

Optimisation de la sélection des opérateurs d'un algorithme Adaptive Large Neighborhood Search par de l'apprentissage par renforcement profond*

Matthieu Fagot ^{1,2}, Laure Brisoux Devendeville ¹, Corinne Lucet¹

¹ Laboratoire MIS (EA 4290), Université de Picardie Jules Verne, France
{m.fagot, laure.devendeville, corinne.lucet}@u-picardie.fr

² Smile Pickup, France
mfagot@smilepickup.com

Mots-clés : *ALNS, Apprentissage par renforcement profond, DeepRL, VRPTW, heuristique*

1 Introduction

L'entreprise Smile Pickup gère un ensemble de points relais pour gros colis ainsi que la logistique du transport de ceux-ci. Nous avons donc développé un algorithme permettant de générer un plan de transport prenant en compte les contraintes spécifiques de l'entreprise [1].

Nous avons choisi l'Adaptive Large Neighborhood Search de Ropke et Pisinger [2] pour sa robustesse, ses performances et sa praticité. Bien que beaucoup utilisé pour résoudre les problèmes de tournées de véhicules, certaines études montrent que l'apport du module adaptatif peut être très limité en fonction des problèmes étudiés [3].

En 2023, plusieurs articles combinant l'ALNS avec de l'apprentissage par renforcement profond sont sortis [4]. Notre approche se différencie par l'utilisation de caractéristiques spécifiques à la solution étudiée pour sélectionner l'opérateur à appliquer. Nous pensons que l'optimisation de la sélection des opérateurs en fonction de l'instance, de la solution ou de l'évolution de l'algorithme pourrait permettre d'optimiser l'exploration de l'espace des solutions et ainsi accélérer l'algorithme.

Cet article se subdivise en trois parties : la présentation de l'Adaptive Large Neighborhood Search (ALNS), la combinaison de l'ALNS avec l'apprentissage par renforcement et les perspectives à suivre sur cette nouvelle méthode.

2 Adaptive Large Neighborhood Search (ALNS)

L'algorithme Adaptive Large Neighborhood Search [2] fonctionne en itérant sur une solution initiale et en appliquant une série d'opérateurs pour explorer l'espace de recherche.

Pour cela deux types d'opérateurs sont appliqués successivement. Tout d'abord, un opérateur de destruction est appliqué pour dégrader la solution en y retirant des points. Ensuite une heuristique de construction est appliquée afin d'optimiser la réinsertion des éléments retirés pour trouver un minimum local.

La sélection des opérateurs est une étape importante de l'algorithme ALNS. La sélection des bons opérateurs peut aider l'ALNS à trouver des solutions de meilleure qualité plus rapidement.

Le module adaptatif de l'ALNS consiste à associer des poids à chaque opérateur. Les opérateurs dont les poids sont élevés sont plus susceptibles d'être sélectionnés.

* Ce projet est soutenu par le projet PICKOPT (CIFRE n° 2021/0599 entre Smile Pickup et le Laboratoire MIS)

3 Apprentissage par renforcement pour la sélection des opérateurs

L'apprentissage par renforcement est une approche d'apprentissage automatique permettant à un agent d'apprendre à optimiser les récompenses renvoyées par l'environnement étudié.

A chaque itération, l'agent dispose de l'état de l'environnement. C'est à partir de celui-ci qu'il choisit une action qui est ensuite appliquée à l'environnement en échange d'une récompense.

Ainsi, l'agent optimise la séquence d'actions appliquées. Dans notre cas, l'agent est le module de sélection tandis que les actions sont les opérateurs et que le reste de l'algorithme d'ALNS correspond à l'environnement.

Le Deep Reinforcement Learning (DRL) est un sous-domaine de l'apprentissage par renforcement qui utilise des réseaux de neurones profonds pour représenter l'agent. Dernièrement, le Deep Reinforcement Learning (DRL) a permis d'obtenir d'excellents résultats dans différents domaines comme le montre Mnih [5] et son algorithme le Deep Q-Learning (2013) ou Schulman [6] avec le Proximal Policy Optimisation (2017).

En 2023, Kallestad [4] a proposé un algorithme combinant ainsi ALNS et DRL. L'idée étant de représenter l'environnement par un vecteur de caractéristiques représentatives comme le score actuel, le meilleur score ou le nombre d'itérations depuis la dernière amélioration. L'ensemble de ces caractéristiques modélise l'état d'exécution de l'algorithme.

Notre méthode se différencie de celle de Kallestad par le choix et la provenance des caractéristiques représentant l'environnement. Notre ensemble modélise non seulement l'état d'exécution de l'algorithme mais aussi la solution actuelle et l'instance étudiée. Nous utilisons par exemple : le nombre d'intersections par tournées, le nombre de points par tournées, la compacité et la profondeur des tournées.

4 Conclusion

Durant la conférence de la ROADEF, nous présenterons donc cette nouvelle méthode pour améliorer l'algorithme Adaptatif Large Neighborhood Search par l'utilisation d'un réseau de neurones entraîné avec de l'apprentissage par renforcement pour la sélection des opérateurs.

Références

- [1] M. Fagot, L. B. Devendeville, and C. Lucet, "Adaptative local search for a pickup and delivery problem applied to large parcel distribution," in *Optimization and Learning : 6th International Conference, OLA 2023, Malaga, Spain, May 3-5, 2023, Proceedings*, p. 186, Springer Nature.
- [2] S. Ropke and D. Pisinger, "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows," *Transportation science*, vol. 40, no. 4, pp. 455–472, 2006.
- [3] R. Turkeš, K. Sørensen, and L. M. Hvattum, "Meta-analysis of metaheuristics : Quantifying the effect of adaptiveness in adaptive large neighborhood search," *European Journal of Operational Research*, vol. 292, no. 2, pp. 423–442, 2021.
- [4] J. Kallestad, R. Hasibi, A. Hemmati, and K. Sørensen, "A general deep reinforcement learning hyperheuristic framework for solving combinatorial optimization problems," *Eur. J. Oper. Res.*, vol. 309, no. 1, pp. 446–468, 2023.
- [5] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 2013.
- [6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv :1707.06347*, 2017.