

Agents évolutionnaires pour l'optimisation multiobjectif

Evolutionary agents for multiobjective optimization

Alain Berro¹

Olivier Heguy²

¹Université Toulouse I Sciences Sociales (UT1)

²Institut de Recherche en Informatique de Toulouse (IRIT)

1 Place Anatole France
Manufacture des tabacs
31042 Toulouse cedex, France

berro@univ-tlse1.fr

heguy@irit.fr

Résumé

Dans cet article, nous présentons une méthode multiagent associée à une méthode de recherche locale inspirée des stratégies d'évolution pour résoudre des problèmes d'optimisation multiobjectifs. En comparaison avec les méthodes basées sur des algorithmes génétiques, cette méthode utilise peu de paramètres de réglage. De plus un décideur peut facilement comprendre l'influence de ces paramètres sur le résultat.

La conception de cette méthode nous a amené à représenter la frontière de Pareto par un ensemble de zones Pareto-optimales et non de points. Cette représentation apporte une information supplémentaire permettant éventuellement de choisir entre deux solutions non dominées.

Mots-clef

Optimisation multiobjectif, algorithme évolutionnaire

Abstract

In this article, we present an agent-based method associated with a local research inspired by strategies of evolution to solve multiobjective problems. In comparison with GA-based methods this method uses few parameters. Moreover a decision maker can easily understand the influence of these parameters on the result. The conception of this method led us to represent the Pareto optimal set with zones and not with points. This representation gives additional information which allows to choose between two non-dominated solutions

Keywords

Multiobjective optimization, evolutionary algorithm

1 Introduction

L'idée d'utiliser la dominance au sens de Pareto a été proposée par Goldberg [6] pour résoudre les problèmes proposés par Schaffer [14]. Il suggère d'utiliser le concept d'optimalité de Pareto pour respecter l'intégralité de chaque critère car il refuse d'agréger à priori les valeurs

de différents critères. L'utilisation d'une sélection basée sur la notion de dominance de Pareto va faire converger la population vers un ensemble de solutions efficaces. Ce concept ne permet pas de choisir une alternative plutôt qu'une autre mais il apporte une aide précieuse au décideur.

1.1 Les méthodes non élitistes

Depuis 1993, un grand nombre de méthodes multiobjectifs utilisent la dominance de Pareto pour rechercher les solutions d'un problème.

Fonseca et Fleming ont proposé la méthode MOGA (Multiple Objective Genetic Algorithm) dans laquelle chaque individu de la population est rangé en fonction du nombre d'individus qui le dominent [5]. Ensuite, ils utilisent une fonction de notation permettant de prendre en compte le rang de l'individu et le nombre d'individus ayant le même rang.

Dans NSGA (Non Dominated Sorting Genetic Algorithm), la méthode proposée par Srivinas et Deb [15], le calcul de la fitness s'effectue en séparant la population en plusieurs groupes en fonction du degré de domination au sens de Pareto de chaque individu. L'algorithme se déroule ensuite comme un algorithme génétique classique. Srivinas et Deb utilisent une sélection basée sur le reste stochastique. Mais leur méthode peut être utilisée avec d'autres heuristiques de sélection (tournoi, roulette pipée, etc.)

NPGA (Niche Pareto Genetic Algorithm), la méthode proposée par Horn et Nafpliotis utilise un tournoi basé sur la notion de dominance de Pareto [8]. Elle compare deux individus pris au hasard avec une sous-population de petite taille également choisie au hasard. Si un seul de ces deux individus domine la sous-population il est alors positionné dans la population suivante. Dans les autres cas une fonction de sharing est appliquée pour sélectionner l'individu.

Les approches que nous venons de voir sont dites non élitistes car :

- elles ne conservent pas les individus Pareto-optimaux trouvés au cours du temps,
- elles maintiennent difficilement la diversité sur la frontière de Pareto,
- la convergence des solutions vers la frontière de Pareto est lente.

Pour résoudre les difficultés ci-dessus, de nouvelles techniques ont été développées :

- introduction d'une population externe ou archive permettant de stocker les individus Pareto-optimaux,
- utilisation de techniques de niching, clustering et grid-based pour répartir efficacement les solutions sur la frontière de Pareto,
- préférence pour les solutions non dominées.

