Une Méthode d'Optimisation par Essaimes Particulaire pour le Problème de Collectes et de Livraisons (PCL)

Ali LEMOUARI et Mohamed BENMOHAMED

Laboratoire LIRE, Département Informatique de Constantine
Rue Aïn El-Baye, 25000, Constantine, Algérie
Lemouari_ali@yahoo.fr et ibnm@yahoo.fr

Résumé : Le problème de collectes et livraisons (PDP : Pick-up and Delivery Problem), est un problème d'optimisation qui consiste à trouver un ensemble de tournées, satisfont un ensemble de demandes de transport. Pour réaliser ces tournées, une flotte de véhicules est disponible. Chaque demande de transport est caractérisée par une charge de transport, un ou plusieurs sites d'origine ou de collecte (*pick-up*), et un ou plusieurs sites de destination ou de livraison (*delivery*). Dans ce travail une nouvelle méthode d'optimisation par essaimes particulaires est proposée pour le résolution de ce problème. Les résultats trouvés, montrent que la méthode proposée est favorablement comparée aux algorithmes proposés dans la littérature, et particulièrement l'algorithme mimétique (AM) .

Mots clés : Optimisation par Essaimes Particulaires, Tournés de Véhicules, Problème de Collectes et de Livraisons. Algorithme Mimétique.

1. Introduction

L'optimisation des tournées de véhicules concerne l'optimisation des parcours d'un ou plusieurs véhicules destinés à rendre un service. De nombreuses variantes du problème ont été étudiées. Ces recherches, souvent menées de manière indépendante, ont parfois abouti à des terminologies différentes pour des problèmes proches.

Nous intéressons dans ce travail au problème de transport de personnel entre plates-formes pétrolières. Celui-ci est un problème réel que les compagnies pétrolières rencontrent lorsqu'elles doivent déplacer leur personnel entre sites terrestres très éloignés ou entre des plates-formes en haute mer. Pour faire ces déplacements, les compagnies se servent d'hélicoptères ayant une capacité en nombre de passagers et une autonomie de vol limitées. Le problème consiste à déterminer pour un ensemble donné de demandes de transport, dans quelle tournée de l'hélicoptère elles seront satisfaites, et à quel moment dans la tournée.

Notre travail consistera à proposer une solution au problème en se basant sur la méthode d'optimisation par essaimes particulaires. Ainsi pour le faire quatre section serons développées dans la suite de ce travail : premièrement une présentation du problème avec les contraintes liées, puis les méthodes de résolution utilisés pour résoudre les types de ce problème, la section qui suit est consacrée à un algorithme à base de population (Algorithme Mimétique) qui sert comme base pour comparer notre méthode proposée. La dernière section est réservée à la méthode proposée pour résoudre le PCL.

2. Présentation du Problème

Le problème de Collectes et Livraisons Général - PCLG (GPDP : General Pick-up and Delivery Problem), est un problème d'optimisation qui consiste à trouver un ensemble de tournées, satisfont un ensemble de demandes de transport. Pour réaliser ces tournées, une flotte de véhicules est disponible. Chaque demande de transport est caractérisée par une charge de transport, un ou plusieurs sites d'origine ou de collecte (*pick-up*), et un ou plusieurs sites de destination ou de livraison (*delivery*). Lorsqu'une demande de livraison est prise en charge, elle ne peut être effectuée que par un seul véhicule et sans livraison intermédiaire de cette charge sur un site autre que la destination.
Ces problèmes ont de nombreuses applications parmi lesquelles nous trouvons, par exemple, le transport de marchandises, la distribution de colis, le transport de personnes par taxi, le transport de personnes handicapées et le transport de personnel.
Le problème général (PCL) est défini de la façon suivante : Soit *n* le nombre de demandes de transport. Chaque demande de transport *i* est caractérisée par une charge q_i à transporter de l'ensemble des origines N_i^+ vers l'ensemble des destinations N_i^-. On suppose que cette charge est répartie sur les origines et les destinations de la façon suivante : $q_i = \sum_{j \in N_i^+} q_j = \sum_{j \in N_i^-} q_j$. De cette façon, les sites de collecte ont une charge positive et les sites de livraison une charge négative.

Soit $N^+ = \cup_{i \in N} N_i^+$ l'ensemble de toutes les origines, $N^+ = \cup_{i \in N} N_i^-$ l'ensemble de toutes les destinations, et $N = N^+ \cup N^-$. Soit M l'ensemble des véhicules disponibles. Chaque véhicule $m \in M$ a une capacité maximale Q_m. Soit $M^+ = \{m^+ | m \in M\}$ l'ensemble des sites de départ (dépôts) des véhicules, $M^- = \{m^- | m \in M\}$ l'ensemble des sites d'arrivée des véhicules, et $W = M^+ \cup M^-$. Pour tout $(i, j) \in N \cup W$, on note d_{ij} la distance, t_{ij} le temps et c_{ij} le coût associés au trajet (i, j).

