Dealing with uncertainty: tactical planning by machine learning

Andrea Lodi

Canada Excellence Research Chair
École Polytechnique de Montréal, Québec, Canada
andrea.lodi@polymtl.ca

MINOA Training School - Ischia - June 27, 2019

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

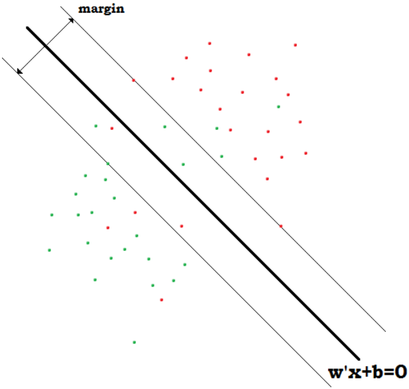
DO4ML: Discrete decisions in SVM

Ramp Loss $g(\xi_i) = (\min\{\xi_i, 2\})^+$

$$\min \frac{\omega^\top \omega}{2} + \frac{C}{n}\left(\sum_{i=1}^{n} \xi_i + 2\sum_{i=1}^{n} z_i\right)$$

$$\begin{aligned} y_i(\omega^\top x_i + b) &\geq 1 - \xi_i - M z_i && \forall i = 1, \ldots, n \\ 0 \leq \xi_i &\leq 2 && \forall i = 1, \ldots, n \\ \omega \in \mathbb{R}^d, b &\in \mathbb{R} \\ z &\in \{0,1\}^n \end{aligned}$$

with $M > 0$ big enough constant.



[Brooks (2011)]

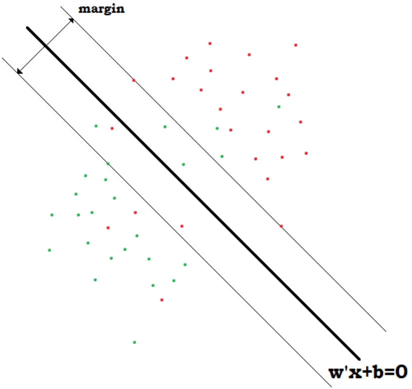
DO4ML: Discrete decisions in SVM

Ramp Loss $g(\xi_i) = (\min\{\xi_i, 2\})^+$

$$\min \frac{\omega^\top \omega}{2} + \frac{C}{n}\left(\sum_{i=1}^{n} \xi_i + 2\sum_{i=1}^{n} z_i\right)$$

$$\begin{aligned} y_i(\omega^\top x_i + b) &\geq 1 - \xi_i - M z_i & \forall i = 1, \ldots, n \\ 0 \leq \xi_i &\leq 2 & \forall i = 1, \ldots, n \\ \omega \in \mathbb{R}^d, b &\in \mathbb{R} & \\ z &\in \{0,1\}^n & \end{aligned}$$

with $M > 0$ big enough constant.



[Brooks (2011)]

Sophisticated methods for dealing with big-M constraints in MIP have been recently devised and integrated within the IBM-Cplex solver, so as decent-size SVM instances above can now be routinely solved to optimality.

[Belotti, Bonami, Fischetti, Lodi, Monaci, Nogales & Salvagnin (2016)]

What can (integer) Optimization do for ML? (reprise)

Just one example: (1) there are ML problems that are naturally casted as MIPs (discrete), but (2) **NOT** solved as MIPs.

Here, the goal is not necessarily to use MIP only. However, leveraging the quality and experience of MIP solving for discrete problems can be a plus (bounds, rewards, interpretability, etc.)

What can (integer) Optimization do for ML? (reprise)

Just one example: (1) there are ML problems that are naturally casted as MIPs (discrete), but (2) **NOT** solved as MIPs.

Here, the goal is not necessarily to use MIP only. However, leveraging the quality and experience of MIP solving for discrete problems can be a plus (bounds, rewards, interpretability, etc.)

An entire field of Interpretable Artificial Intelligence is emerging, where classification problems are solved by decision trees modeled as MIPs.

[Bertsimas et al. (2017), Günlük et al. (2018)]

What can (integer) Optimization do for ML? (reprise)

Just one example: (1) there are ML problems that are naturally casted as MIPs (discrete), but (2) **NOT** solved as MIPs.