1.2 Les méthodes élitistes

En 1998, Zitzler et Thiele proposent la méthode SPEA (Strength Pareto Evolutionary Algorithm) [16]. Cette méthode élitiste utilise une archive pour stocker les individus Pareto-optimaux. Une technique de clustering permet de réduire l'ensemble de Pareto sans supprimer ses caractéristiques, et une technique de niching permet de préserver la diversité de la population. L'avantage de cette méthode est qu'elle n'exige pas de réglage de paramètres de sharing.

En 1999, Knowles et Corne propose la méthode PAES (Pareto Archived Evolution Strategy). Cette méthode a été développée initialement comme une méthode de recherche locale dans un problème de routage d'information off-line. Les premiers travaux de Knowles et Corne ont montré que cette méthode simple objectif fournissait des résultats supérieurs aux méthodes de recherche basées sur une population [9]. Par conséquent, les auteurs ont adapté cette méthode aux problèmes multiobjectifs. Les particularités de cette méthode sont les suivantes :

- elle n'est pas basée sur une population,
- elle n'utilise qu'un seul individu à la fois pour la recherche des solutions,
- elle utilise une population annexe de taille déterminée permettant de stocker les solutions temporairement Pareto-optimales,
- l'algorithme utilisé est simple et inspiré des stratégies d'évolutions [13],
- elle utilise une technique de crowding basée sur un découpage en hypercubes de l'espace des objectifs.

La méthode PESA (Pareto Envelope based Selection Algorithm) a été également proposée par Knowles et Corne [10]. Alors que PAES est basée sur une stratégie d'évolution, PESA est une méthode basée sur les algorithmes génétiques. Elle reprend approximativement le principe de crowding développé dans PAES. La différence essentielle est que la sélection est basée sur un paramètre appelé `squeeze_factor` qui représente la

mesure d'encombrement d'une zone de l'espace des objectifs. Cela permet une bonne répartition des individus mais augmente la dépendance de l'efficacité de la méthode par rapport au facteur de discrétisation de l'espace en hypercubes.

Dans cette deuxième version de NSGA [4], Kalyanmoy Deb tente de résoudre toutes les critiques relevées sur NSGA I : complexité, non élitisme et utilisation du sharing. Pour diminuer la complexité de calcul de NSGA, Deb propose une modification de la procédure de tri de la population en plusieurs frontières. L'autre critique sur NSGA est l'utilisation du sharing, méthode qui exige le réglage d'un ou plusieurs paramètre(s) et qui est également grosse consommatrice de calculs. Dans NSGA II, Deb remplace la fonction de sharing par une fonction de crowding en attribuant deux caractéristiques à chaque individu : la rang de non domination et la distance de crowding. Pour répondre à la critique de non élitisme, Deb utilise une sélection par tournoi, modifie la procédure de passage entre deux générations et définit une notion de préférence entre deux solutions en fonction des deux caractéristiques des individus.

PESA II est une nouvelle technique de sélection basée sur l'utilisation d'hypercubes dans l'espace des objectifs [3]. Au lieu d'effectuer une sélection en fonction de la fitness des individus comme dans PESA, cette méthode effectue une sélection par rapport aux hypercubes occupés par au moins un individu. Après avoir sélectionné l'hypercube, on choisit aléatoirement l'individu dans l'hypercube. Cette méthode se montre plus efficace pour répartir les solutions sur la frontière de Pareto. Cela est dû à sa capacité de choisir avec une plus grande probabilité que le tournoi classique, des individus situés dans des zones désertiques.

En 2001, Coello trouve que les recherches actuelles n'accordent pas assez d'importance à l'efficacité des méthodes d'optimisation multiobjectifs. Il propose une méthode basée sur une population avec un nombre très faible d'individus [2]. Cette technique est basée sur les résultats théoriques obtenus par Goldberg [7], selon lesquels une taille de population très faible suffit à obtenir une convergence indépendamment de la longueur du chromosome. Le mécanisme suggéré par Goldberg est d'utiliser une population très petite et d'appliquer les opérateurs génétiques jusqu'à obtention d'une convergence (identifiée par une perte de diversité génétique). Ensuite on crée une nouvelle population en copiant les meilleurs individus de la population précédente, le reste de la population étant généré aléatoirement. Coello applique ce mécanisme aux problèmes d'optimisation multiobjectifs en utilisant un algorithme génétique avec une petite taille de population associée à une archive et une heuristique de distribution géographique (technique de crowding proposée par Knowles et Corne pour la méthode PAES).

