

## HEURISTIQUES POUR LE PROBLÈME DU VENDEUR *m*-PÉRIPATÉTIQUE

ÉRIC DUCHENNE<sup>1</sup>, GILBERT LAPORTE<sup>2</sup> ET FRÉDÉRIC SEMET<sup>3</sup>

**Abstract.** The *m*-Peripatetic Salesman Problem (*m*-PSP) is defined on a undirected graph  $G = (V, E)$  where  $V = \{1, \dots, n\}$  is the vertex set,  $E = \{(i, j) : i, j \in V, i < j\}$  is the edge set and  $(c_{ij})$  is a cost matrix defined on  $E$ . The *m*-PSP consists of determining *m* edge-disjoint Hamiltonian cycles of least total cost on  $G$ . This article describes seven new heuristics for the *m*-PSP and compares them with the heuristic proposed by Krarup in 1975.

**Résumé.** Le *Problème du Vendeur m-Péripatétique (m-PVP)* est défini sur un graphe non orienté  $G = (V, E)$  où  $V = \{1, \dots, n\}$  est l'ensemble des sommets,  $E = \{(i, j) : i, j \in V, i < j\}$  est l'ensemble des arêtes et  $(c_{ij})$  est une matrice de coûts définie sur  $E$ . Le *m*-PVP consiste à déterminer *m* cycles hamiltoniens sur  $G$  n'ayant aucune arête en commun et dont le coût total est minimal. Cet article décrit sept nouvelles heuristiques pour le *m*-PVP et les compare à celle qui a été proposée par Krarup en 1975.

**Mots Clés.** Problème du Vendeur *m*-Péripatétique, Problème du Voyageur de Commerce, heuristiques.

**Classification Mathématique.** 90C09, 90C59.

---

Received February 19, 2006. Accepted July 17, 2008.

<sup>1</sup> LAMIH-ROI, Université de Valenciennes, Le Mont Houy, 59313 Valenciennes, Cedex 9, France

<sup>2</sup> GERAD and Canada Research Chair in Distribution Management, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, H3T 2A7, Montréal, Canada

<sup>3</sup> LAMIH-ROI, Université de Valenciennes, Le Mont Houy, 59313 Valenciennes, Cedex 9, France; frederic.semet@univ-valenciennes.fr

## 1. INTRODUCTION

Le *Problème du Vendeur  $m$ -Péripatétique* ( $m$ -PVP) est défini sur un graphe non orienté  $G = (V, E)$  où  $V = \{1, \dots, n\}$  est l'ensemble des sommets,  $E = \{(i, j) : i, j \in V, i < j\}$  est l'ensemble des arêtes et  $(c_{ij})$  est une matrice de coûts définie sur  $E$ . Le  $m$ -PVP consiste à déterminer  $m$  cycles hamiltoniens sur  $G$  n'ayant aucune arête en commun et dont le coût total est minimal. Pour éviter les cas triviaux ou inadmissibles, on suppose que  $2 \leq m \leq (n-1)/2$  et  $n \geq 6$ . Dans cet article, un cycle  $C$  est défini comme l'ensemble des arêtes qui le constituent.

Le  $m$ -PVP a été introduit par Krarup [11] en 1975. Des applications de ce problème se retrouvent dans la planification de tournées de veilleurs de nuit qui, pour des raisons de sécurité, doivent utiliser des arêtes différentes dans  $m$  tournées consécutives [17]. Il peut aussi être appliqué à la conception de réseaux de véhicules autoguidés [2,16] et à la détermination de chemins mutuellement disjoints pour le transport de matières dangereuses [9,13].

Les premières recherches sur le  $m$ -PVP ont été réalisées par De Kort. Ce chercheur a démontré que le 2-PVP est NP-difficile en transformant une instance du problème du chemin hamiltonien en un 2-PVP [6]. Par un raisonnement similaire on prouve que le  $m$ -PVP est aussi NP-difficile. De Kort [4] a proposé une borne inférieure sur la valeur  $z^*$  de la solution optimale du  $m$ -PVP en résolvant un problème de transport avec capacités. Des tests réalisés sur des instances avec  $m = 2$  et  $42 \leq n \leq 120$  ont donné lieu à des déviations moyennes de 8.9% par rapport à l'optimum. De Kort [5] a également mis au point un algorithme de branchement et bornes ("branch-and-bound") permettant la résolution d'instances avec  $m = 2$ , comportant jusqu'à 130 sommets dans le cas de coûts aléatoires et jusqu'à 60 sommets dans le cas de coûts euclidiens. Plus récemment, Duchenne, Laporte et Semet [7,8] ont développé des résultats polyédraux ainsi que des algorithmes de séparation et coupes ("branch-and-cut") pour le  $m$ -PVP. Ces algorithmes ont permis la résolution exacte d'instances euclidiennes de TSPLIB [15] comportant jusqu'à 280 sommets pour  $m = 2$  et 26 sommets pour  $m = 12$ .

Comme les temps de calcul des méthodes exactes sont souvent prohibitifs, il est opportun de développer des heuristiques rapides et d'en évaluer le comportement à l'aide de bornes inférieures. Quintero Araujo, Wolfer Calvo et Ould Louly [14] ont développé une telle heuristique mais la difficulté de trouver de bonnes solutions est telle que l'heuristique proposée par Krarup [11] lors de l'introduction du problème semble être plus efficace. Nous présentons une procédure de calcul de borne inférieure à la Section 2 et diverses heuristiques utilisant l'heuristique de Krarup à la Section 3. Les résultats numériques apparaissent à la Section 4 et la conclusion à la Section 5.

