

BORNES SUPERIEURES ET INFERIEURES POUR LE PROBLEME DE TOURNEES DE VEHICULES m -PERIPATETIQUES

Sandra Ulrich NGUEVEU, Christian PRINS, Roberto WOLFLER-CALVO

Institut Charles Delaunay - LOSI
Université de Technologie de Troyes (UTT)
12 rue Marie Curie - 10010 Troyes

ngueveus@utt.fr, christian.prins@utt.fr, roberto.wolfler_calvo@utt.fr

RÉSUMÉ : *Etant donné un graphe non orienté complet, le problème de tournées de véhicules m -péripatétiques (m -PTVP) consiste à déterminer des tournées de véhicules à coût total minimal sur m périodes telles que chaque client soit visité exactement une fois par période. Chaque arête du graphe ne peut être utilisée qu'au maximum une fois au cours des m périodes. Chaque tournée débute et se termine au dépôt. La somme des demandes des clients d'une tournée ne doit pas excéder la capacité maximale des véhicules. Le m -PTVP est un problème nouveau qui apparaît par exemple dans le réapprovisionnement de distributeurs de billets, où un chemin ne doit pas être utilisé plus d'une fois au cours de m périodes, par sécurité. Le coût total à minimiser est égal à la somme des coûts des arêtes qui composent les tournées choisies. Dans cet article, nous proposons une recherche taboue initialisée par une heuristique adaptée de celle de Clarke et Wright, ainsi que deux bornes inférieures. Les deux bornes inférieures sont basées sur les principes du b -couplage parfait et des arbres recouvrants sans arêtes communes. L'analyse des résultats est effectuée non seulement sur le m -PTVP en général, mais aussi sur deux cas particuliers connus: le cas mono-période (véhicule routing problem ou VRP) et le problème du vendeur m -péripatétique (m -PVP).*

MOTS-CLÉS : *métaheuristiques, b -couplage parfait, arbres recouvrants, tournées de véhicules péripatétiques*

1. INTRODUCTION

Le m -PTVP consiste à déterminer les tournées de véhicules disjointes à coût total minimal sur m périodes telles que chaque client soit visité exactement une fois par période. Ce nouveau problème peut être considéré comme une généralisation de deux problèmes NP-difficiles connus en recherche opérationnelle: le véhicule routing problem (VRP) et le problème du vendeur m -péripatétique (m -PVP). En effet, le VRP est un cas particulier de m -PTVP où $m = 1$ puisqu'il consiste à trouver les meilleures tournées de véhicules permettant de visiter chaque nœud une et une seule fois. De même, sachant que le problème du voyageur de commerce (PVC) est un cas particulier du VRP où la capacité des véhicules est infinie, nous pouvons en déduire que tout problème m -PVP correspond à un problème m -PTVP avec une capacité de véhicules infinie.

Le m -PVP consiste à trouver m cycles hamiltoniens disjoints à coût total minimal sur un graphe a priori complet. Ce problème a été introduit par (Krarup,

1975) qui a proposé l'heuristique simple suivante pour le 2-PVP: résoudre un premier PVC sur le graphe de départ, retirer du graphe les arcs utilisés, résoudre à nouveau un PVC sur le graphe restant et prendre comme solution du 2-PSP les deux PVC ainsi trouvés. S'inspirant de la forte relation qui existe entre le problème du voyageur de commerce et l'arbre recouvrant à coût minimal, (Krarup, 1975) a également suggéré l'utilisation d'arbres recouvrants disjoints pour le m -PVP. Une autre borne inférieure a été proposée par (De Kort, 1991) en résolvant des problèmes sous-jacents de transport avec capacité maximum. En ce qui concerne les méthodes exactes, les travaux de (De Kort, 1993) ainsi que (Duchenne *et al.*, 2005) ont permis de résoudre optimalement des instances euclidiennes de tailles inférieures ou égales à soixante nœuds.

Le VRP est une généralisation du PVC où un poids (ou demande) est affecté à chaque nœud (ou client) excepté le nœud-dépôt, chaque cycle doit passer par le dépôt et la somme des poids des nœuds d'un cy-

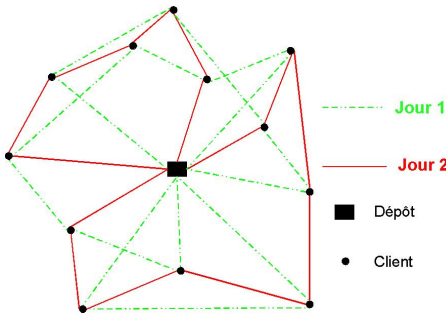


Figure 1: Exemple de solution pour un 2-PTVP

cle ne doit pas dépasser une valeur maximale (capacité maximale des véhicules). Il existe de nombreuses publications sur ce problème parmi lesquelles le livre de (Toth et Vigo, 2002) qui recense les principales avancées sur le sujet. Parmi les méthodes exactes, nous pouvons citer par exemple les procédures de séparation et évaluation basées sur des relaxations simples du VRP telles que le b -couplage ou des relaxations plus élaborées telles que le partitionnement et la relaxation lagrangienne. Plusieurs heuristiques ont été appliquées dont celle de (Clarke et Wright, 1964) qui a l'avantage de réduire à la fois le nombre de tournées et le coût. Son principe consiste à créer une tournée par client, puis à les fusionner de manière à minimiser le coût total, tout en respectant les limites de capacité. Concernant les métaheuristiques, on en distingue principalement deux types: les méthodes à population comme les algorithmes mémétiques (Prins, 2004) et les méthodes qui examinent une solution à la fois comme la recherche taboue (Glover et Laguna, 1993). Un état de l'art des métaheuristiques les plus efficaces pour le VRP a été publié par (Cordeau *et al.*, 2005).