2.1 Problème traité

Notre travail consiste à l'application de la méthode d'optimisation par essaimes particulaires (OEP), au problème PCL, et plus particulièrement aux problèmes des compagnies pétrolières. Chaque jour des personnes doivent être transportées de plate-forme à plate-forme. Chaque demande de transport concerne une ou plusieurs personnes qui désirent être transportées d'une plate-forme de départ à une plate-forme d'arrivée. Ces transports sont faits par des hélicoptères ayant une capacité limitée. L'objectif des compagnies pétrolières est de déterminer les tournées de façon à minimiser les coûts de transport par hélicoptères, tout en satisfaisant les demandes de transport.
Nous disposons d'un jeux de données généré aléatoirement et téléchargeable de www.emn.fr/guéret., ainsi généré des problèmes sur une carte de 50 km de côté pour lesquels 5% des sites ont une capacité d'approche limitée à 5 passagers. L'hélicoptère a une capacité de 13 passagers. Quatre séries de 100 problèmes chacune ont été générées, dans lesquelles le nombre de sites et le nombre de demandes de transport sont variés. Ainsi la première série contient des problèmes avec 10 demandes sur 5 sites **(5S-10D)**, la deuxième des problèmes avec 20 demandes sur 5 sites **(5S-20D)**, la troisième des problèmes de 30 demandes sur 10 sites **(10S-30D)**, et la dernière des problèmes avec 50 demandes sur 20 sites **(20S-50D)**.

2.2 Description et Contraintes liés au problème

Soit n le nombre de demandes de transport à satisfaire. Chaque demande de transport k concerne un seul passager qui doit être transporté d'un site d'origine p_k à un site de destination d_k. Le temps nécessaire pour embarquer ou débarquer un passager sur un site i est appelé temps de service et est noté s_i. Pour satisfaire les n demandes de transport, un seul hélicoptère est disponible. Celui-ci est caractérisé

par une capacité maximale Q en nombre de passagers et une durée maximale de vol T au bout de laquelle il doit se ravitailler en carburant à la base. Une capacité d'approche a_i est affectée à chaque site. Elle correspond au nombre maximal de passagers dans l'hélicoptère au moment de son atterrissage sur un site, ceci pour des raisons de sécurité. Dans ce problème les distances sont euclidiennes. Ainsi pour deux sites i et j, la distance d_{ij} respecte l'inégalité triangulaire $(d_{ij} \leq d_{ik} + d_{kj})$ et $d_{ij} = d_{ji} \quad \forall i, \forall j, \forall k$. Le temps de trajet t_{ij} entre deux noeuds i et j est pré-calculé à partir de la distance entre les sites et de la vitesse moyenne de l'hélicoptère.

Ce problème contient les contraintes classiques du PCL qui sont :

- **Couplage :** Pour chaque demande de transport k, le site de collecte p_k et le site de livraison d_k doivent être visités dans la même tournée.
- **Précédence :** Pour une demande de transport k, le site de collecte p_k doit être visité avant le site de livraison d_k.
- **Capacité :** Le nombre de passagers dans l'hélicoptère ne doit pas dépasser Q.

Les deux contraintes spécifiques suivantes doivent de plus être respectées :

- **Capacité d'approche :** Pour des raisons de sécurité, le nombre maximal de passagers dans l'hélicoptère lors de son atterrissage sur un site peut être limité.
- **Durée maximale des tournées :** Étant donné que l'hélicoptère a une durée de vol limitée T au bout de laquelle il doit retourner se ravitailler en carburant à sa base et qu'il ne peut retourner à la base s'il a des passagers à son bord, la durée des tournées est elle aussi limitée à T.

L'objectif de ce problème est de satisfaire toutes les demandes de transport en minimisant la distance totale parcourue.

3. Méthodes de Résolutions Utilisées

À notre connaissance peu de chercheurs se sont intéressés aux PCL rencontrés dans les problèmes de transport de personnel par hélicoptère. Fiala-Timlin et autres ont résolu un problème de transport de personnel avec plusieurs hélicoptères et des priorités sur les demandes (Fiala-Timlin et al., 1992). L'approche utilisée est une heuristique à trois phases dans laquelle les demandes sont groupées par priorité, ensuite un problème est résolue par priorité grâce à une heuristique d'insertion parallèle.

En ce qui concerne les métaheuristiques, Tang-Montané et autres ont développé une recherche taboue pour résoudre un problème de transport de personnel avec collectes et livraisons simultanées mais indépendantes, c'est-à-dire que chaque demande est soit une collecte, soit une livraison (Tang-Montané et al., 2006). Ils ont utilisé cinq voisinages : déplacement d'une demande d'une tournée à une autre, permutation de demandes entre deux tournées, échange d'une partie de tournée entre deux tournées, croisement entre deux tournées (deux tournées sont divisées en deux et les parties sont échangées en produisant deux nouvelles tournées), et 2-opt. D'autre part, Torres a développé un algorithme génétique qui utilise des algorithmes proches de ceux de (Fiala-Timlin et al., 1992). Avec cette procédure l'auteur résout des problèmes à un seul hélicoptère (Torres, 2004).

