Sélection autonome d'opérateurs par dominance pour la recherche locale

Nadarajen Veerapen **Frédéric Saubion**
LUNAM Université, Université d'Angers, LERIA
{prenom.nom}@univ-angers.fr

Résumé

Cet article présente l'étude d'une méthode de sélection d'opérateurs pour un processus de recherche locale. L'objectif principal est de proposer une méthode générique de recherche locale qui gère la sélection d'opérateurs à partir d'un ensemble d'opérateurs disponibles. Ces opérateurs sont construits à partir de relations de voisinages et de fonctions de sélection de voisin. La sélection d'opérateurs utilise le concept de Pareto dominance en se basant sur des mesures de qualité et de distance. Cette dernière est déterminée à l'aide de deux métriques. L'algorithme de contrôle est codé en langage COMET afin d'être facilement utilisable dans d'autres algorithmes de recherche locale. L'emphase étant mise sur les problèmes de permutation, nous présentons des résultats expérimentaux pour le QAP et le ATSP afin d'évaluer l'efficacité de la méthode.

Abstract

This paper presents a study for the dynamic selection of operators in a local search process. The main purpose is to propose a generic autonomous local search method which manages operator selection from a set of available operators, built on neighborhood relations and neighbor selection functions, using the concept of Pareto dominance with respect to quality and diversity. The latter is measured using two different metrics. This control method is implemented using the COMET language in order to be easily introduced in various constraint local search algorithms. Focusing on permutation-based problems, experimental results are provided for the QAP and ATSP to assess the method's effectiveness.

1 Introduction

Les métaheuristiques ont largement été adoptées comme méthodes de résolution efficaces pour l'optimisation combinatoire et les problèmes de satisfaction de contraintes. Néanmoins, ces approches demandent souvent une expertise conséquente aussi bien du problème que de la méthode de résolution. Une approche récente, la construction de stratégies de contrôle haut niveau, a pour but de rendre les techniques d'optimisation plus facile à utiliser [4].

Du point de vue de la recherche locale (LS), un bon algorithme LS [10] doit explorer l'espace de recherche de façon efficace en quête de la solution optimale. Cela implique de jongler entre deux objectifs généralement divergents : l'intensification (converger vers un optimum local) et la diversification (échantillonner convenablement les différentes parties de l'espace de recherche). L'efficacité de ces deux stratégies dépend largement de la (des) structure(s) de voisinage choisie(s). Cet équilibre peut être contrôlé par le biais d'opérations de base (des mouvements) qui sont appliquées au cours du processus de recherche. De ce fait, un nombre croissant de travaux essaient de construire des algorithmes plus autonomes [8]. Cette tendance a été explorée dans le contexte de la recherche réactive [1], basée sur des travaux fondateurs comme le tabou réactif [2] ou le recuit simulé adaptatif [13]. Le concept général d'un changement systématique de relation de voisinage, basé sur le fait qu'un optimum est défini en fonction d'un voisinage et donc que différents voisinages n'auront pas forcément les mêmes optima, a été développé dans la recherche à voisinage variable (VNS) [9]. Il en existe des versions adaptatives, par exemple [11]. Néanmoins, comme récemment mentionné dans [15], la plupart des algorithmes LS gèrent la diversité et la qualité comme deux objectifs opposés et utilisent donc des étapes alternées de diversification et d'intensification. À titre d'exemple, la recherche locale itérée (ILS) [17] est une succession alternée d'une perturbation et d'une recherche locale (une exploration complète d'un voisinage ou une LS

Cet article est une traduction de l'article *Pareto Autonomous Local Search* [27] présenté à LION5, Rome, 2011.

plus complexe). Les algorithmes LS se focalisent le plus souvent sur la qualité de la solution la plus récente, mais peuvent utiliser des mécanismes de prohibition pour éviter le piège des optima locaux. En accord avec les remarques de [15], nous estimons qu'une meilleure coordination peut être atteinte entre les deux objectifs, qualité et diversité, pouvant être évaluée à travers un compromis fluctuant en réponse à l'état du processus de recherche.

Des travaux récents sur les algorithmes évolutionnaires proposent de nouvelles techniques pour la sélection adaptative d'opérateurs. *Compass* [18] évalue la performance d'un opérateur à travers la scalairisation de l'amélioration de la qualité du parent vers l'enfant, la variation en diversité et le temps d'exécution. Dans [6], un *Bandit Manchot Multi-Bras* est utilisé pour sélectionner l'opérateur qui maximise la somme de deux valeurs, la première représentant la performance de l'opérateur et la seconde assurant que l'opérateur soit sélectionné un nombre infini de fois. En utilisant les principes de *Compass*, un algorithme adaptatif de recherche locale a été présenté dans [22]. Dans ces travaux la performance d'un opérateur est définie par rapport à un compromis statique entre la qualité et la diversité.

Dans cet article nous présentons d'abord un modèle algorithmique général pour la recherche locale définie comme étant un processus de sélection d'opérateurs de mouvement à partir d'un ensemble d'opérateurs disponibles, qui sont une combinaison d'une relation de voisinage et d'une sélection de voisin au sein de ce voisinage. Ensuite l'objectif de l'algorithme est de choisir et d'appliquer un opérateur à la solution actuelle afin de construire progressivement le chemin de recherche. Notre approche est donc double : 1) introduire un nouveau compromis entre qualité et diversité dans la recherche et 2) offrir un framework de contrôle pouvant utiliser des opérateurs génériques afin de résoudre une large palette de problèmes avec comme objectif d'offrir des solutions d'optimisation à des utilisateurs non-experts en les soulageant du travail de conception et de réglage d'algorithmes.

Nous présentons des fonctionnalités de contrôle pour la résolution de problèmes de permutation, c.-à-d. les problèmes dont la configuration peut être modélisée par des permutations. Ce cadre général nous permet de définir différents opérateurs en combinant des voisinages de permutations et des sélecteurs basiques.

À chaque étape de la recherche, les opérateurs sont sélectionnés d'après le principe de Pareto dominance, calculée par rapport aux performances enregistrées de chaque opérateur en termes d'intensification et de diversification. De plus, comme notre objectif est d'offrir un cadre de développement générique aux utilisateurs de recherche locale, nos fonctionnalités de contrôles sont écrites en COMET [26], un langage dédié à la conception d'algorithmes de recherche locale et intégrant la gestion des contraintes. Afin de souligner la généralité de notre contrôleur, nous le testons ensuite sur deux problèmes de permutation bien connus : le problème d'affectation quadratique et le problème du voyageur de commerce asymétrique.

