N° d'ordre : 10/ 2007-M/MT

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENE

FACULTE DE MATHEMATIQUES

MEMOIRE

présenté pour l'obtention du diplôme de **MAGISTER**

EN : MATHEMATIQUES

Spécialité **: Recherche Opérationnelle : Génie Mathématiques**

Par

FASS RAZIKA

SUJET

SUR QUELQUES PROBLEMES DE LA PROGRAMMATION A DEUX NIVEAUX

Soutenu publiquement le 09/10 / 2007, devant le Jury composé de :

BERRACHEDI Abdelhafid,	professeur, USTHB	président
MOULAI Mustapha,	Maitre de conférences, USTHB	Directeur de thèse
AIT HADDADEN, Hacène	professeur, USTHB	Examinateur
BOUROUBI Sadek,	Maitre de conférence, USTHB	Examinateur
CHERGUI Mohamed El Amine,	Chargé de Recherche, USTHB	Examinateur

.

REMERCIEMENTS

En premier lieu, je remercie Dieu le tout puissant de m'avoir prêté main pour la réalisation de ce travail.

À **Mr MOULAÏ Mustapha** j'exprime toute ma gratitude et ma reconnaissance pour la patience et la générosité avec les quelles il a su me guider durant les travaux de recherches. Je le remercie vivement pour son aide, son assistance et sa sympathie dont il a fait preuve.

Mes sincères remerciements vont aussi à **Mr BERRACHEDI Abdelhafid,** professeur à l'U.S.T.H.B, qui m'a fait l'honneur de présider le jury.

Mes remerciements s'adressent également à **Mr AÏT- HADDADENE Hacène,** professeur à l'U.S.T.H.B, **Mr S. BOUROUBI**, maître de conférences à l'U.S.T.H.B et **Mr CHERGUI**, chargé de recherche à l'U.S.T.H.B pour leur gratitude et leur dévouement qui ont permis d'honorer le jury.

DEDICACES

À *ceux qui m'ont donné la vie, je leurs témoigne mon amour et ma reconnaissance,*

À *maman qui m'a appris à être patiente pour surmonter les difficultés,*

À *mon père qui m'a appris que vouloir c'est pouvoir***,**

À *mes sœurs***, *Fatma et Malika***

À *mes frères en particulier AHCENE et KARIM,*

À *mes adorables nièces Hayet et Lynda,*

À *ma belle famille.*

A toutes les personnes qui m'ont connu et qui ont cru en moi.

A tous ceux qui m'ont soutenu de prés ou de loin

A mes amies , Nawel , Nabila , souad ,Sabrina,Sadia Salima, Farida , Fadéla ,Djimi , Karima et Khadidja.

Je dédie ce modeste travail.

Sommaire

Introduction

L’une des principales missions pour laquelle la recherche opérationnelle s’est vouée est d’aider à la prise de décision et à la gestion.

Les modèles traditionnels développés dans le cadre des méthodes quantitatives de gestion considéraient, en général, un critère unique pour lesquels il existe des solutions optimales. Les algorithmes mis au point consistent alors à définir un moyen d’atteindre, le plus rapidement possible, une telle solution. Cependant, dans de nombreux cas, cette modélisation des problèmes ne traduit pas exactement la réalité à appréhender.

Une autre façon de modéliser les problèmes décisionnels a vu le jour dans les années 70, il y a maintenant une trentaine d’années, permettant ainsi une représentation fidèle de la réalité. La nouveauté consiste à résoudre des problèmes de décision hiérarchique.

Contrairement à la recherche d’une solution optimale ou d’une solution Pareto optimale, on est amené à la recherche d’une décision hiérarchique. Cela revient à la recherche d’une solution optimale d’un problème d’optimisation dont les contraintes contiennent un autre problème d’optimisation appelé problème de la programmation à deux niveaux qui est un cas particulier de la programmation multi-niveaux.

La formulation de la programmation à deux niveaux a fait son apparition en 1973 dans un article de J. Bracken et J. Mcgill [10] bien que W. Candler et R. Norton [11] fussent les premiers à avoir utilisé les termes « programmation à deux niveaux et programmation multi niveaux. » et ce n’est qu’au début des années 80 que ce problème commença à recevoir plus d’attention. Inspirés par la théorie des jeux de H. Stackelberg [26] où les stratégies des joueurs sont dépendantes, donc hiérarchiques, plusieurs chercheurs ont contribué à l’enrichissement de la programmation à deux niveaux par de nouveaux concepts de base et de nouvelles méthodes de résolution de ce problème [8].
Comme plusieurs problèmes pratiques nécessitent une prise de décision hiérarchique, le champ d’application de la programmation à deux niveaux est très vaste, citons par exemple

le problème de la gestion des terres inondées, le problème de transport (Network design problem) O. Ben-Ayed, C. Blair, D .Boyce et L. Leblanc [6,7], de gestion [2], de planification [11].

La programmation à deux niveaux constitue donc un domaine très attractif pour les chercheurs. Elle peut être linéaire ou non linéaire avec aspect continu ou discret.