Parmi les méthodes exactes, une approche de partitionnement avec génération de colonnes a été développée par Tijssen pour résoudre un problème avec 51 sites et plusieurs hélicoptères dans lequel la préemption des demandes (possibilité de partitionner la charge d'une demande) est autorisée et dans laquelle à chaque fois qu'une personne monte dans l'hélicoptère, une autre descend (Tijssen, 2000). La méthode fournit de bons résultats mais les temps de calcul sont élevés.

En ce qui concerne l'application du transport de personnel par hélicoptère, nous trouvons peu de travaux et des problèmes comportant des caractéristiques différentes. Les méthodes utilisées pour résoudre ces problèmes, sont également variées.

3.1 Heuristiques D'insertion et d'amélioration

Les heuristiques gloutonnes utilisées pour les problèmes classiques de tournées de véhicules. Ce sont des méthodes d'insertion qui construisent un ensemble de tournées en insérant les demandes de transport les unes après les autres dans un certain ordre. Plusieurs critères peuvent être utilisés pour trier les demandes: (i) Plus Proche Demande (PPD), (ii) Plus Lointaine Demande (PLD), (iii) Plus Proche Centre de Gravité (PPCG), (iv) Plus Lointain Centre de Gravité (PLCG), (v) Meilleure Insertion (MI), et la dernière (vi) Insertion Aléatoire (IA). La complexité globale de ces heuristiques est $O(n^3)$ pour les versions PPD, PLD, PPCG et PLCG et IA, et $O(n^4)$ pour MI.

3.2 Recherches locales

Les méthodes de recherches locales utilisés dans le cadre de tournés de véhicules appartiennent à quatre classes de voisinages : (i) **Voisinage Déplacement d'une Demande,** L'exploration de ce voisinage est de complexité $O(n^3)$. (ii) **Voisinage Permutation de Demandes :** La complexité totale pour explorer ce voisinage est $O(n^4)$. (iii) **Voisinage 2-Opt :** Ce voisinage contient $O(n^2)$ solutions. Chaque voisin est construit en $O(n)$. L'exploration du voisinage coûte donc $O(n^3)$. (iv) **Voisinage Déplacement Site :** Cette méthode de voisinage permet de déplacer une séquence de noeuds consécutifs représentant le même site géographique à un autre emplacement dans la tournée.

3.3 Algorithme Mimétique

Nous décrivant dans cette section un algorithme mimétique proposé dans (Prins, 2004) et utilisé dans (Velasco et al. 2005) pour le transport du personnel par hélicoptère. Il sert comme une base pour comparer notre méthode proposée. Les algorithmes mimétiques construisent une population composée d'un ensemble de solutions. Comme dans les algorithmes génétiques, de plus contiennent une recherche locale qui peut être appliquée dans différentes phases de l'algorithme. Récemment Prins (Prins, 2004) a présenté un AM pour résoudre un problème de tournées de véhicules. Dans son approche l'auteur utilise un chromosome sans délimiteurs de tournées et une procédure ***"Split"*** qui permet de couper un chromosome en tournées optimales en respectant l'ordre de la séquence du chromosome.

3.3.1 Description de l'Algorithme Mimétique

L'algorithme mimétique proposé dans (Prins, 2004) pour les tournées de véhicules, et utilisés après dans (Velasco et al. 2005) pour le PCL. Comme tout algorithme génétique, un AM est défini par : **(i)** un codage des solutions, **(ii)** l'initialisation de la population, **(iii)** un opérateur de croisement, **(iv)** un opérateur de mutation, **(v)** des règles de remplacement, et **(vi)** des critères d'arrêt.
Comme dans plusieurs algorithmes génétiques pour le problème de voyageur de commerce (PVC), un chromosome est une séquence (permutation) S de noeuds, sans délimiteurs de tournées. Le chromosome adopté est une liste de demandes de transport indexées, dans laquelle chaque demande de transport k apparaît deux fois : une fois comme k+ qui indique le nœud de collecte et l'autre fois comme k- pour le noeud de livraison. Un exemple de chromosome est donné par : S = (0, 1+, 1-, 2+, 2-, 3+, 4+, 4, 3-), 0 représentant la base de l'hélicoptère. Afin de couper ce chromosome en tournées, une procédure optimale appelée ***''Split''*** est utilisée, qui permet de déterminer la meilleure solution respectant la séquence et avec respect des contraintes de problème. La complexité de ***split*** est calculé en $O(n^2)$ en utilisant l'algorithme de Bellman pour des graphes acycliques (Cormen et al.1990). À chaque itération, deux parents sont

sélectionnés par la technique du tournoi binaire. Un opérateur de croisement est appliqué à ces deux parents pour générer deux fils. Une fois le croisement des deux parents effectué, un enfant est sélectionné aléatoirement. Il est transformé en une solution composée de plusieurs tournées grâce à la procédure ***Split*** et il est amélioré par une procédure de recherche locale avec une probabilité fixée. Cet enfant remplacera un chromosome de la population. Pour cette recherche locale, les différents voisinages mentionnés en dessus ont été testés.

