An offline-online strategy to improve MILP performance via Machine Learning tools

Asunción Jiménez-Cordero
asuncionjc@uma.es

Joint work with:
Juan Miguel Morales González
Salvador Pineda Morente

oasys.uma.es

New Bridges Between Mathematics and Data Science

November 8th, 2021

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 755705)

Outline

Outline

Mixed Integer
Linear Programs
(MILP)

Machine
Learning
(ML)

Combine knowledge from both worlds

Recent reviews: *Bengio et al. [2021]; Gambella et al. [2021]*

Literature review

- Branch-and-bound methods: *Karapetyan et al. [2017]; Kruber et al. [2017]; Liberto et al. [2016]; Lodi and Zarpellon [2017].*

- End-to-end approaches: *Kool et al. [2019]; Larsen et al. [2018].*

Literature review

- Branch-and-bound methods: *Karapetyan et al. [2017]; Kruber et al. [2017]; Liberto et al. [2016]; Lodi and Zarpellon [2017].*
  - ✓ Optimality guarantees.
  - × Slow.
- End-to-end approaches: *Kool et al. [2019]; Larsen et al. [2018].*

Literature review

- Branch-and-bound methods: *Karapetyan et al. [2017]; Kruber et al. [2017]; Liberto et al. [2016]; Lodi and Zarpellon [2017].*
  - ✓ Optimality guarantees.
  - × Slow.
- End-to-end approaches: *Kool et al. [2019]; Larsen et al. [2018].*
  - ✓ Fast.
  - × Suboptimal/infeasible solutions.

Literature review

- Branch-and-bound methods: *Karapetyan et al. [2017]; Kruber et al. [2017]; Liberto et al. [2016]; Lodi and Zarpellon [2017].*
  - ✓ Optimality guarantees.
  - × Slow.
- End-to-end approaches: *Kool et al. [2019]; Larsen et al. [2018].*
  - ✓ Fast.
  - × Suboptimal/infeasible solutions.

Our approach

- Computational gains.
- Reduce risk of infeasible problems.

Literature review

- Branch-and-bound methods: *Karapetyan et al. [2017]; Kruber et al. [2017]; Liberto et al. [2016]; Lodi and Zarpellon [2017].*
  - ✓ Optimality guarantees.
  - × Slow.
- End-to-end approaches: *Kool et al. [2019]; Larsen et al. [2018].*
  - ✓ Fast.
  - × Suboptimal/infeasible solutions.

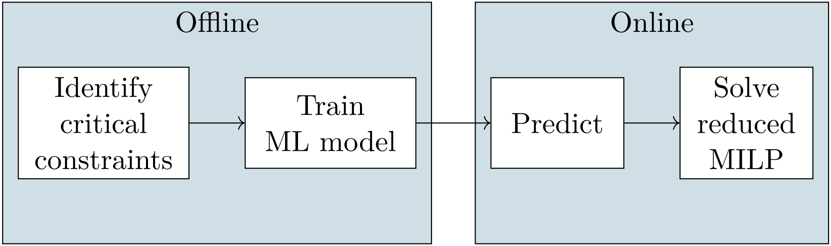
Our approach

- Computational gains.
- Reduce risk of infeasible problems.

Simpler problems

- Remove non-critical constraints.
- Solve a reduced optimization problem.

Our methodology in a nutshell



Outline

MILP

$$(P_{\boldsymbol{\theta}}[\mathcal{J}]) \qquad \begin{cases} \min\limits_{\boldsymbol{z}\in\mathbb{R}^n\times\mathbb{Z}^q} \boldsymbol{c}^T\boldsymbol{z} \\ \quad \text{s.t. } \boldsymbol{a}_j^T\boldsymbol{z} \leq b_j, \quad \forall j \in \mathcal{J} \end{cases}$$

- $\boldsymbol{\theta} = \{\boldsymbol{c}, \boldsymbol{a}_j, b_j, \forall j \in \mathcal{J}\}$.
- $P_{\boldsymbol{\theta}}[\mathcal{J}]$ bounded and feasible.
- Optimal solution $\boldsymbol{z}^*_{\boldsymbol{\theta}}[\mathcal{J}]$ is a singleton.