Notre premier objectif fut la maîtrise des concepts de base utilisés souvent en littérature et la réalisation d'une étude détaillée sur les divers problèmes PDN en passant en revue l'important de la littérature existante en vue d'aboutir à une méthodologie de résolution. Vu la rareté des travaux de recherche relatifs aux problèmes PDN linéaires discrets, notre attention s'est focalisée sur la mise au point d'un critère d'optimalité et d'une méthode d'optimisation en surmontant quelques difficultés de programmation en nombres entiers. Basé sur une technique de coupes planes, l'algorithme proposé permet de déterminer la solution optimale, lorsqu'elle existe, en nombre fini d'étapes. Une illustration numérique est présentée pour expliquer le déroulement de l'algorithme. Comme les problèmes PDN peuvent être non linéaires, une extension intéressante et possible du problème serait de résoudre le cas fractionnaire linéaire en variables entières.

Le mémoire commence par une introduction générale et se termine par une conclusion et une bibliographie suivie d'une annexe. Elle est composée de quatre chapitres dont le premier est principalement consacré à la programmation à deux niveaux où la terminologie et les concepts de base sont donnés pour comprendre les chapitres suivants. Au chapitre 2, on présentera la programmation à deux niveaux linéaire en présence de variables continues et quelques algorithmes de résolution qui constitue le maillon fort pour la recherche de la solution optimale. Le problème linéaire discret concernant la programmation à deux niveaux est exposé au chapitre 3 où on a relevé certaines difficultés rencontrées dans la méthodologie de résolution. Notre contribution est présentée au chapitre 4 sous forme d'une nouvelle méthode de résolution du problème de la programmation à deux niveaux en variables entières.

CHAPITRE 1

" Chaque fois que l'homme découvre la science, il se rend compte de son ignorance"

LA PROGRAMMATION A DEUX NIVEAUX

Définition 1.1 [17, 20, 24]

La programmation à deux niveaux (Bilevel Programming Problem) (BPP) comprend deux problèmes d'optimisation où la région de contraintes du problème du haut niveau est implicitement déterminée par un autre problème d'optimisation. Elle s'applique aux problèmes dont le procédé de décision est hiérarchique.

Le problème (BPP) permet la modélisation de situations où certaines données ou variables du problème ne sont connues qu'après avoir fixé d'autres.

La formulation mathématique du problème est la suivante *:*

$$(1.1) \qquad \begin{cases} \min\limits_{x} F(x,y) & (1) \\ \textit{sujet à } G(x,y) \leq 0 & (2) \\ \qquad \min\limits_{y} f(x,y) & (3) \\ \qquad \textit{sujet à } g(x,y) \leq 0 & (4) \end{cases}$$

Le problème s'écrit aussi :

$$\min_{(x,y)\in\Omega} F(x,y) \quad ou \quad y \in \arg\min_{\tilde{y}\in\Omega(x)} f(x,\tilde{y})$$

tels que Ω et $\Omega(\mathrm{x})$ seront definis par la suite

Remarque:

- $x \in IR^n, y \in IR^m, f, F : IR^n \times IR^m \to IR.$

- (1) et (2) constituent le problème 1^{er} niveau ou haut niveau.
- (3) et (4) forment le problème dit second niveau.
- Le vecteur y est déterminé après que le vecteur x soit fixé.
- Celui qui prend la décision en premier est appelé meneur, le deuxième est dit suiveur.

Définitions

Définition 1.1.1 : *L'ensemble* $\Omega = \{(x, y), \quad G(x, y) \leq 0, g(x, y) \leq 0\}$ *est dit ensemble de contraintes du problème PDN.*

Définition 1.1.2 : *Le domaine réalisable du problème second niveau au point x est l'ensemble définit par :* $\Omega(x) = \{y, \quad g(x, y) \leq 0\}$.

Définition 1.1.3 : *On appelle ensemble de solutions optimales du second niveau au point x, l'ensemble* $M(x) = \left\{y, \quad y \in \arg\min\left[f(x, \tilde{y}), \tilde{y} \in \Omega(x)\right]\right\}$.

Cet ensemble est aussi dit *ensemble des réactions rationnelles du suiveur*. L'ensemble *M(x)* est noté aussi dans la littérature parfois par P(x) ou par Ψ (x).

Définition 1.1.4 : *l'ensemble* $DI = \{(x, y) \in \Omega, \quad y \in M(x)\}$ *est dit domaine induit, il représente l'ensemble de solutions optimales du second niveau et qui vérifient les contraintes du premier niveau.*

Il est aussi noté par **IR (**inducible region). DI représente le domaine sur lequel le meneur optimise.

Définition 1.1.5 : Un point (x, y) est une solution optimale d'un programme à deux niveaux si :

— $(x, y) \in \Omega$,

— $y \in M(x)$,

— $\forall (z,w),(z,w)\in \Omega, w\in M(z), F(x,y)\le F(z,w)$.

Autrement dit, $(x,y)\in DI$ et $\forall (z,w)\in DI : F(x,y)\le F(z,w)$.