3.4 La méthode d'Essaimes Particulaires et le Problème de Tournées.

A notre connaissance il n'y a que peu de travaux, consacrés au problème de tournés de véhicules (voir deux à trois articles), adoptant la méthode d'optimisation OEP au problème de tournés de véhicules. Par exemple dans (Qing et al. 2006), l'OEP est adaptée au problème de tournées de véhicules avec fenêtre horaires. Une des représentations est d'utiliser un vecteur composé de séquence de nœuds pour lequel les tournées sont séparées par le nœud représentant la base (d'ailleurs c'est la même représentation utilisée aussi pour le problème PDP). Supposons que la base est représentée par le nœud 0 un exemple de solution pour 6 clients est représenté comme suit (**Fig. 1**) :

0	5	0	4	3	6	0	2	1	0

Fig. 1 - Séquence de nœuds solution

La représentation d'un individu est basée sur ce principe, et pour lequel les nœuds en extrémité sont supprimer et ceux au milieu sont décalés à gauche, un individu est représenté à l'aide d'un vecteur de dimension $n+k-1$ et a la forme suivante pour le vecteur en figure **Fig 1**.

1	2	3	4	5	6	0	0
8	7	4	3	1	5	2	6

Fig. 2 - Représentation d'un individu

Les deux derniers emplacements dans *index* de tableau de la figure **Fig. 2**, représentent les positions de séparation des tournées. La méthode est appelée uniquement pour des problèmes de petite taille. Il est bien évident que la recherche de la solution à base de cette méthode consiste à trouver les meilleurs positions dans le vecteur *index*, ce qui n'est pas facile avec l'utilisation directe de la méthode OEP.

4. Méthode Proposée

Nous pensons que l'application de la méthode OEP pour des problèmes de tournés, passe par la modification de système d'équation en préservant la philosophie de la méthode. Ce point de vue a été bien appelé par (Allahverdi et Alanzi, 2006) pour le problème d'ordonnancement dans les bases de données distribuées. Nous utilisant ici la même philosophie, avec quelques modifications appropriés afin que la méthode s'applique bien pour le problème de collecte et de livraison (PCL).
La méthode d'Optimisation en Essaims Particulaires (OEP), consiste en un essaime de particules. Ces particules coexistes et en évolution dans l'espace de recherche, basés sur leurs expérience et le savoir partagés avec le voisinage. Chaque particule possède deux paramètres, la position et la vitesse $(x(t), v(t))$, la population vole dans un espace de recherche à base des deux équations suivantes :

$$x(t+1) = x(t) + v(t+1) \quad \textbf{(1)}$$

$$v(t+1) = wv(t) + c_1\varphi_1(pbest - x) + c_2\varphi_2(gbest - x) \quad \textbf{(2)}$$

$v(t)$: est la vitesse au temps t . $x(t)$: la position au temps t . $p(t)$: la meilleure position antérieure obtenue par le particule. $g(t)$: la meilleure position obtenue dans le voisinage de particule. w : le facteur d'inertie. φ_1, φ_2 : deux variables aléatoire dans $[0,1]$. $c_1\varphi_1(p(t)-x(t))$ est la part de l'apprentissage cognitive, et $c_2\varphi_2(g(t)-x(t))$ est la part de l'apprentissage social.

4.1 Représentation de données

Il est bien évident que la représentation des données proposée dans les travaux ci-dessus, et plus particulièrement la représentation de chromosome utilisée dans les algorithmes mimétiques, présente les inconvénients :

- L'application des opérateurs de croisement sur les chromosomes solutions, conduit à la violation des contraintes de problème (contrainte de précédence, contrainte de la durée maximale d'une tournée).
- Quelque soit le traitement proposé pour respecter les contraintes : de précédence et de la durée d'une tournée, le résultat conduit toujours à un nombre important de solutions non réalisables.

Dans ce qui suit nous utilisons une représentation avec les nœuds de collectes seulement. Les autres nœuds seront insérés à l'aide d'une procédure d'insertion des nœuds de livraison, dont la description est la suivante :

4.2 Méthode d'insertion des nœuds de collectes

Considérons l'exemple suivant, avec la séquence des requêtes $\{1,2,3,4\}$, représentant un problème de quatre demandes. La représentation d'une solution quelconque, à l'aide d'un vecteur de dimension $2n$ et à l'aide d'un vecteur de dimension n peut être donnée comme suit :

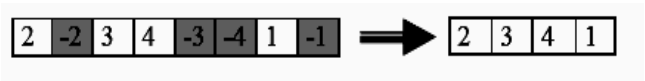

Fig. 3 Représentation à l'aide des nœuds de collectes

Les nœuds $\{-2,-3,-4,-1\}$ seront insérés en respectant l'ordre de la séquence des requêtes, c'est-à-dire l'ordre de traitement des nœuds de collectes $\{2,3,4,1\}$. A chaque insertion d'un nœud de livraison, les contraintes du problème doivent être vérifié. Si l'insertion d'un nouveau nœud de livraison provoque la violation des contraintes, ce dernier sera inséré avec le nœud de collecte correspondant dans la tournée suivante. La construction de la totalité du vecteur évaluable (vecteur contenant les nœuds de collecte et de livraison) s'effectue en parallèle dans un autre vecteur qui sera évalué à l'aide de la procédure ***split*** décrite ci-dessus. Il est à noté que l'utilisation de la procédure ***split*** ici sert pour évaluer uniquement la séquence. Par exemple, pour insérer le nœud de collecte (1) il y a deux possibilités :

- Soit avec le respect de l'ordre des nœuds de livraison, dans ce cas le nœud peut avoir quatre possibilités, à partir de la position entre les nœuds $(4,-2)$, comme le montre la figure **Fig. 4**. Le nombre de possibilité pour insérer un nœud de livraison est inférieur ou égale à n, avec n est le nombre de requêtes.
- Soit sans respect de l'ordre des nœuds de collectes, dans ce cas le nœud est inséré à la première position et le nombre de possibilité pour insérer les nœuds de collecte égale à $2n$.