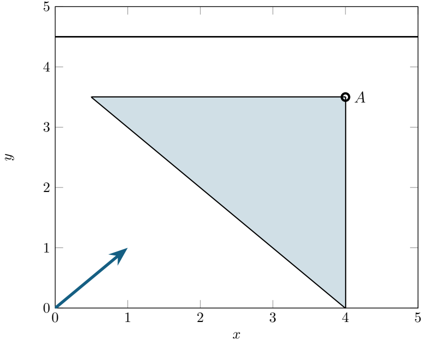
Invariant Constraint Set, $\mathcal{S}$

According to *Calafiore [2010]*:

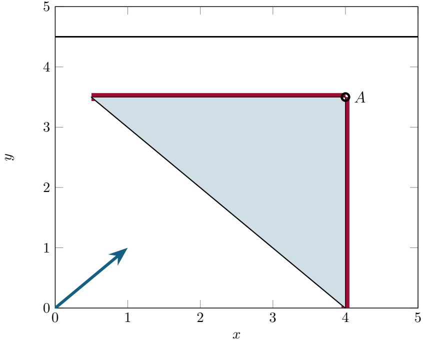
$\mathcal{S} \subset \mathcal{J}$ s.t. $\boldsymbol{c}^T \boldsymbol{z}^*_{\boldsymbol{\theta}}[\mathcal{S}] = \boldsymbol{c}^T \boldsymbol{z}^*_{\boldsymbol{\theta}}[\mathcal{J}]$

The integrality of the decision variables is crucial to find out which constraints belong to $\mathcal{S}$.

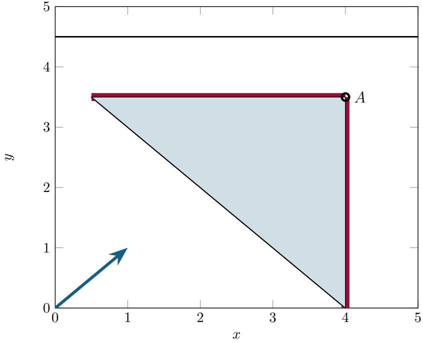
LP vs MILP (Example taken from *Pineda et al. [2020]*)



LP vs MILP (Example taken from *Pineda et al. [2020]*)



LP vs MILP (Example taken from *Pineda et al. [2020]*)

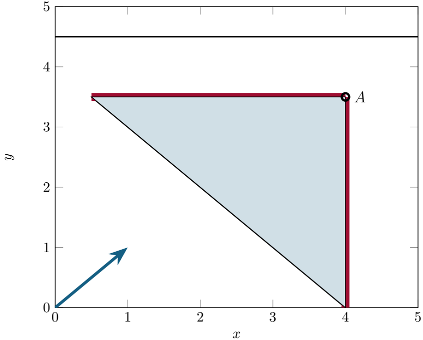


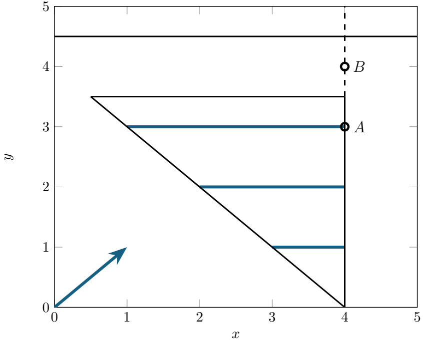
Binding constraints

$$\mathcal{B} = \{j \in \mathcal{J} : \boldsymbol{a}_j^T \boldsymbol{z}_\theta^*[\mathcal{J}] = b_j\}$$

$$\mathcal{S} = \mathcal{B}$$

LP vs MILP (Example taken from *Pineda et al. [2020]*)