Le m -PTVP modélise des problèmes comme le transport de fonds où par sécurité aucun chemin ne doit être utilisé plus d'une fois au cours de m périodes. Il s'applique également aux tournées de gardiennage (Wolfler Calvo et Cordone, 2003). La figure 1 illustre un exemple de solution pour un 2-PTVP.

Nous proposons une formulation mathématique adaptée ainsi qu'une recherche taboue avec facteur de diversification initialisée par une adaptation de l'heuristique de (Clarke et Wright, 1964). Les bornes inférieures sont basées sur le b -couplage parfait et les m arbres disjoints à coût total minimal. En l'absence de jeux d'instances pour le m -PTVP, nous avons testé nos algorithmes sur le VRP et le m -PVP pour nous comparer avec la littérature.

La suite de cet article est organisée tel que suit: la section 2 présente une formulation du m -PTVP basée sur les arêtes, la borne supérieure est décrite au sein

de la section 3 tandis que la section 4 présente les deux bornes inférieures; enfin, l'analyse numérique est effectuée au sein de la section 5, juste avant la conclusion.

2. FORMULATION DU m -PTVP BASEE SUR LES ARETES

Soit un graphe non orienté complet $G = (V, E)$ où $V = \{0..n\}$ est l'ensemble des noeuds (le noeud 0 correspondant au dépôt et $V' = V \setminus \{0\}$) tandis que E est l'ensemble des arêtes. On associe à chaque noeud $i \in V$ une demande d_i et à chaque arête e un coût c_e . De plus, la capacité maximale des véhicules est Q , i.e. la somme des demandes des clients de chaque tournée ne doit pas excéder la valeur Q .

Soient:

- $\mathbb{K} = \{1..m\}$ l'ensemble des périodes du problème m -PTVP
- x_e^k la variable binaire égale à 1 si l'arête e est utilisée dans une tournée lors de la période $k \in \mathbb{K}$, et égale à 0 sinon
- $\delta(S)$ l'ensemble des arêtes ayant une extrémité dans $S \subseteq V'$ et l'autre extrémité hors de S . Par la suite, on remplacera $\delta(\{i\})$ avec $i \in V$ par $\delta(i)$
- $r(S)$ le nombre minimum de véhicules nécessaire pour servir un sous-ensemble de noeuds $S \subseteq V'$ tel que:

$$r(S) = \left\lceil \sum_{i \in S} \frac{d_i}{Q} \right\rceil$$

- $\lambda \in \mathbb{N}$ le nombre minimum de véhicules par période: $\lambda = r(V')$.

Le m -PTVP peut être formulé de la manière suivante:

$$\min \sum_{k \in \mathbb{K}} \sum_{e \in E} c_e x_e^k$$

sous les contraintes

$$\sum_{e \in \delta(i)} x_e^k = 2, \quad \forall k \in \mathbb{K}, \forall i \in V' \quad (1)$$

$$\sum_{e \in \delta(0)} x_e^k \geq 2\lambda, \quad \forall k \in \mathbb{K} \quad (2)$$

$$\sum_{e \in \delta(S)} x_e^k \geq 2r(S), \quad \forall S \subseteq V', S \neq \emptyset, \forall k \in \mathbb{K} \quad (3)$$

$$\sum_{k \in \mathbb{K}} x_e^k \leq 1, \quad \forall e \in E \quad (4)$$

$$x_e^k \in \{0, 1\}, \quad e \in E, k \in \mathbb{K} \quad (5)$$

La fonction-objectif à minimiser correspond au coût total des arêtes qui sont utilisées au cours des m périodes. Les contraintes (1) sont des contraintes de degré, c'est à dire qu'il doit y avoir exactement deux arêtes incidentes à chaque nœud (excepté le dépôt) au cours de chacune des m périodes puisqu'un véhicule a besoin d'une arête pour atteindre un nœud et une autre arête pour en repartir au cours de chaque période.

Les contraintes (2) stipulent que le nombre d'arêtes incidentes au dépôt doit être au moins égal au double du nombre minimum de tournées (λ) nécessaires au cours de chacune des m périodes, car une tournée a besoin de deux arêtes incidentes au dépôt. Les contraintes (3) s'assurent que quelque soit le sous-ensemble S de nœuds, il existe au moins $2 \times r(S) \in \mathbb{N}$ arêtes sortantes (ayant une extrémité dans S et une extrémité hors de S). Ces dernières contraintes jouent deux rôles: faire respecter la capacité maximale des tournées et empêcher des sous-tours (sous-ensembles de nœuds non connectés au dépôt) de se créer. En effet, chaque sous-ensemble de nœuds doit être desservi par au moins $r(S)$ véhicules, ce qui correspond à $2 \times r(S)$ arêtes incidentes. Enfin, les contraintes (4) permettent de s'assurer qu'aucune arête n'est utilisée plus d'une fois au cours des m périodes.

