Modélisation et Optimisation par les Graphes et la Programmation Linéaire

Cours 5 : PL en nombres entiers, méthodes par séparation évaluation

Patrice Perny

patrice.perny@lip6.fr

LIP6 – Sorbonne Université

Master d'informatique – M1 – MOGPL

Plan

- Exemple préliminaire
- Description générale des procédures par séparation et évaluation
- Applications à la PLNE

I) Exemple préliminaire

Exemple

Un problème d'affectation agents/tâches

	t_1	t_2	t_3	t_4
a_1	8	3	1	5
a_2	11	7	1	6
a_3	7	8	6	8
a_4	11	6	4	9

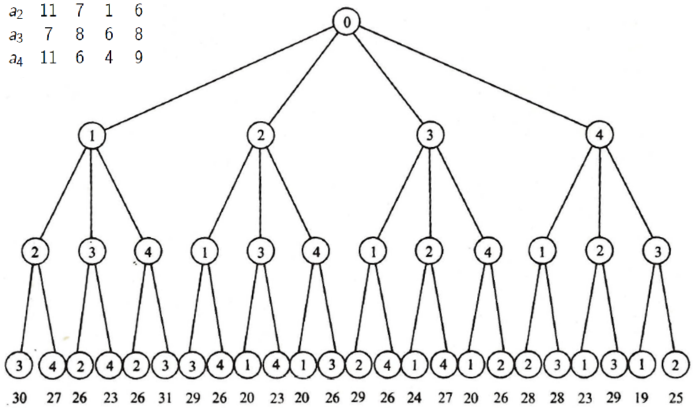
Méthode arborescente pour la résolution :

Au rang i, on choisit quelle tâche on affecte à l'agent i

- énumération complète
- énumération implicite et coupe

Enumération complète

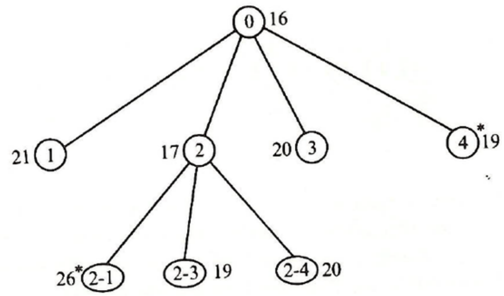
	t_1	t_2	t_3	t_4
a_1	8	3	1	5
a_2	11	7	1	6
a_3	7	8	6	8
a_4	11	6	4	9



Enumération implicite et coupes

	t_1	t_2	t_3	t_4
a_1	8	3	1	5
a_2	11	7	1	6
a_3	7	8	6	8
a_4	11	6	4	9

	1	2	3
2	11	7	1
3	7	8	6
4	11	6	4



II) Description générale des procédures de séparation et d'évaluation

Evaluations

S = ensemble des solutions potentielles

Problème à résoudre : étant donné une application $f : S \to \mathbb{R}$, déterminer $s^* \in S$ tels que :

$$(\mathcal{P}) \qquad f(s^*) = \min_{s \in S}\{f(s)\}$$

Définition

Etant donné un problème d'optimisation combinatoire $\mathcal{P}$, on dit qu'on sait évaluer le sous-ensemble $S' \subset S$ si on sait déterminer un réel $g(S')$ tel que : $g(S') \leq f(s)$ pour tout $s \in S'$. g est appellée évaluation (inférieure) de S' ou borne (inférieure).

Remarque : On considère que $g(S') = +\infty$ quand $S' = \emptyset$

Définitions relative à l'évaluation

Définition

Une évaluation $g(S')$ est dite exacte *si on connaît $s' \in S'$ tel que $g(S') = f(s')$*

Définition

Une évaluation $g'(S')$ est dite meilleure *que $g(S')$ si $g'(S') > g(S')$*

Sous-ensembles stérilisés

Définition

Un sous-ensemble $S' \subseteq S$ est dit stérilisé *dans les cas suivants :*

- *Il a été reconnu que S' est vide*
- *On connaît une solutions $s^* \in S$ au moins aussi bonne que toute solution de S', c'est-à-dire telle que $f(s^*) \leq g(S')$.*

Remarque : Lorsqu'un ensemble S' est stérilisé, il ne peut contenir de solution susceptible d'améliorer la meilleure solution courante. Il n'a donc pas besoin d'être analysé plus avant.