## 2. BORNE INFÉRIEURE

La formulation la plus naturelle du  $m$ -PVP a été proposée par De Kort [5] et est inspirée de la formulation classique du Problème de Voyageur de Commerce

(PVC) [3]. Elle est définie de la manière suivante. Soit  $x_{ek}$  une variable binaire ;  $x_{ek} = 1$  si l'arête  $e \in E$  appartient au cycle  $k$  ; 0 sinon. On définit aussi  $\delta(i) = \{e = (i, j) \in E : j \in V\}$  et  $E(S) = \{e = (i, j) \in E : i \in S, j \notin S\}$ . Le  $m$ -PVP se modélise ainsi :

$$\min \sum_{k=1}^m \sum_{e \in E} c_e x_{ek} \quad (1)$$

sous les contraintes :

$$\sum_{e \in \delta(i)} x_{ek} = 2 \quad (i \in V, k = 1, \dots, m) \quad (2)$$

$$\sum_{e \in E(S)} x_{ek} \leq |S| - 1 \quad (S \subset V, 3 \leq |S| \leq \lfloor n/2 \rfloor, k = 1, \dots, m) \quad (3)$$

$$\sum_{k=1}^m x_{ek} \leq 1 \quad (e \in E) \quad (4)$$

$$x_{ek} \in \{0, 1\} \quad (e \in E, k = 1, \dots, m). \quad (5)$$

Comme pour le PVC, ces contraintes s'expliquent aisément. L'objectif (1) est la minimisation de la longueur des  $m$  cycles. Les contraintes (2) sont les contraintes de degré qui assurent que pour chaque cycle, deux arêtes sont incidentes à chaque sommet. Les contraintes (3) imposent qu'aucun des  $m$  cycles ne contient de sous-tours. Les contraintes (5) assurent que la solution est entière. Les contraintes (4) permettent d'obtenir des cycles à arêtes disjointes en interdisant la présence multiple de la même arête dans la solution.

À partir de la formulation précédente, il est possible d'obtenir une relaxation du  $m$ -PVP par ajout de variables et de contraintes. À cette fin, nous introduisons la variable binaire  $x_e = \sum_{k=1}^m x_{ek}$  où  $x_e = 1$  si l'arête  $e \in E$  appartient à l'un des cycles de la solution, 0 sinon. La formulation à deux indices s'écrit :

$$\min \sum_{e \in E} c_e x_e \quad (6)$$

sous les contraintes :

$$\sum_{e \in \delta(i)} x_e = 2m \quad (i \in V) \quad (7)$$

$$x_e \in \{0, 1\} \quad (e \in E) \quad (8)$$

$$\sum_{e \in E(S)} x_e \leq m(|S| - 1) \quad (S \subset V, 3 \leq |S| \leq n - 3). \quad (9)$$

Le graphe support de la solution de ce modèle est un graphe  $2m$ -régulier, qui étant donné que cette formulation n'est qu'une relaxation du  $m$ -PVP, n'est pas

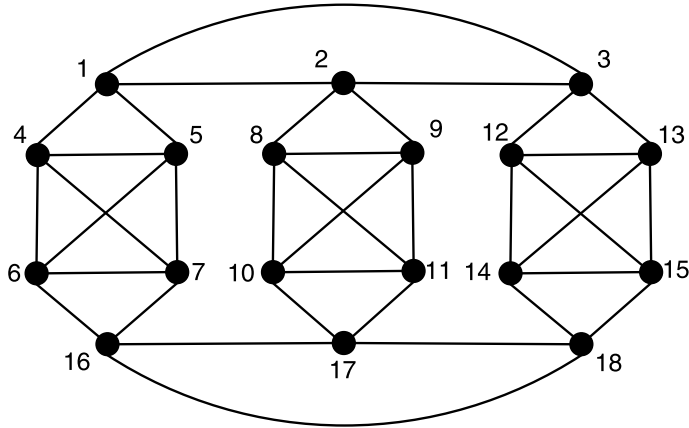


FIGURE 1. Graphe non hamiltonien admissible pour le modèle à 2 indices avec  $m = 2$ .

forcément  $m$ -PVP admissible, autrement dit, il n'admet pas forcément une solution admissible du  $m$ -PVP. C'est par exemple le cas du graphe de la figure 1 où tous les coûts sur les arêtes sont égaux à 1. Ce graphe décrit une solution admissible du modèle à deux indices, mais il n'est pas hamiltonien (Bauer *et al.* [1]) et il n'est donc pas 2-PVP admissible.

Ce modèle à 2 indices, bien que NP-difficile, se résout en un temps raisonnable pour les instances considérées à l'aide d'une méthode de séparation et coupes et produit une borne inférieure  $\underline{z}$  sur  $z^*$  [7]. La qualité de cette borne peut être améliorée par l'ajout de contraintes de 2-couplage et de peignes [8]. Dans la suite de l'article, nous considérons implicitement que ces contraintes font partie du modèle, lequel sera résolu à l'aide d'une méthode de séparation et coupes efficace comme dans [8]. Dans la grande majorité des cas, nous avons  $\underline{z} = z^*$ .

### 3. HEURISTIQUES

Nous présentons dans cette section quatre heuristiques pour la détermination d'une borne supérieure  $\bar{z}$  sur le coût optimal du  $m$ -PVP. La première a été proposée par Krarup [11]. Les trois autres sont nouvelles et améliorent la première heuristique.

