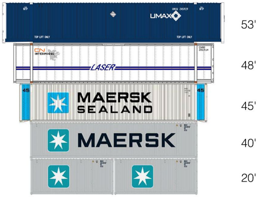
MOTIVATING APPLICATION

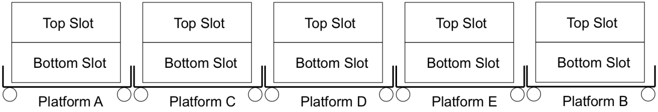
- Booking decisions of intermodal containers on double-stack railcars: **accept/reject fully or partially a booking request in real time**
- Similar to passengers needing a flight booking, containers need a train booking

- The assignment of containers to slots on railcars is a combinatorial optimization problem – the load planning problem (LPP)
- Mantovani et al. (2017) propose an integer linear programming formulation for the double-stack train LPP

MOTIVATING APPLICATION

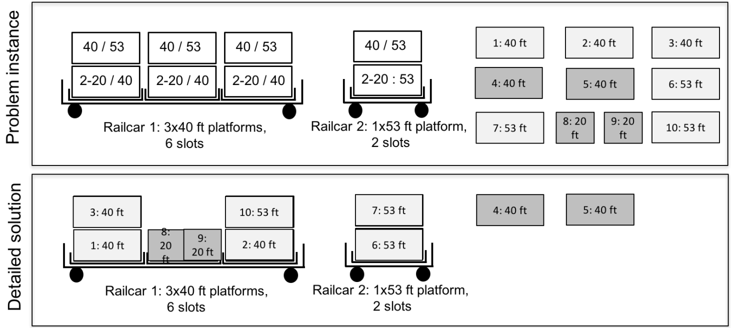
- The LPP crucially depends on detailed characteristics of the containers and the railcars
- **Container weights are not available at the time of the booking**





MOTIVATING APPLICATION

PROBLEM INSTANCE AND DETAILED SOLUTION

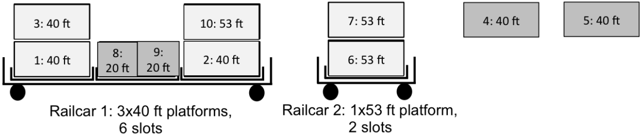


Containers in gray are heavier than those in white.

MOTIVATING APPLICATION

Different Solution Descriptions

Detailed



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

THE METHODOLOGY IN BRIEF

- Predict descriptions of solutions using **supervised learning**
- **Sample** large number of instances under **perfect information**
- Solve the instances using the existing solver to create **labeled data**
- **Train** a ML algorithm using the labeled data

- **Challenges**: input/output structure and how to deal with features that are unknown at the time of prediction

LITERATURE REVIEW

- Application of ML to discrete optimization problems was the focus of an important research effort in the 1980's and 1990's (Smith, 1999) but with limited success
- Recently, growing interest and success in introducing ML techniques in solution methodologies to decision problems
- Similarly, in ML there is a growing interest in addressing problems typically solved by OR methods (Lodi, 2017)
- Two related studies on **supervised learning**
  - Fischetti and Fraccaro (2017) predict optimal objective function value
  - Vinyals et al. (2015) predict detailed solutions to deterministic problems (e.g., TSP)

METHODOLOGY

- Generate data by repeatedly
  - Sample feature vector $\widetilde{\mathbf{x}} = [\widetilde{\mathbf{x}}_{\mathsf{av}}, \widetilde{\mathbf{x}}_{\mathsf{unav}}]$ that represents an instance of the deterministic problem
  - Compute a detailed solution $\widetilde{\mathbf{y}} = \widetilde{f}^*(\widetilde{\mathbf{x}})$ with $\widetilde{f}^*(\cdot)$
- $\bar{\mathbf{y}}$ denotes the synthesis of $\widetilde{\mathbf{y}}$ according to the desired description
- Find the best possible prediction $\mathbf{y} = f(\widetilde{\mathbf{x}}_{\mathsf{av}}; \theta)$ of $\bar{\mathbf{y}} = \bar{f}^*(\widetilde{\mathbf{x}})$
- $f(\cdot;\cdot)$ is a particular ML model and θ is a vector of parameters

METHODOLOGY

- **Aggregation**: how to approximate the output description $\bar{\mathbf{y}}$ resulting from $[\widetilde{\mathbf{x}}_{\text{av}}, \widetilde{\mathbf{x}}_{\text{unav}}]$ based on information on $\widetilde{\mathbf{x}}_{\text{av}}$ only, for example:
  - Over output before training: data $\{(\widetilde{\mathbf{x}}_{\text{av}}^{(i)}, \hat{\mathbf{y}}^{(i)})\ i = 1, \ldots, m\}$ where $\hat{\mathbf{y}}^{(i)} = \int \bar{f}^{*}(\widetilde{\mathbf{x}}_{\text{av}}^{(i)}, \widetilde{\mathbf{x}}_{\text{unav}}^{(i)}) g_2(\widetilde{\mathbf{x}}_{\text{av}}^{(i)}, \widetilde{\mathbf{x}}_{\text{unav}}^{(i)}) dF_{\widetilde{\mathbf{x}}_{\text{unav}}^{(i)}}$
    Choose representative solution, e.g., achieving mean value of objective function given $\widetilde{\mathbf{x}}_{av}^{(i)}$ over the support of $\widetilde{\mathbf{x}}_{unav}^{(i)}$
  - **Over output through training**: data $\{(\widetilde{\mathbf{x}}_{\text{av}}^{(i)}, \bar{\mathbf{y}}^{(i)})\ i = 1, \ldots, m\}$
    Minimize sample approximation of the expected discrepancy between the exact solution $\bar{\mathbf{y}}$ and approximate solution $f(\widetilde{\mathbf{x}}_{av}, \theta)$ based solely on knowledge of $\widetilde{\mathbf{x}}_{av}$

