Une recherche arborescente Monte-Carlo avec biais dynamique pour le problème de tournées de véhicules

Julien Sentuc[1], Jean-Yves Lucas[2], Tristan Cazenave[1]

[1] LAMSADE, Université Paris-Dauphine, PSL, CNRS, 75016 Paris, France
{ julien.sentuc@dauphine.eu, tristan.cazenave@dauphine.psl.eu }

[2] EDF R&D, Département OSIRIS, 7, avenue Gaspard Monge, 91120 Palaiseau, France
jean-yves.lucas@edf.fr

Mots-clés *: Recherche arborescente Monte-Carlo, Simulation imbriquée avec adaptation de la politique, biais dynamique.*

1 Introduction

La recherche arborescente Monte-Carlo (Monte-Carlo Tree Search, ou MCTS) a été appliquée avec succès dans de nombreux domaines, en particulier ceux des jeux et des problèmes combinatoires. Elle trouve sa première application dans la conception de programmes jouant au jeu de Go ([1]). Une variante du MCTS, le Nested Rollout Policy Adaptation (ou NRPA) a été introduite en 2011 ([2]). Elle a permis d'obtenir des résultats de meilleure qualité que les algorithmes précédents de recherche arborescente, sur de nombreux jeux (record du monde du Morpion Solitaire) et problèmes combinatoires classiques tels que le voyageur de commerce avec fenêtre de temps, ou les tournées de véhicules. Le NRPA a donné lieu à son tour à une extension nommée Generalized NRPA (ou GNRPA) ([3]). Dans le NRPA, le choix par tirage aléatoire d'un nœud de l'arbre de recherche se fait à partir de la politique de choix apprise au cours de la recherche arborescente. Dans le GNRPA, ce tirage est fait en pondérant la probabilité de chaque nœud par un biais dynamique heuristique, qui tient compte des caractéristiques du nœud lui-même. La valeur du biais sera donc différente d'un nœud à l'autre.

2 Le problème de tournées de véhicules

Le problème de la construction de tournées de véhicules (ou VRP) est l'un des plus ancien et des plus étudiés de la littérature. Il trouve son origine dans l'article « The truck Dispatching Problem » de Dantzig et Ramser ([4]). C'est un problème crucial pour de nombreuses société qui, chez des clients, livrent des produits, effectuent des interventions de maintenance, ou rendent des visites commerciales. C'est pourquoi de nombreuses variantes de ce problème ont été étudiées, comme le VRP avec contraintes de capacité (chaque véhicule ne peut emporter qu'une quantité limitée de produits à livrer, ou le VRP avec fenêtre de temps (chaque client doit être visité lors d'une période restreinte). Dans ce travail, nous avons utilisé 56 instances de VRP avec fenêtres de temps du benchmark de Solomon. Ce sont les plus difficiles du benchmark, chacune impliquant la visite de 100 clients. La fonction objectif inclut 3 critères : le nombre de clients non visités, affecté d'un poids de 1 000 000, le nombre de véhicules utilisés, affecté d'un poids de 1 000, et enfin la distance totale parcourue, dont le poids est 1.

3 Le biais dynamique

Dans l'algorithme du NRPA, la probabilité de choisir une action c à une étape i de la résolution est liée à un poids w_{ic}. Ce poids est modifié après chaque découverte d'une nouvelle solution : si celle-ci est la nouvelle meilleure solution, le poids de ses actions sont incrémentées, sinon ce sont les poids des actions de la meilleure solution courante qui sont incrémentés. La probabilité de choisir une action c parmi les k actions possibles est : $p_{ic} = \frac{e^{w_{ic}}}{\sum_k e^{w_{ik}}}$. Le GNRPA généralise la manière dont cette probabilité est calculée, en utilisant une température τ et un biais β : $p_{ic} = \frac{e^{\frac{w_{ic}}{\tau}+\beta_{ic}}}{\sum_k e^{\frac{w_{ik}}{\tau}+\beta_{ik}}}$. Dans notre étude nous avons fixé τ=1 (pas de modification de la température), et le biais est calculé en s'inspirant de l'heuristique constructive de Solomon ([5]), basée sur 3 termes : la distance, le temps d'attente, et enfin le retard (qui pénalise le fait de commencer l'intervention très tôt dans la fenêtre de temps du client). Les poids associés à ces 3 termes ont été fixés par une optimisation séquentielle (dans l'ordre des termes mentionné ci-dessus) sur un sous-ensemble des instances de Solomon.

4 Résultats

Nous avons comparé les résultats du GNRPA sur les 56 instances de Solomon avec une implémentation classique du NRPA, avec le module VRP de l'outil OR-Tools de Google, et avec la meilleure solution connue de chaque instance. OR-Tools a été lancé chaque fois avec un temps d'exécution limité à 1800 secondes. Le NRPA et le GNRPA ont été testés avec 3 niveaux et 100 playouts par niveau, en conséquence leur temps d'exécution est inférieur à 1800 secondes. Cela permet des comparaisons pertinentes.

Le GNRPA fournit toujours de meilleures solutions que le NRPA sauf sur une instance. Sur les 56 instances, le GNRPA fournit de meilleures solutions qu'OR-Tools sur 12 instances, alors que celui-ci trouve de meilleures solutions que le GNRPA sur 35 instances. Sur 9 instances, GNRPA et OR-Tools ont fourni la même solution, qui est aussi la meilleure solution connue.

	NRPA	GNRPA	OR-Tools
Meilleure des 3 solutions (sur 56)	0	12	35
Meilleure solution connue (sur 56)	5	10	13

TAB. 1 – comparaison des 3 méthodes sur 56 instances de Solomon

Références

[1] B. Bouzy and T. Cazenave. 2001. Computer Go: An AI oriented survey. *Artificial Intelligence*, 132(1): 39-103.

[2] C.D. Rosin. 2011. Nested Rollout Policy Adaptation for Monte Carlo tree Search. *In IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, 649-654.

[3] T. Cazenave. 2020. Generalized Nested Rollout Policy Adaptation. In Monte Carlo at IJCAI.

[4] G. Dantzig and J. Ramser. 1959. The Truck Dispatching problem. *Management Science*, 6(1): 80-91.

[4] M. Solomon. 1987. Algorithm for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations research*, 35(2): 254-265.