#### 3.1. HEURISTIQUE DE KRARUP

L'heuristique de Krarup [11] est une heuristique constructive gloutonne qui peut être décrite de la façon suivante :

**Étape 1** (Initialisation). Poser  $\bar{z} = 0$  et  $S = \emptyset$ .

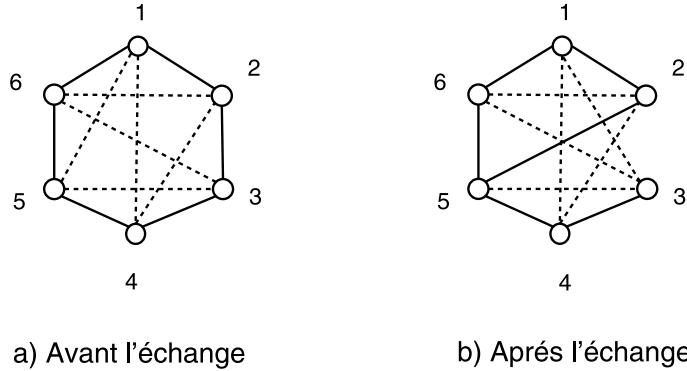


FIGURE 2. Exemple d'échange 2-opt appliqué à deux cycles : le retrait des arêtes  $(2,3)$ ,  $(1,5)$  et leur remplacement par  $(1,3)$ ,  $(2,5)$  donne lieu à une solution inadmissible.

**Étape 2** (Calcul des cycles). Résoudre le PVC sur  $G' = (V, E \setminus S)$ . Soit  $z$  le coût de cette solution et  $S_{PVC}$  l'ensemble des arêtes de cette solution. Mettre à jour  $\bar{z} = \bar{z} + z$  et  $S = S \cup S_{PVC}$ .

**Étape 3** (Test d'arrêt). Si  $m$  cycles ont été calculés, terminer ; sinon, retourner à l'étape 2.

### 3.2. HEURISTIQUE DE PERTURBATION 2-OPT

Cette heuristique applique une perturbation de type 2-opt [12] à la solution produite par l'heuristique de Krarup. Cette perturbation enlève deux arêtes d'un cycle et reconnecte les deux chaînes ainsi créées en utilisant deux autres arêtes. Ce type de mouvement ne peut être appliqué qu'au sein d'un même cycle. En effet, essayer d'appliquer une perturbation 2-opt sur deux cycles distincts entraîne l'obtention d'une solution non admissible, soit parce qu'on obtient un seul cycle qui utilise deux fois chaque sommet, soit parce que les contraintes de degrés ne sont plus respectées (voir Fig. 2). Si les PVC sont résolus de façon optimale dans l'heuristique de Krarup, appliquer ce mouvement ne peut que dégrader la solution finale. L'idée est donc d'utiliser cette perturbation sur le premier cycle de l'heuristique de Krarup afin d'obtenir un premier cycle différent, puis de poursuivre l'heuristique telle que décrite précédemment afin d'obtenir une nouvelle solution. Cette opération est répétée pour tous les choix de deux arêtes possibles dans le premier cycle obtenu par l'heuristique de Krarup ou par la meilleure perturbation antérieure.

**Étape 1** (Initialisation). Calculer une solution de départ de coût  $\bar{z}$  grâce à l'heuristique de Krarup. Sauvegarder cette solution. Soit  $\{e_1, \dots, e_n\}$  l'ensemble des arêtes du premier cycle  $C$ . Poser  $i = 1$  et  $j = 3$ .

- Étape 2** (Mouvement 2-opt). Supprimer les arêtes  $e_i$  et  $e_j$  du cycle  $C$  et les remplacer par deux autres arêtes afin d'obtenir un cycle hamiltonien  $C'$  de coût  $z'$ . Poser  $\bar{z}' = z'$  et  $S = C'$ .
- Étape 3** (Procédure *complément*). Résoudre le PVC sur  $G' = (V, E \setminus S)$ . Soit  $z$  le coût de cette solution et  $S_{PVC}$  l'ensemble des arêtes de cette solution. Mettre à jour  $\bar{z}' = \bar{z}' + z$  et  $S = S \cup S_{PVC}$ . Itérer cette étape jusqu'à l'obtention de  $m$  cycles.
- Étape 4** (Mise à jour et test d'arrêt). Si  $\bar{z}' < \bar{z}$ , sauvegarder cette solution. Poser  $C = C'$ ,  $\bar{z} = \bar{z}'$ ,  $i = 1$  et  $j = 3$ . Sinon, faire  $j = j + 1$ . Si  $j = n + 1$ , faire  $i = i + 1$  et  $j = i + 2$ . Si  $i = n - 1$ , terminer ; sinon, retourner à l'étape 2.

En pratique cet algorithme est très lent. En effet, dans le meilleur des cas, c'est-à-dire si la solution de départ n'est pas améliorée, il donne lieu à  $(m - 1)(n - 3)n/2$  résolutions de PVC. Cependant il est possible d'accélérer cette heuristique grâce à une procédure de restriction de graphe appliquée immédiatement avant l'exécution de l'étape 2. Un graphe restreint peut être obtenu grâce à la formulation à deux indices et une borne supérieure  $\bar{z}$  fournie par l'heuristique de Krarup, par exemple. Afin de le décrire, introduisons les notations suivantes :

- $E_1$  : l'ensemble des arêtes qui ne peuvent pas appartenir à une solution de coût inférieur à  $\bar{z}$  ;
- $E_2$  : l'ensemble des arêtes qui ne sont pas dans  $E_1$  ;
- $\tilde{G}$  : le graphe défini par  $\tilde{G} = (V, E_2)$ . Ce graphe restreint sera utilisé dans certains des algorithmes décrits ci-dessous.

Ces ensembles peuvent être obtenus par la procédure *restriction* suivante :

- Étape 1** (Initialisation). Résoudre le modèle à deux indices. Soit  $x$  la solution de ce modèle. Poser  $E_1 = E_2 = \emptyset$ .
- Étape 2** (Mise à jour de  $E_1$ ). Pour chaque arête  $e \in E$ , si  $x_e = 0$ , résoudre à nouveau le modèle à deux indices en forçant l'utilisation de cette arête. Soit  $z$  le coût de la solution obtenue. Si  $z \geq \bar{z}$ , insérer  $e$  dans  $E_1$ , sinon, l'insérer dans  $E_2$ .
- Étape 3** (Mise à jour de  $E_2$ ). Pour chaque arête  $e \in E$ , si  $x_e = 1$ , insérer  $e$  dans  $E_2$ .

