

Optimisation de tournées avec clients optionnels : hybridation entre recherche de sélections et recherche de séquencements

Trong-Hieu Tran¹, Cédric Pralet², Hélène Fargier¹

¹ IRIT-CNRS, Université de Toulouse, Toulouse, France
{trong-hieu.tran, helene.fargier}@irit.fr

² ONERA/DTIS, Université de Toulouse, F-31055 Toulouse, France
cedric.pralet@onera.fr

Mots-clés : *OPTW, SAT, recherche locale, apprentissage de conflits*

1 Problématique et démarche générale

Nous abordons ici le problème OPTW (*Orienteering Problem with Time Windows*) dans lequel on considère un ensemble de N clients optionnels pouvant être visités dans certaines fenêtres temporelles et qui apportent chacun un certain profit. Ce problème implique deux niveaux de décision, avec d'un côté le problème de la sélection des clients à visiter, et de l'autre le séquençement des visites des clients sélectionnés. Pour traiter ces deux niveaux, une première approche consiste à définir des méthodes spécifiques prenant en compte simultanément les aspects sélection et séquençement, ce qui est fait dans les algorithmes de référence pour le problème OPTW [3]. Une autre approche que nous explorons s'inspire de méthodes de type *SAT modulo theory* [1] dans lesquelles des décisions sur des variables booléennes d'un problème sont prises pour pouvoir ensuite raisonner, sans disjonction, sur un problème spécifique disposant de méthodes de résolution très efficaces. Dans l'approche explorée pour les OPTW, les décisions de sélection de clients sont prises à un premier niveau, et une recherche locale efficace est utilisée sur le problème bas niveau TSPTW (*Traveling Salesman Problem with Time Windows*) pour évaluer très rapidement la faisabilité de la sélection d'un ensemble de clients fixé [2]. En outre, comme dans les approches SMT classiques, des explications peuvent être calculées pour aider la prise de décision au premier niveau. Dans notre cas, ces explications correspondent à des conflits de sélection de clients qui vont venir enrichir une base de conflits.

2 Méthode hybride mise en oeuvre

Dans la méthode définie, la sélection de clients est traitée par une recherche à voisinage large (LNS). Cette dernière part d'une sélection initiale et applique à chaque étape des opérations de destruction et de reconstruction sur cette sélection, tout en s'assurant que la solution retenue est compatible vis-à-vis de conflits de sélection générés en cours de recherche. Chaque conflit de sélection correspond à un sous-ensemble $S \subseteq \{1, \dots, N\}$ tel que visiter tous les clients dans S est jugé infaisable. En introduisant pour chaque $i \in \{1, \dots, N\}$ une variable $x_i \in \{0, 1\}$ décrivant la sélection du client i , le conflit peut être vu comme une clause $\bigvee_{i \in S} \neg x_i$ spécifiant qu'au moins un client de S ne doit pas être sélectionné.

Globalement, la phase de destruction du LNS désélectionne k clients et la phase de réparation sélectionne de nouveaux clients avec l'objectif de maximiser le profit total collecté, tout en prenant en compte les conflits de sélection. Pour cela, l'algorithme de réparation explore des options de réinsertion de clients via une recherche arborescente en profondeur d'abord avec un nombre limité de backtracks. Des mécanismes de propagation de contraintes servent de plus à élaguer l'arbre de recherche en exploitant les conflits de sélection et en comparant la valeur

de la meilleure solution connue avec une borne sur le meilleur coût qu’il est possible d’obtenir à partir de la solution courante. La technique SAT des *2-watched literals* est aussi exploitée pour réaliser les propagations efficacement. Suite à ces élagages, si l’insertion d’un nouveau client j n’est pas interdite, alors un algorithme de recherche incomplet (mais très rapide) pour le problème TSPTW est appelé, de manière à évaluer la faisabilité effective de l’insertion du client j dans le plan courant.

Lors des recherches arborescentes limitées réalisées à chaque itération du LNS, des conflits obtenus par un type d’explication appelé *xpTree* sont mémorisés. Ils permettent de représenter les sous-arbres déjà explorés. Il serait possible d’oublier ces conflits en cours de recherche pour prendre en compte le fait que la faisabilité des insertions effectives de clients est évaluée sur la base d’une méthode incomplète. De plus, après chaque convergence vers une solution localement optimale, l’algorithme cherche à expliquer pourquoi les clients non visités ne peuvent pas être insérés dans le plan courant. Pour cela, une méthode de programmation dynamique capable de produire des conflits est exploitée. Ces conflits, obtenus par le type d’explication *xpDP* (*explain with Dynamic Programming*), servent à améliorer les recherches arborescentes futures. Pour limiter le temps dédié à ces explications, on fixe une taille maximum pour les conflits recherchés et on fixe pour chaque client j un nombre maximum de fois où l’absence du client j dans une solution localement optimale doit être expliquée.

3 Résultats expérimentaux

Le tableau 1 donne les résultats obtenus sur 76 instances standards du problème OPTW, regroupées dans 8 classes. Plus précisément, il donne, sur la base de 10 résolutions pour chaque instance, l’écart moyen en pourcentage obtenu en 1 minute par rapport à la meilleure solution connue. Globalement, on voit que les performances dépendent de la taille des fenêtres temporelles considérées (approche efficace pour les instances du groupe 1 qui impliquent des fenêtres plus courtes). Une raison à cela est que pour les instances avec fenêtres longues, les conflits produits contiennent beaucoup plus de clients et génèrent beaucoup moins d’élagage de l’arbre de recherche. Les résultats montrent aussi que dans tous les cas, le fait de partitionner le problème en raisonnant d’abord sur la sélection (indépendamment des aspects temporels), puis sur le séquençement est une approche viable, qui permet de réutiliser directement des approches bas niveau efficace pour traiter le problème TSPTW. Dans le futur, nous envisageons nous inspirer des méthodes de recherche tabou en utilisant des conflits temporaires interdisant de reconsidérer la sélection simultanée d’un petit nombre de clients. Nous envisageons également étendre l’approche à d’autres problèmes de tournées impliquant des décisions de sélection, comme les TSPTW avec fenêtres multiples ou les *Team OPTW* qui mettent en jeu plusieurs véhicules.

	no xp	xpDP	xpTree	xpDP+xpTree		no xp	xpDP	xpTree	xpDP+xpTree
C1	0.54	0.51	0.51	0.61	C2	0.86	0.86	0.91	1.03
R1	1.09	0.98	1.11	0.95	R2	2.33	2.13	3.12	3.09
RC1	0.98	0.78	0.83	0.60	RC2	2.92	2.98	3.53	3.47
PR1	2.45	2.20	2.41	2.00	PR2	5.03	5.08	5.36	5.23

TAB. 1 – Ecart moyen obtenu suivant l’utilisation de chaque type d’explication pour LNS

Références

- [1] Clark W. Barrett, Roberto Sebastiani, Sanjit A. Seshia, and Cesare Tinelli. Satisfiability modulo theories. In *Handbook of Satisfiability*, volume 185, pages 825–885. 2009.
- [2] C. Pralet. Iterated maximum large neighborhood search for the traveling salesman problem with time windows and its time-dependent version. *To appear in Computer & Operations Research*.
- [3] V. Schmid and J. F. Ehmke. An effective large neighborhood search for the team orienteering problem with time windows. In *International Conference on Computational Logistics*, 2017.