Séparation

Définition

Un sous-ensemble $S' \subseteq S$ est dit séparé *en les sous-ensembles $S'_1, S'_2, \ldots, S'_k$ si*

- $S'_i \subseteq S', \quad i = 1, 2, \ldots, k$
- $S'_1 \cup S'_2 \cup \ldots \cup S'_k = S'$

Remarques :

- la définition n'exclut pas le fait que certains des sous-ensembles S'_i puissent être vides
- souvent les S'_i forment une partition de S'

Sépartion et optimalité

Proposition

Soit $\mathcal{F}$ une famille de sous-ensembles de S dont l'union est égale à S alors :

- *Si $S' \in \mathcal{F}$ est séparé en $S'_1, S'_2, \ldots, S'_k$ alors $\mathcal{F}' = \mathcal{F} \cup \{S'_1, \ldots, S'_n\} \setminus \{S'\}$ est encore une famille de sous-ensembles de S dont l'union est S.*
- *Si tous les ensembles de la famille $\mathcal{F}$ ont une évaluation exacte et si S' est un sous-ensemble d'évaluation minimum, alors :*
  1. *si $g(S') < \infty$ la solution $s' \in S'$ telle que $f(s') = g(S')$ est une solution optimale de $\mathcal{P}$*
  2. *si $g(S') = \infty$ alors $S = \emptyset$*

Schéma des algorithmes par séparation et évaluation

Algorithm 1: Algorithme SE pour minimiser f

$v \leftarrow +\infty$; $\mathcal{F} \leftarrow \{S\}$;
while $\mathcal{F} \neq \emptyset$ **do**
  CHOIX : choisir un ensemble $S' \in \mathcal{F}$ qui n'a pas été évalué, si un tel S' existe, sinon choisir un $S' \in \mathcal{F}$ évalué ;
  if *S' n'a pas été évalué* **then**
    EVALUATION : associer à S' son évaluation $g(S')$ (borne inf);
    if $g(S') \geq v$ **then** $\mathcal{F} \leftarrow \mathcal{F} \setminus \{S'\}$;
    else
      if *l'évaluation est exacte* **then**
        $s^* \leftarrow arg\ \min_{s \in S'}\{f(s)\}$; $v \leftarrow f(s^*) = g(S')$; $\mathcal{F} \leftarrow \mathcal{F} \setminus \{S'\}$
      end
    end
  end
  else
    SÉPARATION : créer les sous-ensembles $S'_1, \dots, S'_k$; $\mathcal{F} \leftarrow \mathcal{F} \cup \{S'_1, \dots, S'_k\} \setminus \{S'\}$
  end
end

Stratégies pour la procédure de Choix

a) Selectionnner l'ensemble S' dont l'évaluation est la plus faible (heuristique pour explorer la solution la plus prometteuse)
b) Sélectionner l'ensemble S' le plus récemment créé (gestion en pile), parcours en profondeur d'abord (backtracking)
c) Sélectionner l'ensemble S' le plus anciennement créé (gestion en file), parcours en largeur d'abord

Avantage et inconvénient du parcours en profondeur :

- obtenir rapidement une solution réalisable ce qui est intéressant pour stériliser des sous-ensembles
- Ne tient pas compte de la fonction d'évaluation. On peut perdre du temps à explorer dans le détail des sous-ensembles ne contenant pas de solution très intéressante.