Le reste de l'article est organisé en quatre sections. La section 2 pose les définitions permettant de manipuler les voisinages, les sélecteurs et les opérateurs pour les problèmes de permutation. Dans la section 3 nous présentons le cadre du contrôle, deux mesures de distance et la méthode de sélection Pareto. Ceci est suivi par le protocole de test et les résultats à la section 4. Enfin, la section 5 clos l'article par la conclusion ainsi que quelques possibilités de recherches futures.

2 Voisinages, sélecteurs et opérateurs

L'objectif de cette section est de présenter une description formelle des problèmes basés sur les permutations ainsi que les opérateurs qui leur sont associés. Dans [23], les auteurs considèrent formellement différents voisinages et définissent des distances associées à ces voisinages. Comme mentionné ci-dessus, puisque notre but est de gérer dynamiquement les opérateurs en fonction de leur comportement et de leurs propriétés, ces métriques nous intéressent tout particulièrement. Néanmoins, dans [23], les auteurs ne considèrent que des méthodes utilisant un unique opérateur et les métriques pouvant être utilisées pour évaluer la diversité de la trajectoire de la recherche locale dépendent entièrement de l'opérateur.

Ici nous avons pour objectif d'offrir une description simple et générique de ce que sont les voisinages et les opérateurs qui peuvent être utiles pour définir de nouveaux opérateurs et gérer leur application en fonction de leur impact sur le processus de recherche. Un deuxième objectif est de proposer un framework capable de comparer des voisinages et des sélecteurs dans le contexte d'une procédure de recherche multiopérateur.

2.1 Définitions générales

Cette section a pour but de définir clairement ce que sont un voisinage, la sélection d'un voisin et donc un opérateur, ainsi que les différentes notions associées au processus de recherche.

2.1.1 Voisinage

Soit $\mathcal{S}$ l'espace des solutions candidates. Une relation de voisinage est une relation binaire irréflexive

$\mathcal{N} \subseteq \mathcal{S}^2$ sur l'espace de recherche. Dans la majorité des cas, la relation est aussi symétrique.

2.1.2 Trajectoire de recherche

Soit une relation de voisinage $\mathcal{N}$, nous définissons l'ensemble des trajectoires de recherches par $\mathcal{P}_\mathcal{N} = \{s_1 \cdots s_n \in \mathcal{S}^* | \forall i > 1, (s_{i-1}, s_i) \in \mathcal{N}\}$, où $\mathcal{S}^*$ représente classiquement l'ensemble des mots construits sur $\mathcal{S}$. Donc, toute paire (s, s') d'éléments de $\mathcal{S}$, telles que $(s, s') \in \mathcal{N}^+$, définie une classe d'équivalence sur l'ensemble $\mathcal{P}_\mathcal{N}$, qui correspond à tous les chemins reliant s à s'. Nous notons cet ensemble $\mathcal{P}_\mathcal{N}/(s, s')$. Dans la plupart des cas, le voisinage est complet, c.-à-d. $\forall s, s' \in S, \mathcal{P}_\mathcal{N}/(s, s') \neq \emptyset$.

2.1.3 Distances

En fait, la relation de voisinage définit la structure déclarative de l'espace de recherche. Nous pouvons ainsi définir la distance entre s et s' comme $d_\mathcal{N}(s, s') = min_{p \in \mathcal{P}_\mathcal{N}/(s,s')} |p|$, où $|p|$ est la longueur classique d'un mot. Par définition, nous imposons que $d_\mathcal{N}(s, s) = 0$. Notons qu'il est possible de requérir que $\mathcal{N}$ soit symétrique afin que d soit une distance.

2.1.4 Combinaison de voisinages

Afin d'exprimer des structures de voisinages plus complexes, nous notons $\mathcal{N} \circ \mathcal{N}'$ la composition et $\mathcal{N} \cup \mathcal{N}'$ l'union qui sont les constructeurs de voisinage les plus communément utilisés. Un voisinage composé avec lui-même est noté $\mathcal{N}^2$ et $\mathcal{N}^{n+1} = \mathcal{N} \circ \mathcal{N}^n$.

2.1.5 Paysage de recherche

Considérons maintenant le paysage de recherche, nous introduisons d'abord la relation d'ordre $<$ sur $\mathcal{S}$ correspondant à l'ordre induit par la fonction objectif du problème. Notons que nous considérons ici uniquement les problèmes de minimisation sans perte de généralité.

2.1.6 Paysage opérationnel

Il nous faut maintenant introduire la structure opérationnelle de la recherche locale afin d'explorer la relation de voisinage.

Dans ce contexte, un sélecteur est une fonction qui procède à une sélection sur un voisinage, éventuellement guidé par la relation d'ordre $<$ et défini par $\sigma : \mathcal{S} \times 2^{\mathcal{S}^2} \mapsto \mathcal{S}$ (ici la sélection retourne un unique voisin), tel que $(s, \sigma(s, \mathcal{N})) \in \mathcal{N}^=$ (la clôture réflexive de $\mathcal{S}$ afin d'inclure l'identité). Un opérateur est alors défini par une paire $(\mathcal{N}, \sigma)$.

Considérons à nouveau les chemins induits par un opérateur $\mathcal{P}_o = \bigcup_{n>1} \{s_1 \cdots s_n \in \mathcal{S}^* | o = (\mathcal{N}, \sigma), \forall i > 1, s_i = \sigma(s_{i-1}, \mathcal{N})\}$ Afin de simplifier la notation, nous utilisons $o(s) = \sigma(s, \mathcal{N})$ quand $o = (\mathcal{N}, \sigma)$ puisque o peut être considéré comme une fonction sur S. Nous notons $o \circ o'$ la composition entre opérateurs, o^2 la composition de o avec lui-même et $o^{n+1} = o \circ o^n$.

Notons, ici, que nous avons seulement l'inclusion $\mathcal{P}_o \subseteq \mathcal{P}_\mathcal{N}$, puisque certains chemins du voisinage ne peuvent pas forcément être construits par les opérateurs dès lors qu'ils contiennent un processus de sélection entre les voisins. De plus, même s'il existe un chemin dans $\mathcal{P}_o$ de s à s', il n'existe pas forcément un chemin de s' à s. Du fait de cet aspect non symétrique des opérateurs l'utilisation d'une distance simple sur les chemins créés par les opérateurs n'est pas forcément évidente. Nous pouvons maintenant gérer la recherche locale à voisinages multiples en composant ou en réunissant des relations de voisinage.