Here, the goal is not necessarily to use MIP only. However, leveraging the quality and experience of MIP solving for discrete problems can be a plus (bounds, rewards, interpretability, etc.)

An entire field of Interpretable Artificial Intelligence is emerging, where classification problems are solved by decision trees modeled as MIPs.

[Bertsimas et al. (2017), Günlük et al. (2018)]

MIP (mostly, Combinatorial Optimization) sub-structure are present in Structured Prediction problems. Namely, these are the ML problems in which some constraints on the structure of the prediction have to be satisfied.

What can (integer) Optimization do for ML? (reprise)

Just one example: (1) there are ML problems that are naturally casted as MIPs (discrete), but (2) **NOT** solved as MIPs.

Here, the goal is not necessarily to use MIP only. However, leveraging the quality and experience of MIP solving for discrete problems can be a plus (bounds, rewards, interpretability, etc.)

An entire field of Interpretable Artificial Intelligence is emerging, where classification problems are solved by decision trees modeled as MIPs.

[Bertsimas et al. (2017), Günlük et al. (2018)]

MIP (mostly, Combinatorial Optimization) sub-structure are present in Structured Prediction problems. Namely, these are the ML problems in which some constraints on the structure of the prediction have to be satisfied.

A classical example is word alignment (a key step in machine translation), where matching and transportation structures can be effectively exploited.

[Lacoste-Julien et al. (2006, 2013, . . .)]

DO4LM: Learning by Column Generation

Besides formulating learning / classification problems by IP, one can apply sophisticated IP techniques to the learning phase.

This is the case of training a choice model in assortment optimization, where given a subset of the consumer's behaviors, one has to find the probability distribution (λ_k) that explains at best the training set, i.e., the observed sales.

DO4LM: Learning by Column Generation

Besides formulating learning / classification problems by IP, one can apply sophisticated IP techniques to the learning phase.

This is the case of training a choice model in assortment optimization, where given a subset of the consumer's behaviors, one has to find the probability distribution (λ_k) that explains at best the training set, i.e., the observed sales.

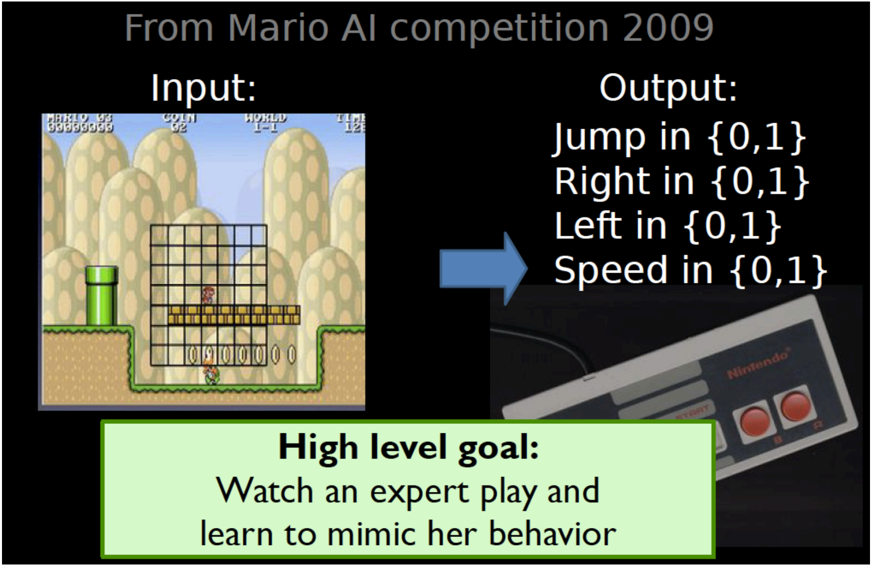
This can be done in a very elegant way by Column Generation

$$
\begin{aligned}
\min_{\boldsymbol{\lambda}, \boldsymbol{\epsilon}^+, \boldsymbol{\epsilon}^-} \quad & \mathbf{1}^T \boldsymbol{\epsilon}^+ + \mathbf{1}^T \boldsymbol{\epsilon}^- \\
\text{s.t.} \quad & \boldsymbol{A}\boldsymbol{\lambda} + \boldsymbol{\epsilon}^+ - \boldsymbol{\epsilon}^- = \boldsymbol{v} \\
& \mathbf{1}^T \boldsymbol{\lambda} = 1 \\
& \boldsymbol{\lambda}, \boldsymbol{\epsilon}^+, \boldsymbol{\epsilon}^- \geq 0
\end{aligned}
$$