$\rightarrow$ en général on utilise un mixte des stratégies a) et b), par exemple en utilisant une évaluation en profondeur avec priorité au meilleur d'abord

Le cas d'une maximisation

Algorithm 2: Algorithme SE pour maximiser f

$v \leftarrow -\infty$; $\mathcal{F} \leftarrow \{S\}$;
while $\mathcal{F} \neq \emptyset$ **do**
  CHOIX : choisir un ensemble $S' \in \mathcal{F}$ qui n'a pas été évalué, si un tel S' existe, sinon choisir un $S' \in \mathcal{F}$ évalué ;
  if *S' n'a pas été évalué* **then**
    EVALUATION : associer à S' son évaluation $g(S')$ (borne sup) ;
    if $g(S') \leq v$ **then** $\mathcal{F} \leftarrow \mathcal{F} \setminus \{S'\}$;
    else
      if *l'évaluation est exacte* **then**
        $s^* \leftarrow arg\ \max_{s \in S'}\{f(s)\}$; $v \leftarrow f(s^*) = g(S')$; $\mathcal{F} \leftarrow \mathcal{F} \setminus \{S'\}$
      end
    end
  end
  else
    SÉPARATION : créer les sous-ensembles $S'_1, \dots, S'_k$; $\mathcal{F} \leftarrow \mathcal{F} \cup \{S'_1, \dots, S'_k\} \setminus \{S'\}$
  end
end

On développe en priorité les sous-ensemble S' qui ont la plus grande valeur $g(S')$ (évaluation supérieure)

III) PSE POUR LA PLNE

Exemple 1

$$\max\ z = 2x_1 + x_2$$

$$\mathcal{P}\left\{\begin{array}{ccccccc} 4 & x_1 & & & & \geq & 1 \\ & & & 4 & x_2 & \geq & 1 \\ 2 & x_1 & + & 2 & x_2 & \leq & 9 \end{array}\right.$$

$$x_1, x_2 \in \mathbb{N}$$

Solution réelle

$x_1^* = 4.25, x_2^* = 0.25$ et $z^* = 8.75$

Solution entière

$x_1^* = 3, x_2^* = 1$ et $z^* = 7$

Exemple 2

$$\max\ z = x_1 + 5x_2$$

$$\mathcal{P}\left\{\begin{array}{cccccc} x_1 & + & 10 & x_2 & \leq & 20 \\ x_1 & & & & \leq & 2 \end{array}\right.$$

$$x_1, x_2 \in \mathbb{N}$$

Solution réelle

$x_1^* = 2, x_2^* = 9/5$ et $z^* = 11$

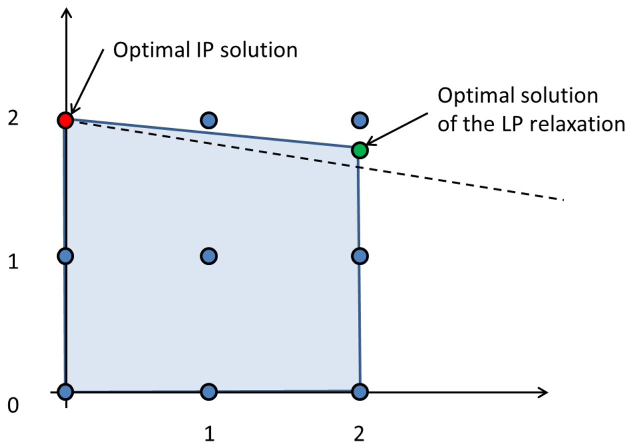
(2, 2) non réalisable

(2, 1) donne $z = 7$ sous-optimale

Solution entière

$x_1^* = 0, x_2^* = 2$ et $z^* = 10$

Représentation graphique de l'exemple 2



Résolution d'un PLNE par sép/éval

- Evaluation g donnée par une relaxation continue du problème
- algorithme par séparation évaluation

1. Initialisation : Calculer la solution $\hat{x}$ du PL en variables continues. Si entière FIN sinon 2.
2. Séparation : Séparer sur une variable $\hat{x}_k$ non entière de la solution courante $\hat{x}$ (par exemple celle de partie fractionnaire la plus grande) $\rightarrow$ ajout de l'une ou l'autre des contraintes :

   $$x_k \leq \lfloor \hat{x}_k \rfloor \quad \text{ou} \quad x_k \geq \lceil \hat{x}_k \rceil$$

   $\rightarrow$ on sépare le sommet courant en considérant deux nouveaux sommets correpondants à l'ajout de l'une ou l'autre contrainte
3. Evaluation : Calculer les valeurs des versions relaxées des deux sous-problèmes corespondants aux deux nouveaux sommets, choisir le sommet le plus prometteur parmi les différents sommets ouverts et aller en 2.