Il suffit ensuite d'appliquer l'heuristique de perturbation 2-opt sur le graphe restreint  $\tilde{G}$ . Cette restriction a pour effet de réduire considérablement le nombre de mouvements de type 2-opt et donc de résolutions de PVC et de rendre cette heuristique plus efficace.

Une autre amélioration réside en l'utilisation d'une borne inférieure sur le coût des cycles restant à calculer lors de la procédure *complément* à l'étape 3. En effet, grâce à la formulation à deux indices, on peut obtenir une telle borne. Si le coût des cycles déjà calculés plus la borne inférieure sur le coût des cycles restant à calculer est supérieur au coût  $\bar{z}$  de la meilleure solution connue, alors il est inutile de poursuivre la procédure *complément* qui ne permettra pas d'obtenir une meilleure solution. Tout comme la première amélioration, celle-ci permet de réduire significativement le nombre de résolutions de PVC.

Quatre autres versions de l'heuristique de perturbation 2-opt ont également été développées afin d'en diminuer le temps de calcul. Ces différentes heuristiques explorent un moins grand nombre de solutions et sont donc plus rapides mais aussi moins performantes. L'étape la plus coûteuse de l'heuristique de perturbation 2-opt étant l'étape 3, qui nécessite un très grand nombre de résolution de PVC, les modifications interviennent à ce niveau. Trois des nouvelles heuristiques réduisent le nombre de résolutions de PVC en n'appliquant pas systématiquement l'étape 3. Les deux dernières heuristiques, exploitent le fait que la borne inférieure obtenue par le modèle à 2 indices est très proche de l'optimum. Plutôt que de chercher une solution meilleure que la précédente, on peut donc tenter de trouver une solution plus proche de la valeur de cette borne inférieure. Les heuristiques peuvent être décrites de la façon suivante :

**2-opt 50%:** L'étape 3 de l'heuristique de perturbation 2-opt est appliquée avec une probabilité de 0.5.

**2-opt mn:** Pour un cycle donné, l'étape 3 de l'heuristique de perturbation 2-opt n'est appliquée qu'avec une probabilité égale à  $\min\{1, 50/mn\}$ .

**2-opt-borne inf:** Soit  $\underline{z}$  la valeur de la borne inférieure fournie par le modèle à 2 indices. Si la somme du coût des cycles déjà calculés et de la borne inférieure sur le coût des cycles restant à calculer est supérieure à  $\underline{z} + (\bar{z} - \underline{z})/2$ , la procédure *complément* n'est pas poursuivie.

**2-opt-borne inf mn:** Cette version combine les deux versions précédentes.

### 3.3. HEURISTIQUE DE PERTURBATION INVERSÉE

Comme expliqué dans la section précédente, il est très difficile d'améliorer une solution complète composée des  $m$  cycles. Par contre, il est possible de compléter une solution partielle en s'inspirant de l'heuristique de Krarup. Comme pour l'heuristique précédente, l'heuristique de perturbation inversée modifie le cycle de départ de l'heuristique de Krarup et donc la solution finale obtenue.

À l'étape 1 de cette heuristique, on considère  $S_1 = \{h_1^1, \dots, h_1^m\}$ , l'ensemble des cycles fournis par l'heuristique de Krarup. L'heuristique proposée exploite le fait que le dernier des cycles de cette solution, le cycle  $h_1^m$ , peut contenir des arêtes qui n'appartiennent pas à la solution optimale. Afin d'éviter que ces arêtes n'apparaissent à nouveau dans une solution, nous devons donc trouver une autre façon de générer ce cycle. Nous avons donc décidé de choisir les  $r$  arêtes les plus courtes de ce cycle,  $r$  étant un paramètre, et de fixer le coût de ces arêtes à 0 avant de résoudre un PVC qui donnera lieu à un nouveau cycle  $h_2^1$ . De cette façon le cycle  $h_2^1$  contiendra certainement les meilleures arêtes du cycle  $h_1^m$  ainsi que des arêtes des premiers cycles de la solution  $S_1$ . Il suffit ensuite d'appliquer la procédure *complément* décrite à l'étape 3 du premier algorithme de la Section 3.2 sur le cycle  $h_2^1$  avec la matrice des coûts non modifiée pour obtenir une nouvelle solution  $S_2 = \{h_2^1, \dots, h_2^m\}$ .

Cette première idée permet d'obtenir une nouvelle solution mais il est possible de la développer davantage. En effet, on peut conserver le premier cycle  $h_2^1$  ainsi obtenu et ensuite de répéter l'opération pour obtenir les cycles subséquents. On obtiendra ainsi d'autres solutions parmi lesquelles il suffira de conserver la meilleure.

- Étape 1** (Initialisation). Appliquer l'heuristique de Krarup afin d'obtenir la solution  $S_1 = \{h_1^1, \dots, h_1^m\}$  de coût  $\bar{z}$ . Poser  $i = 1$ .
- Étape 2** (Modification des coûts). Fixer le coût des  $r$  plus petites arêtes du cycle  $h_i^m$  à 0 et calculer le cycle  $h_{i+1}^i$  dans  $\bar{G} = (V, \bar{E})$  avec  $\bar{E} = E \setminus \{h_1^2, \dots, h_i^{i-1}\}$  si  $i \geq 2$  et  $\bar{E} = E$  si  $i = 1$ .
- Étape 3** (Calcul des cycles suivants). Utiliser la procédure *complément* sur les cycles  $h_2^1, \dots, h_{i+1}^i$  afin d'obtenir la solution  $S_{i+1} = \{h_2^1, \dots, h_{i+1}^i, h_{i+1}^{i+1}, \dots, h_{i+1}^m\}$  de coût  $z$ .
- Étape 4** (Test d'arrêt). Si  $z < \bar{z}$ , poser  $\bar{z} = z$  et sauvegarder la solution  $S_{i+1}$ . Incrémenter  $i$ . Si  $i < m$ , retourner à l'étape 2; sinon, terminer l'algorithme.

