Résolution du *Capacitated Arc Routing Problem* (CARP) avec LocalSolver

Bienvenu Bambi

LocalSolver, 24 avenue Hoche, 75008 Paris, France
bbambi@localsolver.com

Mots-clés : *tournées de véhicules*

1 Introduction

Déterminer un trajet efficace de parcours des routes d'une zone tout en respectant des contraintes liées au réseau routier constitue un problème de planification de tournées de véhicules difficile dans un environnement urbain complexe. Cela était l'enjeu d'un projet client de LocalSolver. Le problème d'optimisation était de calculer un itinéraire qui visite toutes les rues d'une agglomération en respectant différentes contraintes opérationnelles de façon à minimiser la distance parcourue par les véhicules. Pour résoudre ce problème industriel, la zone à parcourir peut être modélisée par un graphe orienté dont les arcs représentent les sens de parcours des routes et les sommets sont les intersections entre les routes. Il s'agit alors de résoudre un problème d'*Arc Routing* avec Localsolver.

LocalSolver est un solveur d'optimisation mathématique basé sur différentes techniques de recherche opérationnelle, combinant des méthodes exactes, telles que la programmation linéaire, non linéaire et par contraintes, et heuristiques, comme la recherche locale [1]. On s'intéresse ici au problème d'*Arc Routing* qui est décrit de la façon suivante. Considérons un graphe orienté $G = (V, E)$ où V est l'ensemble de ses sommets, E l'ensemble de ses arcs et $R \subseteq E$ l'ensemble des arcs à visiter (arc requis). Une flotte de K véhicules identiques est basée à un sommet désigné, le dépôt. Chaque arc $e \in E$ du graphe induit un coût c_{ij} lorsqu'un véhicule l'emprunte. Enfin, une tournée de véhicules doit commencer et finir au dépôt désigné. L'objectif de l'*Arc Routing Problem* est alors de déterminer une tournée de coût minimal qui visite chaque arc requis au moins une fois.

2 Modélisation

La modélisation de l'*Arc Routing Problem* au sein de LocalSolver est une extension des modèles de *vehicule routing* tel que le *CVRP* : chaque véhicule est modélisé par une variable de liste $routes[k]$ dont la valeur est égale à l'ensemble ordonné des arcs requis qu'il visite.

En pratique, étant donné une route de l'agglomération à parcourir, on ne veut la visiter que dans un des deux sens (pour les routes qui ne sont pas à sens unique). Ainsi si on note r la route, $r^1, r^2 \in E$ les arcs associés à ses deux sens de parcours, la contrainte de respect de visite de cette route s'écrit alors :

```
for [r in 0..|R|-1] {
        constraint contains(routes, r^1)
          + contains(routes, r^2) == 1;
}
```

Le modèle de l'*Arc Routing Problem* peut facilement être étendu au *Capacitated Arc Routing Problem* dans lequel les véhicules ont une capacité maximale Q et les arcs à visiter ont une

demande à satisfaire q_r. Une tournée de véhicule doit alors vérifier que la demande totale satisfaite sur son itinéraire ne dépasse pas sa capacité maximale Q.

3 Résultats

Le tableau suivant compare les résultats de LocalSolver à ceux de la littérature pour un groupe d'instances [2]. On utilise un benchmark de 34 instances que l'on répartit en 3 catégories : petites (98 arêtes), moyennes (190 arêtes), grandes (375 arêtes).

	Ecart aux bornes inf.		Ecart aux bornes sup.	
	60s	600s	60s	600s
instances à 98 arêtes	1.30%	1.06%	0.13%	-0.10%
instances à 190 arêtes	2.46%	1.59%	0.78%	-0.07%
instances à 375 arêtes	5.83%	4.66%	-0.35%	-1.46%

TAB. 1 – Comparaison des performances de LocalSolver par rapport aux résultats de la littérature

4 Conclusion

La résolution du *Capacitated Arc Routing Problem* avec LocalSolver permet d'obtenir d'excellents résultats sur les instances de la littérature. On observe des améliorations sur toutes les catégories d'instances (0.22% de *gap* par rapport aux meilleures solutions connues en 60 secondes) et des écarts aux bornes inférieures très faibles. Les extensions prévues visent à améliorer les bornes obtenues sur les problèmes de *Capacitated Arc Routing*.

Références

[1] F. Gardi, T. Benoist, J. Darlay, B. Estellon, et R. Megel. *Mathematical Programming Solver Based on Local Search*, Wiley, 2014

[2] J. Brandão, et R. Eglese. *A Deterministic Tabu Search Algorithm for the Capacitated Arc Routing Problem (CARP)*, Computers & Operations Research., 2008