Apprentissage par renforcement profond pour résoudre des problèmes d'ordonnancement d'atelier avec incertitude

Guillaume Infantes[1], Stéphanie Roussel[2], Pierre Pereira[1],
Antoine Jacquet[1], Emmanuel Benazera[1]

[1] Jolibrain, Toulouse, France
{guillaume.infantes, pierre.pereira, antoine.jacquet,
emmanuel.benazera@jolibrain}@jolibrain.com
[2] ONERA-DTIS, Université de Toulouse, France
stephanie.roussel@onera.fr

Mots-clés : *Ordonnancement atelier, incertitude, apprentissage par renforcement profond*

1 Problématique

L'ordonnancement d'atelier (*Job-Shop Scheduling Problem* - JSSP) est un problème classique d'optimisation combinatoire dans lequel n jobs $J_1, \dots J_n$ doivent être ordonnancés sur m machines de manière à minimiser des critères comme le makespan ou le retard. Chaque job J_i est composé de m opérations notées O_{i1}, ..., O_{im}, une par machine, qui ont chacune une durée et doivent être réalisées dans un ordre donné.

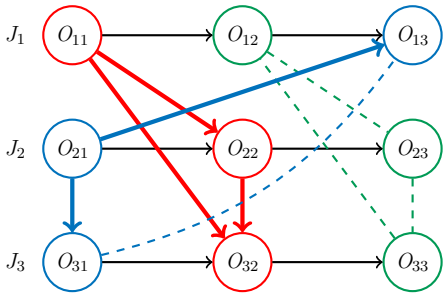
Pour traiter des problèmes plus réalistes, nous nous intéressons à la résolution des JSSPs avec des durées d'opérations incertaines. Plus précisément, nous associons à chaque opération une distribution de probabilité pour sa durée. Notre objectif est de générer un ordonnancement des tâches qui soit robuste, c'est-à-dire qui minimise le makespan obtenu en moyenne (Stochastic JSSP). Plusieurs approches basées sur des métaheuristiques ou des algorithmes génétiques ont été proposées dans la littérature, comme décrit dans [1]. Nous proposons une nouvelle approche, Wheatley [1], qui combine les réseaux de neurones en graphes (Graph Neural Network - GNN) avec des techniques d'apprentissage par renforcement et des récompenses éparses.

2 Description de l'approche

Comme dans [3], la formalisation du problème sous la forme d'un processus décisionnel de Markov repose sur le graphe disjonctif classiquement utilisé pour les JSSPs. Un état du processus est représenté par un graphe dans lequel certaines opérations ont été ordonnancées. Comme illustré sur la figure 1a, O_{11} est avant O_{22} mais l'ordre entre O_{13} et O_{31} n'est pas encore décidé. Une action consiste à sélectionner une opération dont tous les prédécesseurs ont déjà été sélectionnés, par exemple O_{13}, et la mettre à la suite de ce qui est déjà sélectionné sur la machine correspondante, O_{21} dans l'exemple. La seule récompense non nulle est celle associée à l'état dans lequel toutes les opérations sont sélectionnées. Elle est égale au makespan calculé avec des durées aléatoirement tirées pour chaque opération. Dans ce modèle, les actions sont donc déterministes et l'incertitude vient uniquement de la récompense de l'état final.

Dans notre approche d'apprentissage par renforcement, l'agent sélectionne des opérations, observe les ordonnancements correspondants et reçoit la récompense finale. L'objectif est de trouver une politique qui minimise le makespan moyen sur un ensemble de problèmes test qui ne sont pas utilisés pendant la phase d'apprentissage. Pour cela, nous considérons une politique

1. Code Open Source - https://github.com/jolibrain/wheatley/



(a) Exemple d'état du MDP

Taille	Wheatley 15x15	MOPNR	OR-Tools/CP-SAT average	OR-Tools/CP-SAT real
10x10	**1232**	1252	1258	*1002*
15x15	**1889**	1988	1935	*1469*
20x15	**2188**	2314	2335	*1756*
20x20	**2608**	2708	2759	*2071*

(b) Résultats sur instances de Taillard

paramétrique et utilisons l'algorithme de type acteur-critique on-policy Proximal Policy Optimization (PPO) [2], avec le masquage d'actions. L'acteur est la politique stochastique actuelle et est directement dérivé des logits de nœuds de l'agent. Le critique estime la qualité de l'état actuel à partir du logit du graphe de notre agent.

Le graphe disjonctif correspondant à chaque état est capturé par le GNN qui permet de représenter les différents arcs clés (précédence, conflits, etc.). Pour que les messages passent entre les nœuds, nous ajoutons des arcs inverse aux précédences, ainsi que des arcs entre opérations utilisant la même machine. Du point de vue de l'embedding, chaque arc a pour attribut son type. Chaque nœud du graphe a pour attribut les caractéristiques de l'opération associée, dont les attributs de la distribution de probabilité de durée. Le GNN renvoie une valeur associée à chaque nœud ainsi qu'une valeur pour le graphe global.

3 Expérimentations

Nous avons testé l'approche sur des instances de Taillard avec de l'incertitude sur les durées via une distribution de probabilité triangulaire. Nous avons entraîné Wheatley sur des instances de taille 15×15 et l'évaluons sur des instances de taille différente. Nous comparons le résultat obtenu par des règles de priorité (par exemple MOPNR). Nous comparons avec CP-SAT d'OR-Tools avec lequel nous calculons l'ordonnancement obtenu sur chaque machine avec la durée moyenne des opérations et que nous appliquons à la manière d'un Schedule Generation Scheme (SGS) avec la vraie durée de ces dernières (time-out de 10 secondes). Nous utilisons également ce solveur avec les vraies durées pour avoir une borne inférieure de la valeur du makespan. Comme illustré sur la table 1b dans laquelle chaque case est la moyenne des makespan obtenu sur 100 instances d'une taille donnée, Wheatley est le plus proche en moyenne de la borne inférieure. Ces résultats montrent que Wheatley est une approche particulièrement performante pour traiter le SJSSP. De plus, les modèles appris sont capable de généraliser puisque les résultats obtenus sont de bonne qualité sur des instances plus grandes que celles utilisées pour l'entraînement. Les travaux futurs consistent à étendre l'approche pour des problèmes avec resource (Resource-Constrained Project Scheduling - RCPSP).

Références

[1] Mohammed Boukedroun, David Duvivier, Abdessamad Ait-el Cadi, Vincent Poirriez, and Moncef Abbas. A hybrid genetic algorithm for stochastic job-shop scheduling problems. *RAIRO : Operations Research (2804-7303)*, 57(4), 2023.

[2] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[3] Cong Zhang, Wen Song, Zhiguang Cao, Jie Zhang, Puay Siew Tan, and Xu Chi. Learning to dispatch for job shop scheduling via deep reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1621–1632. Curran Associates, Inc., 2020.