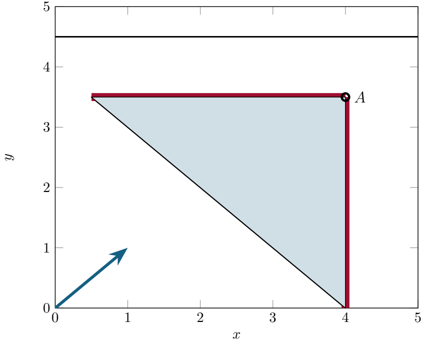


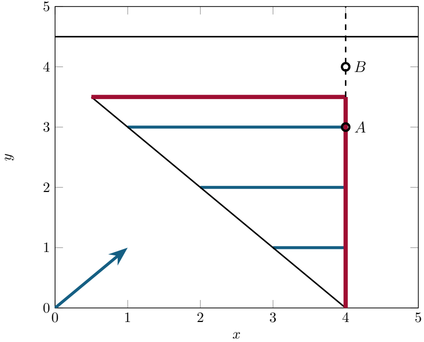
Binding constraints

$$\mathcal{B} = \{j \in \mathcal{J} : \boldsymbol{a}_j^T \boldsymbol{z}_\theta^*[\mathcal{J}] = b_j\}$$

$$\mathcal{S} = \mathcal{B}$$

LP vs MILP (Example taken from *Pineda et al. [2020]*)



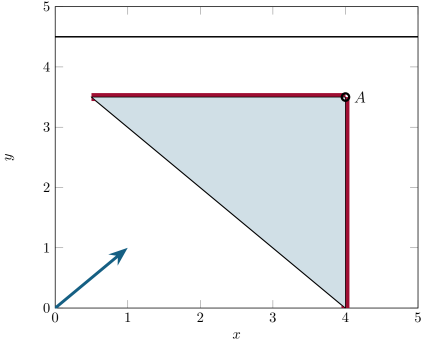


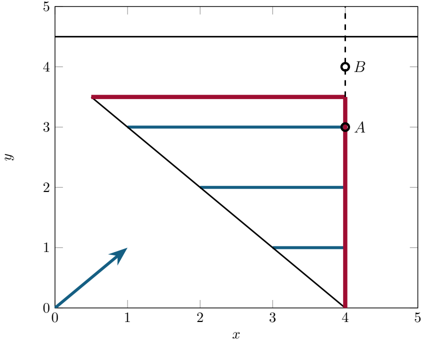
Binding constraints

$$\mathcal{B} = \{ j \in \mathcal{J} : \boldsymbol{a}_j^T \boldsymbol{z}_\theta^*[\mathcal{J}] = b_j \}$$

$$\mathcal{S} = \mathcal{B}$$

LP vs MILP (Example taken from *Pineda et al. [2020]*)





Binding constraints

$\mathcal{B} = \{j \in \mathcal{J} : \boldsymbol{a}_j^T \boldsymbol{z}_\theta^*[\mathcal{J}] = b_j\}$

Some non-binding constraints

also belong to $\mathcal{S}$.

$\mathcal{S} = \mathcal{B}$

$\mathcal{S} \supset \mathcal{B}$

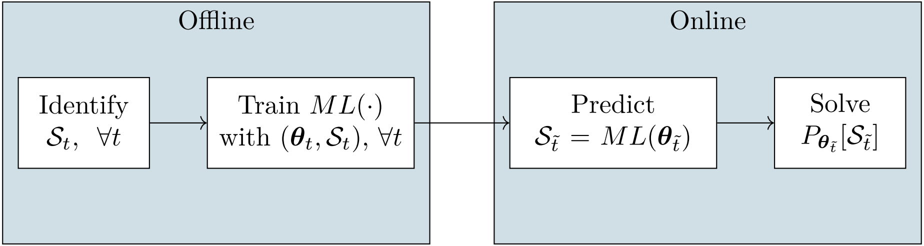
- Finding $\mathcal{S}$ is challenging in MILPs.
- For each train instance t, we look for $\mathcal{S}_t$, including some of the non-binding constraints.
- And reduced MILP $P_{\theta_t}[\mathcal{S}_t]$ is solved.

How to find $\mathcal{S}_t$?