Memetic¹-PAES, méthode proposée par Knowles et Corne, est une extension de la recherche locale (1+1)-PAES combinée avec l'utilisation d'une population [11]. Périodiquement, celle-ci utilise le croisement pour combiner les optima locaux trouvés par la procédure (1+1)-PAES. La mise à jour de l'archive est un peu plus compliquée que pour PAES. Bien que l'algorithme soit plus complexe que les précédents, cette nouvelle méthode de Knowles et Corne apporte des perspectives intéressantes. L'utilisation combinée d'une recherche locale et d'une population permet d'associer la capacité exploratoire locale à une capacité d'exploration globale par deux mécanismes indépendants.

1.3 Discussion

La mise en œuvre de la plupart des méthodes d'optimisation multiobjectifs présentées ci-dessus exige en priorité de maîtriser le fonctionnement d'un algorithme génétique et de comprendre l'interaction de ses différents paramètres pour répondre aux questions suivantes : Quel génotype ? Quelle notation ? Quelle fonction de sélection ? Quelle heuristique de partage ? Quel critère d'arrêt ?

De plus, l'usage d'une population externe pour maintenir les meilleurs individus exige la détermination de stratégies de mise à jour et de réutilisation basées dans la plupart des cas sur une heuristique de remplacement géographique.

La définition d'un critère d'arrêt est une difficulté supplémentaire. Le critère le plus usité pour stopper la recherche d'un algorithme génétique est la perte de diversité génétique. Or celui-ci n'a plus de sens dans des méthodes qui maintiennent la diversité génétique de la population pour permettre une recherche plus efficace. Dans la méthode Memetic-PAES, les auteurs définissent un critère d'arrêt basé sur le nombre d'échecs consécutifs de la recherche locale.

La relation entre la valeur d'un paramètre et son action sur la résolution du problème est donc très difficile à contrôler sans une parfaite connaissance de la méthode employée. De plus, beaucoup de ces paramètres ont peu de sens pour un décideur.

Dans la méthode que nous avons réalisée, nous nous sommes attachés à réduire ces difficultés tout en conservant les qualités de convergence et de maintien de la diversité du processus d'optimisation. Nous avons développé une méthode ayant un paramétrage simple et compréhensible pour un non initié.

Dans toutes les méthodes présentées ci-dessus la frontière de Pareto est décrite par un ensemble de points non dominés. Mais elles n'apportent aucune information supplémentaire qui permettrait au décideur de choisir une

solution par rapport à une autre. Par exemple, dans la figure ci-dessous, supposons que le triangle représente la zone Pareto-optimale théorique (non connu à priori du décideur). Un décideur ne pourra pas choisir entre les points A et B car ils sont tous les deux non dominés. Or nous pouvons constater sur le schéma que le point B est préférable au point A car ce dernier est très proche de la limite de la zone optimale. En choisissant le point B, le décideur aura une marge de sécurité² plus importante que pour le point A.

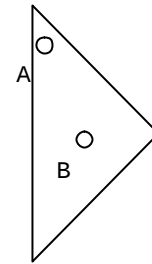


Fig. 1. Zone Pareto-optimale et deux points non dominés.

Notre méthode présentée ci-dessous permet de donner une évaluation de la plus petite distance entre un point Pareto-optimal et la limite de la zone Pareto-optimale sans que cette zone soit connue a priori. Cette caractéristique vient du fait que notre processus d'optimisation ne recherche pas des points Pareto-optimaux mais des zones Pareto-optimales.

2 Description de la méthode

Nous présentons dans cette partie la méthode d'optimisation multiobjectif que nous avons développée. Initialement cette méthode a été réalisée pour rechercher les optima d'une fonction dans un environnement dynamique [1] et possède les caractéristiques suivantes :

- une auto-adaptation du nombre d'agents nécessaires à la recherche,
- deux paramètres de réglage sont utilisés,
- une évaluation de la marge d'erreur sur la position de l'optimum.