À la suite de nombreux tests, le paramètre  $r$  a été fixé à  $n/5$ , ce qui semble donner les meilleurs résultats.

### 3.4. HEURISTIQUE DE KRARUP RESTREINTE

Les heuristiques proposées précédemment sont basées sur l'heuristique de Krarup. Cependant, nous avons remarqué grâce à la procédure *restriction* de la Section 3.2 qu'il arrive que cette heuristique utilise des arêtes qui ne peuvent pas appartenir à une solution optimale. Nous avons donc décidé de répéter l'heuristique de Krarup sur le graphe réduit  $\tilde{G}$  après l'avoir appliquée une première fois afin d'obtenir la borne supérieure nécessaire à cette restriction.

L'idée utilisée ici peut encore être développée. En effet, comme il est probable que les arêtes aux coûts les plus élevés n'appartiennent pas à de bonnes solutions, il peut être intéressant de restreindre davantage le graphe  $\tilde{G}$ . Il faut toutefois faire attention de conserver un nombre suffisant d'arêtes pour pouvoir obtenir au moins une solution. Pour diminuer le nombre d'arêtes nous avons décidé de réduire le degré des sommets en prenant garde de ne pas descendre en-dessous de  $2m$ . À chaque itération nous décrétons donc le degré du sommet le plus élevé en éliminant l'arête de coût maximum, tout en faisant attention à ce que cela ne fasse pas diminuer le degré de l'autre sommet adjacent en-dessous de  $2m$ . Bien entendu, il est inutile de recalculer une nouvelle solution si l'arête supprimée n'appartient pas à la solution de l'heuristique de Krarup.

- Étape 1** (Initialisation). Appliquer l'heuristique de Krarup afin d'obtenir une solution de coût  $\bar{z}$  et utiliser le modèle à deux indices pour obtenir une borne inférieure  $\underline{z}$ .
- Étape 2** (Restriction). Restreindre le graphe à l'aide de la borne supérieure de coût  $\bar{z}$ . Soit  $\tilde{G} = (V, E_2)$  le graphe résultant.
- Étape 3** (Suppression d'une arête). Sélectionner un sommet de degré maximum dans  $\tilde{G}$ , supprimer une arête de  $E_2$  de coût maximum adjacente à



ce sommet sauf si cela fait diminuer le degré du sommet adjacent en dessous de  $2m$ .

**Étape 4** (Calcul de solution). Si l'arête supprimée n'appartient pas à la solution de l'heuristique de Krarup, retourner à l'étape 3. Sinon, appliquer l'heuristique de Krarup au graphe restreint  $\tilde{G}$ .

**Étape 5** (Test d'arrêt). Si aucune solution n'existe dans le graphe, terminer. Sinon, soit  $z$  le coût de la solution. Si  $z = \underline{z}$ , la solution optimale a été obtenue, terminer. Si  $z < \bar{z}$ , poser  $\bar{z} = z$  et retourner à l'étape 2. Sinon, retourner à l'étape 3.

#### 4. RÉSULTATS NUMÉRIQUES

Les algorithmes que nous venons de décrire ont été codés en C++ et ont été exécutés sur un Pentium IV Extreme Edition 3.4 Ghz. Tous les modèles linéaires, utilisés pour obtenir la borne inférieure, ont été résolus à l'aide de CPLEX 9.0 et les solutions du PVC ont été obtenues grâce à l'algorithme LKH [10]. Cet algorithme est une heuristique efficace pour la résolution du PVC symétrique qui fournit, pour les instances que nous considérons, une solution optimale ou très proche de celle-ci. Une limite de 3600 s a été utilisée lors de ces tests. Les instances pour lesquelles la solution finale de l'heuristique n'a pas été obtenue avant cette limite ne sont pas comptabilisées dans les moyennes. Pour ces instances, une solution est malgré tout connue et est, dans le pire des cas, la solution de l'heuristique de Krarup.

Les tests ont été exécutés sur des instances euclidiennes aléatoires et sur des instances de TSPLIB [15]. Les instances euclidiennes aléatoires ont été obtenues en générant  $n$  points à coordonnées entières de façon aléatoire selon une loi uniforme discrète dans un carré de 100 par 100 et en calculant la distance euclidienne entre chaque paire de points. Dix instances de chaque taille ont été générées et pour chaque instance les tests ont été effectués pour  $m = 2, \dots, 9$ .

Les résultats obtenus avec ces toutes les heuristiques sont résumés dans le Tableau 1 pour les instances euclidiennes aléatoires et dans le Tableau 2 pour les instances de TSPLIB. Un tableau supplémentaire 3 résume les résultats obtenus pour les deux heuristiques les plus rapides, l'heuristique de perturbation inversée et l'heuristique de perturbation 2-opt-borne inf mn, sur des instances de TSPLIB d'une taille plus importante. Les Tableaux 1, 2 et 3 indiquent les ratios entre les valeurs des heuristiques de Krarup, de 2-optimalité, de perturbation inversée ou de Krarup restreinte et la valeur de la borne inférieure. Les temps de calcul nécessaires pour chaque heuristique sont également présentés, à l'exception de celui de l'heuristique de Krarup qui, étant donné la taille relativement petite des graphes et l'efficacité de LKH, s'exécute en moins d'une seconde.

