Résolution heuristique du *Stacker Crane Problem* préemptif et asymétrique à l'aide d'une *Arbre-représentation* des tournées

M. Lacroix[2*], H. Kerivin[1], A. Quilliot[2], H. Toussaint[2]

[1] Department of Mathematical Sciences, Clemson University, CLEMSON, O-326 Martin Hall, Clemson, SC 29634 - USA.
kerivin@exchange.clemson.edu

[2] LIMOS, CNRS UMR 6158, Université Blaise-Pascal - Clermont-Ferrand II, Complexe Scientifique des Cézeaux, 63177 Aubière, Cedex - France
{lacroix, quilliot, toussain}@isima.fr

Mots-Clés : *Stacker Crane Problem, relais, tournée de véhicules, recherche locale, heuristique.*

1 Introduction

Parmi tous les problèmes de Ramassage et Livraison (*Pickup and Delivery*) le *Stacker Crane Problem* est caractérisé par le fait de disposer d'un unique véhicule, ne transportant qu'une demande à la fois. Il a tout d'abord été introduit par Frederickson et *al.* dans [1]. Il peut se définir de la manière suivante : étant donnés un graphe G dont les arcs sont valués par des longueurs (ou coûts) depuis un nœud « dépôt » spécifique et un ensemble de demandes K, il s'agit de déterminer la tournée d'un unique véhicule V, de telle sorte que chaque demande $k \in K$ soit transportée de son nœud origine o_k à son nœud destination d_k. Le véhicule V ne peut transporter qu'une demande à la fois. Il s'agit donc de trouver une tournée Γ dans G, la plus courte possible, qui commence et finit au dépôt et permettant à V de satisfaire toutes les demandes. Dans le **SCP** préemptif (**SCPP**), chaque charge peut être déchargée à n'importe quel nœud du graphe avant d'être rechargée. Ce processus peut être effectué plusieurs fois avant que la charge arrive au nœud destination. On parle de **SCP** asymétrique lorsque la fonction qui à tout arc (x, y) fait correspondre un coût $c(x, y)$ est asymétrique. Nous nous intéresserons au *Stacker Crane Problem* Préemptif Asymétrique (**SCPPA**).

2 Résultats structuraux

On appelle **arbre biparti ordonné** un arbre T tel que : d'une part, ses nœuds peuvent être ordonnés en deux classes A et B de telle manière que la classe A a ses fils dans la classe B et vice versa ; d'autre part, pour chaque nœud x dans T qui n'est pas une feuille l'ensemble des fils associés

* nouvelle adresse : lacroix@lamsade.dauphine.fr, Laboratoire LAMSADE, Université Paris-Dauphine, Place du Maréchal de Lattre de Tassigny

à x est ordonné linéairement et décrit sous forme de liste. On peut établir un *théorème de restriction* dont la signification est que l'on peut restreindre la recherche de solutions optimales pour le **SCPPA** à un sous-domaine, dont les éléments sont des tournées dotées de propriétés additionnelles, qui nous permettent de les représenter sous forme d'un arbre T biparti ordonné. Les demandes forment l'une des 2 classes et les sommets relais forment la seconde.

3 Heuristiques à base d'arbres pour le SCPPA

Les algorithmes décrits ici se déduisent facilement de la représentation à l'aide d'arbres des solutions optimales du **SCPPA**. Ils sont basés sur deux types d'opérateurs : opérateurs d'insertion et opérateurs de transformation locale. Les opérateurs d'insertion opèrent sur un arbre biparti ordonné T. Ces opérateurs insèrent une demande k dans T. L'algorithme 1 présente un algorithme de construction utilisant ces opérateurs. Il s'insère parfaitement dans un schéma de type Monte-Carlo. Les opérateurs de transformation locale modifient un arbre biparti ordonné T. L'algorithme 2 est un algorithme de descente utilisant ces opérateurs.

Algorithme 1 : **SCPPA**-Insertion

Définir aléatoirement un ordre ρ sur les demandes de K ;
$T = \{Depot\}$;
Pour $k \in K$ suivant l'ordre ρ **faire**
  Choisir un opérateur d'insertion I et le jeu de paramètres u associé tels que l'insertion de k dans T via $I(u)$ induise une augmentation du coût $Cout\_arbre(T)$ la plus faible possible ;
  Appliquer $I(u)$ à T ;

Algorithme 2 : **SCPPA**-Descente

Initialiser l'arbre T avec **SCPPA**-Insertion ;
Initialiser la valeur seuil filtrante H ;
stop $\leftarrow$ faux ;
tant que $stop = faux$ **faire**
  Rechercher (en utilisant le filtre H) un opérateur I et un jeu de paramètres u tels que l'application de $I(u)$ sur T réduise $Cout\_arbre(T)$ $(I1)$;
  si *la recherche a échouée* **alors** $H \leftarrow H/2$;
  si H *devient trop petit* **alors** stop $\leftarrow$ vrai ;

4 Conclusion

Nous avons traité ici un problème de Ramassage et Livraison et nous avons montré qu'il était possible de le transformer en un problème de construction d'arbre peu contraint, de telle sorte que nous puissons le résoudre grâce à des algorithmes efficaces de type glouton et descente.

Références

[1] G.N. Frederickson, M.S. Hecht, C.E. Kim. Approximation Algorithms for Some Routing Problems. SIAM Journal on Computing, 7 :178, 1978.