Une condition nécessaire pour qu'un problème de m -PTVP admette une solution réalisable est que:

$$m \leq \frac{n}{2 \times \lambda}.$$

En effet, il faut un minimum de λ tournées par période pour respecter les contraintes de capacité, ce qui équivaut à $2 \times \lambda$ arêtes connectées au dépôt. Sachant qu'il existe au total n arêtes différentes connectées au dépôt dans le graphe initial, nous pouvons en déduire que le nombre maximum (M) de périodes possibles est $M = \lfloor \frac{n}{2 \times \lambda} \rfloor$.

3. BORNE SUPERIEURE

Nous avons mis au point un algorithme de recherche taboue avec facteur de diversification pour trouver les bornes supérieures du m -PTVP. Il s'agit d'une méthode de recherche locale qui explore le voisinage d'une solution pour y trouver de meilleures solutions. La particularité de cette métaheuristique est qu'elle accepte de passer temporairement par des solutions moins bonnes, si cela peut lui permettre d'atteindre des solutions bien meilleures. De plus, une liste dite "taboue" stocke en permanence les dernières solutions visitées au cours de la recherche et empêche l'algorithme d'y retourner. Les composantes classiques d'une recherche taboue sont donc les suivantes:

la solution initiale pour initialiser la recherche, la définition du voisinage au sein duquel s'effectuera la recherche ainsi que la manière dont ce voisinage sera exploré et enfin la gestion de la liste taboue.

3.1. Solution initiale: Adaptation de l'heuristique de Clarke et Wright

Sachant que l'heuristique de (Clarke et Wright, 1964) de base permet de trouver une solution au 1-PTVP, nous avons choisi de l'appliquer m fois, en retirant du graphe les arêtes déjà utilisées après chaque itération, pour trouver une solution initiale au m -PTVP, à la manière de (Krarup, 1975). Dans la pratique, une importante pénalité est affectée aux arêtes déjà utilisées pour dissuader l'algorithme de les réutiliser, à moins qu'il n'y ait pas d'autre alternative.

3.2. Exploration du voisinage

Le voisinage d'une solution est l'ensemble des solutions qui peuvent être obtenues en appliquant une transformation locale, par exemple la permutation de nœuds ou les mouvements k -opt. Un mouvement k -opt consiste à retirer k arêtes non consécutives qui seront remplacées par k autres arêtes de manière à obtenir une nouvelle solution réalisable.

Notre approche est basée sur deux constats:

- Comme l'ont fait remarquer (Toth et Vigo, 1998), les arêtes de faible coût ont plus de chances que les arêtes chères de se retrouver dans la solution optimale.
- Le coût total de la solution d'un m -PTVP dépend uniquement de l'ensemble des arêtes utilisées, indépendamment de la période au cours de laquelle chaque arête est effectivement utilisée.

Par conséquent, notre approche consiste à insérer d'une manière simple au sein de la solution courante des arêtes à faible coût qui ne sont utilisées dans aucune des m périodes. Les arêtes choisies peuvent être affectées à n'importe laquelle des m périodes. La figure 1 illustre les huit différentes manières, inspirées des mouvements 2-opt classiques, d'introduire une arête [A, B] au sein d'une période. Il y a donc $8 \times m$ mouvements possibles par arête. Les mouvements qui impliquent deux tournées ne sont admis que si les contraintes de capacité sont respectées: la demande totale de chacune des deux nouvelles tournées obtenues doit rester inférieure ou égale à la capacité maximale des véhicules Q .

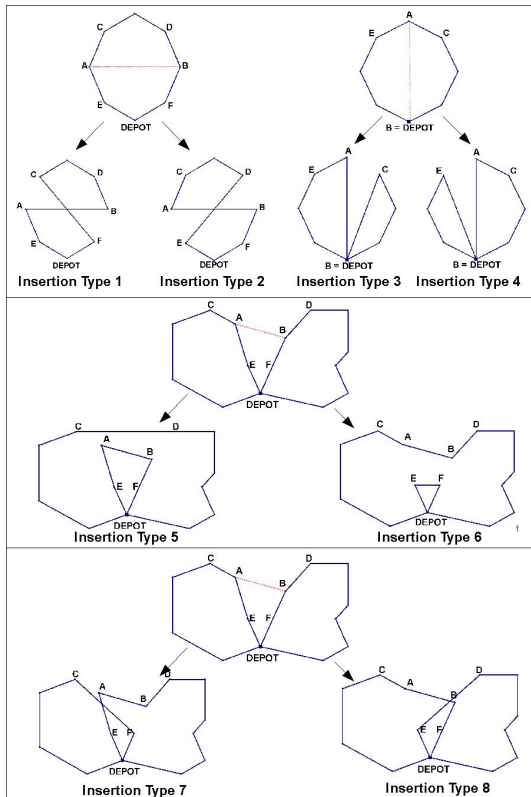


Figure 2: Voisinage - 8 mouvements possibles pour insérer l'arête $[A,B]$ au cours d'une période $k \in \mathbb{K}$

En plus des mouvements 2-opt classiques, ce voisinage autorise la fusion (si une arête connecte les extrémités de deux tournées) ou la scission de tournées (illustrée par les cas 7 et 8 de la figure 2). Après chaque itération, l'algorithme remplace la solution actuelle avec la meilleure solution non taboue du voisinage.