Les entêtes de ces tableaux sont les suivants :

$n$  : nombre de sommets du graphe,  
 $m$  : nombre de cycles à arêtes disjointes dans la solution,  
 $\bar{z}/\underline{z}$  : ratio de la borne supérieure sur la borne inférieure,

Inversée : heuristique de perturbation inversée,  
 Restreinte : heuristique de Krarup restreinte,  
 2-opt : heuristique perturbation 2-opt,  
 2-opt 50% : heuristique perturbation 2-opt 50 %,  
 2-opt mn : heuristique perturbation 2-opt mn,  
 2-opt-borne inf : heuristique perturbation 2-opt-borne inf,  
 2-opt-borne inf mn : heuristique perturbation 2-opt-borne inf mn,  
 Secondes : temps CPU en seconde utilisé pour cette instance.

À l'analyse des résultats obtenus pour les instances euclidiennes aléatoires, nous constatons que ces heuristiques sont très précises. Cela est principalement dû à la précision de l'heuristique de Krarup qui fournit une solution éloignée en moyenne de 0.45% de la valeur de la borne inférieure. Les autres heuristiques étant des heuristiques d'amélioration s'appliquant sur la solution de l'heuristique de Krarup, leur ratio est forcément plus petit que celui de l'heuristique de Krarup. L'heuristique de perturbation 2-opt est celle qui fournit les meilleures solutions avec un ratio  $\bar{z}/z$  égal à 1.0022, correspondant à une diminution d'écart de plus de 50% par rapport à l'heuristique de Krarup. L'heuristique de perturbation inversée réduit l'écart d'optimalité de l'heuristique de Krarup d'environ 25% moyennant des temps de calcul très modestes. On peut également observer que les heuristiques de perturbation inversée et de perturbation 2-opt-borne inf mn permettent d'obtenir de bonnes solutions pour de grandes instances en un temps raisonnable.

On peut remarquer que l'heuristique la plus précise est l'heuristique 2-opt et la moins précise est celle de perturbation inversée. Malheureusement, malgré cette constatation, aucune des heuristiques ne domine les autres, mais les heuristiques de Krarup restreinte et 2-opt-borne inf sont dominées. En effet, l'heuristique la plus rapide est l'heuristique de perturbation inversée et la plus lente est l'heuristique 2-opt. Le principal problème de cette dernière est que le temps de calcul augmente fortement lorsque  $m$  augmente. Cela est dû au fait que pour  $m$  grand, on doit résoudre de nombreux PVC.

Les résultats obtenus pour les instances de TSPLIB sont sensiblement équivalents même si quelques différences peuvent être observées. Ces différences peuvent être expliquées en partie par le nombre d'instances sur lesquelles la moyenne est calculée. Pour les instances euclidiennes aléatoires, la moyenne est effectuée sur 310 instances alors que pour les instances de TSPLIB, celle-ci n'est effectuée que sur 28 instances. Les instances fournissant des résultats éloignés de la moyenne ont donc plus d'influence sur celle-ci que pour les instances euclidiennes aléatoires.

## 5. CONCLUSION

Nous avons développé plusieurs heuristiques pour le Problème du Vendeur  $m$ -Péripatétique. Chacune des heuristiques améliore la qualité de la solution fournie par l'heuristique classique de Krarup. Dans le meilleur des cas, la réduction de



TABLEAU 2. Résultats numériques pour les instances de TSPLIB.