Exemple

$$\max \; z = 3x_1 + 2x_2$$

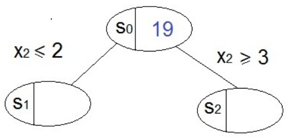
$$\mathcal{P} \left\{ \begin{array}{rcrcrcr} -2 & x_1 & + & 2 & x_2 & \leq & 7 \\ 2 & x_1 & + & 3 & x_2 & \leq & 18 \\ 9 & x_1 & - & 2 & x_2 & \leq & 36 \end{array} \right.$$

$$x_1, x_2 \in \mathbb{N}$$

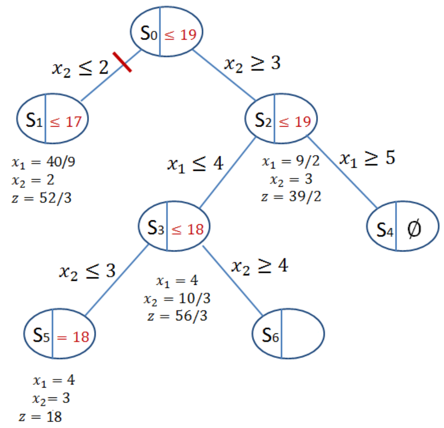
Résolution en variables continues :

$x_1 = 144/31 \simeq 4.65, \quad x_2 = 90/31 =\simeq 2.90$

$z = 612/31 \simeq 19.74$



Résolution par Séparation et Evaluation



IV) Programation linéaire en variables 0-1

Modélisation par la PL en var 0-1

Exemple : Un problèmes de selection de projets : comment utiliser au mieux un budget fixé pour financer des projets au sein d'une entreprise. 4 projets sont envisagés de manière non exclusive.

Projets	retour sur invest.	Année 1	Année 2	Année 3	Année 4
P_1	90000	15000	20000	20000	15000
P_2	40000	10000	15000	20000	5000
P_3	10000	10000	0	0	4000
P_4	37000	15000	50000	10000	10000
	ressources prév.	40000	50000	40000	35000

Modélisation par un PL en variables 0-1

$x_i \in \{0,1\}$: vaut 1 ssi on sélectionne le projet P_i
Modélisation du problème de sélection de projets :

$$\max\ z = 90x_1 + 40x_2 + 10x_3 + 37x_4$$

$$\mathcal{P}\left\{\begin{array}{rcrcrcrcl} 15x_1 & + & 10x_2 & + & 10x_3 & + & 15x_4 & \leq & 40 \\ 20x_1 & + & 15x_2 & + & & + & 50x_4 & \leq & 50 \\ 20x_1 & + & 20x_2 & + & & + & 10x_4 & \leq & 40 \\ 15x_1 & + & 5x_2 & + & 4x_3 & + & 10x_4 & \leq & 35 \end{array}\right.$$