2.2 Permutations

Concentrons-nous maintenant sur les permutations qui correspondent à l'encodage que nous utiliserons dans nos problèmes. Notre objectif est de proposer un panorama détaillé des opérateurs possiblement utilisables dans ce contexte.

Soit $\Pi(n)$ l'espace de recherche, c.-à-d. l'ensemble de toutes les permutations de l'ensemble $\{1, \ldots, n\}$. Si $\pi \in \Pi(n)$ et $1 \leq i \leq n$, alors π_i dénote l'élément i dans π.

Comme décrit dans [23] nous pouvons utiliser un ensemble de relations de base de voisinage induites par les permutations de base possibles.

Swap $\mathcal{N}_S$:
$(s, s') \in \mathcal{N}_S$ ssi $s = (\pi_1, \ldots, \pi_i, \pi_{i+1}, \ldots, \pi_n)$ et $s' = (\pi_1, \ldots, \pi_{i+1}, \pi_i, \ldots, \pi_n)$ pour un i.

Échange $\mathcal{N}_E$:
$(s, s') \in \mathcal{N}_E$ ssi $s = (\pi_1, \ldots, \pi_{i-1}, \pi_i, \pi_{i+1}, \ldots, \pi_{j-1}, \pi_j, \pi_{j+1}, \ldots, \pi_n)$ et $s' = (\pi_1, \ldots, \pi_{i-1}, \pi_j, \pi_{i+1}, \ldots, \pi_{j-1}, \pi_i, \pi_{j+1}, \ldots, \pi_n)$ pour un i et un j.

Insertion $\mathcal{N}_I$:
$(s, s') \in \mathcal{N}_I$ ssi $s = (\pi_1, \ldots, \pi_{i-1}, \pi_i, \pi_{i+1}, \ldots, \pi_{j-1}, \pi_j, \pi_{j+1}, \ldots, \pi_n)$ et $s' = (\pi_1, \ldots, \pi_{i-1}, \pi_{i+1}, \ldots, \pi_{j-1}, \pi_i, \pi_j, \pi_{j+1}, \ldots, \pi_n)$ pour un i et un j.

Échange d'arêtes $\mathcal{N}_{EE}$:
$(s, s') \in \mathcal{N}_{EE}$ ssi $s = (\pi_1, \ldots, \pi_{i-1}, \pi_i, \pi_{i+1}, \ldots, \pi_{j-1}, \pi_j, \pi_{j+1}, \ldots, \pi_n)$ et $s' = (\pi_1, \ldots, \pi_i, \pi_j, \pi_{j-1}, \ldots, \pi_{i+1}, \pi_{j+1}, \ldots, \pi_n)$ pour $i + 1 < j$.

$\mathcal{N}^+$ est la clôture transitive de $\mathcal{N}$.

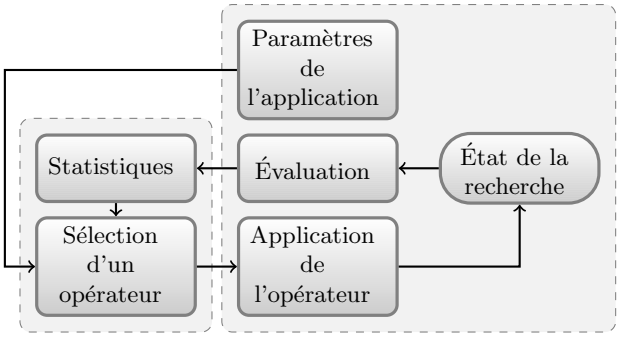


FIGURE 1 – Schéma général

Il apparait clairement que le voisinage construit par $\mathcal{N}_S$ peut également être construit par $\mathcal{N}_E$ et $\mathcal{N}_I$. Des relations d'ordres peuvent donc être définies pour classifier les voisinages afin de mettre l'emphase sur les relations entre les distances qu'ils induisent (voir [23] pour d'avantage de détails).

Nous pouvons maintenant proposer plusieurs fonctions classiques de sélections afin de construire des opérateurs.

Aléatoire : σ_R tel que $\sigma(s, \mathcal{N})$ est n'importe quel s' choisi aléatoirement tel que $(s, s') \in \mathcal{N}$.

Meilleure amélioration : $\sigma_B I$ tel que $\sigma(s, \mathcal{N})$ est un élément minimal s' selon l'ordre $<$, tel que $(s, s') \in \mathcal{N}$.

Meilleure amélioration k : $\sigma_B Ik$ tel que $\sigma(s, \mathcal{N})$ est un élément uniformément sélectionné $s' \in K$, K étant l'ensemble des k-meilleurs éléments selon l'ordre $<$, tel que $(s, s') \in \mathcal{N}$.

Amélioration : σ_I tel que $\sigma(s, \mathcal{N})$ est n'importe quel élément s' tel que $(s, s') \in \mathcal{N}$ et $s' < s$.

Tournois k : $\sigma_T k$ tel que $\sigma(s, \mathcal{N})$ est un élément s' tel que K est un sous-ensemble de k éléments en relation avec s dans $\mathcal{N}$ et s' est le meilleur de ces k éléments.

3 Contrôle d'opérateurs pour la recherche locale

L'objectif de notre méthode est de sélectionner, à partir d'un ensemble donné d'opérateurs, un opérateur approprié à appliquer à chaque itération (Fig. 1).Ceci requiert l'évaluation de l'efficacité des opérateurs basée sur leur comportement antérieur et la sélection d'un opérateur capable de faire progresser le processus de recherche, à des fins soit d'intensification soit de diversification.

Notre objectif est d'avoir une approche aussi générique que possible pour résoudre les problèmes de permutation. À cette fin, notre méthode de résolution se décompose en quatre modules distinct.

- Définition du Problème (DP)
- Gestionnaire de Chemin (GC)
- Gestionnaire d'Opérateurs (GO)
- Conteneur d'Opérateurs (CO)

Du point de vue de l'utilisateur, résoudre un nouveau problème de permutation nécessite seulement de définir une procédure de lecture les données de l'instance et la spécification de la fonction objectif et des contraintes. L'utilisateur peut, s'il le souhaite, ajouter de nouveaux opérateurs au Conteneur d'Opérateurs. Il peut également définir une méthode d'initialisation de la solution initiale s'il ne veut pas utiliser une solution générée aléatoirement.

