Machine learning for optimization

M2 MODO

*Course project - 2025/2026**

Guidelines

- Students may discuss the project with their classmates, but the submitted version must be worked out, written, and submitted individually.
- Projects should be sent to Clément Royer via email (`clement.royer@lamsade.dauphine.fr`) or Teams before the deadline. Please use a compressed folder with all necessary files.
- For Part A of the project, students may submit a version of the solution notebooks used in class. All questions from the final (Extension) parts should be answered.
- For Part B, the format is open. A short PDF document (slides or report) is the suggested format but creativity is encouraged, as long as the document does answer the questions from Part B.
- The deadline to send sources is **March 1st, 2026 11.59pm AOE**.

Part A - Following up on the lab sessions

Assignment 1: Implicit layers and automatic differentiation Fill out the last part of the notebook on implicit layers and automatic differentiation to implement a trust-region layer.

Assignment 2: Unfolded solvers Fill out the last part of the notebook on unfolded solvers to learn a good regularization parameter along with a good stepsize.

Assignment 3: Graph neural networks Fill out the last part of the notebook on graph neural networks to analyze the ability of a GNN to predict dual variables.

*Revised version of January 28, 4.30pm. Latest version available at https://www.lamsade.dauphine.fr/~croyer/ensdocs/MLO/ProjMLO.pdf

Part B - Paper reading

Assignment 4: Machine learning and MILPs Pick two of the references below, that present a strategy to learn branching rules for MILPs, and compare the proposed methodologies along the following axes:

- What is the machine learning task at hand?
- What dataset is considered, and how is it built?
- What was the computing setup necessary to obtain the results?

In addition, feel free to mention other aspects of the papers (discussion items, comparison with the literature,...) that you deem interesting for the course.

Reference list (with links to open access versions)

- A. M. Alvarez, Q. Louveaux and L. Wehenkel, *A machine learning-based approximation of strong branching* (Informs Journal on Computing, 2017)
- M.-F. Balcan, T. Dick, T. Sandholm, and E. Vitercik. *Learning to branch* (ICML 2018)
- H. He, H. Daumé III, and J. Eisner. *Learning to search in branch and bound algorithms.* (NeurIPS 2014)
- E. B. Khalil, P. Le Bodic, L. Song, G. Nemhauser, and B. Dilkina. *Learning to branch in mixed integer programming.* (AAAI 2016).