Algorithm 1 Identifying an invariant constraint set for each instance t

0) Initialize $\mathcal{S}_t = \mathcal{B}_t$.
1) Solve $P_{\boldsymbol{\theta_t}}[\mathcal{S}_t]$ with solution $\boldsymbol{z}^*_{\boldsymbol{\theta}_t}[\mathcal{S}_t]$.
2) If $\boldsymbol{z}^*_{\boldsymbol{\theta}_t}[\mathcal{S}_t]$ is infeasible for $P_{\boldsymbol{\theta_t}}[\mathcal{J}]$, go to step 3). Otherwise, stop.
3) $\mathcal{S}_t := \mathcal{S}_t \cup \{j \in \mathcal{J} \setminus \mathcal{S}_t : j$ is the most violated constraint$\}$, go to step 1).

Recap


Offline
Identify
$\mathcal{S}_t, \ \forall t$
Train $ML(\cdot)$
with $(\boldsymbol{\theta}_t, \mathcal{S}_t)$, $\forall t$
Online
Predict
$\mathcal{S}_{\tilde{t}} = ML(\boldsymbol{\theta}_{\tilde{t}})$
Solve
$P_{\boldsymbol{\theta}_{\tilde{t}}}[\mathcal{S}_{\tilde{t}}]$

Advantages

- Based on constraint generation. Crucial non-binding constraints are guaranteed to be included.
- Reduce risk of infeasible problems.
- Independent on the ML method used.
- Identifying $\mathcal{S}$ and training ML is performed offline.

Outline

Experimental Setup

- Binary classification problem. $k$nn.
- Label $s_j^t = \pm 1$ depending on inclusion on $\mathcal{S}_t$.
- Two approaches: $\mathcal{B}$-learner and $\mathcal{S}$-learner.
- Synthetic and real-world applications.

Unit Commitment problem

$$\begin{cases} \min\limits_{\boldsymbol{x}\in\mathbb{R}^n,\, \boldsymbol{y}\in\{0,1\}^n} \sum_{i=1}^n c_i x_i \\ \text{s.t. } \sum_{i=1}^n x_i = \sum_{i=1}^n d_i, \\ -f_j \leq \sum_{i=1}^n a_{ij}(x_i - d_i) \leq f_j, \quad j = 1, \dots, m \\ l_i y_i \leq x_i \leq u_i y_i, \quad i = 1, \dots, n \end{cases}$$

- $\boldsymbol{\theta} = \mathbf{d}$.
- $n = 96$.
- $m = 120$ (240 constraints).
- $T = 8640$ (Leave-one-out).

			k				
		Bench.	5	10	20	50	100
$\mathcal{B}$-learner	constraints	240	[0, 23]	[0, 23]	[0, 26]	[0, 26]	[0, 28]
	% infeasible	0%	45.71%	38.02%	31.73%	23.11%	16.29%
	MILP time pr.	100%	24.59%	27.05%	28.72%	31.31%	33.43%
$\mathcal{S}$-learner	constraints	240	[0, 26]	[0, 28]	[0, 29]	[0, 30]	[0, 32]
	% infeasible	0%	7.38%	2.78%	1.20%	0.55%	0.28%
	MILP time pr.	100%	33.96%	34.97%	36.45%	36.14%	37.79%

- Larger values of k imply more constraints (more conservative).
- Computational gains in both approaches.
- Few extra constraints in $\mathcal{S}$-learner.
- Large improvements with regard to infeasible problems in $\mathcal{S}$-learner.
- Adding constraints is not enough ($k = 5$ vs $k = 50$).

More details

Offline constraint screening
for online mixed-integer optimization

Asunción Jiménez-Cordero*, Juan Miguel Morales, Salvador Pineda

OASYS Group. Ada Byron Research Building,
Arquitecto Francisco Peñalosa St., 18, 29010,
University of Málaga, Málaga, Spain

Abstract

Mixed Integer Linear Programs (MILP) are well known to be NP-hard problems in general, and therefore, tackling and using their solution in online applications is a great challenge. Some Machine Learning techniques have been proposed in the literature to alleviate their computational burden. Unfortunately, these techniques report that a non-negligible percentage of the resulting machine-learning-based solutions are infeasible.