Nous avons donc adapté cette méthode pour des problèmes d'optimisation multiobjectifs en utilisant la dominance au sens de Pareto pour effectuer la comparaison entre les différentes solutions et pour diriger les agents dans l'espace de recherche.

Notre méthode multiagent utilise une recherche locale inspirée des stratégies d'évolution. Le but d'un agent est de trouver une zone Pareto-optimale puis de l'agrandir.

Les apports originaux de cette méthode sont l'introduction de la notion de zone de dominance d'un agent et

¹ Le terme "memetic algorithms" a été utilisé pour la première fois par Pablo Moscato [15]. Les "memetic algorithms" combinent une heuristique de recherche locale avec des opérateurs de croisement sur une population.

² Les résultats obtenus par calcul ne sont pas forcément appliquer tel quel. Par exemple ils peuvent être arrondis ou légèrement modifiés pour correspondre à une contrainte réelle non modélisée dans la méthode.

l'utilisation d'un facteur de précision comme critère d'arrêt de l'algorithme. Lors de la conception de cette méthode, nous nous sommes également attachés à réduire le nombre de paramètres de réglage et à rendre ces paramètres intelligibles pour le décideur.

2.1 Le processus d'évolution

Le processus d'évolution est basé sur la création aléatoire d'agents logiciels indépendants qui vont ensuite coloniser une zone Pareto-optimale. Chaque agent possède une stratégie de recherche locale qui lui permet de trouver une zone Pareto-optimale. Le système est composé de deux parties : le système de contrôle et les agents.

2.1.1 Le système de contrôle

Le système de contrôle a deux rôles différents :

- il crée périodiquement des agents qui sont positionnés aléatoirement dans l'espace de recherche,
- il élimine les agents inefficaces³.

La génération aléatoire d'agents apporte à la méthode une bonne capacité d'exploration de l'espace de recherche car chaque point de l'espace a la même probabilité d'être atteint.

L'élimination des agents inefficaces permet de réduire le nombre d'agents présents dans l'espace de recherche. Cette suppression permet d'accélérer le processus de recherche sans diminuer sa capacité de prospection.

Le système de contrôle ne possède pas de critère d'arrêt, il effectue perpétuellement le cycle d'exécution suivant :

- a. Création aléatoire d'un ou plusieurs agent(s).
- b. Evolution de chaque agent.
- c. Elimination des agents inefficaces.
- d. Retour en a.

2.1.2 Les agents

Chaque agent du système a les propriétés suivantes :

- Il a une position $P(x_1, \dots, x_n)$ dans l'espace de recherche.
- Il développe autour de lui une zone de recherche de taille variable dans laquelle il effectue sa recherche de la zone Pareto-optimale.
- Après avoir trouver une zone Pareto-optimale, il tente de l'agrandir au maximum.
- Il est doué d'autonomie, c'est-à-dire qu'il n'est pas dirigé par l'utilisateur ou par le système de contrôle. Il possède une fonction d'évolution qui lui permet de modifier sa position dans l'espace de recherche.
- Il ne possède qu'une représentation partielle de son environnement. Il ne connaît pas les fonctions objectifs qu'il va explorer mais il connaît les domaines de valeurs des variables constituant l'espace de recherche.
- Dans certaines conditions l'agent génère un autre agent qui va visiter une zone voisine.
- Il ne communique pas avec les autres agents mais il perçoit leur présence.

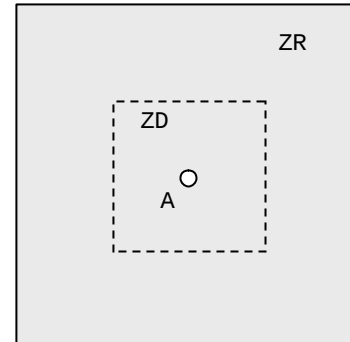


Fig. 2. Le cercle blanc représente la position de l'agent A. La zone de dominance ZD d'un agent est une région autour de sa position dans laquelle toutes les solutions sont non dominées. La zone de recherche ZR d'un agent, zone grise, est une région de taille variable dans laquelle l'agent recherche des solutions non dominées. Nous appellerons ZE la zone $ZR \setminus ZD$.

