

Une méthode ALNS appliquée à la planification dans le domaine de la santé. *

G erard Olivier^{1,2}, Brisoux Devendeville Laure², Lucet Corinne², Darras Sylvain¹

¹ Laboratoire MIS (UR 4290), Universit  Picardie Jules Verne, France
{olivier.gerard, laure.devendeville, corinne.lucet}@u-picardie.fr

² Evolucare Technologies, France
{o.gerard, s.darras}@evolucare.com

Mots-cl s : *ALNS, planification, sant , optimisation.*

1 Introduction

L'am lioration du syst me de soin est l'un des plus grand d fis que de nombreux pays auront   relever dans les ann es   venir. Parmi les axes d'am lioration possibles, celui qui nous int resse dans cet article est la planification de rendez-vous m dicaux. Ce probl me NP-difficile appartient   la cat gorie Resource Constrained Project Scheduling Problem (RCPSP) [1, 3] et a  t  le sujet de nombreuses  tudes ces derni res ann es, particuli rement dans le domaine de la sant . R soudre un probl me RCPSP revient   trouver les meilleurs affectations de ressource et les meilleures dates de d but   un ensemble d'activit s. Dans le domaine de la sant , cela consiste   trouver des dates de d but et des ressources m dicales (personnel m dical, salles et  quipements)   un ensemble de rendez-vous n cessitant g n ralement la pr sence d'un patient sp cifique. Pour tester nos approches, nous disposons de plusieurs sc narios  labor s avec l'aide de planificateurs de plusieurs  tablissements de sant  en France. Dans cet article, nous proposons une adaptation de la m thode Adaptive Large Neighborhood Search (ALNS)   la r solution de notre probl me.

2 D finition du probl me

Notre probl me est d fini comme suit : un ensemble A de rendez-vous doit  tre planifi  sur un horizon H . Chaque rendez-vous $a \in A$ est caract ris  par sa dur e, sa fen tre de faisabilit  $[ES_a, LS_a] \subseteq H$, et une quantit  $qtreq_a^\pi$ de ressources de propri t  π requise.

Chaque ressource r appartenant   l'ensemble fini des ressources renouvelable R peut poss der diff rentes propri t s $\pi \in \Pi$, qualifiant par exemple diff rentes comp tences.   chaque rendez-vous sont  galement associ s deux coefficients intervenant dans la priorit  qu'il lui sera port e pour sa planification : un coefficient d'importance $Essential_a$ et un coefficient d'urgence $Emergency_a$. Des contraintes de pr c dence s'appliquent entre les rendez-vous et certaines ressources peuvent  tre pr affect es afin de repr senter les pr f rences.

La qualit  d'une solution Sol est  valu e par le nombre de rendez-vous $a \in \overline{Sol}$ qui n'ont pu  tre planifi s, pond r  par leur importance $Essential_a$ et par la somme des retards des rendez-vous $a \in Sol$ par rapport   ES_a , pond r  par leur urgence $Emergency_a$. Notre objectif est de trouver une solution valide tout en minimisant la fonction objective f d finie dans l' quation 1.

$$f = \sum_{a \in \overline{Sol}} Essential_a + \sum_{a \in Sol} \frac{t_a - ES_a}{LS_a - ES_a} \times Emergency_a \quad (1)$$

*Ce projet est soutenu par le projet LORH (CIFRE N  2018/0425 entre Evolucare et le Laboratoire MIS)

3 Méthode de résolution

Pour construire la solution initiale, un algorithme glouton est utilisé. Les rendez-vous à placer sont choisis dans un ordre aléatoire. Pour un rendez-vous a , le glouton cherche le premier timeslot t_a sur lequel un ensemble de ressource R_a correspondant exactement aux ressources requises par a est disponible. Puis, afin de se déplacer dans l'espace de recherche nous appliquons la méthode ALNS. Cette méthode a été introduite dans [5] et a été appliquée à de nombreux problèmes de planification [2, 4]. Celle-ci, à chaque itération, choisit un mouvement parmi un ensemble de mouvements pré-sélectionnés, et l'applique à la solution courante. Cette sélection dépend du poids w_m^s associé à chaque mouvement m , résultat des récompenses obtenues par m en fonction des améliorations auxquelles celui-ci a contribué. Il s'agit de la couche adaptative de l'algorithme ALNS. Afin de stabiliser le processus d'apprentissage, la recherche est organisée en blocs d'itérations appelés segments. A l'issue de chaque segment, les poids w_m^s sont mis à jour.

Pour notre problème de planification, les différents mouvements détruisent k rendez-vous et tentent d'en reconstruire ensuite le plus possible. Le choix des rendez-vous à détruire varie d'un mouvement à l'autre : il peut être aléatoire, basé sur la difficulté à placer certains rendez-vous ou cibler des rendez-vous dont la destruction permet le placement d'un autre choisi auparavant. Les méthodes de reconstruction sont elles aussi différentes, et peuvent soit tenter de replacer les rendez-vous de manière optimale soit replacer le plus de rendez-vous possible avec l'algorithme glouton cité précédemment. Lorsque la recherche stagne et pour s'échapper des minimums locaux, des mouvements de diversifications sont appliqués à la solution. Nous utilisons alors des mouvements d'échange de rendez-vous et de décalage à gauche pour perturber une solution. Un restart peut également être appliqué si la période sans amélioration se prolonge.

4 Expérimentations & Résultats

Pour tester notre méthode nous avons généré des instances inspirées d'instances réelles impliquant jusqu'à plusieurs centaines de rendez-vous et des dizaines de ressources différentes. Nous avons comparé les résultats de l'ALNS aux solutions optimales obtenues avec un modèle de programmation linéaire 0-1 implémenté sous CPLEX sur les instances les plus petites, CPLEX ne donnant pas l'optimalité sur les instances de grandes tailles. Le temps de calcul de l'ALNS est limité à deux minutes et celui de CPLEX à deux heures. Nous obtenons une différence moyenne de qualité de solution de 1,32 par rapport à l'optimalité sur ces instances. Nous avons aussi comparé notre approche avec et sans couche adaptative (ALNS et LNS). Nous avons pu constater une amélioration moyenne de 8,12% avec la couche adaptative sur toutes les instances, avec une amélioration pouvant aller jusqu'à 36,02% sur certaines instances d'une même famille. Les résultats seront détaillés lors de la conférence.

Références

- [1] P. Baptiste, P. Laborie, C. LePape, and W. Nuijten. Constraint-based scheduling and planning. In *Foundations of artificial intelligence*, volume 2, pages 761–799. Elsevier, 2006.
- [2] E.F. Bueno. *Mathematical modeling and optimization approaches for scheduling the regular-season games of the National Hockey League*. PhD thesis, École Polytechnique de Montréal, 2014.
- [3] D.S. Johnson and M.R. Garey. *Computers and intractability : A guide to the theory of NP-completeness*. WH Freeman, 1979.
- [4] L.F. Muller. An Adaptive Large Neighborhood Search Algorithm for the Multi-mode RCPSP. *DTU Manag. Eng*, 3 :25, 2011.
- [5] S. Ropke and D. Pisinger. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation science*, 40(4) :455–472, 2006.