

# Diversité, Dispersion Réactive et Croisement Informé dans l'Approche Génétique pour la Coloration de Graphe

Daniel Cosmin Porumbel,<sup>1,2</sup> Jin-Kao Hao,<sup>1</sup> et Pascale Kuntz<sup>2</sup>

<sup>1</sup> Leria, Université d'Angers, 2, Boulevard Lavoisier, 49045 Angers Cedex 01 – France  
porumbel,hao@info.univ-angers.fr

<sup>2</sup> Lina, Polytech'Nantes, rue Christian Pauc, BP 50609, 44306 Nantes Cedex 3 – France.  
pascale.kuntz@univ-nantes.fr

**Mots-Clés :** *Diversité population, écart individus, algorithme génétique, croisement informé.*

## 1 Introduction et Objectifs

Les algorithmes génétiques ont traditionnellement atteint de très bons résultats pratiques pour de nombreux problèmes de recherche opérationnelle et d'optimisation combinatoire, y compris la coloration de graphe. Dans cette communication, nous nous concentrons sur deux facteurs cruciaux dans les algorithmes génétiques : la diversité de la population et l'opérateur de croisement.

En effet, un challenge connu dans le calcul évolutionniste concerne le risque de convergence prématurée et le manque de diversité. Pour faire face à ce type de problèmes, nous contrôlons *l'écart* au sein de la population en utilisant une *mesure de distance* entre les individus. Nous introduisons une procédure générique d'écart qui permet d'une part de décider les individus acceptables dans la population et ceux qui doivent être enlevés, et d'autre part de contrôler l'application des mutations.

Un objectif essentiel est d'imposer, à tout moment, une *distance minimum* entre toute paire d'individus. En utilisant des travaux précédents [2], nous considérons que les individus les plus adaptés sont regroupés dans des sphères de rayon fixe (noté  $R$ ) ; ce rayon définit ainsi la distance minimum imposée dans la population. De plus, une attention particulière est portée à l'opérateur de remplacement et aux mutations. En fait, les mutations sont utilisées essentiellement dans une phase de dispersion réactive qui est déclenchée uniquement lorsqu'il n'est plus possible d'assurer la diversité uniquement par croisement et recherche locale[1].

Un autre challenge dans les algorithmes évolutionnistes consiste à concevoir un opérateur de croisement pertinent pour le problème en question, i.e. qui préserve les bonnes caractéristiques des parents et perturbe les mauvais. Pour la coloration de graphe, on considère une coloration comme une partition de l'ensemble de sommets : le croisement sélectionne certaines classes de couleurs qui sont assemblées pour engendrer l'enfant. Cette sélection est traditionnellement faite uniquement en fonction de la taille des classes, mais nous introduisons ici une mesure "bien informée" pour évaluer les classes, et ainsi pour mieux choisir l'information transmise à la génération suivante.

## 2 L’algorithme Evo-Div

Les opérations principales de notre algorithme (ci-après appelé Evo-Div) sont les suivantes :

**Tant que** *condition d’arrêt* **non** atteinte :

A. **Répéter**

1.  $O = \text{Croisement}(\text{parentsAléatoires}(Population, n))$  /\*  $n \geq 2$  \*/
2. **Si**  $rejets \geq maxRejets$  **alors**  $O = \text{mutation}(O)$
3.  $O = \text{rechercheLocale}(O, maxIter)$
4.  $rejets++$  /\*initialement,  $rejets = 0$ \*/

**Jusqu’à ce que**  $\text{acceptEnfant}(Population, O)$

B.  $Population = Population \cup \{O\} - \text{indivRemplacé}(Population)$

Il existe de nombreux indicateurs pour mesurer la diversité, mais nous employons deux indices à base de distances : (i) l’écart moyen  $S$  représente la distance moyenne d’un individu à l’autre, et (ii) l’écart *minimum*  $S_{\min}$  représente la distance minimum entre deux individus. Notre procédure générique d’écart a deux objectifs : (i) garder  $S_{\min}$  au dessus de  $R$ , (ii) rendre l’écart moyen  $S$  aussi élevé que possible. En principe, le premier point est abordé par la procédure de rejet de l’enfant ( $\text{acceptEnfant}$ ) et le deuxième par l’opérateur de remplacement ( $\text{indivRemplacé}$ ). Notons que ces deux routines essaient uniquement d’influencer le processus naturel de reproduction, sans sacrifier la qualité. Il serait facile d’assurer la diversité en acceptant des petites détériorations de qualité via des mutations aléatoires. Une nouveauté d’Evo-Div est le fait que la mutation est appliquée uniquement lorsque le processus naturel de reproduction a échoué  $maxRejets$  fois. Normalement  $maxRejets$  est très grand et les mutations sont ainsi très rares. La phase dispersion décroît  $maxRejets$  afin de permettre plus de mutations dans certaines situations spécifiques – e.g. si tous les individus restent pendant plusieurs générations dans une vallée de grande taille autour d’un plateau.

Pour définir le croisement nous introduisons une fonction de score qui permet de choisir les classes utilisées pour la construction de l’enfant. Le premier critère de score privilégie les classes avec le moins d’arêtes en conflit. Cependant, comme Evo-Div rencontre souvent des classes sans conflit lorsqu’il converge vers des colorations “presque optimales”, un deuxième critère est nécessaire : si l’on doit choisir entre deux classes avec le même nombre de conflits, on préfère la classe la plus grande. Comme les tailles de classes peuvent être assez homogènes, un troisième critère est employé pour renforcer la discrimination. Il se base sur l’hypothèse selon laquelle un sommet de haut degré est plus contraint et ainsi plus difficile à colorier. Par conséquent, Evo-Div considère qu’une classe bien coloriée avec des sommets de plus haut degrés a une valeur plus élevée qui favorise son héritage.

**Conclusion** Nous avons développé un nouvel algorithme évolutionniste (Evo-Div) qui est capable de converger rapidement vers de bonnes configurations à l’aide d’un croisement informé tout en évitant une convergence prématurée. Nous avons testé expérimentalement Evo-Div pour le problème de coloration et montré qu’il permet de retrouver toutes les meilleures solutions – sauf 2 – du challenge DIMACS et d’atteindre une nouvelle borne supérieure pour une instance. Les principes développés sont suffisamment généraux pour être appliqués à d’autres problèmes d’optimisation combinatoire.

## Références

- [1] C.D. Porumbel, J.K. Hao, and P. Kuntz. Diversity control and multi-parent recombination for evolutionary graph coloring algorithms. In *EvoCOP*, volume 5482 of *LNCS*, pages 121–132, 2009.
- [2] C.D. Porumbel, J.K. Hao, and P. Kuntz. A search space “cartography” for guiding graph coloring heuristics. *Computers & Operations Research*, 37 :769–778, 2010.