2.1.3 La fonction d'évolution d'un agent

Au cours de sa vie, un agent a deux états possibles. Il peut être valide ou non valide. Un agent est valide s'il a développé autour de lui une zone de dominance, $ZD \neq \emptyset$. Pour modifier sa position dans l'espace de recherche l'agent va analyser la dominance des points de son voisinage, c'est-à-dire à l'intérieur de sa zone de recherche, en effectuant la procédure ci-dessous :

- a. Découpage de la zone de recherche en une grille aléatoire de taille dépendante du paramètre force.
- b. On calcule pour chaque point i de la grille le nombre de points j qui le dominent. On obtient ainsi une matrice de dominance.
- c. Si l'agent A est non valide

Alors

Il recherche une zone autour de lui dont les points sont non dominés.

Si cette zone existe

Alors

L'agent peut développer sa zone de dominance ZD.

Sinon

L'agent calcule les sommes des lignes et des colonnes de la matrice de dominance, puis il choisit les plus petites valeurs trouvées pour déterminer les coordonnées du point sur lequel il va se déplacer. Il agrandit également sa zone de recherche ZR.

Sinon

Il effectue la somme s_i des valeurs des points de la matrice de dominance situés dans sa zone de dominance ZD, et la somme s_e des valeurs des points situés dans ZE.

³ Un agent inefficace est un agent situé dans la zone de dominance d'un autre agent.

Si $s_i = 0$
 Alors
 Il tente d'agrandir ZD en effectuant un éventuel déplacement.
 Si $s_e \neq 0$
 Alors
 Il crée un autre agent dans ZE, cet agent va aller explorer le voisinage.
 Sinon
 Il recalcule sa zone de dominance ZD et sa zone de recherche ZR.

2.2 Le rôle des paramètres

Ce système est très simple à paramétrer. Après avoir défini la ou les fonction(s) à optimiser ainsi que les dimensions de l'espace de recherche, l'utilisateur doit spécifier deux paramètres qui vont influencer la rapidité de recherche et la précision de calcul des solutions optimales.

- **force** : détermine le nombre de points que l'agent va tester à chaque étape de son évolution. Plus force sera élevé plus le calcul sera lent mais la recherche de l'agent sera plus efficace.
- **epsilon** : joue le rôle d'un critère d'arrêt. Lorsque l'agent trouve une zone de dominance ZD, il tente à chaque cycle de temps d'agrandir cette zone au maximum. S'il n'arrive pas à agrandir cette zone d'une valeur epsilon pendant un nombre force de cycles de temps, il stoppe sa recherche.

2.3 Les caractéristiques de la méthode

2.3.1 Comparaison entre les solutions non dominées

Cette méthode permet de comparer les solutions Pareto-optimales en fonction de la grandeur de la zone de dominance autour de l'agent.

Lorsqu'un agent a développé une zone de dominance alors sa position $P(x_1, \dots, x_n)$ dans l'espace de recherche est une solution non dominée. La taille de la zone de dominance est représentée par la moitié de la largeur d'un côté. En comparant les tailles de deux zones, on peut alors savoir quelle solution est la plus éloignée du bord de la zone Pareto-optimale. En choisissant cette solution, le décideur s'assurera une plus grande marge de sécurité pour appliquer cette valeur calculée au problème réel.

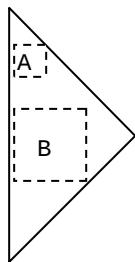


Fig. 3. Le triangle représente la zone Pareto-optimale théorique. Les zones pointillées représentent les zones de dominance des agents A et B. On constate que la zone de B étant plus grande de celle de A alors la solution représentée par l'agent B permet de prendre une décision avec une plus grande marge de sécurité car cette solution est éloignée de la limite de la zone Pareto-optimale.

2.3.2 Une technique de répartition géographique endogène aux agents.

Pour inciter les agents valides à se disperser dans l'espace et pour éviter que leurs zones de dominance se recouvrent, chaque agent A ajoute la valeur 1 à tout point de son voisinage inclus dans la zone de dominance d'un autre agent B si la taille de la zone de dominance de B est supérieure à celle de A. Ainsi, l'agent A détecte qu'une certaine partie de sa zone de dominance est déjà sous influence d'un autre agent donc A devra réduire sa zone de dominance. Ce mécanisme de répulsion permet de disperser les agents valides sur l'ensemble de la zone Pareto-optimale. Ainsi nous n'avons pas besoin d'utiliser une heuristique de répartition (sharing).