NUMERICAL RESULTS

INPUT-OUTPUT STRUCTURE

- We consider two container types (40 ft and 53 ft) and 10 railcar types (10 most numerous ones in the North American fleet)
- Solution description encoded as fixed-size vector $\bar{\mathbf{y}}$ (size 12) where each component corresponds to the number of railcars and containers **used in the solution**
- Feature vector $\widetilde{\mathbf{x}}_{\text{av}}$ has the same fixed size as $\bar{\mathbf{y}}$ where each element corresponds to the number of **available** railcars and containers

CN CHAIR IN OPTIMIZATION OF RAILWAY OPERATIONS

NUMERICAL RESULTS

DATA GENERATION

- Sample problem instances: sets of railcars and containers from predefined types, container weights $\widetilde{\mathbf{x}}_{\text{unav}}$ are sampled from empirical distribution conditional on type
- Two sampling procedures
  - 1-stage (1S) random sampling
  - 2-stage (2S) random sampling: at the first stage sample railcar and container types, at the second stage sample container weights conditional on stage one (100 instances)
- The purpose of the different sampling procedures is to analyze the two aggregation methods, here we present results for implicit (through training) aggregation only

NUMERICAL RESULTS

DATA GENERATION

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]

Sampling procedure	Data class	# instances	Percentiles time (s)		
			P_5	P_{50}	P_{95}
1S	A	20M	0.007	0.48	1.67
2S	A	20M	0.011	0.64	2.87
2S	B	20M	0.02	1.26	3.43
2S	C	20M	0.72	2.59	6.03
2S	D	10M	2.64	5.44	20.89

NUMERICAL RESULTS

ML ALGORITHM

- Multilayer perceptron (MLP): approximately 7 layers and 500 rectified linear units (ReLU) per layer
  Random hyperparameter search
- Classification (ClassMLP): softmax units in output layer
  Constraints on input-output in training, pseudo-likelihood maximization assuming outputs conditionally mutually independent given inputs
- Regression (RegMLP): linear units in output layer and rounding to the nearest integer
  Constraints on input-output at testing, minimization of sum of absolute errors

NUMERICAL RESULTS

ML ALGORITHM – TRAINING

- Mini-batch stochastic gradient descent
- Learning rate adaptation by Adam (adaptive moment estimation) method
- Regularization by early stopping
- 2 to 10 hours on GPU

VALIDATION ERRORS

- Performance measure: sum of mean absolute prediction error (MAE) over slots and containers

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

where $s_j, \; j = 1, \ldots, 10$, equals the number of slots on railcar type j and $s_{11} = s_{12} = 1$

CN CHAIR IN OPTIMIZATION OF RAILWAY OPERATIONS

VALIDATION ERRORS

- Average performance of both MLP models are very good, regression slightly better than classification

  MAE of only 2.1 containers/slots for instances with up to 150 containers and 200 slots and small standard deviation
- MLP results are considerably better than the benchmarks
- The marginal value of using 100 times more observations is fairly small (modest increase in MAE from 0.985 to 1.304)
- Prediction times are negligible, milliseconds or less and with very little variation

VALIDATION ERRORS

Data # examples	2S-Athr 200K	1S-Athr 20M	2S-ABCthr 600K
ClassMLP	**1.481** (0.018)	**0.965** (0.002)	**2.312** (0.014)
LogReg	5.956 (0.029)	5.887 (0.003)	9.051 (0.027)
RegMLP	**1.304** (0.017)	**0.985** (0.002)	**2.109** (0.014)
LinReg	18.306 (0.094)	18.372 (0.009)	39.907 (0.084)
HeurV	14.733 (0.075)	14.753 (0.008)	27.24 (0.083)
HeurS	17.841 (0.083)	17.842 (0.008)	31.448 (0.089)

EXTRANEOUS ERRORS

- Do the models trained and validated on simpler instances (classes A, B, C) **generalize to harder instances** (class D: up to 300 containers and 200 slots) without specific training and validation?
  - Performance is still good

    MAE of 2.85 (training on class A)
    MAE of 0.32 (training on classes A, B and C)
  - Important variability across models with different hyperparameters when only trained on class A

    MAE varies between 0.74 and 9.05

EXTRANEOUS ERRORS

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)

CONCLUSION

- We proposed a supervised ML approach for predicting descriptions of solutions to discrete stochastic optimization problems
- The motivating application was the train load planning problem
- The ML algorithm was trained on a large number of deterministic problems
- Missing information in input was addressed with aggregation methods
- Results showed that solutions can be predicted with high accuracy in very short computing time – much shorter than solving the full deterministic problem

FUTURE WORK

- Compare the solutions with those to a stochastic program solved by sample average approximation
- Results for other aggregation methods
- Active learning
- Other input-output structures (more detailed solution descriptions)