Le programme est écrit en COMET [26]. Nous estimons que ce langage de programmation destiné à la recherche locale basée sur les contraintes nous offre des perspectives intéressantes puisqu'il facilite la programmation d'algorithmes de recherche locale et qu'il fournit des mécanismes simples de manipulation de voisinages et de sélecteurs. Notre programme se base sur ces fonctionnalités intuitives et peut être considéré comme une extension de COMET. Nous sommes convaincus que la généricité et la simplicité (modulo le minimum de connaissances nécessaire afin de définir le problème) alliées à la facilité intrinsèque d'utilisation de COMET est un pas positif vers la démocratisation de l'utilisation de logiciels d'optimisation.

DP (problème de permutation)
Ajouter les opérateurs au CO
Initialiser le GC
Initialiser le GO
$s \leftarrow$ solution initiale
$s^* \leftarrow s$
répéter
  $op \leftarrow$ sélectionner un opérateur
  $s \leftarrow op(s)$
  Mise à jour du GC avec s
  Mise à jour du GO avec les mesures de s
  si *s est meilleur que s^** **alors** $s^* \leftarrow s$
jusqu'à *condition de fin*
retourner s^*

Algorithme 1 : Description algorithmique globale

Notre approche se veut aussi générique que possible mais fait face à certaines limites. En pratique (pour des opérateurs plus complexes que de simples échanges), l'utilisateur doit donner la fonction permettant de calculer le delta de l'évaluation d'une solution candidate puisque qu'elle dépend de la fonction objectif.

3.1 Métriques

Comme mentionné dans l'introduction, un aspect important du contrôle est d'estimer l'équilibre diversification/intensification en utilisant des métriques qui

peuvent évaluer l'efficacité d'un opérateur par rapport à la trajectoire de recherche afin de choisir le prochain opérateur. Nous proposons de traiter simultanément deux critères, la qualité et la diversité, afin de gérer cet équilibre comme un compromis.

3.1.1 Qualité

La qualité est mesurée directement grâce à la fonction objectif. Le changement relatif de qualité en utilisant un opérateur *op* sur une solution s est donné par

$$\Delta Q = \frac{eval(op(s)) - eval(s)}{eval(s) + 1}$$

3.1.2 Distance

La diversité est un concept intuitif lorsque l'on considère des populations de solutions dans le contexte des algorithmes évolutionnaires. Elle l'est moins en recherche locale où une seule solution est produite à chaque itération. Cette notion a été explorée par exemple dans [24] et dans [14]. Nous pourrions considérer la diversité du chemin de la recherche (la séquence de solutions déjà trouvées) ou une fenêtre glissante de celui-ci. À la place nous choisissons d'essayer de mesurer la différence entre le chemin et la solution candidate actuelle $c = op(s)$. Nous proposons deux perspectives différentes : 1) le degré de dissimilitude entre le chemin et c au niveau des variables elles-même ; 2) l'éloignement de c par rapport au chemin en termes de nombres d'opérations les séparant.

La distance L_1 (de Manhattan) entre deux vecteurs p et q est définie par $d_1(p,q) = \sum_{i=1}^{n} |p_i - q_i|$. Nous utilisons une métrique simple mesurant la distance L_1 entre les représentations de la solution candidate et le barycentre du chemin. Les dimensions des points représentant les solutions sont les variables binaires, $x_{a,b}$, la valeur desquelles est 1 dans la solution candidate, où $x_{a,b} = 1$ implique que la variable a à pour valeur b pour les problèmes d'affectation ou que a est suivie de b pour les problèmes d'ordonnancement. Le barycentre correspond donc à la fréquence de $x_{a,b} = 1$ dans le chemin.

Plus formellement, soit $X = \{1, \ldots, n\}$ tel qu'il existe une bijection g de X dans l'ensemble des variables représentant les solutions où Y est le domaine de ces variables. Soit $f_s : X \to Y$ la fonction représentant les valeurs données aux variables de la solution s. Soit $P_{i,j}$ le chemin de l'itération i jusqu'à j, $i \leq j$. Alors

$$d_1^P(c, P_{i,j}) = \frac{1}{n} \times \sum_{k=1}^{n} \left(1 - \frac{occ(P_{i,j}, x_{k,f_c(k)})}{|P_{i,j}|}\right)$$

où $occ(P_{i,j}, x_{a,b})$ correspond au nombre de fois où $x_{a,b} = 1$ dans $P_{i,j}$.

Nous utilisons également la distance de base de voisinage présentée à la section 2, l'idée étant de comparer le chemin de recherche au chemin optimal qui aurait pu être obtenu en utilisant cette relation de voisinage de référence.

Nous définissons ainsi la distance

$$d_{\mathcal{N}}^P(p_k, P_{i,j}) = \sum_{l=i}^{j} \frac{|P_{l,k}|}{d_{\mathcal{N}}(p_l, p_k)} \quad , i \leq j \leq k$$

En utilisant un opérateur suffisamment simple, $d_{\mathcal{N}}^P$ peut être utilisée pour évaluer les caractéristiques exploratoires d'opérateurs plus complexes. À cette fin, nous suggérons l'utilisation de l'opérateur Échange, $\mathcal{N}_E$. Un algorithme permettant de calculer $d_{\mathcal{N}_E}$ est présenté dans [23].

3.2 Sélection d'opérateurs

Nous présentons maintenant le processus de sélection utilisé pour choisir l'opérateur de mouvement à chaque étape. Soient deux vecteurs u et v de même cardinalité p et considérant un problème de maximisation, u *domine* v si $u_k \geq v_k, \forall k \in \{1, \ldots, p\}$ avec au moins une inégalité stricte. Le terme Pareto-dominance est fréquemment employé pour décrire cette relation.

Considérons la population des vecteurs bidimensionnels représentant la performance de chaque opérateur. Dans cet article, la performance correspond à la moyenne ΔQ et d^P calculée pour chaque opérateur sur une fenêtre glissante, spécifique à chaque opérateur, de longueur m. La performance initiale de chaque opérateur est calculée en testant chacun d'eux une fois sur la solution initiale au début de la recherche. Si un opérateur n'a pas été utilisé au cours des m dernières opérations, la fenêtre glissante de cet opérateur ne sera pas vide : elle contiendra au moins un élément (et au plus m éléments) calculés avant ces m itérations.

L'opérateur à utiliser à chaque itération de l'algorithme est sélectionné selon une probabilité proportionnelle à son utilité [20]. Nous définissons l'utilité d'un opérateur comme le nombre d'opérateurs que celui-ci domine augmenté d'un ϵ afin de conserver une utilité non nulle.