$$x_1, x_2, x_3, x_4 \in \{0,1\}$$

La résolution donne pour solution optimale
$x^* = (1,1,1,0), z^* = 140$

Contraintes additionnelles

- Exclusion mutuelle : Je prends un et un seul projet parmi $\{P_1, P_2, P_3\}$: $x_1 + x_2 + x_3 = 1$
- Cardinalité : Je ne prends pas plus de 2 projets parmi $\{P_2, P_3, P_4\}$: $x_2 + x_3 + x_4 \leq 2$
- Si je prends P_1 je suis obligé de prendre P_3 : $x_1 \leq x_3$
- Je prends P_2 si et seulement si je prends P_4 : $x_2 = x_4$
- Je prends P_4 seulement si j'ai P_1 et P_2 : $x_4 \leq x_1$ et $x_4 \leq x_2$
- Je prends P_1 ou P_4 mais pas les deux : $x_1 + x_4 = 1$
- Si je prends P_1 et P_4 alors je dois prendre P_2 : $x_1 + x_4 - 1 \leq x_2$
- une des deux contraintes suivantes $f(x) \leq a$, $g(x) \leq b$ au moins doit être vérifiée :
  $f(x) \leq a + yM$, $g(x) \leq b + zM, y + z \leq 1, y, z \in \{0,1\}$

Tout cela s'écrit à l'aide de variables binaires et de contraintes linéaires !

Résolution d'un PL en 0-1

BB avec bornes supérieures calculées par résolution du problème relaxé en continu. Séparation du type : $x_i = 0$ ou $x_i = 1$

$$\max\ z = 9x_1 + 5x_2 + 6x_3 + 4x_4$$

$$\mathcal{P}\left\{\begin{array}{rcrcrcrcl} 6x_1 & + & 3x_2 & + & 5x_3 & + & 2x_4 & \leq & 10 \\ & & & & x_3 & + & x_4 & \leq & 1 \\ -x_1 & & & + & x_3 & & & \leq & 0 \\ & & -x_2 & + & x_3 & & & \leq & 0 \end{array}\right.$$

$$x_1, x_2, x_3, x_4 \in \{0,1\}$$

La résolution donne pour solution optimale
$x^* = (1,1,0,0), z^* = 14$

Résolution de problèmes de sac-à-dos

Applications : sélection de projets, de personnel, allocation de crédits, problèmes de configuration

Formulation :

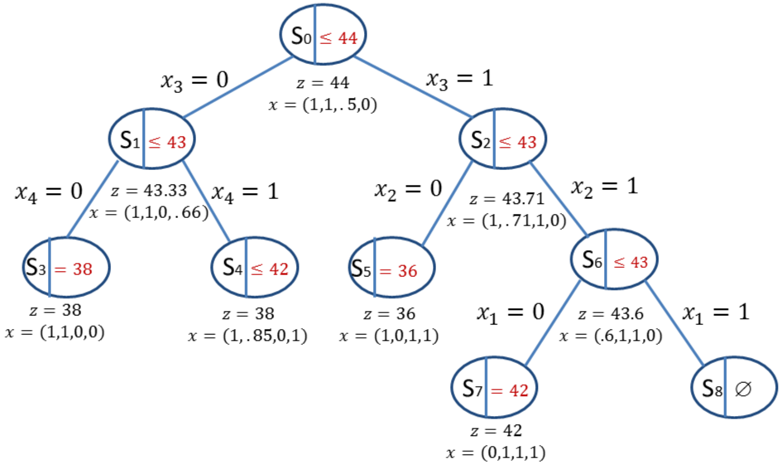
$$\begin{array}{l} \max \sum_{i=1}^n u_i x_i \\ \sum_{i=1}^n w_i x_i \leq W \\ x_i \in \{0,1\}, i = 1, \ldots, n \\ \\ w_i \geq 0, i = 1, \ldots, n \end{array}$$

Résolution par B&B :

On utilise la solution de la relaxation continue comme borne. Il est à noter que la version continue du problème se résout trivalement sans recours à la PL. Il suffit d'ordonner les variables selon les rapports u_i/w_i et de les affecter dans l'ordre induit par ces rapports, jusqu'à saturer la contrainte de poids (la dernière variable peut alors être fractionnaire).

Exemple

$$\max z = 16x_1 + 22x_2 + 12x_3 + 8x_4$$
$$5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14$$
$$x_i \in \{0, 1\}, i = 1, \ldots, 4$$



Solution : $x^* = (0, 1, 1, 1)$, $z^* = 42$