3.3. Liste taboue

La liste taboue est une mémoire à court terme qui évite de retourner à des solutions déjà visitées. En théorie cette liste devrait donc contenir l'ensemble des solutions déjà visitées. Dans la pratique, seules certaines caractéristiques sont enregistrées pour limiter la durée d'exécution et l'espace mémoire requis. A chaque itération t , les arêtes retirées de la solution sont déclarées taboues jusqu'à l'itération $t + \tau$.

Lors d'un 2-opt classique, il est interdit d'insérer des arêtes taboues dans la solution courante (à l'exception du critère d'aspiration, décrit au paragraphe suivant). Par contre, nous avons utilisé un "tabou partiel" pour notre algorithme: une fois un arc non tabou à insérer trouvé, l'algorithme recherche la meilleure manière de l'insérer parmi les $8 \times m$ possibilités, même si la

deuxième arête entrante est taboue.

- Critère d'aspiration

Nous avons autorisé l'application d'un mouvement totalement tabou (qui insère deux arêtes taboues dans la solution courante) si et seulement si la solution obtenue est la meilleure rencontrée jusqu'à cet instant.

3.4. Pénalisation des solutions infaisables

Pour donner à notre algorithme la capacité de s'initialiser à partir d'une solution non réalisable, de s'extraire de minima locaux, et également d'atteindre n'importe quelle région de l'espace des solutions, nous avons retiré la contrainte (4) et nous avons ajouté à la fonction-objectif une pénalité

$$\alpha \times \max(0, (\sum_{k \in \mathbb{K}} x_e^k - 1)).$$

Par conséquent, une arête pourra être utilisée plus d'une fois, au cours de deux périodes différentes par exemple, mais dans ce cas son utilisation sera pénalisée. Cette pénalité permet à la métaheuristique d'atteindre n'importe quelle solution en passant si nécessaire par des solutions infaisables. La valeur de α doit toutefois être suffisamment grande pour que l'algorithme préfère les solutions réalisables aux solutions infaisables lorsqu'un tel choix doit être fait.

3.5. Facteur de diversification

Nous nous sommes inspirés de l'idée de (Taillard, 2002) qui consiste à rajouter une pénalisation au coût des arêtes selon la fréquence de leur utilisation au cours de la recherche locale, partant du principe que les arêtes trop souvent utilisées peuvent bloquer l'algorithme dans une région et l'empêcher de visiter l'espace des solutions. Toutefois, nous ne souhaitons pas pénaliser les arêtes les plus utilisées, n'étant pas certains que ces arêtes ne fassent pas partie de la solution optimale recherchée. Notre algorithme, après γ itérations infructueuses, recherche la meilleure manière d'insérer dans la solution courante l'arête la moins chère parmi celles qui n'ont encore jamais été utilisées dans une des solutions déjà visitées.

Ce facteur nous a permis d'améliorer considérablement nos résultats sur le m -PTVP, en particulier de trouver de très bonnes solutions dans des cas où sans ce facteur nous ne parvenions même pas à trouver des solutions réalisables. Ce facteur de diversification permet également d'améliorer les performances de notre algorithme dans les cas particulier du VRP et du m -PVP.

4. BORNES INFÉRIEURES

4.1. Les m arbres recouvrants disjoints de coût total minimal

L'arbre recouvrant de coût minimal étant une borne inférieure du 1-PVP (PVC), nous pouvons en déduire qu'un ensemble de m arbres recouvrants sans arêtes communes et de coût total minimal (m -ARCM) fournit une borne inférieure au m -PVP, et par conséquent au m -PTVP. Notre première borne inférieure du m -PTVP est donc basée sur le calcul de ces m -ARCM.

L'algorithme de résolution que nous avons utilisé a été proposé par (Roskind et Tarjan, 1985), inspirés par celui de (Kruskal, 1956). En effet, l'algorithme de Kruskal pour l'arbre recouvrant de coût minimal est bien connu: partant de la liste des arêtes triées par coûts croissants et d'une forêt réduite à n noeuds isolés, il consiste à prendre les arêtes une par une et à les ajouter à la forêt si aucun cycle n'est créé. Le principe de l'algorithme de Roskind et Tarjan pour le m -ARCM est assez voisin, mais fait croître m forêts en parallèle. Lors de l'insertion d'une arête, en cas de création d'un cycle, une procédure tente de déplacer les arêtes qui le constituent vers d'autres forêts, pour faire disparaître ce cycle. En cas d'échec de cette procédure, l'arête initiale est rejetée. Cet algorithme polynomial peut être implémenté en $O(m^2 \times n^2)$.

