Recherche rapide d'arbres de Steiner

Eric Bourreau
Rodolphe Giroudeau

David Aubert
Deguilhem Julien
Université de Montpellier II

20 jan 2015

1. Résolution de programmes linéaires

1.1. Génération de colonnes

1.1.1. Principe général

La génération de colonnes est une technique utlisée pour résoudre un programme linéaire en nombre réels, lorsque les variables sont trop nombreuses pour être toutes énumérées ou pour une résolution directe par un solveur. La génération de colonnes utilise en autres, la solution duale et les coûts réduits. [2]

1.1.1.a. Dualité

Pour un problème linéaire apellé *Primal*, le Dual est la "*transposée*" du *Primal*: les variables du Primal deviennent les constantes du Dual et vice et versa.
Le problème de maximisation du primal est le problème de minimisation du dual et réciproquement.

Exemple :

Problème Primal :
$Min{-x_1} + 4x_2$

$$\begin{cases} -x_1 \geq -7 \\ 3x_1 - 2x_2 \geq 4 \\ x_1 + x_2 \geq 5 \end{cases}$$

Problème Dual :
$Max - 7u_1 + 4u_2 + 5u_3$

$$\begin{cases} -u_1 + 3u_2 + u_3 \leq -1 \\ -2u_2 + u_3 = 4 \end{cases}$$

1.1.1.b. Coûts réduits

Pour résoudre un PL il faut une base. Tel que pour un problème de minimisation il y ait:

A$x = d$
$x \geq 0$

Ici, il est possible de transformer les contraintes d'inégalités en égalités avec l'ajout de variables d'écarts.

Pour chaque variables d'écarts, il y a un coût: c_j *(Coût associé à la variable d'écart j).* Son coût réduit est l'écart qu'il y a entre son coût et

1.1.1.c. Génération de colonnes

1.2. Branch and Price

1.2.1. Principe général

Un algorithme de branch and Price est l'équivalent d'un algorithme de Branch and Bound et de génération de colonnes.
La différence se fait par incrémentation. A chaque noeud, une résolution de PL. De ce fait on ne garantit pas à chaque noeud une solution optimal ou même valide.
La première étape d'un branch and price est de générer ce qui s'appele le ***Master Problem***.
De ce ***Master Problem*** devront être générés des ***sous-problèmes***.
Chaque ***sous-problèmes*** sera résolu et à chaque itération, une résolution de PL sera effectué et chaque solution entrainera une génération de colonne et sera rajouté à la solution.
Lorsque tout les ***sous-problèmes*** générés admettent une solution entièrement négative c'est la fin de l'algorithme.

1.2.2. exemple

Job*(i)*			
	1	2	3
Machine $i = 1$	5	7	3
Machine $i = 2$	2	10	5

i	C_i
1	5
2	8

Job*(i)*			
	1	2	3
Machine $i = 1$	3	5	2
Machine $i = 2$	4	3	4

Soit k_1 et k_2 l'ensemble des solutions possibles pour les machines 1 et 2 tel que $k_1 =$ (1,0,0), (0,1,0), (0,0,1), (1,0,1)

$k_2 =$ (1,0,0), (0,1,0), (0,0,1), (1,1,0), (0,1,1)

Le ***Master Problem*** pourra être representé comme ci-suit:
max z = $5y_1^1 + 7y_1^2 + 3y_1^3 + 8y_1^4 + 2y_2^1 {+} 10y_2^2 + 5y_2^3 + 12y_2^4 {+} 15y_2^5$

Voici une décomposition par job:
$y_1^1 + 0 + 0 + y_1^4 + 0 + 0 + y_2^4 + 0 = 1$ (u_1)
$0 + y_1^2 + 0 + 0 + 0 + y_2^2 + 0 + y_2^4 + y_2^5 = 1$ (u_2)
$0 + 0 + y_1^3 + y_1^4 + 0 + 0 + y_2^3 + 0 + y_2^5 = 1$ (u_3)

Ici au lieu d'exprimer les **Machines** par **Job**, est exprimé les **Job** par **Machines**. C'est le Dual.

References

[1] Dominique Feillet, *Résolution de problèmes de tournées par Branch and Price.* Laboratoire d'informatique, Avignon, Mars 2008.

[2] Hélène Toussaint, *Introduction au Branch cut and price et aux solveurs SCIP (Solving constraint integer programs).* Rapport de recherche LIMOS /RR-13-07, 19 avril 2013.

[3] Mads Kemhet Jepsen, *Column generation and Branch and Price.*

[4] Mohan Akella, Sharad Gupta, Avijit Sarkar, *Branch and Price column generation for solving huge integer programs.* Bufallo University (SUNY),