[Bertsimas and Misic (2015)]

and the challenge is to make it practical for relevant sizes of the number of products. [Jena, Lodi, Palmer, Sole (2017, 2019)]

LM4DO: Learning to Search



[Langford and Daumé III, 2015]

LM4DO: Learning to Search (2)

The most notable outcome of the Learning to Search paradigm is the recent bulk of work on replacing MIP to solve combinatorial optimization problems by ML, so called End-to-end Learning.

LM4DO: Learning to Search (2)

The most notable outcome of the Learning to Search paradigm is the recent bulk of work on replacing MIP to solve combinatorial optimization problems by ML, so called End-to-end Learning.

Not surprising, the first attempts have been done in the Traveling Salesman Problem context and two papers stand:

- Supervised learning trained by precomputed (by an "expert") TSP solutions [Vinyals et al., 2015]
- Reinforcement learning with tour length as a reward function [Bello et al., 2017]

LM4DO: Learning to Search (2)

The most notable outcome of the Learning to Search paradigm is the recent bulk of work on replacing MIP to solve combinatorial optimization problems by ML, so called End-to-end Learning.

Not surprising, the first attempts have been done in the Traveling Salesman Problem context and two papers stand:

- Supervised learning trained by precomputed (by an "expert") TSP solutions [Vinyals et al., 2015]
- Reinforcement learning with tour length as a reward function [Bello et al., 2017]

Currently, none of the approaches is competitive in any way with specifically designed algorithms but the research, admittedly, led to interesting ML architectures (that can be applied elsewhere). [Khalil et al., 2017]

Machine Learning and Uncertainty

- I believe dealing with uncertainty is one of the topics that can benefit the most from ML.

Machine Learning and Uncertainty

- I believe dealing with uncertainty is one of the topics that can benefit the most from ML.

- In particular, we will deal with the uncertainty in planning and, specifically, the uncertainty associated with planning at a tactical (or strategic) level.

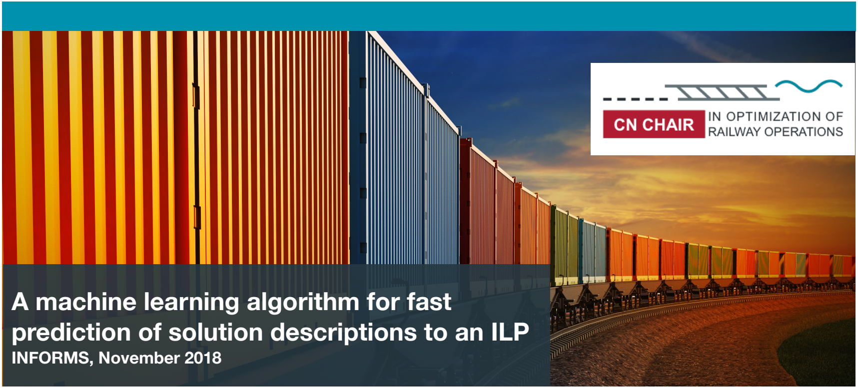
Machine Learning and Uncertainty

- I believe dealing with uncertainty is one of the topics that can benefit the most from ML.
- In particular, we will deal with the uncertainty in planning and, specifically, the uncertainty associated with planning at a tactical (or strategic) level.
- In other words, we are planning, for example the resources needed to deliver goods, at the time in which the amount of goods to be delivered is not known yet.