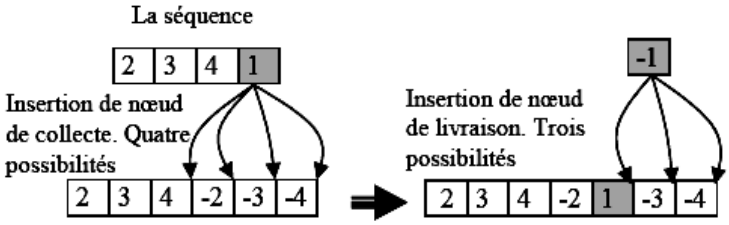

Fig. 4 Exemple d'insertion des nœuds représentant la requête n°1.

Dans le but de trouver une meilleure insertion, pour tous les nœuds de livraison insérés. Les nœuds de livraisons insérés en premiers sont déplacés vers les nouvelles positions crées, comme le montre l'exemple de la figure **Fig. 5**. Le test de toutes les positions de l'ensemble des nœuds de livraisons s'effectué en $O(n^2)$.

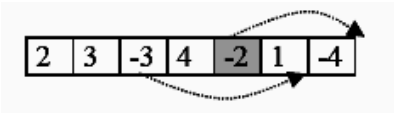

Fig. 5 Exemple de déplacement des noeuds.

Pour les deux cas, le nombre de possibilité totale d'insertion des nœuds de livraison NL, en tenant compte de toutes les possibilités d'insertion des nœuds de collectes.

Soit n : le nombre de nœuds de requête.

- **Premier Cas**

$$NL = \frac{1}{6}n(n+1)(n+2) + n^2 \quad \textbf{(3)}$$

La complexité dans les pires des cas est $O(n^3)$.

- **Deuxième Cas 2**

$$NL = \frac{1}{6}n(n+1)(4n+1) + n^2 \quad \textbf{(4)}$$

La complexité dans les pires des cas est $O(n^3)$.

Algorithme 6.2 – Algorithme d'insertion des nœuds.

1. Entrée : Vecteur de collectes : *Pickup* % contient les nœuds de collecte (ou requêtes)
2. Insérer la première requête (nœud de collecte et de livraison) dans un vecteur *Result*.
3. Pour $i=2,..,N$ Faire
   {Prendre *Pickup [i]* %insertion d'un nœud de collecte ;
   K=indice du dernier nœud de collecte inséré.
4. Pour $j=k+1,..,size(result)$ Faire {
5. Insérer *Pickup [i]* dans l'emplacement *result[j]* ;
6. % insertion de nœud de livraison
7. Pour $l=j+1,..,size(result)$ Faire {
8. Insérer le nœud de livraison dans l'emplacement result[l] ;
9. f= Evaluer la séquence obtenue ;
10. If f<min {prendre positions des nœuds de collecte et de livraison} ; }
11. Supprimer le nœud de collecte inséré ;}
12. Insérer les nœuds de collecte et de livraison dans les positions calculés ;
13. Si les contraintes sont respectées alors nouveau nœud
14. Sinon nouvelle tournée}
15. Evaluer result ;

Algorithme 1 – Algorithme d'insertion des nœuds.

4.3 Adaptation de l'OEP pour le PCL

Adapter la méthode OEP pour le problème de collectes et de livraisons, avec préservation de la philosophie de la méthode, nécessite en premier une réécriture de système équations de base de l'OEP (équation (1) et (2)) on se basant uniquement sur les deux parts d'accélération ou proprement dit sur les parts d'apprentissages. Ce qui fait sur les deux mémoires de l'individu et de la population.
Il est évident que les deux parties de l'accélération, cognitive ou social, permet le changement dans la position des individus vers le meilleure dans l'expérience, et le meilleur de la population par $c_1\varphi_1(pbest - x)$, et $c_2\varphi_2(gbest - x)$ respectueusement.

