

# Etude de l'impact du couple codage et opérateur de croisement pour la résolution de problèmes de permutation

Hind Mohammed Ali<sup>1</sup>, Wahabou Abdou<sup>2</sup>, Pascal Chatonnay<sup>1</sup>, Christelle Bloch<sup>1</sup>, François Spies<sup>1</sup>

<sup>1</sup> Univ. Bourgogne Franche-Comté - University of Franche-Comté - FEMTO-ST Institute, UMR CNRS 6174

1 Cours Leprince-Ringuet - 25200 Montbéliard, France

{hind.mohammed\_ali,pascal.chatonnay,christelle.bloch, francois.spies}@univ-fcomte.fr

<sup>2</sup> LE2I - UMR CNRS 6306, University of Burgundy, Dijon, France

{wahabou.abdou}@u-bourgogne.fr

**Mots-clés** : *Algorithmes génétiques, codage, croisement, problèmes de permutation.*

## 1 Introduction

Dans la littérature, les algorithmes génétiques (ou AGs) sont souvent utilisés pour résoudre des problèmes d'optimisation dits « de permutation », le plus connu étant le problème de voyageur de commerce (ou *Traveling Salesman Problem* – TSP). Mais, les auteurs étudient souvent la performance de l'AG pour résoudre un problème donné. Ils fournissent généralement une comparaison empirique entre leur AG et des travaux antérieurs, pour des problèmes de référence reconnus, et exprimée dans l'espace des objectifs. Ce travail, lui, utilise aussi des indicateurs dans l'espace des variables. Le but est de comprendre des phénomènes sous-jacents dans le comportement interne de l'AG. Après avoir situé notre travail par rapport à l'état de l'art, nous étudions ces aspects pour un codage indirect, appelé « numérique – symbolique » ou CNS, basé sur le rang occupé par chaque symbole de la permutation dans une liste de référence. L'étude est menée en comparaison avec le codage de permutation classique. Des exemples et des tests illustrent comment l'utilisation d'opérateurs de croisement 1-point (ou dérivés) donne des comportements différents en fonction du codage.

## 2 Travaux connexes et nouvelle contribution

Une première classe d'articles étudiant le comportement interne des AGs, regroupe les travaux de référence classiques introduisant des fondements théoriques (ou « *schema theory* »). Ils considèrent un codage binaire et un croisement 1-point et démontrent que la performance de l'AG est liée à la conservation de « blocs intéressants » transmis des parents aux enfants. Ils en déduisent aussi que la position des gènes dans le chromosome (et donc des valeurs prises par ces gènes, ou allèles) doit être étudiée avec soin, en particulier en termes de distance, de proximité entre les gènes. Quand deux variables ne sont pas indépendantes elles doivent être le plus près possible l'une de l'autre dans le chromosome pour moins risquer de « casser » un bloc intéressant au moment du croisement.

Une deuxième classe d'articles, celle qui comporte sans doute le plus de références, étudie d'autres codages et d'autres opérateurs ou paramètres de l'algorithme. Mais généralement la performance de l'algorithme est mesurée sous forme d'indicateurs globaux, exprimés dans l'espace des objectifs (tels que l'espace mémoire, le temps d'exécution, ou l'écart par rapport à l'optimum connu pour des problèmes de référence), mais sans fournir d'analyse détaillée du fonctionnement interne.

Une troisième classe utilise des indicateurs dans l'espace des objectifs mais adapte dynamiquement l'évolution. Ainsi [2] modifie la probabilité d'utilisation de couples d'opérateurs (croisement, mutation) selon l'impact de leur utilisation sur le front de Pareto pour le TSP multi-objectifs.

Une quatrième classe fait cette étude plus détaillée dans l'espace des variables (des individus) pour d'autres types de codage et opérateurs que ceux de la théorie initiale. Ainsi, [1] donne une analyse détaillée et des résultats expérimentaux comparant plusieurs opérateurs de croisement pour le codage de permutation. Les auteurs soulignent aussi qu'en fonction du type de problème de permutation considéré, les caractéristiques à conserver ne sont pas les mêmes. Pour le TSP, il s'agit de conserver le lien entre des valeurs successives (ou « *edge* »), alors que pour l'ordonnement d'une machine c'est la position des valeurs dans l'individu qui prime. Enfin dans certains problèmes, notamment en ordonnancement avec temps de « *setup* » dépendants de la séquence, les deux sont importants. Ces auteurs utilisent donc plusieurs indicateurs pour mesurer la capacité d'un croisement à conserver ces caractéristiques (*edge based indicator*, *position based indicator*, ...) et étudient également l'impact sur la qualité des individus dans l'espace des objectifs.

Le travail proposé ici utilise la même démarche pour un autre codage, CNS, et le compare au codage de permutation. CNS est basé sur une liste de référence, nommée « *alphabet* », qui contient tous les symboles du problème (les éléments constituant la permutation). L'ordre de cette liste peut être aléatoire ou heuristique. Chaque valeur codée dans un individu donne la position du symbole dans l'alphabet pour la procédure de décodage. Celle-ci parcourt l'individu, trouve le symbole associé à chaque valeur et retire alors le symbole de la liste. Par exemple, si l'alphabet est A,B,C,D,E,F le codage 1-1-1-1-1-1 représente la permutation A-B-C-D-E-F. CNS convient pour tous les problèmes de permutation, y compris ceux dont les caractéristiques à conserver par croisement concernent aussi bien la position que les valeurs successives. Son principal avantage est que l'utilisation directe d'un croisement 1-point ne produit pas d'individu non faisable. Alors que, pour le codage de permutation, ce croisement peut dupliquer le même symbole dans un individu, ce qui nécessite soit l'utilisation d'un croisement plus complexe, tel que PMX, soit une procédure de réparation. Mais utiliser CNS n'est pas sans conséquence sur le fonctionnement interne de l'AG. Pour CNS, l'enjeu de la transmission des « blocs intéressants » concerne tout autant les positions que les successions de valeurs. Ainsi, par exemple, le même « bloc » contenant les valeurs successives 1-3 n'aura pas la même valeur décodée selon sa position dans l'individu. Pour l'alphabet pris en exemple, dans l'individu 1-3-2-1-1-1 ce bloc représente la succession de symboles A-D, alors que dans l'individu 1-1-1-3-1-1 il représente C-F. Ce travail étudie les avantages et inconvénients de CNS associé au croisement 1-point, par comparaison avec le codage de permutation et sur la base d'indicateurs similaires à ceux utilisés par [1].

### 3 Conclusions et perspectives

Ce papier propose l'étude de l'impact du choix d'un couple de codage et d'opérateur de croisement sur le comportement d'un AG. Il compare des caractéristiques liées à la transmission de propriétés entre parents et enfants, aux problématiques éventuelles de réparation, de diversité au sein des générations. Plus globalement il étudie le parcours de l'espace des variables et de celui des objectifs, par comparaison entre deux types de codage, dont le codage de référence par permutation.

### Références

- [1] Marie-Claude Portmann, and Antony Vignier. Performances Study on crossover operators keeping good schemata for some scheduling problems. *Proceedings of the Second Genetic & Evolutionary Computation Conference - GECCO'2000*, 2000, Las Vegas, USA, Morgan Kaufmann, pp.331-338, 2000.
- [2] Semya Elaoud, Jacques Teghem, and Ta Loukil. Multiple crossover genetic algorithm for the multiobjective traveling salesman problem. *Electronic Notes in Discrete Mathematics*, 36:939-949, Elsevier, 2010.