4 Expérimentations

Nous testons notre méthode sur le *problème d'affectation quadratique* (QAP) et le *problème du voyageur de commerce asymétrique* (ATSP). Ces problèmes sont choisis car leurs solutions sont facilement représentables par un vecteur de variables. Le QAP modélise le problème qui revient à trouver le coût minimum

d'affectation d'unités de productions à des emplacements, les coûts correspondant à la somme des produits distance-flot [16]. Le ATSP demande de trouver le chemin hamiltonien de poids minimum dans un graphe orienté [7]. La solution initial pour le QAP est générée aléatoirement et l'heuristique de construction « voisin le plus proche » est utilisée pour le ATSP. Dans ces expériences nous utilisons dix opérateurs :

- O1 ($\sigma_I, \mathcal{N}_E$), première amélioration pour l'échange entre deux variables.
- O2 ($\sigma_B I, \mathcal{N}_E$), meilleure amélioration pour l'échange entre deux variables.
- O3 ($\sigma_B I5, \mathcal{N}_E$), choix aléatoire parmi les cinq meilleurs échanges entre deux variables.
- O4 $(\sigma_B I, \mathcal{N}_E)^2$, deux meilleurs échanges consécutifs. Les variables échangées à l'étape une sont interdites à l'étape deux.
- O5 $(\sigma_B I, \mathcal{N}_E)^3$, trois meilleurs échanges consécutifs. Les variables échangées aux étapes précédentes sont interdites à l'étape suivante.
- O6 ($\sigma_T 3!, \mathcal{N}_E^2$), meilleur échange entre 3 variables choisies aléatoirement.
- O7 ($\sigma_T 4!, \mathcal{N}_E^3$), meilleur échange entre 4 variables choisies aléatoirement.
- O8 ($\sigma_T 5!, \mathcal{N}_E^4$), meilleur échange entre 5 variables choisies aléatoirement.
- O9 ($\sigma_T 6!, \mathcal{N}_E^5$), meilleur échange entre 6 variables choisies aléatoirement.
- O10 ($\sigma_R, \mathcal{N}_E^3$), trois échanges aléatoires consécutifs entre deux variables.

Comme décrit plus loin, ces dix opérateurs ne donnent pas de bons résultats pour le ATSP. Nous ajoutons donc l'opérateur O11. Le mouvement 3-opt [7] ($\sigma_B I, \mathcal{N}_{EE}^2$) sélectionne le meilleur voisin obtenu en cassant 3 arêtes et en reconstruisant de nouvelles arêtes de sorte qu'aucun sous-chemin ne soit inversé.

Dans ces expériences, l'accent est mis sur des voisinages construits à partir de $\mathcal{N}_E$, qui semble être un bon voisinage intermédiaire. D'autres voisinages pourraient également être utilisés (d'ailleurs ils peuvent également être exprimés en termes de $\mathcal{N}_E$). Des travaux futurs pourraient étudier des ensembles plus importants de combinaisons comme cela a été fait dans le cadre des algorithmes évolutionnaires dans [19].

4.1 Protocole expérimental

Les instances de test proviennent de QAPLIB [3] pour le QAP et de TSPLIB [21] pour le ATSP. Chaque couple (algorithme, instance) est reproduit 30 fois. La taille de la fenêtre glissante est arbitrairement fixée à 100 et $\epsilon = 1$ pour le processus de sélection. Chaque exécution tourne au maximum pendant 40 000 itérations. Nous utilisons le test non paramétrique des rangs-signés de Wilcoxon [5, 25]. Soit deux algorithmes A et B, l'hypothèse nulle est la suivante : les médianes des distributions des solutions générées par A et B sont équivalentes. L'hypothèse est rejetée avec un seuil de confiance de 95%.

4.2 Résultats et discussion

Le tableau 1 présente les résultats pour le QAP et le tableau 2 ceux du ATSP. Le pourcentage moyen d'écart entre la meilleure valeur connue (BKV) et le choix proportionnel aux valeurs d'utilité suivantes sont donnés : distribution uniforme, qualité, nombre de solutions Pareto-dominées en utilisant la distance d_1^P (ParDom d_1^P) et la distance $d_{\mathcal{N}_E}^P$ (ParDom $d_{\mathcal{N}_E}^P$). Les résultats pour la recherche tabou robuste (RoTS) pour le QAP sont aussi donnés à titre de comparaison (bien entendu des travaux plus récents en recherche locale obtiennent de meilleurs résultats que RoTS, par exemple [12]). Notre objectif ici est simple de donner un marqueur de référence, réécrit en COMET, et de montrer que notre méthode, utilisant un ensemble non optimisé d'opérateurs est capable d'atteindre des résultats intéressants. Les meilleurs résultats pour chaque instance sont indiqués en gras (les résultats de RoTS ne sont pas pris en compte car ils sont meilleurs ou équivalents sur toutes les instances sauf deux). Dans le tableau 2, la colonne ParDom10 contient les résultats lorsque seuls les dix même opérateurs que pour le QAP sont utilisés avec la distance d_1^P.

Pour le QAP ParDom d_1^P et ParDom $d_{\mathcal{N}_E}^P$ partagent la majorité des résultats en gras. Cependant, selon le test de Wilcoxon, ParDom d_1^P semble être le meilleur algorithme lorsqu'il est comparé à la sélection uniforme et à celle basée sur la qualité. A contrario, les résultats de ParDom $d_{\mathcal{N}_E}^P$ ne sont pas statistiquement significatifs. Les résultats semblent donc montrer que ParDom d_1^P est meilleur que ParDom $d_{\mathcal{N}_E}^P$ (l'hypothèse nulle ne peut toutefois pas être rejetée si on les compare l'un à l'autre.

Pour le ATSP, l'utilisation des dix opérateurs utilisés pour le QAP n'est pas concluante. Cela s'explique aisément puisqu'aucun d'entre eux ne prend en compte la nature cyclique des solutions. L'ajout de l'opérateur 3-opt produit une amélioration notable. Avec un unique opérateur spécifique au ATSP, la population des opérateurs demeure très biaisée envers le ATSP. Ceci résulte en une amélioration bien plus visible que dans le cas du QAP lorsque l'on compare les trois méthodes de sélection non triviale avec la sélection uniforme. Ici, ParDom d_1^P et ParDom $d_{\mathcal{N}_E}^P$ partagent les meilleurs résultats avec toutefois la majorité pour ParDom $d_{\mathcal{N}_E}^P$. En termes statistiques, les deux méthodes de sélection par Pareto dominance sur onze opérateurs

Table 1 – Résultats expérimentaux pour le QAP.