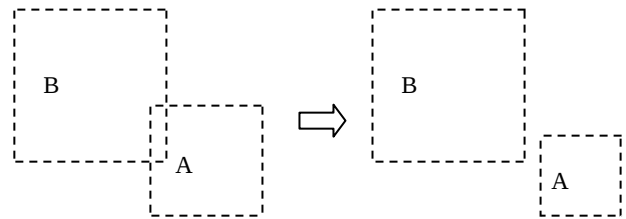


Fig. 4. Dans cet exemple, l'agent A détecte que sa zone de dominance recouvre partiellement la zone de dominance de B. A réduit sa zone pour éviter cette redondance.

De plus la représentation des solutions sous forme d'une zone évite d'avoir à utiliser une technique de clustering pour réduire les zones encombrées par un nombre important de solutions non dominées.

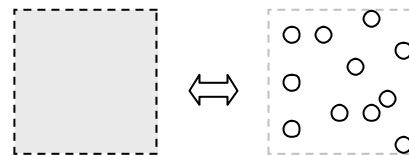


Fig. 5. Une zone de dominance correspond à un cluster.

2.3.3 Capacité d'exploration et d'exploitation

Lorsqu'un agent a trouvé une zone Pareto-optimale, il acquiert la capacité de création. Les nouveaux agents vont ainsi explorer le voisinage à la recherche d'autres zones. La répétition de ce phénomène va permettre par diffusion de visiter tout l'espace de recherche.

Pour accélérer l'exploration de l'espace de recherche, le système crée périodiquement des agents. Ainsi tous les points de l'espace ont la même probabilité d'être atteint.

Un autre avantage de cette création aléatoire est de permettre une recherche dans des espaces non connexes.

2.3.4 Auto régulation du système

Le système de contrôle élimine les agents qui pénètrent dans la zone de dominance d'un autre agent. Ainsi le système régule tout seul le nombre d'agents dans l'espace de recherche. Plus le nombre d'agents valides augmente, plus le total des zones sous influences augmente donc l'espace de recherche à explorer diminue ainsi que le nombre total d'agents. Cela permet d'augmenter la vitesse d'exploration de l'espace. Par contre, l'utilisateur d'un algorithme génétique doit fixer un nombre d'individus dans la population. Cela implique une certaine connaissance du problème.

3. Les résultats

Ce paragraphe présente les résultats obtenus sous les problèmes de Viennet suivants :

Viennet 1.

$\min (f_1(x, y), f_2(x, y), f_3(x, y))$
 avec $-4 \leq x, y \leq 4$
 $f_1(x, y) = x^2 + (y-1)^2$
 $f_2(x, y) = x^2 + (y+1)^2$
 $f_3(x, y) = (x-1)^2 + y^2 + 2$

Viennet 2

$\min (f_1(x, y), f_2(x, y), f_3(x, y))$
 avec $-3 \leq x, y \leq 3$
 $f_1(x, y) = (x-2)^2/2 + (y+1)^2/13 + 3$
 $f_2(x, y) = (x+y-3)^2/36 + (-x+y+2)^2/8 - 17$
 $f_3(x, y) = (x+2y-1)^2/175 + (2y-x)^2/17 - 13$

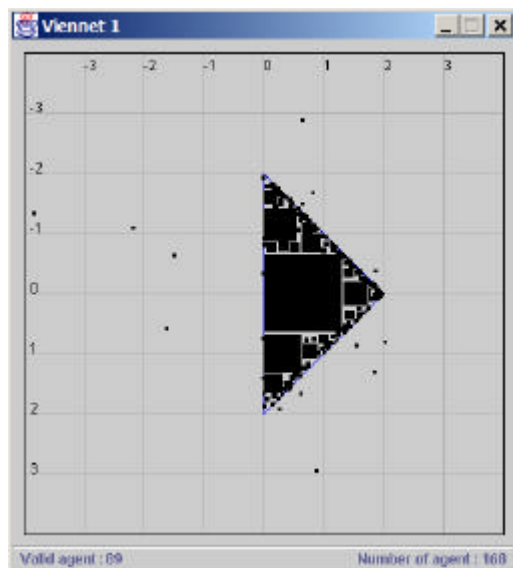


Fig. 6a. Ensemble Pareto-optimal pour le problème de Viennet 1.