Machine Learning and Uncertainty

- I believe dealing with uncertainty is one of the topics that can benefit the most from ML.
- In particular, we will deal with the uncertainty in planning and, specifically, the uncertainty associated with planning at a tactical (or strategic) level.
- In other words, we are planning, for example the resources needed to deliver goods, at the time in which the amount of goods to be delivered is not known yet.
- Of course, we can do that by stochastic optimization but the catch is that we want to do it in real time.

Learning to Search for Tactical Planning



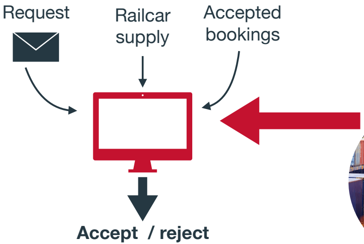
Eric Larsen Université de Montréal,CIRRELT
Sébastien Lachapelle Université de Montréal, CIRRELT
Yoshua Bengio Université de Montréal, Mila
Emma Frejinger Université de Montréal, CIRRELT
Simon Lacoste-Julien Université de Montréal, Mila
Andrea Lodi École Polytechnique

MOTIVATING APPLICATION

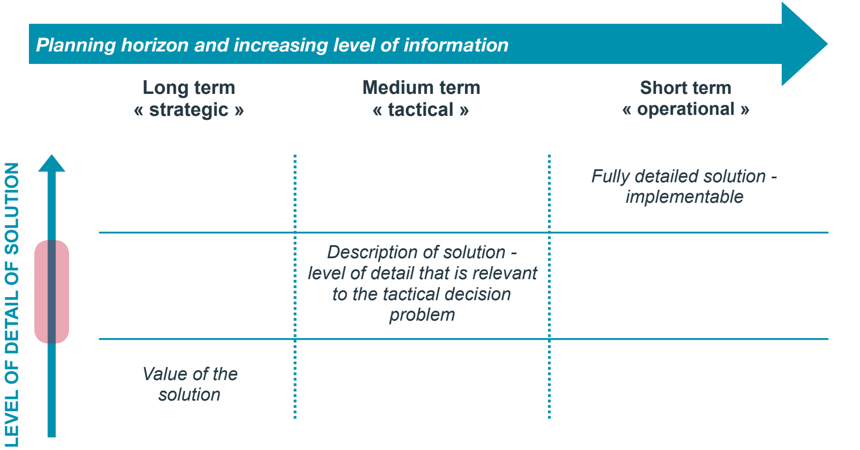
Planning horizon and increasing level of information

Longer term « tactical »	Shorter term « operational »
Accept / reject container bookings	**Load planning for double-stack trains**

OUR APPLICATION



CONTEXT



CN CHAIR IN OPTIMIZATION OF RAILWAY OPERATIONS

intermodal.iro.umontreal.ca | Page 4

Fast ML Prediction for Stochastic ILPs

CONTEXT

Planning horizon and increasing level of information

	Longer term « tactical »	**Shorter term « operational »**
	Compute description of solution to operational problem under imperfect information	Operational problem of interest: Compute solution under perfect information
COMPUTING TIME BUDGET: seconds to minutes		*Reasonable computing time - within the time budget for the operational problem*
COMPUTING TIME BUDGET: < milli-seconds	*Much shorter than the time it takes to solve the full problem under perfect information*	

CONTEXT

Planning horizon and increasing level of information

Longer term
« tactical »

Compute description of solution to operational problem under imperfect information

Shorter term
« operational »

Operational problem of interest: Compute solution under perfect information

High solution precision
Reasonable computing time

Solve deterministic optimization problem
mathematical programming

High-level solution
Very short computing time

Stochastic programming

Machine learning
predict the tactical solution descriptions

CN CHAIR IN OPTIMIZATION OF RAILWAY OPERATIONS

intermodal.iro.umontreal.ca | Page 6

CONTEXT

Planning horizon and increasing level of information

	Longer term « tactical »		**Shorter term « operational »**	
Problem instance	Imperfect information	$\tilde{\mathbf{x}}_{\mathbf{av}}$	Perfect information	$\tilde{\mathbf{x}} = [\tilde{\mathbf{x}}_{\mathbf{av}}, \tilde{\mathbf{x}}_{\mathbf{unav}}]$
Solution			Deterministic problem	$\tilde{\mathbf{y}}^*(\tilde{\mathbf{x}}) :\equiv \arg \inf_{\mathbf{y} \in \mathscr{Y}(\tilde{\mathbf{x}})} C(\tilde{\mathbf{x}}, \mathbf{y})$