Sachant qu'un arbre recouvrant de coût minimal ne prend en compte ni des contraintes de degré sur les noeuds, ni des contraintes de capacité sur les tournées, nous constatons donc que les m -ARCM nous fournissent une solution qui ne respecte pas forcément les contraintes (1), (2) et (3). Ceci nous laisse envisager plusieurs pistes d'amélioration telles que le remplacement des m -arbres recouvrants disjoints par m 1-arbres recouvrants disjoints ou la génération de contraintes pour une méthode de coupes (contraintes de degré par exemple).

4.2. Le b -couplage parfait

Notre deuxième borne inférieure est obtenue en résolvant le sous-problème de b -couplage parfait qui découle du m -PTVP. Le b -couplage parfait fait partie de la classe des problèmes linéaires résolus polynomialement par (Edmonds et Johnson, 1970), et utilisés par (Miller et Wright, 1995) sur le VRP. Ce problème peut être déduit de la formulation basée sur les arêtes du m -PTVP de la manière suivante:

- Premièrement, relaxer les contraintes de capacité (3) puis sommer les contraintes (1) et (2) sur $k \in \mathbb{K}$:

$$\min \sum_{k \in \mathbb{K}} \sum_{e \in E} c_e x_e^k$$

s. c.

$$\begin{aligned} \sum_{k \in \mathbb{K}} \sum_{e \in \delta(i)} x_e^k &= 2m, & \forall i \in V' \\ \sum_{k \in \mathbb{K}} \sum_{e \in \delta(0)} x_e^k &\geq 2m\lambda \\ \sum_{k \in \mathbb{K}} x_e^k &\leq 1, & \forall e \in E \\ x_e^k &\in \{0, 1\}, & e \in E, \quad k \in \mathbb{K} \end{aligned}$$

- Ensuite, effectuer le changement de variable $\sum_{k \in \mathbb{K}} x_e^k = y_e$ avec $y_e \in \{0, 1\}$, $\forall e \in E$:

$$\min \sum_{e \in E} c_e y_e$$

s. c.

$$\sum_{e \in \delta(i)} y_e = 2m, \quad \forall i \in V' \quad (6)$$

$$\sum_{e \in \delta(0)} y_e \geq 2m\lambda \quad (7)$$

$$y_e \leq 1, \quad \forall e \in E \quad (8)$$

$$y_e \in \{0, 1\}, \quad e \in E \quad (9)$$

- Enfin, supprimer la contrainte (8) devenue redondante puisque par définition $0 \leq \sum_{k \in \mathbb{K}} x_e^k = y_e \leq 1$. Si en plus la contrainte $\sum_{e \in \delta(0)} y_e \geq 2m\lambda$ est remplacée par l'égalité $\sum_{e \in \delta(0)} y_e = 2m\lambda$, alors la formulation obtenue est celle d'un problème appelé b -couplage parfait.

$$\min \sum_{e \in E} c_e y_e$$

s. c.

$$\sum_{e \in \delta(i)} y_e = b_i \quad \text{avec} \quad b_i = \begin{cases} 2m & \forall i \in \{1 \dots n\} \\ 2m\lambda & \text{if } i = 0 \end{cases} \quad (10)$$

$$y_e \in \{0, 1\}, \quad \forall e \in E \quad (11)$$

Ce problème étant polynomial, nous pouvons espérer obtenir des bornes inférieures du m -PTVP en un temps de calcul limité. Contrairement au m -ARCM, le b -couplage parfait prend partiellement en compte les contraintes de degré (1) et (2): la solution obtenue respecte le nombre total d'arêtes incidentes à chaque noeud, mais cette contrainte n'est pas forcément respectée au cours de chacune des m périodes. En revanche, le b -couplage parfait ne conserve aucune propriété de connectivité ou de capacité (3).

5. RESULTATS NUMERIQUES ET ANALYSES

5.1. Implémentation

Le m -PTVP étant un problème nouveau, il n'existe aucune instance spécifique dans la littérature. Pour pouvoir quand même évaluer nos algorithmes, nous avons effectué les tests numériques sur des instances VRP de (Augerat, 1995) et de (Christofides *et al.*, 1979) disponibles sur le site internet suivant <http://neo.lcc.uma.es/radi-aeb/WebVRP/>, ainsi que certaines instances de la TSPLIB (Reinelt, 1995) déjà utilisées pour le m -PVP (Duchenne *et al.*, 2005).

L'algorithme de recherche taboue a été codé en C, celui pour le m -ARCM a été codé en Delphi et enfin les résultats de b -couplage parfait ont été obtenus grâce aux solveurs Glpk et Cplex. Les calculs ont duré en moyenne moins de dix secondes sur un PC Pentium de 3,39 GHz et 1 Go de mémoire.