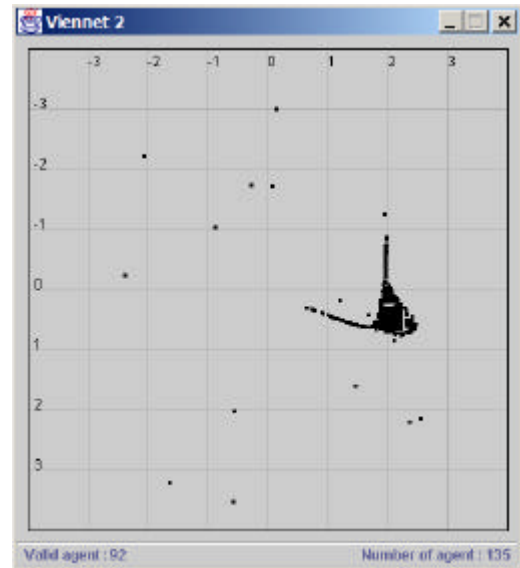


Fig. 6b. Ensemble Pareto-optimal pour le problème de Viennet 2.

On peut remarquer les différentes tailles des zones de dominance des agents. Les points noirs dispersés dans la figure sont des agents non valides qui explorent l'espace de recherche.

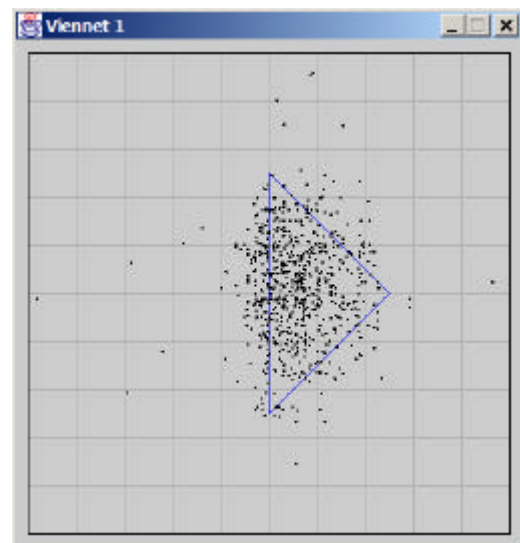


Fig. 7. Optimal Pareto set obtenu avec la méthode NPGA (1000 individus, taux de sélection 65%, mutation 5%, taille de la sous-population 10).

La première caractéristique que nous pouvons remarquer est la différence de représentation des solutions non dominées. Dans la méthode NPGA, les solutions sont représentées par des points et dans notre méthode les solutions sont représentées par des zones. Cette représentation affine la description de l'ensemble Pareto-optimal.

Dans le problème de Viennet 1, la somme de l'espace occupé par les zones d'influences (Fig 6a.) des agents

représente 90% de la zone Pareto-optimale théorique. Par comparaison, il est difficile d'estimer la zone recouverte par tous les points de la représentation de la figure 7.

Dans les figures 6 nous pouvons remarquer que la distribution des solutions non dominées est relativement homogène en comparaison avec la représentation utilisée dans la figure 7.

Enfin le mode de représentation que nous utilisons permet également au décideur de fixer le nombre d'agents qui effectueront la recherche. Cela permet d'accélérer les processus de recherche. Le résultat final est moins précis mais les premières solutions obtenues sont éloignées du bord de la zone Pareto-optimale théorique. Ces solutions sont relativement sûres pour le décideur.

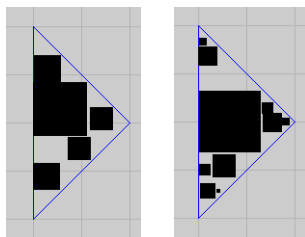


Fig. 8. Sur la figure de gauche il y a 5 agents et la somme totale des zones d'influences représente 60% de la zone Pareto-optimale théorique. Sur la figure de droite, il y a 10 agents pour une couverture de 68%.