Tactical solution description $\bar{\mathbf{y}}^* = g(\tilde{\mathbf{y}}^*(\tilde{\mathbf{x}}))$

PROBLEM AND OBJECTIVE

- Problem: $\bar{\mathbf{y}}^*$ is a solution to a two-stage stochastic program that we need to solve for any value of $\widetilde{\mathbf{x}}_{\mathbf{av}}$ in very short computing time
- Challenge: we would not be able to solve the stochastic program within the time budget for the application at hand
- Objective: find best possible prediction

$$\mathbf{y} = \underbrace{f}_{\text{Machine learning model}}(\widetilde{\mathbf{x}}_{\mathbf{av}}; \underbrace{\boldsymbol{\theta}}_{\text{Parameters}}) \quad \text{of} \quad \bar{\mathbf{y}}^*$$

CN CHAIR IN OPTIMIZATION OF RAILWAY OPERATIONS

METHODOLOGY

Problem

Two-stage stochastic programming formulation

Optimal prediction conditional on $\widetilde{\mathbf{x}}_{\mathbf{av}}$, expectation over distribution of $\widetilde{\mathbf{x}}_{\mathbf{unav}}$

Optimal solution to deterministic problem for given $\widetilde{\mathbf{x}} = [\widetilde{\mathbf{x}}_{\mathbf{av}}, \widetilde{\mathbf{x}}_{\mathbf{unav}}]$

Data

Problem instances and solutions (perfect information)

Machine learning training data

Training & predictions

Training and validation

Assess prediction performance

METHODOLOGY

Problem

$$\bar{\mathbf{y}}^*(\tilde{\mathbf{x}}_{\mathbf{av}}) :\equiv \arg \inf_{\mathbf{y} \in \mathscr{Y}(\tilde{\mathbf{x}}_{\mathbf{av}})} E_{\tilde{\mathbf{x}}_{\mathbf{unav}}} \{ \| \mathbf{y} - g(\tilde{\mathbf{y}}^*(\tilde{\mathbf{x}}_{\mathbf{av}}, \tilde{\mathbf{x}}_{\mathbf{unav}})) \| \mid \tilde{\mathbf{x}}_{\mathbf{av}} \}$$

$$\tilde{\mathbf{y}}^*(\tilde{\mathbf{x}}_{\mathbf{av}}, \tilde{\mathbf{x}}_{\mathbf{unav}}) :\equiv \arg \inf_{\mathbf{y} \in \mathscr{Y}(\tilde{\mathbf{x}}_{\mathbf{av}}, \tilde{\mathbf{x}}_{\mathbf{unav}})} C(\tilde{\mathbf{x}}_{\mathbf{av}}, \tilde{\mathbf{x}}_{\mathbf{unav}}, \mathbf{y})$$

Data

Problem instances and solutions (perfect information)

Machine learning training data

Training & predictions

Training and validation

Assess prediction performance

METHODOLOGY

Problem

$$\bar{\mathbf{y}}^*(\widetilde{\mathbf{x}}_{\mathsf{av}}) :\equiv \arg \inf_{\mathbf{y}\in\mathscr{Y}(\widetilde{\mathbf{x}}_{\mathsf{av}})} E_{\widetilde{\mathbf{x}}_{\mathsf{unav}}}\{\|\mathbf{y} - \boxed{g(\widetilde{\mathbf{y}}^*(\widetilde{\mathbf{x}}_{\mathsf{av}}, \widetilde{\mathbf{x}}_{\mathsf{unav}}))\| \mid \widetilde{\mathbf{x}}_{\mathsf{av}}\}}$$

$$\boxed{\widetilde{\mathbf{y}}^*(\widetilde{\mathbf{x}}_{\mathsf{av}}, \widetilde{\mathbf{x}}_{\mathsf{unav}})} :\equiv \arg \inf_{\mathbf{y}\in\mathscr{Y}(\widetilde{\mathbf{x}}_{\mathsf{av}},\widetilde{\mathbf{x}}_{\mathsf{unav}})} C(\widetilde{\mathbf{x}}_{\mathsf{av}}, \widetilde{\mathbf{x}}_{\mathsf{unav}}, \mathbf{y})$$