Instance	BKV	Uniforme	Qualité	ParDom d_1^P	ParDom $d_{\mathcal{N}_E}^P$	RoTS
bur26a	5426670	0.000244	0.001629	**0.000000**	0.004015	0.000000
bur26c	5426795	0.000061	**0.000000**	0.000059	0.000002	0.000000
bur26f	3782044	**0.000000**	**0.000000**	**0.000000**	**0.000000**	0.000000
chr25a	3796	11.790306	10.353003	10.189673	**9.381454**	7.093783
els19	17212548	**0.000000**	**0.000000**	**0.000000**	**0.000000**	0.000000
kra30a	88900	**0.470416**	0.488939	0.499888	0.730034	0.067267
kra30b	91420	0.110698	0.124335	**0.063881**	0.098666	0.023408
nug20	2570	**0.000000**	**0.000000**	**0.000000**	**0.000000**	0.000000
nug30	6124	0.12279556	0.091444	0.057478	**0.050947**	0.014370
sko42	15812	0.163167	0.148832	**0.090817**	0.115608	0.029598
sko49	23386	0.266655	0.194703	**0.186265**	0.193962	0.125203
sko56	34458	0.212781	**0.196955**	0.229497	0.292762	0.118753
tai30a	1818146	1.131385	1.178607	0.794332	**0.633736**	0.512898
tai35a	2422002	1.538266	1.391353	0.943254	**0.745479**	0.762013
tai50a	4941410	1.847374	1.815764	1.377229	**1.363935**	1.391181
tai30b	637117113	0.150888	0.107800	**0.103892**	0.129518	0.026246
tai50b	458821517	**0.173836**	0.186702	0.269760	0.537427	0.150598
wil50	48816	0.076696	**0.074429**	0.079400	0.090216	0.053425

Table 2 – Résultats expérimentaux pour le ATSP.

Instances	BKV	ParDom10	Uniforme	Qualité	ParDom d_1^P	ParDom $d_{\mathcal{N}_E}^P$
br17	39	**0.000000**	**0.000000**	**0.000000**	**0.000000**	**0.000000**
p43	5620	0.202847	0.009490	0.002372	0.001779	**0.000593**
ry48p	14422	4.309620	0.661721	0.347155	0.204086	**0.168955**
ft53	6905	12.608255	1.108858	0.517982	0.186338	**0.172822**
ft70	38673	5.891276	0.749791	0.455787	0.080849	**0.048268**
ftv33	1286	7.550544	**0.000000**	**0.000000**	**0.000000**	**0.000000**
ftv35	1473	5.489930	0.495587	0.072415	0.067889	**0.031681**
ftv38	1530	6.141612	0.718954	0.429194	0.305011	**0.259259**
ftv44	1613	9.070056	0.725356	0.378177	**0.237652**	0.252118
ftv47	1776	11.006006	0.478604	0.191441	0.138889	**0.114489**
ftv55	1608	14.195688	0.972222	0.213516	0.136816	**0.093284**
ftv64	1839	17.130687	1.386623	0.781222	**0.554649**	0.580025
ftv70	1950	17.042735	1.540171	0.919658	**0.635897**	0.637607
ftv90	1579	25.429597	2.180705	1.253958	0.975301	**0.821195**
ftv100	1788	24.571216	2.839299	1.498881	1.168904	**1.047726**
ftv110	1958	31.089547	4.375213	2.667688	2.378277	**2.311883**
ftv120	2166	25.386273	3.464143	2.368421	2.136042	**2.132964**
ftv130	2307	22.831961	4.838896	2.781390	2.417281	**2.265569**
ftv140	2420	32.836088	4.720386	3.286501	3.004132	**2.965565**
ftv150	2611	32.370739	5.581514	3.993361	**3.150772**	3.292481
ftv160	2683	35.242887	5.998261	3.467511	3.473723	**3.334576**
ftv170	2755	33.393829	5.929825	3.680581	**3.097816**	3.553539
kro124p	36230	18.522587	2.327813	1.358451	1.150520	**1.055479**
rbg323	1326	7.986425	0.072901	0.012569	**0.000000**	**0.000000**
rbg358	1163	9.203210	0.005732	**0.000000**	**0.000000**	**0.000000**
rbg403	2465	1.150778	**0.000000**	**0.000000**	**0.000000**	**0.000000**
rbg443	2720	1.455882	**0.000000**	**0.000000**	**0.000000**	**0.000000**

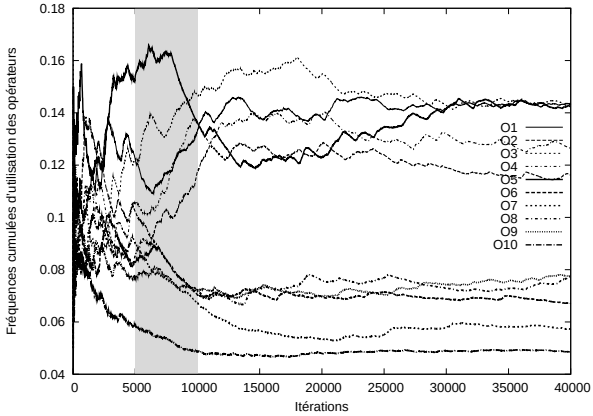
sont meilleures que la sélection uniforme ainsi que celle basée sur la qualité. Lorsque l'on compare ParDom d_1^P à ParDom $d_{\mathcal{N}_E}^P$, l'hypothèse nulle peut être rejetée. Cela démontre clairement que ParDom $d_{\mathcal{N}_E}^P$ est la mieux adaptée au ATSP.

La figure 2(a) représente la fréquences cumulées de l'application des opérateurs pour une exécution arbitraire du QAP et la figure 3 représente la même chose pour le ATSP. On peut observer que les opérateurs sont clairement séparés en deux groupes : un dont la fréquence est supérieure à la moyenne (0,1) et l'autre inférieure. Le détail nous révèle que le premier est le groupe qui améliore le plus la qualité alors que le second est celui qui perturbe le plus les solutions sans apporter de changement positif à la qualité.