Les en-têtes suivants seront communs à tous les tableaux de résultats:

- m : nombre de périodes
- Opt : valeur de la solution optimale
- Bi_1 : borne inférieure basée sur les m -ARCM
- Bi_2 : borne inférieure basée sur le b -couplage parfait
- CW : solution initiale obtenue après l'application de l'heuristique adaptée de Clarke et Wright
- T : borne supérieure obtenue par la recherche taboue sans facteur de diversification et sans critère d'aspiration
- TDA : borne supérieure obtenue par la recherche taboue avec facteur de diversification et avec critère d'aspiration
- Pour $i = CW, T$ ou TDA , nous définissons la déviation $\Delta i = \frac{i-Opt}{Opt}$

5.2. Résultats sur le 1-PTVP (VRP)

Les tableaux 1 et 2 montrent les résultats obtenus pour le VRP sur des instances de (Augerat, 1995) ou de (Christofides *et al.*, 1979). Au sein du tableau 1, la colonne BCM contient à titre indicatif les résultats de la borne inférieure de (Baldacci *et al.*, 2007) basée sur la génération de q -routes, seule borne inférieure du VRP qui puisse être obtenue en quelques secondes.

Instance	m	Opt	$\frac{Bi_1}{Opt}$ (%)	$\frac{Bi_2}{Opt}$ (%)	$\frac{BCM}{Opt}$ (%)
B-n45-k5	1	751	48,58	65,51	99,4
B-n50-k7	1	741	39,56	70,85	99,7
B-n68-k9	1	1272	24,7	51,1	98,9
P-n50-k7	1	554	63,02	89,71	98,9
P-n65-k10	1	792	55,16	87,5	98,6
P-n76-k4	1	593	79,65	95,11	98,8
P-n101-k4	1	681	82,56	96,62	97,9
Vrpnc1	1	524,61	71,77	92,57	-
Vrpnc11	1	1042,11	44,66	45,56	-
Vrpnc12	1	819,56	50,92	74,49	-
Vrpnc2	1	835,26	56,55	83,93	-
Vrpnc3	1	826,14	68,06	91,68	-
Vrpnc4	1	1028,42	61,77	85,92	-
Vrpnc5	1	1291,29	53,59	77,9	-
Moyenne	1		57,18	79,18	98,89

Tableau 1. Résultats des bornes inférieures sur le VRP

Les résultats montrent que les bornes inférieures Bi_1 et Bi_2 ne sont pas très performantes puisqu'elles n'atteignent respectivement que 57,18% et 79,18% de l'optimum, alors que la borne inférieure BCM atteint en moyenne 98,89% de l'optimum. Cette faible performance peut s'expliquer par le fait que les m -ARCM ne tiennent compte ni des contraintes de capacité ni des contraintes de degré. Le b -couplage parfait produit de meilleurs résultats parce qu'il prend partiellement en compte les contraintes de degré (en s'assurant qu'il y a bien 2 arêtes incidentes à chaque nœud).

Le tableau 2 illustre la performance de notre borne supérieure TDA, tout en mettant en évidence l'impact des différents éléments introduits. Le réglage de paramètres est le suivant: $\alpha = 1000, \tau = n, \gamma = 75$.

Instance	m	Opt	ΔCW (%)	ΔT (%)	ΔTDA (%)
B-n45-k5	1	751	0,8	0	0
B-n50-k7	1	741	0,67	0	0
B-n68-k9	1	1272	2,91	1,26	1,1
P-n50-k7	1	554	5,42	0,18	0
P-n65-k10	1	792	6,31	0,5	0
P-n76-k4	1	593	9,95	0,67	0,84
P-n101-k4	1	681	10,28	1,47	0,15
Vrpnc1	1	524,61	11,44	0	0,06
Vrpnc11	1	1042,11	2,5	0,46	0,46
Vrpnc12	1	819,56	1,71	0,34	0,21
Vrpnc2	1	835,26	7,78	1,46	1,13
Vrpnc3	1	826,14	7,6	1,12	0,59
Vrpnc4	1	1028,42	10,15	2,59	0,81
Vrpnc5	1	1291,29	9,67	4,02	2,28
Moyenne	1		6,23	1,01	0,55

Tableau 2. Résultats des bornes supérieures sur le VRP

Notre borne supérieure TDA obtient de bons résultats sur le VRP puisque la déviation moyenne avec l'optimum n'est que de 0,55%. Le tableau 2 montre également la contribution des différents éléments introduits puisque la solution initiale CW n'est qu'à 6,23% de l'optimum, et que la recherche taboue simple T maintient une déviation de 1,01%.

5.3. Résultats sur le m -PVP

Les tableaux 3 et 4 montrent les performances de nos algorithmes pour le m -PVP avec des instances TSPLIB (Reinelt, 1995). Au sein du tableau 3, la colonne $m * PVC$ contient les résultats de la borne inférieure simple mentionnée par (Duchenne *et al.*, 2005) obtenue en multipliant par m la solution optimale du PVC.