Data

Problem instances and solutions (perfect information)

$$(\ \widetilde{\mathbf{x}}^{(i)}\ ,\ \widetilde{\mathbf{y}}^{*(i)}\ )\quad i = 1,\ldots,m$$

Machine learning training data

$$(\ \widetilde{\mathbf{x}}_{\mathsf{av}}^{(i)}\ ,\ \bar{\mathbf{y}}^{*(i)}\ )\quad i = 1,\ldots,m$$

Training & predictions

Training and validation

Assess prediction performance

METHODOLOGY

Problem

$$\bar{\mathbf{y}}^*(\tilde{\mathbf{x}}_{\text{av}}) :\equiv \arg \inf_{\mathbf{y}\in\mathscr{Y}(\tilde{\mathbf{x}}_{\text{av}})} E_{\tilde{\mathbf{x}}_{\text{unav}}}\{\|\mathbf{y} - g(\tilde{\mathbf{y}}^*(\tilde{\mathbf{x}}_{\text{av}}, \tilde{\mathbf{x}}_{\text{unav}}))\| \mid \tilde{\mathbf{x}}_{\text{av}}\}$$

$$\tilde{\mathbf{y}}^*(\tilde{\mathbf{x}}_{\text{av}}, \tilde{\mathbf{x}}_{\text{unav}}) :\equiv \arg \inf_{\mathbf{y}\in\mathscr{Y}(\tilde{\mathbf{x}}_{\text{av}}, \tilde{\mathbf{x}}_{\text{unav}})} C(\tilde{\mathbf{x}}_{\text{av}}, \tilde{\mathbf{x}}_{\text{unav}}, \mathbf{y})$$

Data

Problem instances and solutions (perfect information)

$$(\tilde{\mathbf{x}}^{(i)}, \tilde{\mathbf{y}}^{*(i)}) \quad i = 1, \ldots, m$$

Machine learning training data

$$(\tilde{\mathbf{x}}_{\text{av}}^{(i)}, \bar{\mathbf{y}}^{*(i)}) \quad i = 1, \ldots, m$$

Training & predictions

Training and validation, e.g.,

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{m'} \sum_{i=1}^{m'} L\left(f(\tilde{\mathbf{x}}_{\text{av}}; \theta), \bar{\mathbf{y}}^{*(i)}\right)$$

Assess prediction performance, e.g.,

$$\textbf{MAE}_{\textbf{test}} = \frac{1}{n} \sum_{i=1}^{n} \left| f(\tilde{\mathbf{x}}_{\text{av}}^{(i)}; \hat{\theta}) - \bar{\mathbf{y}}^{*(i)} \right|$$

APPLICATION - OPERATIONAL PROBLEM

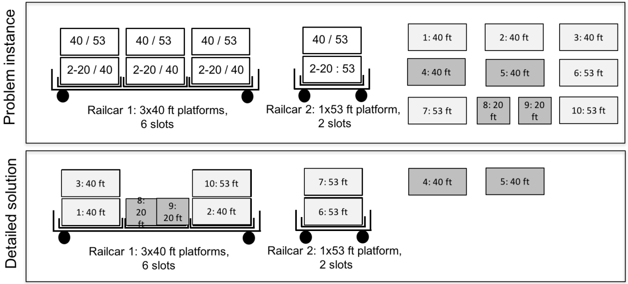
Load planning problem (LPP) for double-stack trains

- The assignment of containers to slots on railcars is a combinatorial optimization problem that depend on, e.g.,
  - Railcar types
  - Container types and their **weight**
- ILP formulation (Mantovani et al., 2018)

Fast ML Prediction for Stochastic ILPs

APPLICATION - OPERATIONAL PROBLEM

Load planning problem (LPP) for double-stack trains



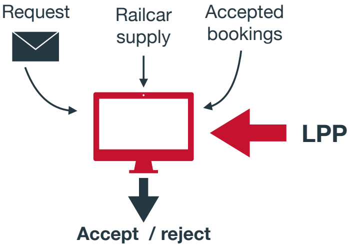
Containers in dark gray are heavier than the others