Soit x, $pbest$, et $gbest$ trois vecteurs, représentant des tournées de véhicules pour le problème de collecte et de livraison. Avec x : la solution actuelle. $pbest$: la meilleure solution obtenue par l'individu dans le passé. $gbest$: la meilleure solution obtenue par la totalité de l'essaim.
Les équations **(1)** et **(2)** seront modifiés comme suit :

- La solution actuelle x obtenue par un individu sera modifiée, en déplaçant l'individu vers son meilleure position visité jusqu'ici *pbest* , c'est-à-dire son expérience par la grandeur suivante :

$$V_1 = c_1\varphi_1 d_1 \quad \textbf{(5)}$$

- Le résultat obtenu à l'aide de l'équation **(5)**, c'est-à-dire la nouvelle position de l'individu sera soumise à des changements à base de la deuxième part de l'accélération. Par conséquent l'individu est déplacé vers la meilleure position obtenue par la totalité de l'essaim $gbest$, comme suit :

$$V_2 = c_2\varphi_2 d_2 \quad \textbf{(6)}$$

Où d_1 et d_2 sont deux scalaires appartenant à l'intervalle $[0\ 1]$ implémentant la différence entre les couples de vecteurs $(pbest, x)$ et $(gbest, (pbest, x))$, et V_1, V_2 sont les vitesses de déplacements.

L'exemple suivant explique l'utilisation des équations **(5)** et **(6)**. Soit X_1, X_2 deux vecteurs représentant les nœuds d'un graphe. Chacun des nœuds est codé en un entier positif :

X_1	2	1	3	4	6	7	9	8	10	5
X_2	9	5	6	7	3	4	1	8	10	2

Trouver une mesure de différence entre deux vecteurs peut s'effectuer à l'aide de la *distance de Hamming.* L'inconvénient majeur de cette distance est lorsque l'un des vecteurs est le décalage circulaire de l'autre, la distance entre les deux vecteurs est maximale pourtant ces derniers sont les mêmes. Pour cela nous cherchons tout d'abords à trouver une mesure de ressemblance $R(X_2, X_1)$ entre deux vecteurs, cette mesure représente l'ensemble des arrêtes appartenant au deux vecteurs. Formellement soit A_1, A_2 les arrêtes représentant X_1, et X_2.

$$A_1 = \{(2,1)\ (1,3)\ (3,4)\ (4,6)\ (6,7)\ (7,9)\ (9,8)\ (8,10)\ (10,5)\}$$
$$A_2 = \{(9,5)\ (5,6)\ (6,7)\ (7,3)\ (3,4)\ (4,1)\ (1,8)\ (8,10)\ (10,2)\}$$
$$R(X_2, X_1) = (\sum_{i=1}^{N-1} n_i)/(N-1) \qquad \textbf{(7)}$$
$$n_i = \begin{cases} 1 & Si\ l'arrête\ i \in A_1, i \in A_2 \\ 0 & Sinon \end{cases}$$

Dans l'exemple $R(X_2, X_1) = 1/3$, la différence entre les deux vecteurs est :

$$d(X_2, X_1) = 1 - R(X_2, X_1) \qquad \textbf{(8)}$$

Nous appelons cette distance la *distance des paires*. Dans l'exemple la différence est $2/3$. Une fois la différence entre les deux vecteurs est calculée, la vitesse de changement de l'un des deux vecteurs vers l'autre est obtenue à l'aide de l'équation $V = c\varphi d$. Admettons que $c = 1$, et $\varphi = 0.5$, alors la vitesse de déplacement de vecteur X_1 vers X_2 est de $V = 1/3$.

Afin d'appliquer les changements sur le vecteur X_1, une variable aléatoire ρ est générée pour chaque arrête dans X_2. Si $\rho < V$ l'arrête en question est sélectionnée pour participer au vecteur *résultat*. Si à la fin le vecteur *résultat* n'est pas encore complet, celui-ci est complété à partir des éléments de X_1, ce dernier est parcouru d'une manière circulaire du dernier élément trouvé dans le vecteur *résultat*. Admettons que $\rho = \{0.3, 0.5, 0.2, 0.9, 0.1, 0.2, 0.3, 0.7, 0.05\}$, le vecteur *résultat* sera :

Résultat	9	5	7	4	1	8	2	3	6	10

Sélectionné de X_2 *Complété de* X_1

L'algorithme suivant, explique l'utilisation de la méthode pour le problème de collecte et de livraison.

Algorithme 6.3 - Algorithme d'OEP pour le PCL

1. Initialisation
   a. Définir une taille pour l'essaim (nombre d'individus), et initialiser $(X, Pbest)$ les individus aléatoirement.
   b. Initialiser les paramètres c_1, c_2, φ_1, φ_2
   c. Evaluer ces individus ; puis calculer le meilleur dans l'essaim, le $gbest$.

2. While (Nbe_Iteration ≠ Maximum) Do
3. For $(i = 1,\ Taille\ Essaimes)$ Do {
4. Evaluer $V_1 = c_1\varphi_1 d_1$, en déplaçant X vers $Pbest$ par V_1 .
5. Evaluer $V_2 = c_2\varphi_2 d_2$, en déplaçant le résultat obtenue, vers $Gbest$ par V_2 . Le résultat donnera X à l'instant $(t+1)$.
6. Modifier le vecteur résultat en ajoutant les nœuds de livraison à l'aide de la procédure ***Insertion_Noeuds***. Les résultats obtenus, représentent des solutions réalisables.
7. évaluer l'individu.
8. If $Eval(X) < Eval(pbest)$ alors Remplacer $pbest$ par X
9. If $Eval(X) < Min$ alors
10. $Min = Eval(X)$; $position = i$ (i : est l'individu d'ordre i) }
11. $Gbest$ = L'individu d'ordre $Position$. }

Algorithme 2 – Algorithme d'optimisation par essaimes pour le PCL.

