

Tournées de Véhicules avec Conflits

Khaoula Hamdi, Nacima Labadi, Alice Yalaoui

ICD - LOSI - Université de Technologie de Troyes, 12 rue Marie Curie 10010 Troyes, France
khaoula.hamdi, nacima.labadi, alice.yalaoui@utt.fr

Mots-Clés : *tournées de véhicules, matières dangereuses, conflits.*

1 Problématique

Le problème de tournées de véhicules est un problème combinatoire classique qui a été intensément étudié tout au long des dernières décades et pour lequel plusieurs méthodes exactes et approchées ont été développées [1]. D'autre part, il existe de nombreux travaux consacrés au transport des matières dangereuses. Ces derniers introduisent souvent la notion du risque qui a été modélisée de différentes manières et considèrent pour la plupart un chemin origine-destination. L'association des tournées de véhicules au transport des matières dangereuses est très peu fréquente. Il existe quelques travaux qui se penchent sur cette problématique et qui considèrent le risque comme une grandeur à minimiser.

Dans ce travail, on considère le risque d'une manière différente. En effet, les matières dangereuses sont regroupées en plusieurs classes selon leurs natures. Certaines sont incompatibles entre elles et leur présence dans un même véhicule peut engendrer un sérieux danger. Dans [2,3], les auteurs ont abordé la notion de compatibilité entre produits dans le cadre d'études de cas industriels, et peu de détails ont été donné sur la gestion de cette contrainte. Nous avons défini formellement le problème de tournées de véhicules avec conflits (VRPC) pour la première fois dans [4].

2 Développements

Pour résoudre ce problème, trois heuristiques classiques connues pour le problème de tournées de véhicules ont été adaptées :

L'algorithme de Clark et Wright : on commence par affecter chaque client à un véhicule différent. Ensuite, on évalue le gain résultant de la fusion de chaque couple de tournées. La fusion qui permet d'obtenir le meilleur gain sans générer de conflit est alors effectuée.

L'algorithme de meilleure insertion : à chaque itération, on évalue le coût supplémentaire engendré par l'insertion de chaque client non servi dans toutes les positions possibles. L'insertion qui donne le coût minimal sans générer des conflits est réalisée.

Heuristique de type cluster first - route second : l'heuristique commence par regrouper les clients en sous-ensembles en se guidant de la borne inférieure du 1-arbre recouvrant de coût minimum, tout en prenant en compte les conflits. Ensuite, on construit l'enveloppe convexe pour chaque sous-ensemble.

Les clients du sous-ensemble, qui n'appartiennent pas à l'enveloppe, sont insérés en rallongeant, le moins possible, la tournée résultante.

Une nouvelle heuristique basée sur la répartition des matières dangereuses en classes est également proposée. Les clients dont les demandes appartiennent à une même classe sont tout d'abord groupés dans un même ensemble, ensuite, une règle de priorité est appliquée pour construire les tournées. Tout d'abord, on définit le degré de chaque classe comme étant le nombre de classes en conflit avec elle. Les classes sont triées suivant un ordre décroissant des degrés. On propose trois variantes de cette heuristique. Dans chacun des trois algorithmes, l'insertion des clients se fait par meilleure insertion.

Variante 1 : la priorité est donnée à la classe de plus haut degré. Quand tous les clients de cette classe sont insérés, on passe à la classe suivante dans la liste triée.

Variante 2 : les classes de degré zéro (classes compatibles avec toutes les autres classes) sont retirées de la liste. À chaque étape, on considère les clients de la classe de plus haut degré et ceux des classes de degré zéro. Quand les clients de la classe de plus haut degré sont affectés, on passe à la classe suivante de la liste triée, avec les clients restants des classes de degré zéro.

Variante 3 : on commence par construire un sous-ensemble contenant la classe de plus degré et toutes les classes compatibles avec elle (et compatibles entre elles). Quand tous les clients de ce sous ensemble sont insérés, on recommence le processus avec les classes restantes.

3 Résultats et conclusions

Deux benchmark ont été créés pour tester les performances des heuristiques, en ajoutant les données sur les conflits à des instances du VRP classique. Le premier benchmark est généré en utilisant la méthode proposée par Soriano et Gendreau [5]. Le deuxième benchmark, proposé en [3] repose sur l'appartenance des demandes aux classes de matières dangereuses et crée les conflits suivant les compatibilités réelles entre les matières.

Les trois heuristiques adaptées ont été testées sur les deux benchmarks, alors que les variantes de la nouvelle heuristique ne sont testées que sur le deuxième benchmark. On a constaté que l'algorithme de Clark et Wright permet d'obtenir un coût inférieur par rapport aux autres heuristiques, alors que la variante 1 de la nouvelle heuristique est plus performante quant au nombre de véhicules utilisés.

Pour améliorer les résultats obtenus par les heuristiques décrites dans ce résumé, une métaheuristique a été développée et est actuellement en cours de finalisation. Nous envisageons également de concevoir la première borne inférieure pour le VRP avec conflits.

Références

- [1] P. Toth et D. Vigo, The Vehicle Routing Problem, Monographs and Discrete Mathematics and Applications. *Society for Industrial & Applied Mathematics*, U.S., 2002.
- [2] A. Ceselli, G. Righini et M. Salani, A Column Generation Algorithm for a Rich Vehicle Routing Problem. *Transportation Science*, Vol. 43, pp 56-69, 2009.
- [3] G. Giuseppe, G. Galliano et G. Stecca, An Evolutionary Algorithm for Vehicle Routing Problem with Real Life Constraints. *The 41st CIRP Conference on Manufacturing Systems*, Tokyo, 2008.

- [4] K. Hamdi, N. Labadi and A. Yalaoui, Vehicle routing problem with conflicts. AIRO 2009, Siena, p 22, 2009.
- [5] M. Gendreau et P. Soriano, Tabu Search algorithms for the maximum clique problem. In : DS Johnson, MA Trick, editors. Cliques Colouring and Satisfaisability. *DIMACS Series in Discrete Mathematics and Theoretical Computer Sciences*, Vol. 26. Providence : American Mathematical Society, pp 221-242, 1996