CN CHAIR IN OPTIMIZATION OF RAILWAY OPERATIONS

APPLICATION - TACTICAL PROBLEM

Accept / reject container bookings

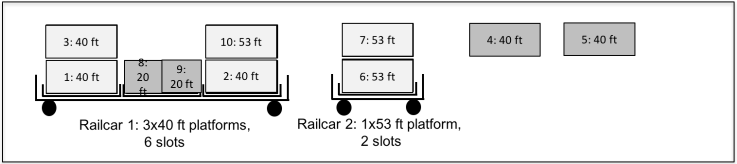
- Similar to passengers needing to book a seat on a flight, containers need a train booking
- Container weights are unknown at the time of booking
- A accept/reject decision does not require a full solution (assignment) and must be done in very short time (real-time system)



APPLICATION - TACTICAL PROBLEM

Accept / reject container bookings

Detailed solution



Aggregate solution

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

INPUT-OUTPUT STRUCTURE

- 2 container types: 40 ft and 53 ft
- 10 railcar types (10 most numerous in the North American fleet)
- Solution description $\bar{\mathbf{y}}$ is encoded as fixed size vector (size 12)
  - Each element corresponds to the number of railcars and containers **used in the solution**
- Feature vector $\tilde{\mathbf{x}}_{\mathbf{av}}$ has the same size as the output vector
  - Each element corresponds to number of **available** railcars/ containers

DATA GENERATION

- 1-stage (1S) random sampling
- 2-stage (2S) random sampling
  - Stage 1: container/railcar types. Stage 2: weights conditional on stage 1.

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

TEST ERROR

- Average performance of the MLP model is very good
  - 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 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

FEASIBILITY

- We assess numerically if a **feasible operational solution** exists to a given predicted tactical solution description
  - All decision variables of the LPP depend on the weights but the ML algorithm is blind to weights and to the structure of the constraints
- We assess the share of instances that satisfy the weight constraints (analogy with a chance constraint formulation)
  - Train algorithm on 1S-A
  - Algorithm predicts tactical solution descriptions for 200K first stage instances of 2S-A (no weights)
  - For each of the 200K instances, there are 100 full information instances, we solve these with CPLEX using the LPP formulation but constrained to the tactical solution predicted by the algorithm

FEASIBILITY

	Sample ratio feasible	Std err sample ratio (gaussian approx of binomial)
ClassMLP	0.975	0.00035
LogReg	0.614	0.00109
RegMLP	0.966	0.00041
LinReg	0.742	0.00098
HeurV	0.324	0.0011
HeurS	0.400	0.0011

There exists a feasible operational solution for a given predicted tactical solution in 96.6% of the instances

This share is much lower for linear regression or the deterministic heuristics (74.2% and 40% respectively)

Variance very close to zero

Conclusions

We have shown that standard deep learning techniques can be applied to predict the description of a solution of a discrete optimization problem under imperfect information, which is generally the case of the tactical level of a planning problem.

Conclusions

We have shown that standard deep learning techniques can be applied to predict the description of a solution of a discrete optimization problem under imperfect information, which is generally the case of the tactical level of a planning problem.

We have shown an application in transportation of this general methodology in which the need of giving an answer in real time motivated our approach.

In the same context, the description of the solution (computed as fast as possible) can be used in an outer algorithm that is searching the space of the optimal train scheduling.

Conclusions

We have shown that standard deep learning techniques can be applied to predict the description of a solution of a discrete optimization problem under imperfect information, which is generally the case of the tactical level of a planning problem.

We have shown an application in transportation of this general methodology in which the need of giving an answer in real time motivated our approach.

In the same context, the description of the solution (computed as fast as possible) can be used in an outer algorithm that is searching the space of the optimal train scheduling.

I believe we are just experiencing the first steps for the use of machine learning techniques for discrete optimization under uncertainty, another example being reoptimization. [Lodi, Mossina & Rachelson, 2019]