AG41 - Transbordement

Staine Florian

27 mai 2016

Chapitre 1

Objectif

L'objectif de ce projet est de résoudre un problème d'optimisation exacte, dérivé d'un problème de transport.

1.1 Problème

En plus de devoir transiter d'un fournisseur à un client, les ressources doivent passer par une plateforme de transbordement.

Chaque arc possède :
- une capacité maximale
- un coût fixe ainsi qu'un coût unitaire d'utilisation
- un temps de transport

Chaque plateforme possède :
- un coût unitaire de transbordement
- un temps de transbordement

Le coût total est constitué de la somme de plusieurs coûts :
En effet, si un arc transporte un ou plusieurs produit, son coût fixe sera pris en compte, en plus du coût unitaire multiplié par le nombre de produits transportés. Pour chaque produit transitant par une plateforme, un coût unitaire de transbordement sera ajouté au coût total.

Nous avons aussi d'autres contraintes que dans le problème initial.
Tout d'abord, les arc ont tous une capacité maximale, ce qui signifie que qu'il ne peut transiter plus de produit que la capacité maximale. Les arc sont utilisés au maximum une fois, ils ne peut pas y avoir des aller-retour pour transporter plus de ressources.
Un temps maximal de transport devra être respecté pour le transport de toutes les ressources. De plus, tous les produits partent au même moment et sont transportés en parallèles. Cela signifie que pour chaque cheminement emprunté pour atteindre les clients, les temps de transport plus le temps de transbordement doivent être inférieurs au temps maximum autorisé.

Chapitre 2

Modèle mathématique

2.1 Paramètres

n : nombre total de nœuds
n_f : nombre de nœuds fournisseurs
n_p : nombre de nœuds plates-formes
n_c : nombre de nœuds clients

b_i : demande ou disponibilité du nœud
g_i : Coût unitaire de transbordement
s_i : Temps de transbordement

$u_{i,j}$: Capacité de l'arc (i, j)
$c_{i,j}$: Coût fixe de l'arc (i, j)
$h_{i,j}$: Coût unitaire de l'arc (i, j)
$t_{i,j}$: Temps de transport de l'arc (i, j)

2.2 Variables

$x_{i,j}$: nombre de produit transportés par l'arc (i, j)
$y_{i,j,k}$: Nombre de produits transportés par la route (i, j) (j, k)

2.3 Objectif

$min\ z = \sum_{i=1,\ j=1,\ x_{i,j}\neq 0}^{n_f,\ n_p} (x_{i,j} \cdot h_{i,j} + c_{i,j} + x_{i,j} \cdot g_j) + \sum_{j=1,\ k=1,\ xi,j\neq 0}^{n_p,\ n_c}(x_{j,k} \cdot h_{j,k} + c_{j,k})$

2.4 Contraintes

$(C1) :\ \forall (i, j) \in n^2,\ x_{i,j} \leq u_{i,j}$
$(C2) :\ \forall i \in n,\ \sum_{j=1}^{n}(x_{i,j} - x_{j,i}) = -b_i$
$(C3) :\ \forall (i, j) \in n^2,\ \sum_{k=1}^{n_c} y_{i,j,k} + \sum_{k=1}^{n_f} y_{k,j,i} = x_{i,j}$
$(C4) :\ \forall (i, j, k) \in n^3,\ y_{i,j,k} \neq 0,\ t_{i,j} + t_{j,k} + s_j \leq T$

Chapitre 3

Résolution algorithmique

3.1 Algorithme utilisé

Pour résoudre ce problème, un algorithme de type Branch and Bound sera utilisé. Cela consiste à séparer en deux l'ensemble des solutions selon s'il ont fait un choix ou son contraire. Certains choix sont évités s'ils possèdent une borne minimal de l'objectif plus grand qu'une solution déjà existante.

3.2 Problèmes à résoudre

En premier lieu il est nécessaire de calculer une première solution. Cela permet d'éviter toutes les solutions plus mauvaises que cette première. Plus cette première solution sera proche de la solution finale, plus il sera facile de converger rapidement vers la solution optimale car un plus grand nombre de choix seront évités.

Ensuite, une borne minimum la plus élevée possible doit être définie pour permettre d'éliminer les solution moins efficace que la solution déjà existante. Tout le temps de résolution du problème est dépendant de ces deux algorithmes, qui sont le choix d'une première solution efficace et le calcul d'une borne inférieure.

Ensuite, il est possible d'éliminer certains choix lorsqu'il n'est pas possible de résoudre le problème à partir d'un sous-ensemble en solutions, cela permet encore un gain de temps non négligeable dans la résolution.

3.3 Méthodes utilisées

3.3.1 Solution initiale

Il n'y a pas à proprement parler de solution initialie qui est trouvée grace à une méthode différente.
Cependant, une sélection différente est faite sur les arcs selon si on a déjà trouvé une solution initiale ou pas.
En effet, les chemins sont choisis de manière à maximiser le nombre de ressouces transportés, ce qui permet de converger plus rapidement vers une solution initialie, même si celle-ci n'est pas d'excellente qualité.

Même si le principe de l'algorithme est évidement de couper le plus haut possible dans l'arbre des solutions pour éviter de perdre du temps dans des solutions loins de l'optimum, l'ordre des choix n'a pas d'incidence sur la meilleur solution trouvée du fait que toutes les possibilités sont envisagés puis testés.

Lorsqu'une solution initialie à été trouvée, l'algorithme se concentre sur la recherche des branches les moins chères permettant de se diriger vers la solution optimale.

3.3.2 Borne minimale

3.3.3 Solutions déduites

3.3.4 Bugs - Problèmes rencontrés

— Solution optimale non trouvée
—