By linking Mathematical Optimization and Machine Learning, this paper proposes an offline-online strategy to solve MILPs in a fast and reliable manner. The offline step seeks to identify the so-called support constraints of past instances of the target MILP and uses them to train a Machine Learning model of our choice. This model is subsequently used online to generate a reduced MILP that is significantly easier to solve.

Through synthetic and real-life MILPs, we show that our approach dramatically decreases the risk of obtaining solutions from the reduced MILP that are infeasible in the original formulation without any extra cost in terms of computational time. Hence, our methodology is appropriate for online applications where feasibility is of particular importance.

Keywords: Machine Learning, Mixed Integer Linear Programming, constraint screening, offline-online strategy

Available at:

https://www.researchgate.net/publication/350371853_Offline_constraint_screening_for_online_mixed-integer_optimization

SCAN ME

Outline

Conclusions

- Approach which combines MILPs and ML.
- Offline-online strategy.
- Reduce risk of infeasible problems.
- Reduce computational burden.
- Tested on synthetic and real-world applications.

Conclusions

- Approach which combines MILPs and ML.
- Offline-online strategy.
- Reduce risk of infeasible problems.
- Reduce computational burden.
- Tested on synthetic and real-world applications.

Further research

- Other input parameters.
- Introduce expert-knowledge information.

References

Bengio, Y., Lodi, A., and Prouvost, A. (2021). Machine learning for combinatorial optimization: A methodological tour d'horizon. *European Journal of Operational Research*, 290(2):405–421, doi:10.1016/j.ejor.2020.07.063.

Calafiore, G. C. (2010). Random convex programs. *SIAM Journal on Optimization*, 20(6):3427–3464, doi:10.1137/090773490.

Gambella, C., Ghaddar, B., and Naoum-Sawaya, J. (2021). Optimization problems for machine learning: A survey. *European Journal of Operational Research*, 290(3):807–828, doi:10.1016/j.ejor.2020.08.045.

Karapetyan, D., Punnen, A. P., and Parkes, A. J. (2017). Markov chain methods for the bipartite boolean quadratic programming problem. *European Journal of Operational Research*, 260(2):494–506, doi:10.1016/j.ejor.2017.01.001.

Kool, W., Van Hoof, H., and Welling, M. (2019). Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475*.

Kruber, M., Lübbecke, M. E., and Parmentier, A. (2017). Learning when to use a decomposition. In Salvagnin, D. and Lombardi, M., eds., *Integration of AI and OR Techniques in Constraint Programming*, pages 202–210. Springer International Publishing.

Larsen, E., Lachapelle, S., Bengio, Y., Frejinger, E., Lacoste-Julien, S., and Lodi, A. (2018). Predicting tactical solutions to operational planning problems under imperfect information. *arXiv preprint arXiv:1807.11876*.

Liberto, G. D., Kadioglu, S., Leo, K., and Malitsky, Y. (2016). DASH: Dynamic approach for switching heuristics. *European Journal of Operational Research*, 248(3):943–953, doi:10.1016/j.ejor.2015.08.018.

Lodi, A. and Zarpellon, G. (2017). On learning and branching: A survey. *TOP*, 25(2):207–236, doi:10.1007/s11750-017-0451-6.

Pineda, S., Morales, J. M., and Jiménez-Cordero, A. (2020). Data-driven screening of network constraints for unit commitment. *IEEE Transactions on Power Systems*, 35(5):3695–3705, doi:10.1109/TPWRS.2020.2980212.

An offline-online strategy to improve MILP performance via Machine Learning tools

Asunción Jiménez-Cordero

asuncionjc@uma.es

Joint work with:

Juan Miguel Morales González

Salvador Pineda Morente

Thank you very much for your attention!

oasys.uma.es

New Bridges Between Mathematics and Data Science

November 8th, 2021

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 755705)