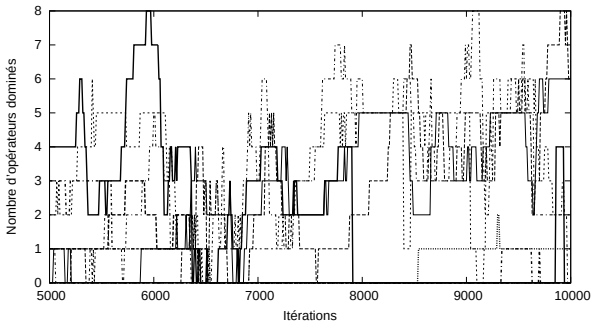
L'opérateur qui est sélectionné le plus souvent est celui qui démontre sa constance dans l'amélioration de la solution tout en modifiant un certain nombre de variables au cours de ses cent dernières applications. Comme on peut le voir à la figure 2(a), l'opérateur O5 se comporte très bien au début de la recherche. Sa performance baisse ensuite pour augmenter à nouveau jusqu'au niveau des autres « bons » opérateurs, la recherche stagnant à la fin. Ceci illustre le fait qu'un opérateur n'est pas toujours le meilleur tout au long de la recherche et qu'il est important pour le mécanisme de sélection d'être influencé par l'étape à laquelle se trouve la recherche. A contrario, dans la figure 3, l'opérateur O11 est constamment l'opérateur le plus souvent sélectionné. C'est un résultat auquel on pouvait s'attendre puisque c'est le seul opérateur spécifique au ATSP. Il se comporte donc forcément mieux que les autres opérateurs. Les opérateurs qui améliorent la qualité pour le QAP gardent toutefois un intérêt pour le ATSP.

Afin d'expliquer pourquoi les opérateurs améliorant la qualité sont sélectionnés le plus souvent, on peut remarquer que la modification d'une solution pour la rendre meilleure requiert également la modification de ses variables et donc de la distance depuis la dernière solution. Les opérateurs améliorant la qualité sont donc plus enclin à dominer les opérateurs dont la seule action est de causer des perturbations dans la solution.

La figure 2(b) montre le nombre d'opérateurs dominés par chaque opérateur sur une partie de la recherche de la figure 2(a) (région grisée). On peut observer qu'à certains moments de la recherche un opérateur domine presque tous les autres, ayant ainsi la probabilité la plus grande d'être sélectionné. Cette probabilité accrue se reflète dans les courbes de fréquences cumulées. On peut observer que pendant les quelques centaines d'itérations avant l'itération 7 000, aucun opérateur n'est considéré comme bien bien supérieur aux



(a) Fréquences cumulées de sélection.



(b) Nombre d'opérateurs dominés correspondant aux itérations grisées en (a).

FIGURE 2 – Une exécution de l'instance tai50a (QAP).

autres. Ceci transforme le processus de sélection en une simple sélection uniforme. Une sélection plus discriminante émerge ensuite, des opérateurs différents étant apparemment mieux adaptés à la suite de la recherche.

Une autre observation intéressante, plus particulièrement à la figure 3, est que l'augmentation significative de la fréquence de sélection d'un opérateur implique souvent l'inverse pour un autre opérateur.

La figure 4 est un instantané d'une partie de la recherche sur une instance de QAP. Elle représente la meilleure valeur obtenue, la valeur courante de la fonction objectif et la distance du chemin à la nouvelle solution. Elle met en valeur que le contrôle, gérant le compromis entre qualité et diversité, est capable d'échapper à des minima locaux et également d'atteindre de bonnes solutions. La corrélation entre notre mesure de distance et la qualité apparait aussi clairement.

Bien que cet article ne contienne pas de résultats détaillés, nous pouvons noter que si cinq clones d'un opérateur ne faisant rien sont rajoutés, l'écart de per-

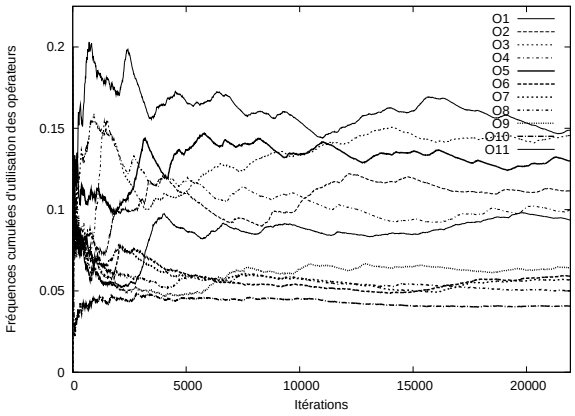


FIGURE 3 – Une exécution de l'instance ftv55 (ATSP).

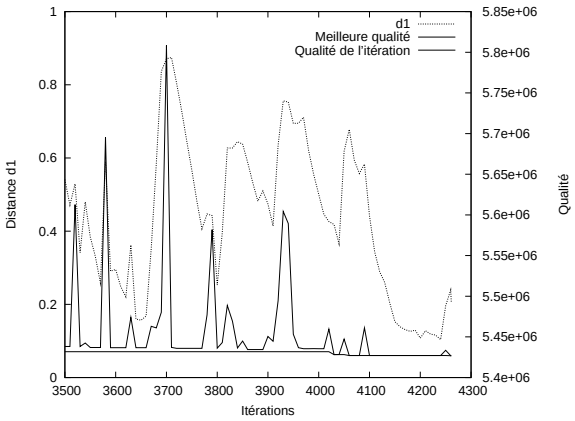


FIGURE 4 – Partie finale de la recherche pour une exécution de l'instance bur26a (QAP).

formance entre les méthodes proposées de sélection et la sélection uniforme s'agrandit. Si l'on utilise que le meilleur opérateur pour l'intensification et le meilleur opérateur pour la diversification avec ϵ égale à 0,1 (1 étant trop proche de la sélection uniforme), les résultats sont moins bons que pour dix opérateurs.

5 Conclusion

Dans cet article nous avons présenté une méthode générique de gestion d'opérateurs de recherche locale pour les problèmes de permutation ainsi que deux métriques pour mesurer la distance entre une solution et une partie du chemin de recherche. Les résultats des tests sur le QAP et le ATSP montrent que l'algorithme fonctionne mais qu'il peut être grandement amélioré.

Le processus de sélection favorise les opérateurs qui maintiennent un compromis coopératif entre l'intensification et la diversification, c.-à-d. les opérateurs au milieu du front Pareto. La prochaine étape de notre travail sera d'inclure un élément réactif au processus. Une autre question importante est la gestion des redémarrages (et pas seulement de petites perturbations), soit comme une redémarrage « externe » dépendant d'une certaine condition soit comme un restart « interne », c.-à-d. un opérateur en lui-même. Il va de soi que l'évaluation de la généricité de notre approche sur d'autres problèmes est requise, et plus particulièrement des problèmes avec des contraintes et des problèmes qui requièrent l'exploration de solutions non réalisables.