Instance	n	m	Krarup		Restrainte		Inversée		2-opt		2-opt 50%		2-opt mn		2-opt borne inf		2-opt borne inf mn	
			$\bar{z}/\underline{z}$	Seconds	$\bar{z}/\underline{z}$	Seconds	$\bar{z}/\underline{z}$	Seconds	$\bar{z}/\underline{z}$	Seconds	$\bar{z}/\underline{z}$	Seconds	$\bar{z}/\underline{z}$	Seconds	$\bar{z}/\underline{z}$	Seconds	$\bar{z}/\underline{z}$	Seconds
gr21	21	2	1.0025	1.0025	1.0025	0	1.0025	0	1.0025	0	1.0025	0	1.0025	0	1.0025	0	1.0025	0
gr21	21	3	1.0006	1.0006	1.0006	1	1.0006	0	1.0001	0	1.0001	0	1.0001	1	1.0002	0	1.0002	0
gr21	21	4	1.001	1.001	1.001	1	1.001	0	1	1	1	2	1	1	2	1	1	2
gr21	21	5	1.0075	1.0075	1.0075	1	1.0075	0	1.0041	7	1.0041	7	1.0041	7	1.0075	7	1.0075	6
swiss42	42	2	1.0058	1.0058	1.0058	19	1.0058	0	1.0026	0	1.0026	0	1.0026	0	1.0026	0	1.0026	0
swiss42	42	3	1.0049	1.0049	1.0049	18	1.0049	0	1.0018	9	1.0018	6	1.0018	3	1.0018	7	1.002	3
swiss42	42	4	1.0029	1.0029	1.0029	32	1.0027	0	1.0002	18	1.0002	13	1.0012	4	1.0002	6	1.0014	3
swiss42	42	5	1.0006	1.0006	1.0006	27	1.0006	1	1	1	1	3	1	1	1	3	3	3
gr48	48	2	1.0125	1.0125	1.0125	36	1.0125	0	1.0109	5	1.0116	4	1.0116	4	1.0109	3	1.0116	2
gr48	48	3	1.0043	1.0043	1.0043	44	1.0043	3	1.0027	17	1.0027	14	1.0035	6	1.0032	12	1.0035	6
gr48	48	4	1.0054	1.0054	1.0054	49	1.0054	1	1.0036	43	1.0036	34	1.0049	9	1.0054	24	1.0054	6
gr48	48	5	1.0028	1.0028	1.0028	78	1.0027	2	1.0006	50	1.0006	43	1.0012	10	1.0006	40	1.0012	7
gr96	96	2	1.0065	1.0065	1.0065	833	1.0065	16	1.0061	88	1.0061	69	1.0064	36	1.0061	52	1.0064	31
gr96	96	3	1.008	1.008	1.008	779	1.008	11	1.0062	380	1.0062	305	1.0065	73	1.008	210	1.008	39
gr96	96	4	1.0056	1.0056	1.0056	796	1.0049	6	1.0035	763	1.0035	628	1.0038	112	1.0056	586	1.0056	72
gr96	96	5	1.0036	1.0036	1.0036	1032	1.0023	8	1.0018	960	1.0018	786	1.0027	173	1.0023	632	1.002	68
gr70	70	2	1.0118	1.0118	1.0118	241	1.0118	1	1.0087	18	1.0087	13	1.0087	8	1.0118	4	1.0118	3
gr70	70	3	1.006	1.006	1.006	188	1.006	2	1.0025	98	1.0025	77	1.0028	25	1.0032	71	1.0028	15
gr70	70	4	1.0058	1.0058	1.0058	225	1.0059	2	1.0028	194	1.0028	150	1.0028	33	1.0028	101	1.0028	16
gr70	70	5	1.0031	1.0031	1.0031	281	1.0031	5	1.0013	279	1.0013	215	1.0015	36	1.0015	225	1.0015	31
hn105	105	2	1.0042	1.0042	1.0042	1194	1.0042	3	1.0033	26	1.0033	23	1.0035	20	1.0042	23	1.0042	19
hn105	105	3	1.0069	1.0069	1.0069	974	1.0069	2	1.0031	518	1.0049	466	1.0064	76	1.0039	273	1.0069	50
hn105	105	4	1.0079	1.0079	1.0079	1263	1.0067	7	1.005	1499	1.005	1312	1.0052	198	1.0067	961	1.0067	126
hn105	105	5	1.0079	1.0079	1.0079	1654	1.004	7	1.0048	3080	1.0048	2913	1.0057	221	1.0079	1882	1.0079	127
gr120	120	2	1.0064	1.0064	1.0064	1804	1.0064	6	1.0049	230	1.0049	160	1.0063	62	1.0049	86	1.0049	38
gr120	120	3	1.0023	1.0023	1.0023	1955	1.0023	121	1.0016	749	1.0016	537	1.0016	216	1.0016	482	1.0023	179
gr120	120	4	1.0022	1.0022	1.0022	2297	1.0022	8	1.0015	1393	1.0015	1065	1.0019	216	1.002	1035	1.0022	149
gr120	120	5	1.0029	1.0029	1.0029	2696	1.0029	27	1.0011	2592	1.0011	2178	1.0013	259	1.0011	1693	1.0013	159
Moyenne			1.0050	1.0045	661	1.0047	9	1.0031	465	1.0032	393	1.0036	65	1.0039	300	1.0041	41	

TABLEAU 3. Résultats numériques pour les grandes instances de TSPLIB.

Instance			Krarup		Inversée		2-opt borne inf mn	
Code	$n$	$m$	$\bar{z}/z$	$m$	$\bar{z}/z$	Secondes	$\bar{z}/z$	Secondes
kroA100	100	2	1,0117	2	1,0117	6	1,0117	38
kroA100	100	3	1,0112	3	1,0102	10	1,0112	89
kroA100	100	4	1,0045	4	1,0045	9	1,0045	93
kroA100	100	5	1,0019	5	1,0019	9	1,0019	85
kroB100	100	2	1,0149	2	1,0126	14	1,0149	29
kroB100	100	3	1,005	3	1,005	4	1,0042	37
kroB100	100	4	1,0023	4	1,0022	8	1,001	42
kroB100	100	5	1,0018	5	1,0018	7	1,001	86
kroC100	100	2	1,0116	2	1,0116	4	1,011	19
kroC100	100	3	1,0114	3	1,0114	9	1,0114	91
kroC100	100	4	1,0048	4	1,0048	13	1,0048	125
kroC100	100	5	1,0027	5	1,0027	9	1,0027	144
kroD100	100	2	1,0071	2	1,0071	3	1,0047	13
kroD100	100	3	1,0043	3	1,0043	4	1,0043	21
kroD100	100	4	1,007	4	1,0053	4	1,004	60
kroD100	100	5	1,0092	5	1,0055	8	1,0092	177
kroE100	100	2	1,0116	2	1,0116	7	1,0116	10
kroE100	100	3	1,0055	3	1,0055	33	1,0048	86
kroE100	100	4	1,004	4	1,004	12	1,004	81
kroE100	100	5	1,0023	5	1,0023	11	1,0023	126
eil101	101	2	1,0048	2	1,0048	5	1,0048	23
eil101	101	3	1,0041	3	1,0036	8	1,0024	52
eil101	101	4	1,0019	4	1,0016	8	1,0011	68
eil101	101	5	1,001	5	1,001	10	1,0008	128
gr137	137	2	1,0073	2	1,0073	32	1,0073	66
gr137	137	3	1,0062	3	1,0062	14	1,0043	122
gr137	137	4	1,0029	4	1,0029	50	1,0021	135
gr137	137	5	1,0041	5	1,003	21	1,0041	267
kroA150	150	2	1,0093	2	1,0093	28	1,0093	106
kroA150	150	3	1,006	3	1,0051	16	1,006	195
kroA150	150	4	1,0051	4	1,0049	30	1,0051	320
kroA150	150	5	1,0047	5	1,0047	22	1,0047	482
kroB150	150	2	1,0098	2	1,0098	18	1,0098	139
kroB150	150	3	1,0101	3	1,0101	22	1,007	204
kroB150	150	4	1,0056	4	1,0056	46	1,0056	353
kroB150	150	5	1,0029	5	1,0023	30	1,0029	444
rat195	195	2	1,005	2	1,005	52	1,005	144
rat195	195	3	1,0019	3	1,0019	35	1,0015	363
rat195	195	4	1,0018	4	1,0018	135	1,0011	748
rat195	195	5	1,0007	5	1,0007	568	1,0007	1323
kroA200	200	2	1,0105	2	1,0105	26	1,0105	421
kroA200	200	3	1,0058	3	1,0058	48	1,0055	658
kroA200	200	4	1,0033	4	1,0033	129	1,0033	1190
kroA200	200	5	1,0021	5	1,0021	135	1,0021	1238
kroB200	200	2	1,0093	2	1,0093	322	1,0093	701
kroB200	200	3	1,0081	3	1,0081	52	1,0081	784
kroB200	200	4	1,0034	4	1,0034	96	1,0034	708
kroB200	200	5	1,0019	5	1,0019	55	1,0019	1213
pr264	264	2	1,0041	2	1,0041	237	1,0036	3548
pr264	264	3	1,0054	3	1,0054	671	1,0054	4050
pr264	264	4	1,005	4	1,005	685	1,005	756
pr264	264	5	1,0033	5	1,0033	787	1,0033	807
pr299	299	3	1,0084	3	1,0084	941	1,0084	5790
pr299	299	5	1,0011	5	1,0011	512	1,0011	5374
lin318	318	2	1,009	2	1,009	169	1,009	5381
lin318	318	3	1,0081	3	1,0081	116	1,0081	3559
lin318	318	4	1,0041	4	1,0041	143	1,0041	5555
lin318	318	5	1,004	5	1,004	176	1,004	9107
Moyenne			1,0057		1,0054	114	1,0052	999

