

Étude de paysages de fitness pour l'ordonnancement d'atelier de type jobshop

Israël Tsogbetse¹, Julien Bernard², Hervé Manier¹, Marie-Ange Manier¹

¹ FEMTO-ST, CNRS, UTBM, 90010 Belfort, France

{israel.tsogbetse, herve.manier, marie-ange.manier}@utbm.fr

² FEMTO-ST, CNRS, UFC, 25000 Besançon, France

julien.bernard@femto-st.fr

Mots-clés : *optimisation, job shop, codage, paysage de fitness*

1 Introduction

L'optimisation combinatoire, dès lors que la taille des problèmes augmente, a recours aux méthodes dites approchées dont les métaheuristiques. De nombreuses études dans la littérature se focalisent sur l'efficacité des opérateurs ou de l'hybridation de diverses méthodes [1, 2]. Sur l'ensemble des paramètres caractérisant les métaheuristiques, le codage des solutions est souvent choisi sans véritable justification ni étude de pertinence. Or ce paramètre joue un rôle important dans la définition et le parcours de l'espace de recherche, au même titre que des opérateurs de voisinage par exemple [3, 4]. Par ailleurs, la pertinence d'un codage peut être remise en cause lorsqu'on change de variante au sein d'une même classe de problème. Dans cette étude, on réalise une première analyse sur les paysages de fitness générés pour un problème d'ordonnancement de type job shop de base, en fonction de différents codages. Les codages sont testés avec plusieurs métriques qui renseignent sur leurs caractéristiques fondamentales.

2 Problème job shop, codages et métriques

Le problème étudié consiste en un ensemble de jobs, chaque job est constitué d'opérations liées par une relation d'ordre total (contraintes de précedence ou de gamme), chaque opération devant être réalisée sur une ressource (machine) donnée. Le problème consiste à trouver un ordre et des dates d'exécution de l'ensemble des opérations sur les machines, en respectant les contraintes de précedence, et en minimisant le temps total d'exécution (C_{\max}).

Pour ce problème, on définit un codage qui permet de représenter chaque solution possible. Trois types de codage sont étudiés : *par job* : liste d'entiers dont la taille correspond au nombre total d'opérations, chaque élément est un numéro de job ; *par opération* : liste d'entiers distincts dont chaque élément est un numéro d'opération ; *par machine* : une liste de permutation pour chaque machine comprenant les opérations qu'elle doit exécuter ;

Pour les évaluer, on considère une méthode de génération de solutions par voisinage qui utilise un opérateur classique de type swap. L'évaluation des codages se fait via quatre métriques [5] : *distribution de types de point* : classification des solutions au regard de leur voisinage ; *densité des points de fitness* : histogramme des fitness des solutions ; *degré de neutralité* : proportion de voisins ayant la même fitness ; *fonction d'autocorrélation des fitness (rugosité)* : corrélation des fitness avec une marche aléatoire entre deux pas éloignés d'une distance ℓ qu'on fait varier.

	Job	Opé	Mach
lop	0.35	0.65	0.13
swv07	0.21	0.73	0.37
ta62	0.31	0.85	0.5

TAB. 1 – Degré de neutralité pour les instances `lop`, `swv07` et `ta62` suivant les codages utilisés

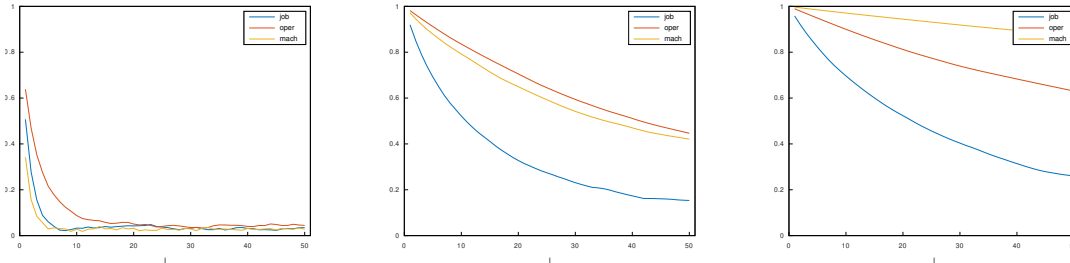


FIG. 1 – Fonction d'autocorrélation pour les instances `lop`, `swv07` et `ta62` suivant les codages utilisés

3 Expérimentations et résultats

Des tests ont été réalisés sur un ensemble de benchmarks de la littérature, dont un extrait est fourni ici (tableau 1 pour le degré de neutralité, et figure 1 pour l'autocorrélation) pour trois instances et les codages utilisés. La taille des instances varie de 2×3 (2 jobs, 3 opérations par job) à 100×20 . L'analyse du tableau 1 montre que pour chaque instance, la neutralité peut varier de manière significative en fonction du codage utilisé, avec une différence marquée du codage par opération par rapport aux deux autres. Les codages pour lesquels un voisinage de type swap a une plus forte probabilité de donner un ordonnancement identique semblent donner un degré de neutralité plus élevé. Sur la figure 1, on observe que la fonction d'autocorrélation a des valeurs proches pour les codages par opération et par machine (pour lesquels il existe des solutions infaisables), alors qu'elle a des valeurs plus faibles pour le codage par job (pour lequel il n'y a pas d'infaisables).

4 Conclusion et perspectives

Les valeurs des métriques sont nettement impactées par le choix du codage et la taille des instances. Les prochains travaux consisteront à tester d'autres opérateurs de voisinage sur ces codages. Ensuite les couples codages-voisinages seront analysés pour le problème de job shop avec des contraintes additionnelles, dont des contraintes de transport.

Références

- [1] D.C. Mattfeld and C. Bierwirth and H. Kopfer. A search space analysis of the Job Shop Scheduling Problem. *Annals of Operations Research*, 86441–453, 1999.
- [2] Kenneth Sörensen and Marc Servaux and Fred Glover. A history of metaheuristics. *Handbook of heuristics*, 1-18, 2018.
- [3] Franz Rothlauf and David E. Goldberg. Redundant representations in evolutionary computation. *Evolutionary Computation*, 11 381-415, 2003.
- [4] Ivan Vlašić and Marko Durasević and Domagoj Jakobović. A comparative study of solution representations for the unrelated machines environment. *Computers & Operations Research*, 123 (105005), 2020.
- [5] Katherine M. Malan and Andries P. Engelbrecht. A survey of techniques for characterising fitness landscapes and some possible ways forward *Information Sciences*, 241 : 148-163, 2013.