Remerciements. Ces travaux sont financés par Microsoft Research à travers le PhD Scholarship Programme.

Références

[1] Roberto Battiti, Mauro Brunato, and Franco Mascia. *Reactive Search and Intelligent Optimization.* Springer Publishing Company, Incorporated, 2008.

[2] Roberto Battiti and Giampietro Tecchiolli. The Reactive Tabu Search. *INFORMS JOURNAL ON COMPUTING*, 6(2) :126–140, January 1994.

[3] Rainer E. Burkard, Stefan E. Karisch, and Franz Rendl. QAPLIB – A Quadratic Assignment Problem Library. *Journal of Global Optimization*, 10(4) :391–403, June 1997.

[4] Edmund Burke, Graham Kendall, Jim Newall, Emma Hart, Peter Ross, and Sonia Schulenburg. Hyper-Heuristics : An Emerging Direction in Modern Search Technology. In Fred Glover and Gary Kochenberger, editors, *Handbook of Metaheuristics*, volume 57, pages 457–474. Springer New York, 2003.

[5] Marco Chiarandini, Luís Paquete, Mike Preuss, and Enda Ridge. Experiments on metaheuristics : Methodological overview and open issues. Technical Report DMF-2007-03-003, The Danish Mathematical Society, 2007.

[6] Luis DaCosta, Álvaro Fialho, Marc Schoenauer, and Michèle Sebag. Adaptive operator selection with dynamic multi-armed bandits. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 913–920, Atlanta, GA, USA, 2008. ACM.

[7] Gregory Gutin and Abraham P. Punnen. *The traveling salesman problem and its variations.* Springer, 2002.

[8] Y. Hamadi, E. Monfroy, and F. Saubion. *Hybrid Optimization : The Ten Years of CPAIOR*, chapter What Is Autonomous Search ? Springer, 2010.

[9] Pierre Hansen and Nenad Mladenović. Variable Neighborhood Search. In Fred Glover and Gary Kochenberger, editors, *Handbook of Metaheuristics*, volume 57, pages 145–184. Springer New York, 2003.

[10] Holger Hoos and Thomas Stützle. *Stochastic Local Search : Foundations & Applications*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[11] Bin Hu and Günther R. Raidl. Variable Neighborhood Descent with Self-Adaptive Neighborhood-Ordering. In *Proc. of the 7th EU Meeting on Adaptive, Self-Adaptive and Multilevel Metaheuristics*, 2006.

[12] Mohamed Saifullah Hussin and Thomas Stützle. Hierarchical Iterated Local Search for the Quadratic Assignment Problem. In *Hybrid Metaheuristics, 6th International Workshop, HM 2009,*, volume 5818 of *Lecture Notes in Computer Science*, pages 115–129. Springer, 2009.

[13] Lester Ingber. Adaptive simulated annealing (ASA) : Lessons learned. *Control and Cybernetics*, 25, 1996. Control and Cybernetics 25 (1996) 33-54.

[14] Alexandre Linhares. The structure of local search diversity. In *Math'04 : Proceedings of the 5th WSEAS International Conference on Applied Mathematics*, pages 1–5, Stevens Point, Wisconsin, USA, 2004. World Scientific and Engineering Academy and Society (WSEAS).

[15] Alexandre Linhares and Horacio Hideki Yanasse. Search intensity versus search diversity : a false trade off ? *Applied Intelligence*, 32(3) :279–291, 2010.

[16] Eliane Maria Loiola, Nair Maria Maia de Abreu, Paulo Oswaldo Boaventura-Netto, Peter Hahn, and Tania Querido. A survey for the quadratic assignment problem. *European Journal of Operational Research*, 176(2) :657–690, January 2007.

[17] Helena Lourenço, Olivier Martin, and Thomas Stützle. Iterated Local Search. In Fred Glover and Gary Kochenberger, editors, *Handbook of Metaheuristics*, volume 57, pages 320–353. Springer New York, 2003.

[18] J. Maturana and F. Saubion. A compass to guide genetic algorithms. In G. Rudolph et al., editor, *Parallel Problem Solving from Nature - PPSN X, 10th International Conference Dortmund, Germany, September 13-17, 2008, Proceedings*, volume 5199 of *LNCS*, pages 256–265. Springer, 2008.

[19] Jorge Maturana, Frédéric Lardeux, and Frédéric Saubion. Autonomous operator management for evolutionary algorithms. *Journal of Heuristics*, 2010.

[20] Alexander Nareyek. Choosing search heuristics by non-stationary reinforcement learning. In *Metaheuristics : computer decision-making*, pages 523–544. Kluwer Academic Publishers, 2004.

[21] Gerhard Reinelt. TSPLIB – A Traveling Salesman Problem Library. *INFORMS JOURNAL ON COMPUTING*, 3(4) :376–384, January 1991.

[22] Julien Robet, Frédéric Lardeux, and Frédéric Saubion. Autonomous Control Approach for Local Search. In *Engineering Stochastic Local Search Algorithms. Designing, Implementing and Analyzing Effective Heuristics*, pages 130–134. Springer-Verlag, Berlin, Heidelberg, 2009.

[23] Tommaso Schiavinotto and Thomas Stützle. A review of metrics on permutations for search landscape analysis. *Comput. Oper. Res.*, 34(10) :3143–3153, 2007.

[24] Alain Sidaner, Olivier Bailleux, and Jean-Jacques Chabrier. Measuring the Spatial Dispersion of Evolutionary Search Processes : Application to Walksat. In *Artificial Evolution, 5th International Conference, Evolution Artificielle, EA 2001*, volume 2310 of *LNCS*, pages 77–90. Springer, 2001.

[25] Peter Sprent. *Applied Nonparametric Statistical Methods*. Chapman & Hall, London, 1989.

[26] Pascal Van Hentenryck and Laurent Michel. *Constraint-Based Local Search*. The MIT Press, 2005.

[27] Nadarajen Veerapen and Frédéric Saubion. Pareto Autonomous Local Search. In Xin Yao and Carlos A. Coelho Coelho, editors, *Proc. Fifth International Conference on Learning and Intelligent Optimization (LION5), Rome, Jan 17-21, 2011*, Springer, Lecture Notes on Computer Science, to appear.