Instance	m	Opt	$\frac{Bi.1}{Opt}$ (%)	$\frac{Bi.2}{Opt}$ (%)	$\frac{m*PVC}{Opt}$ (%)
bays29	1	2020	77,08	96,39	100
	2	4694	81,49	100	86,07
fri26	1	937	79,08	94,24	100
	2	2218	80,12	99,55	84,49
gr17	1	2085	68,15	80,77	100
	2	4915	70,66	98,92	84,84
	3	9005	66,93	100	69,46
gr21	4	13668	67,77	100	61,02
	1	2707	79,83	100	100
	2	6900	77,81	99,72	78,46
	3	12486	76,69	100	65,04
gr24	1	1272	79,48	96,46	100
	2	3147	78,52	100	80,84
	3	5614	76,72	100	67,97
	4	8435	77,46	100	60,32
Moyenne			75,85	97,74	82,57

Tableau 3. Résultats des bornes inférieures sur le m -PVP

Nos bornes inférieures $Bi.1$ et $Bi.2$ sont performantes sur le m -PVP puisqu'elles atteignent en moyenne 75,85% et 97,74% de l'optimum. Ces bornes atteignent même 75,42% et 99,82% pour $m > 1$ et surclassent naturellement le $m * PVC$ (qui n'atteint que 73,85% si $m > 1$). Ces résultats sont cohérents avec les résultats obtenus sur le VRP puisque les contraintes de capacité n'existent pas pour un m -PVP.

Le tableau 4 montre les performances obtenues par notre borne supérieure TDA tout en mettant en évidence l'impact des différents éléments introduits. Le réglage de paramètres est le suivant: $\alpha = 1000$, $\tau = n$ et $\gamma = 50, 150, 275, 425$ lorsque $m = 1, 2, 3, 4$.

Instance	m	Opt	ΔCW (%)	ΔT (%)	ΔTDA (%)
bays29	1	2020	5,56	0,42	0
	2	4694	8,37	0	0,09
fri26	1	937	2,7	0	0
	2	2218	8,2	0,09	0
gr17	1	2085	2,66	0	0
	2	4915	4,38	0,43	0
	3	9005	3,19	0,18	0,13
	4	13668	3,46	0,23	0,09
gr21	1	2707	3,87	0	0
	2	6900	11,88	0,25	0
	3	12486	9,82	0,30	0,02
gr24	1	1272	2,9	0	0
	2	3147	6,84	0,35	0
	3	5614	9,42	0,57	0,07
	4	8435	11,88	0,25	0,27
Moyenne			6,34	0,21	0,05

Tableau 4. Résultats des bornes supérieures sur le m -PVP

Notre borne supérieure TDA obtient de bons résultats sur le m -PVP avec une déviation moyenne par rapport à l'optimum d'à peine 0,05%.

5.4. Résultats sur le m -PVRP

Le tableau 5 récapitule les performances de nos bornes supérieures et inférieures sur le m -PTVP avec des instances de type VRP de (Augerat, 1995) et de (Christofides *et al.*, 1979), pour un nombre de périodes $m > 1$. La colonne $m * VRP$ contient les résultats de la borne inférieure simple qui peut être obtenus en multipliant par m la solution optimale du VRP. Le réglage des paramètres est le suivant: $\alpha = 1000$, $\tau = n$ et $\gamma = 150, 275, 425, 600, 800, 1025$ lorsque $m = 2, 3, 4, 5, 6, 7$.

Instance	m	TDA (%)	$\frac{Bi.1}{TDA}$ (%)	$\frac{Bi.2}{TDA}$ (%)	$m * VRP$
B-n45-k5	2	1602	51,75	75,97	93,76
	3	2653	55,14	88,84	84,92
	4	4151	60,87	95,86	72,37
B-n50-k7	2	1573	42,96	82,77	94,21
	3	2628	47,46	94,56	84,59
B-n68-k9	2	2672	27,83	71,18	95,21
	3	4312	31,4	89,91	88,5
P-n50-k7	2	1264	65,37	98,02	87,66
	3	2207	64,42	98,64	75,31
P-n65-k10	2	1761	58,72	96,99	89,95
	3	3055	57,91	97,41	77,77
	4	1377	81,82	97,75	86,13
P-n76-k4	3	2374	81,12	97,64	74,94
	4	3564	80,48	97,47	66,55
	2	1575	82,96	98,73	86,48
	3	2702	81,82	98,37	75,61
P-n101-k4	4	4038	80,88	98,66	67,46
	5	5528	80,85	98,9	61,6
	6	7178	80,85	99	56,92
	7	8965	80,78	99,1	53,17
	2	1219,27	72,61	95,63	86,05
	3	2127,77	71,61	98,2	73,97
	4	3720,13	50,43	61,16	84,04
Vrpncl1	4	5275,64	52,83	72,06	79,01
	5	7124,1	53,72	78,66	73,14
	6	9238,82	54,28	84,08	67,68
	7	11668	55,14	86,65	62,52
	2	1821,79	53,55	82,19	89,97
	3	3000,82	56,45	88,68	81,93
	4	4474,18	57,45	92,2	73,27
Vrpncl2	2	1852,48	60,82	94,24	90,18
	2	1874,57	69,7	96,6	88,14
Vrpncl3	3	3182,68	69,47	97,7	77,87
	4	4707,64	69,37	98,51	70,2
	5	6453,78	69,25	98,41	64
	2	2381,38	65,82	94,62	86,37
	3	4023,86	66,96	96,88	76,67
Vrpncl4	4	5983,11	66,6	97,33	68,75
	5	8141,82	66,74	98,06	63,16
	6	10576,2	66,51	98,19	58,34
	2	2947,26	58,07	88,74	87,63
	3	4935,02	60,9	92,83	78,5
Vrpncl5	4	7275,41	61,65	94,42	70,99
	5	9949,03	61,65	94,93	64,9
	Moyenne		62,98	91,35	77,34

Tableau 5. Résultats pour le m -PTVP

Les résultats montrent que la borne $m * VRP$ peut être performante pour des faibles valeurs de m appliquées à de petites instances. La borne $Bi.2$ produit en général les valeurs les plus élevées, ce qui reste

cohérent avec tous les résultats trouvés jusque-là. La borne Bi_{-1} , dont la performance augmente lorsque le nombre de routes nécessaires par période diminue (par exemple l'instance P-n101-k4 ne nécessite que 4 routes par période), devrait être considérablement améliorée lorsque les contraintes de degré auront été prises en compte.