4. Conclusion

Une méthode multiagent a été proposée. En comparaison avec les méthodes basées sur les algorithmes génétiques, notre méthode utilise peu de paramètres et ceux-ci sont compréhensibles pour le décideur.

Nous avons également proposé une représentation différente des solutions non dominées. Cette représentation nous évite d'utiliser des techniques de sharing pour disperser les solutions non dominées sur l'ensemble de la zone Pareto-optimale, ou une technique de clustering pour réduire les zones encombrées. De plus nous pouvons comparer deux solutions non dominées à l'aide de la mesure sur la taille des zones d'influences.

Les premiers tests de cette méthode ont été satisfaisants mais nous savons que cela n'est pas suffisant. Nous devons :

- effectuer des tests sur des problèmes plus compliqués dans lesquels la zone Pareto-optimale théorique est discontinue, avec une distribution non uniforme ou non connexe des solutions,
- effectuer des comparaisons avec d'autres méthodes qui utilisent une archive pour stocker les solutions intermédiaires,
- réduire la complexité de la méthode,

- ajouter la communication entre agent pour augmenter l'efficacité de la recherche.

Bibliographie

- [1] Berro, A., Duthen, Y.: Search for optimum in dynamic environment : a efficient agent-based method. GECCO'2001 Workshop on Evolutionary Algorithms for Dynamic Optimization Problems, San Francisco, California, (2001) 51-54
- [2] Coello, C., A., C., Toscano, G., T.: Multiobjective Optimization using a Micro-genetic Algorithm. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001), San Francisco, California (2001) 274-282
- [3] Corne D., W., and al.: PESA II : Region-based Selection in Evolutionary Multiobjective Optimization. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001), San Francisco, California (2001) 283-290
- [4] Deb, K.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multiobjective Optimization : NSGA II. Parallel problem Solving from Nature – PPSN VI, Springer Lecture Notes in Computer Science (2000) 849-858
- [5] Fonseca, C., M., Fleming, P., J.: Genetic Algorithm for Multiobjective Optimization : Formulation, Discussion and Generalization. In Proceedings of the Fifth International Conference on Genetic Algorithms, San Mateo, California, (1993) 416-423
- [6] Goldberg, D., E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, Massachusetts (1989)
- [7] Goldberg, D., E.: Sizing Populations for Serial and Parallel Genetic Algorithms. In J. David Schaffer Ed., Proceedings of the Third International Conference on Genetic Algorithm, San-Mateo, California (1989) 70-79
- [8] Horn, J., Nafpliotis, N.: Multiobjective Optimisation using the Niche Pareto Genetic Algorithm. Illigal TR. n° 93005 (1993)
- [9] Knowles, J., D., Corne, D., W.: The Pareto Archived Evolution Strategy : A New Baseline Algorithm for Multiobjective Optimisation. Congress on Evolutionary Computation, Washington (1999) 98-105
- [10] Knowles, J., D., Corne, D., W., Oates, M., J.: The Pareto-Envelope based Selection Algorithm for Multiobjective Optimization. In Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature (PPSN VI), Berlin (2000) 839-848
- [11] Knowles, J., D., Corne, D., W.: M-PAES, A Memetic Algorithm for Multiobjective Optimization. In Proceedings of the 2000 Congress on Evolutionary Computation (2000) 325-332
- [12] Moscato, P.: On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. Technical Report N°826, California Institute of Technology, Pasadena, California, USA (1989)

- [13] Rechenberg, I.: Evolutionsstrategie : Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Stuttgart (1973)
- [14] Schaffer, D.: Multiple Objective Optimisation with Vector Evaluated Genetic Algorithm. In genetic Algorithm and their Applications : Proceedings of the First International Conference on Genetic Algorithm (1985) 93-100
- [15] Srivinas, N., Deb, K.: Multiobjective Optimization using Nondominated Sorting in Genetic Algorithms. Technical Report, Departement of Mechanical Engineering, Institute of Technology, India, (1993)
- [16] Zitzler, E., Thiele, L.: An Evolutionary Algorithm for Multiobjective Optimization : The Strength Pareto Approach. TIK-Report n° 43 (1998)