Nantes Université
Faculté des Sciences et des Techniques

Master informatique
Parcours “Optimisation en Recherche Opérationnelle (ORO)”

Optimisation discrète et combinatoire

Elements de solution au TP

Xavier GANDIBLEUX

25 novembre 2022

Remarque :
les écrits qui suivent sont issus de différents documents et n'ont pas fait l'objet d'une uniformisation des notations.

1 Quelques informations préalables tirées de l'algorithme de Balas

Le papier qui présente cet algorithme est :

E. Balas, An Additive Algorithm for Solving Linear Programs with Zero-One Variables, *Operations Research* 13, 4 (1965) 517-546.

Parmi les conditions initiales nécessaires pour appliquer l'algorithm :

— toutes les variables doivent être binaires;
— la fonction objectif doit être à minimiser;
— tous les coefficients de la fonction objectif doivent être non-négatifs;
— les variables doivent être organisées selon l'ordre des coûts croissants de la fonction objectif.

Du fait des particularités du problème et des conditions initiales, l'algorithme suit les règles suivantes :

— il est préféré de mettre toutes les variables à 0 pour obtenir la plus petite valeur de z
(arguments : la fonction est à minimiser; tous les coefficients sont non-négatifs)
— si il n'est pas possible de mettre toutes les variables à 0 sans violer une ou plusieurs contraintes, alors c'est la variable de plus petit index qui est mise à 1
(argument : variables organisées selon l'ordre des coûts croissants)

Les règles habillant le branch-and-bound :

— branchement (branching) :
une variable prend soit 1 ou 0 comme valeur
— évaluation (bounding) :
examine le coût de la solution immédiatement suivante la moins couteuse, qui pourrait conduire à une solution réalisable
  — si la variable d'indice v est mise à 1 :
  une solution réalisable pourrait être trouvée et la borne est obtenue avec $\sum_{j=1}^{v} c_j x_j$. Du fait de l'ordre sur les coûts, c'est le moins cher possible.
  — si la variable d'indice v est mise à 0 :
  Avec $v = 0$, avec des inégalités $\geq$, impossible avec ce branchement d'espérer trouver une solution réalisable. Seul en passant une variable à 1 on peut espérer une solution réalisable et la moins cher des variables est $v + 1$ et donc la borne est $\sum_{j=1}^{v} c_j x_j + c_{v+1}$.

Pour tester si une solution élaborée par la procédure de bounding est réalisable, il suffit de prendre les variables fixées et mettre à zero les variables libres, et enfin de vérifier les contraintes

  - — si toutes les contraintes sont satisfaites :
    la solution issue du bounding est réalisable, le noeud est sondé (optimalité) ;
  - — si au moins une contrainte est non réalisable :
    compte-tenu des variables fixées, impossible de satisfaire la contrainte quelque soit les valeurs des variables libres, le noeud est sondé (infaisabilité)
  - — sinon (au moins une contrainte n'est pas satisfaite) :
    la solution n'est pas réalisable, le noeud est ouvert.

— choix du noeud (selection) :
la stratégie en profondeur d'abord (noeud le plus profond) est appliqué

2 Quelques informations préalables tirées du chapitre de Martello & Toth

— A chaque noeud :
En supposant que b bins sont initialisés et avec c la capacité d'un bin, on maintient :

$$\left(\bar{c}_{i_1}, \ldots, \bar{c}_{i_b}\right)$$

les capacités résiduelles des bins triées par valeurs croissantes,
et on établit :

$$\bar{c}_{i_{b+1}} \equiv c_{b+1} = c$$

la capacité (nominale) du prochain bin à ouvrir

— Phase de branchement :
Affecte l'item libre j^* de plus grand poids au bin

$$i_s, \ldots, i_b, i_{b+1}$$

avec

$$s = \min\left\{h : 1 \leq h \leq b+1\ ,\ \bar{c}_{i_h} + w_{j^*} \leq c\right\}$$

— Sondage avec la borne L_1 :

$$L_1 = \left\lceil \frac{\sum_{j=1}^{n} w_j}{c} \right\rceil$$

3 Quelques informations préalables tirées de diverses autres sources

— L'ensemble des indices N en un noeud S se décompose en $N = N^0(S) \cup N^1(S) \cup N^F(S)$ avec
$N^0(S)$, indices des variables fixées à 0
$N^1(S)$, indices des variables fixées à 1
$N^F(S)$, indices des variables libres

4 Lecture de l'algorithme de Eilon & Christofides

— Cet algorithme est adapté du schéma énumératif général de Balas, 1965, pour résoudre des problèmes d'optimisation linéaires en variables binaires.

— L'arbre de décision est binaire. Un noeud a au plus 2 descendants, soit un item est assigné a un bin, soit un item n'est pas assigné à un bin.

— le calcul de $\bar{z}$ est obtenu avec BFD (best-fit-decreasing)

— n : nombre d'items
N : nombre de bins
$\rightarrow n \times N$ variables $x_{i,j}$ renumérotées $x_1, \ldots, x_{n \times N}$

— S : ensemble de k variables fixées à 0 ou 1
$S = N^0(S) \cup N^1(S)$
il reste donc $n \times N - k$ variables libres, qui peuvent être rassemblées dans $N^F(S)$

— soit un sous-problème constitué des variables $N^F(S)$
Si il est résolu :

1. les meilleures valeurs pour les variables libres sont déterminées et donc
$N^0(S)$, $N^1(S)$, $N^F(S)$ forment une solution qui minimise z
on peut donc mettre à jour éventuellement $\bar{z}$
2. pas de solution admissible atteignable

— comme les $V_j > 0$ alors la meilleure valeur des variables libres est 0

— si une solution x est non-réalisable (attention, les cas "sondé par impossibilité" et "sondé par dominance" ne sont pas considéré dans ce cas de figure) , elle vient avec un sous-ensemble $N^F(S)$

un élément de $N^F(S)$ est transféré soit sur $N^1(S)$, soit sur $N^0(S)$, formant un nouveau sous-problème,

cet élément est choisi tq l'item avec le plus grand poids est assigné au bin présentant la plus petite capacité résiduelle

— exclusion de parties complète de l'arbre (contenant des solutions non-réalisables)
calculer la relaxation linéaire du sous-problème → borne duale
si borne duale > borne primale $\bar{z}$ alors branche coupée
si existe pas de solution alors branche coupée

— si $N^F(S) = \emptyset$ plus de sous-problème dans la branche