La déviation de la borne supérieure par rapport à la meilleure borne inférieure est en général inférieure à 10%. Ceci nous laisse penser que notre recherche taboue avec facteur de diversification est performante pour le m -PTVP. Nous pouvons espérer réduire cette déviation de manière encore plus significative une fois que les différentes améliorations envisagées auront été appliquées aux bornes inférieures, et que les paramètres optimaux auront été trouvés pour notre borne supérieure.

CONCLUSION

Le problème de tournées de véhicules m -péripatétiques est un problème nouveau avec des applications très concrètes telles que le transport de fonds. Nous avons présenté une modélisation en deux indices et deux bornes inférieures pour lesquelles plusieurs pistes d'amélioration sont déjà envisagées (par exemple, ajouter les contraintes de degré sur les m -ARCM). Nous avons également présenté notre algorithme de recherche taboue avec facteur de diversification qui fournit de bons résultats sur le m -PTVP en général, et sur deux cas particuliers connus: le VRP et le m -PVP. Le travail en cours inclut l'optimisation des paramètres de la recherche taboue, le développement d'une nouvelle borne très prometteuse basée sur le partitionnement et l'application de méthodes exactes.

REFERENCES

- Augerat, P., 1995. *Approche polyédrale du problème de tournées de véhicules*. Thèse de doctorat, Institut National Polytechnique de Grenoble, France.
- Baldacci R., Christofides N. and Mingozzi A., 2007. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, DOI 10.1007/s10107-007-0178-5.
- Christofides N., Mingozzi A. and Toth P., 1979. *The vehicle routing problem*. IN: Christofides N., Mingozzi A. and Toth P. (Eds.), *Combinatorial Optimization*, Wiley, Chichester, p. 315-338.
- Clarke G. and Wright J., 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12, p. 568-581.
- Cordeau J.F., Gendreau M., Hertz A., Laporte G. and Sormany J.S., 2005. *New heuristics for the vehicle routing problem*. IN: Langevin A. and Riopel D. (Ed.), *Logistics systems: design and optimisation*, Wiley, p. 279-298.
- De Kort J.B.J.M., 1991. Lower bounds for symmetric k -peripatetic salesman problems. *Optimization*, 22, p. 113-122.
- De Kort J.B.J.M., 1993. A Branch and bound algorithm for symmetric 2-peripatetic salesman problems. *European Journal of Operational Research*, 70, p. 229-243.
- Duchenne E., Laporte G. and Semet F., 2005. Branch-and-Cut algorithms for the undirected m -Peripatetic Salesman Problem. *European Journal of Operational Research*, 162, p. 700-712.
- Edmonds J. and Johnson E.L., 1970. Matching: A well-solved class of integer linear programs. *Combinatorial structures and their applications*, Calgary International Conference, Gordon and Breach, p. 89-92.
- Glover F. and Laguna M., 1993. *Tabu search*. IN: Reeves C.R. (Ed.), *Modern Heuristic Techniques for Combinatorial Problems*, Blackwell, Oxford, U.K., p. 70-150.
- Krarup J., 1975. *The peripatetic salesman and some related unsolved problems*. IN: Roy, B. (Ed.), *Combinatorial Programming Methods and Applications*, Reidel, Dordrecht, p. 173-178.
- Kruskal J.B., 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the american mathematical society*, 7(1), p. 48-50.
- Miller D.L. and Wright J., 1995. A matching based exact algorithm for capacitated vehicle routing problems. *ORSA Journal on Computing*, 7, p. 298-320.
- Prins C., 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers and Operations Research*, 31, p. 1985-2002.
- Roskind J. and Tarjan R. E., 1985. A note on finding minimum-cost edge-disjoint spanning trees. *Mathematics of Operations Research*, 10, p. 701-708.
- Taillard E.D., 1993. Parallel Iterative Search Methods for vehicle routing Problem. *Networks*, 23, p. 661-673.

Toth P. and Vigo D., 1998. *The granular tabu search (and its application to the vehicle routing problem)*. Università di Bologna, DEIS, Technical report OR/98/9, Italy.

Toth P. and Vigo D., 2002. *The Vehicle Routing Problem*, Philadelphia, SIAM.

Wolfer Calvo R. and Cordone R., 2003. A heuristic approach to the overnight security service problem. *Computers and Operations Research*, 30, p. 1269-1287.

Reinelt G., 1995, Universität Heidelberg, disponible sur le site <http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95/>