

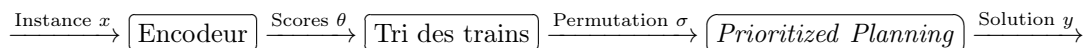
Recherche d'itinéraires dans un réseau ferroviaire : apprendre à mieux optimiser

- **Guillaume Dalle** (CERMICS, Ecole des Ponts, Marne-la-Vallée, France)
- **Axel Parmentier** (CERMICS, Ecole des Ponts, Marne-la-Vallée, France)

Mots-clé : optimisation combinatoire, apprentissage machine, différentiation automatique, recherche d'itinéraires

Résumé : Dans tout réseau ferroviaire de grande taille, une gestion efficace du trafic est essentielle pour maximiser la ponctualité et réagir aux incidents. Le défi principal est de construire puis d'ajuster l'itinéraire de chaque train en évitant les conflits d'infrastructure : sur des rails, difficile de se dépasser ou de se croiser. Ce problème, appelé *Multi-Agent Path Finding*, est NP-dur, si bien que pour les instances de grande taille, seules les méthodes heuristiques génèrent des solutions en un temps raisonnable [1]. L'une d'entre elles, appelée *Prioritized Planning*, ordonne les trains au préalable, puis calcule leurs trajets de façon séquentielle grâce à un algorithme de plus court chemin (typiquement A*).

La performance de cette heuristique dépend grandement de la permutation choisie, c'est pourquoi nous en proposons une version *data-driven*, inspirée des travaux de [2]. En effet, les grilles horaires varient peu d'un jour à l'autre, et l'on peut donc s'appuyer sur l'historique des instances passées (et de leurs solutions) afin de mieux résoudre celles du futur. Notre procédure est décrite par le schéma suivant :



La phase d'apprentissage consiste à estimer les paramètres de l'encodeur, qui combine des *features* d'intérêt pour associer un score d'importance à chaque train. Cependant, ni le tri ni le *Prioritized Planning* ne sont des opérations différentiables, ce qui exclut l'utilisation de méthodes de gradient lors de l'entraînement.

Pour pallier cette difficulté, nous interprétons le tri comme un Programme Linéaire, dont nous perturbons le vecteur de coût avec un bruit aléatoire [3] pour lisser son comportement. Nous modifions également la cible de l'apprentissage, en prédisant non pas l'ensemble des chemins mais simplement la permutation des trains. Cela permet de considérer le *Prioritized Planning* comme une étape de post-traitement, qu'il n'est plus nécessaire de différencier. En revanche, il faut alors constituer le dataset de façon adaptée, par exemple en résolvant ses instances grâce à une recherche locale sur le permutahedron.

Nous implémentons notre algorithme dans le langage Julia, ce qui nous permet de tirer parti de son écosystème de différenciation algorithmique. Les tests sont pratiqués sur le challenge Flatland [4], un environnement simplifié de simulation ferroviaire issu d'une collaboration entre les chemins de fer suisses, allemands et français. A court terme, nous prévoyons d'étendre notre approche au cas stochastique, afin de faciliter l'adaptation du plan de transport suite à une défaillance matérielle.

Références :

- [1] Stern, R. (2019). *Multi-Agent Path Finding – An Overview*. Artificial Intelligence : 5th RAAI Summer School, Dolgoprudny, Russia, July 4–7, 2019, Tutorial Lectures Lecture Notes in Computer Science (G. S. Osipov, A. I. Panov and K. S. Yakovlev, ed) pp 96–115. Springer International Publishing, Cham.
- [2] Parmentier, A. and T'Kindt, V. (2021). *Learning to solve the single machine scheduling problem with release times and sum of completion times*. arXiv :2101.01082 [cs, math].
- [3] Berthet, Q., Blondel, M., Teboul, O., Cuturi, M., Vert, J.-P. and Bach, F. (2020). *Learning with differentiable perturbed optimizers*. Advances in neural information processing systems vol 33, (H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan and H. Lin, ed) pp 9508–19. Curran Associates, Inc.
- [4] Mohanty, S., Nygren, E., Laurent, F., Schneider, M., Scheller, C., Bhattacharya, N., Watson, J., Egli, A., Eichenberger, C., Baumberger, C., Vienken, G., Sturm, I., Sartoretti, G. and Spigler, G. (2020). *Flatland-RL : Multi-Agent Reinforcement Learning on Trains*. arXiv :2012.05893 [cs].