4.4 Résultats et Testes

Les algorithmes décrits ci-dessus ont été réalisés en **Java Eclipse©** testés sur une machine Pentium VI d'horloge 3Mhz et sur le jeu de données aléatoire décrit ci-dessus téléchargeable sur www.emn.fr/guéret. Le choix des paramètres dépend de problème en question. Les valeurs suivantes ont été fixées pour chaque type de problème $c_1 = 1/2, c_2 = 1/2$. La vitesse de déplacement D_i avec $i = 1,2$, variée d'un problème à l'autre. Pour les problèmes de types $5s10r$ la vitesse $D_i \in [1/2 \quad 9/10]$. Les problèmes de types $5s20r$ $D_i \in [4/10 \quad 6/10]$. Les problèmes de types $10s30r$ $D_i \in [2/10 \quad 4/10]$. et les problèmes de types $20s50r$ $D_i \in [0 \quad 2/10]$. Le choix des intervalles dépend de la dimension du vecteur solution. La taille de l'essaim est fixée au nombre de demandes et elle varié de 10 à 50 individus.

Type de Problème	5s10r	5s20r	10s30r	20s50r
Temps d'Exécution Moyen	5,6	20,5	73,97	344,97

Table 6.1 – Temps moyen pour 20 instances pour chaque problème.

Problème 5s20r

#Instance	Mimétique Algorithme	OEP Algorithme	Gain Obtenu
5s10r_1	384,69	384,69	
5s10r_2	356,29	356,29	
5s10r_3	368,26	**361,95**	**6,31**
5s10r_4	301,70	301,70	
5s10r_5	317,36	317,36	
5s10r_6	386,54	386,54	
5s10r_7	326,37	326,37	
5s10r_8	334,97	334,97	
5s10r_9	284,89	284,89	
5s10r_10	314,87	314,87	
5s10r_11	364,75	364,75	
5s10r_12	323,78	**323,77**	**0,01**
5s10r_13	320,55	320,55	
5s10r_14	354,34	354,34	
5s10r_15	280,73	280,73	
5s10r_16	299,10	299,10	
5s10r_17	362,45	362,45	
5s10r_18	288,27	288,27	
5s10r_19	306,76	306,76	
5s10r_20	268,90	268,90	

Table 6.3 - Résultat des tournées pour 20 instances de (5s20r).

Problème 5s20r

#Instance	Mimétique Algorithme	OEP Algorithme	Gain Obtenu
5s10r_1	384,69	384,69	
5s10r_2	356,29	356,29	
5s10r_3	368,26	**361,95**	**6,31**
5s10r_4	301,70	301,70	
5s10r_5	317,36	317,36	
5s10r_6	386,54	386,54	
5s10r_7	326,37	326,37	
5s10r_8	334,97	334,97	
5s10r_9	284,89	284,89	
5s10r_10	314,87	314,87	
5s10r_11	364,75	364,75	
5s10r_12	323,78	**323,77**	**0,01**
5s10r_13	320,55	320,55	
5s10r_14	354,34	354,34	
5s10r_15	280,73	280,73	
5s10r_16	299,10	299,10	
5s10r_17	362,45	362,45	
5s10r_18	288,27	288,27	
5s10r_19	306,76	306,76	
5s10r_20	268,90	268,90	

Table 6.3 - Résultat des tournées pour 20 instances de (5s20r).

Problème 10s30r

#Instance	Mimétique Algorithme	OEP Algorithme	Gain Obtenu
5s10r_1	464,26	464,26	
5s10r_2	524,91	**524,61**	**0,29**
5s10r_3	514,21	514,21	
5s10r_4	533,39	533,48	**0,09**
5s10r_5	458,10	458,10	
5s10r_6	528,74	528,74	
5s10r_7	814,80	**510,42**	**4,38**
5s10r_8	576,27	**572,82**	**3,44**
5s10r_9	494,21	494,21	
5s10r_10	641,58	**631,67**	**9,91**
5s10r_11	489,45	**487,98**	**1,47**
5s10r_12	508,15	508,15	
5s10r_13	507,28	507,28	
5s10r_14	490,51	490,51	
5s10r_15	464,17	464,17	
5s10r_16	482,09	**481,50**	**0,58**
5s10r_17	477,61	**473,83**	**3,78**
5s10r_18	474,66	**463,13**	**11,52**
5s10r_19	442,07	442,07	
5s10r_20	498,29	**496,68**	**1,60**

Table 6.4 - Résultat des tournées pour 20 instances de (10s30r).