l'écart d'optimalité atteint 50%. Si on tient compte simultanément de la qualité de la solution et du temps de calcul, aucune des heuristiques proposées ne domine les autres.

## REMERCIEMENTS

Ce travail a été en partie réalisé grâce à une subvention du Conseil de recherche en Sciences naturelles et en Génie du Canada, numéro OGP0039682. Nous remercions également la Région Nord-Pas de Calais et les fonds européens FEDER pour le soutien fourni. Nous remercions deux arbitres anonymes pour leurs commentaires judicieux.

## RÉFÉRENCES

- [1] D. Bauer, H.J. Broersma and H.J. Veldman, On smallest non-Hamiltonian regular tough graphs. *Congressus Numerantium* **70** (1990) 95–98.
- [2] J. Blazewicz, R.E. Burkard, G. Finke and G.J. Woeginger, Vehicle scheduling in two-cycle flexible manufacturing systems. *Math. Comput. Model.* **20** (1994) 19–31.
- [3] G.B. Dantzig, D.R. Fulkerson and S.M. Johnson, Solution of a large-scale traveling-salesman problem. *Oper. Res.* **2** (1954) 393–410.
- [4] J.B.J.M. De Kort, Lower bounds for symmetric  $m$ -peripatetic salesman problems. *Optimization* **22** (1991) 113–122.
- [5] J.B.J.M. De Kort, A branch-and-bound algorithm for symmetric 2-peripatetic salesman problems. *Eur. J. Oper. Res.* **70** 229–243.
- [6] J.B.J.M. De Kort, Sensitivity analysis for symmetric 2-peripatetic salesman problems. *Oper. Res. Lett.* **13** (1993) 79–84.
- [7] É. Duchenne, G. Laporte and F. Semet, Branch-and-cut algorithms for the undirected  $m$ -peripatetic salesman problem. *Eur. J. Oper. Res.* **162** (2005) 700–712.
- [8] É. Duchenne, G. Laporte and F. Semet, The undirected  $m$ -peripatetic salesman problem : polyhedral results and new algorithms. *Oper. Res.* **55** (2007) 949–965.
- [9] R. Gopalan, R. Batta and M.H. Karwan, The equity constrained shortest path problem. *Comput. Oper. Res.* **17** (1990) 297–307.
- [10] K. Helsgaun, An effective implementation of the Lin-Kernighan traveling salesman heuristic. *Eur. J. Oper. Res.* **126** (2000) 106–130.
- [11] J. Krarup, The peripatetic salesman and some related unsolved problems, in *Combinatorial Programmin. Methods and Applications*, edited by Roy B. D. Reidel Publ., Dordrecht (1975) 173–178.
- [12] S. Lin, Computer solutions of the traveling salesman problem. *Bell Syst. Tech. J.* **44** (1965) 2245–2269.
- [13] L. Lindner-Dutton, R. Batta and M.H. Karwan, Equitable sequencing of a Given set of hazardous materials shipments. *Transportation Science* **25** (1991) 124–137.
- [14] C. Quintero Araujo, R. Wolfler Calvo and M.A. Ould Louly, Développement de méthodes heuristiques pour le 2-voyageur de commerce péripatétique. *6e Conférence Francophone de MODélisation et SIMulation - MOSIM 06*, Rabat, Maroc.
- [15] G. Reinelt, TSPLIB – A traveling salesman problem library. *ORSA J. Comput.* **3** (1991) 376–384.
- [16] M.A. Venkataramanan and K.A. Wilson, A branch-and-bound algorithm for flow-path design of automated guided vehicle systems. *Nav. Res. Logist.* **38** (1991) 431–445.
- [17] R. Wolfler Calvo and R. Cordone, A heuristic approach to the overnight security service problem. *Comput. Oper. Res.* **30** (2003) 1269–1287.