Problème 20s50r

#Instance	Mimétique Algorithme	Essaim Particulaire	Gain Obtenu
5s10r_1	756,24	756,24	
5s10r_2	874,77	**871,50**	**3,26**
5s10r_3	946,76	**945,78**	**0,97**
5s10r_4	785,48	**785,40**	**0,08**
5s10r_5	802,29	802,49	
5s10r_6	810,08	**808,79**	**1,28**
5s10r_7	981,88	981,88	
5s10r_8	764,38	764,38	
5s10r_9	911,73	911,73	
5s10r_10	896,61	**881,74**	**14,86**
5s10r_11	945,94	**942,39**	**3,55**
5s10r_12	864,65	864,65	
5s10r_13	847,32	847,32	
5s10r_14	898,62	**897,75**	**0,86**
5s10r_15	787,66	787,66	
5s10r_16	770,71	770,71	
5s10r_17	812,92	812,92	
5s10r_18	794,36	**793,17**	**1,18**
5s10r_19	906,32	**902,46**	**3,85**
5s10r_20	752,65	**752,03**	**0,61**

Table 6.5 - Résultat des tournées pour 20 instances de (20s50r).

Les résultats obtenus pour les quatre types de problèmes de collectes et de livraisons, montrent que la méthode proposée permet l'amélioration dans les résultats pour la majorité des instances des problèmes. Le taux de changement varie de 0% pour les problèmes de petite taille (5s10r) jusqu'à 50% pour les problèmes de grande taille (20s50r).

Le temps d'exécution dépend de la taille du problème et varie de quelques secondes pour les problèmes de petite taille (5s10r) jusqu'à quelques minutes (20s50r).

5. Conclusion

La méthode d'optimisation par essaimes particulaires partage beaucoup de similarité avec la algorithmes génétiques dans le sens où les propriétés d'un individu sont influencés par les caractéristiques des autres. La méthode a été appelée efficacement dans ces dernières années pour les problèmes dites d'optimisation continue. Néanmoins son application aux problèmes discrets s'avère difficile pour les deux raisons suivantes : **(i) Modélisation :** L'application de la méthode aux problèmes discret nécessite une phase d'adaptation de modèle d'équations de l'OEP. **(ii) Voisinage :** L'OEP de base est basée sur deux topologies de voisinage social, ***global best*** pour lequel l'individu est influencé par un voisinage global et ***local best*** pour lequel l'individu est influencé par un voisinage local. Chacun des deux types de voisinage présente des avantages et des inconvénients. Une bonne utilisation de la méthode nécessite la définition de la topologie de voisinage qui convient au problème.
Nous avons tenues compte des deux derniers raisons dans notre application de l'OEP au problème de collecte et de livraison, ainsi le système d'équation de base de l'OEP est modifié afin qu'elle s'adapte mieux au problème, en deuxième un opérateur (qui n'ai pas détaillé ci-dessus) est définie qui permet l'échanges entre les nœuds des deux tournées choisisses aléatoirement afin de remédier à la faiblesse de la topologie ***global best*** en face les optimums locaux.
Les résultats ainsi présentés dans les tableaux précédents montrent l'efficacité de telle approche pour les problèmes de tournées de véhicules et de collectes et livraison, et que la méthode proposée est favorablement comparée aux algorithmes génétiques combinés avec les recherches locales.
En perspective nous retenons d'appeler la méthode à des instances de grande taille par exemple à partir de deux cents nœuds, et aussi d'appeler la méthode aux problème de tournés des véhicules en utilisant les benchmarks de la littérature.

References

(Allahverdi et al., 2006) A. Allahverdi, F.S Alanzi, ''A PSO and Tabu Search Heuristic for the Assembley Scheduling Problem of the two Stage Distribution Database Application''. Computer and Operations Research 33, 1065-1080, 2006.
(Fiala-Timlin et al., 1992) M.T. Fiala-Timlin, W.R. Pulleyblank. ''Precedence constrained routing and helicopter scheduling: Heuristic design.'' Interfaces, 22 :100–111, 1992.
(Moscato, 1999) P. Moscato. "Memetic algorithms : A short introduction." In D. Corne, M. Dorigo, and F. Glover, editors, New Ideas in Optimization, pages 219–234. McGraw-Hill, 1999.
(Prins, 2004) C. Prins. "A simple and effective evolutionary algorithm for the vehicle routing problem." Computers and Operations Research, 31 :1985–2002, October 2004.
(Qing et al. 2006) Z. Qing, Q. Limin, L. Yingchun, Z. Shanjun "An improved particle swarm optimization algorithm for vehicle routing problem with time window". IEEE, Congres on Evolutionary Computation, CEC'06, 1386-1390. 2006.
(Tang-Montané et al., 2006) F. A. Tang-Montané and R. Diéguez-Galvao. "A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service". Computers and Operations Research, 33 :595–619, 2006.
(Tijssen, 2000) G.A. Tijssen. "Theoretical and practical aspects of linear optimization". PhD thesis, Graduate School/Research Institute, Systems, organization and management. University of Groningen, Netherlands, 2000.
(Torres, 2004) F. Torres. "Algoritmo genético basado en una heurística de inserción para el problema de ruteo de helicópteros". XII CLAIO-04, Congreso Latino-Iberoamericano de Investigación de Operaciones, La Habana, Cuba, 2004.
(Velasco et al. 2005) N. Velasco, P. Dejax, , C. Guéret, C. Prins "A Memetic Algorithm for a Pick-up and Delivery Problem by Helicopter", MIC'05, Vienna (Austria), 2005.