Remarques :

1- D'après les définitions précédentes le problème (1.1) peut être reformulé de la manière suivante :

$$\begin{aligned} &\min_{x,y} F(x,y) \\ &\textit{sujet à} \quad G(x,y)\le 0 \\ &\qquad\qquad\quad y\in M(x) \end{aligned}$$

2- le domaine induit est généralement non convexe et dans le cas de la présence de contraintes de haut niveau il peut devenir discontinu ou vide.

Exemple 1.1 : [26]

$$\begin{aligned} &\min_{x,y} x-2y \\ &\text{sujet à} \quad -x+3y-4\le 0 \\ &\qquad\qquad \min_{y} x+y \\ &\qquad\qquad \text{sujet à} \quad x-y\le 0 \\ &\qquad\qquad\qquad\quad -x-y\le 0 \end{aligned}$$

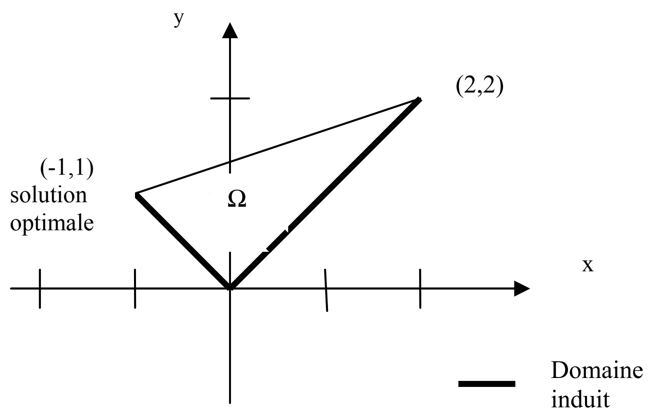


Figure de l'exemple 1.1

$\Omega = \{(x,y), -x+3y-4 \le 0, y \ge x, y \ge -x\}$ (ensemble des contraintes de la PDN),

$\Omega(x) = \{y, y \ge |x| \}$ (ensemble de solutions réalisables du second niveau au point x),

$M(x) = |x|$ (ensemble de solutions optimales du problème second niveau au point x).

Domaine induit : $DI = \{(x,y), -x+3y-4 \le 0, y \in M(x)\}$.

$$DI = \{(x,y), y = -x, -1 \le x \le 0\} \cup \{(x,y), y = x, 0 \le x \le 2\}.$$

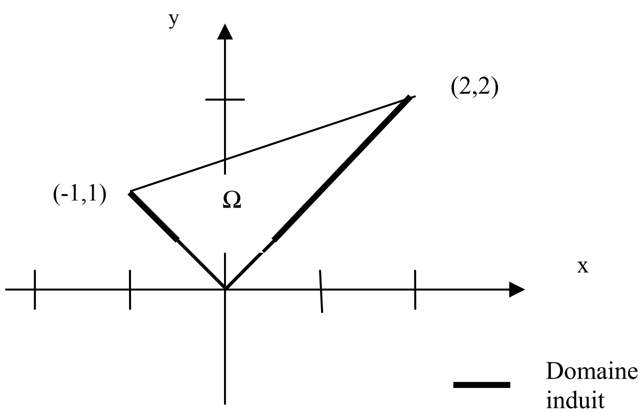
La solution optimale globale est $(x,y) = (-1,1)$.

Si on change les contraintes du problème haut niveau par

$$-x+3y-4 \le 0$$
$$-y+\frac{1}{2} \le 0$$

Le domaine induit devient :

$\left\{(x;y), y=-x, -1 \le x \le -\frac{1}{2}\right\} \cup \left\{(x,y), y = x, \frac{1}{2} \le x \le 2\right\}$ (ensemble discontinu).



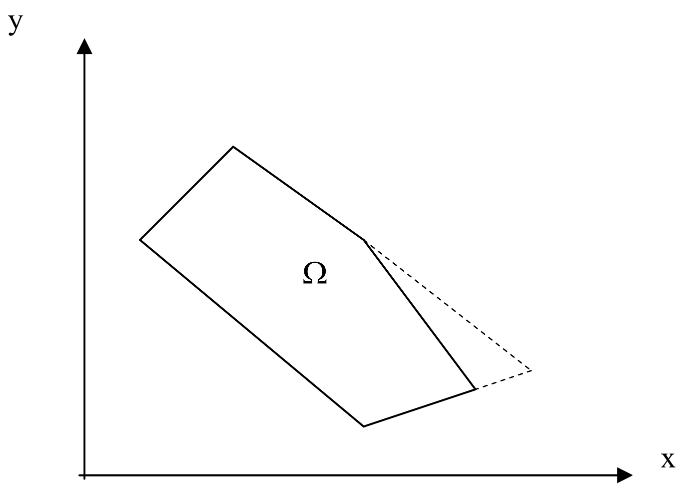
On a deux solutions locales $(-1,1),(2,2)$ et une solution globale $(-1,1)$.

Cet exemple illustre les différents états de la PDN tels que : la non convexité, la discontinuité du DI et l'existence de plusieurs minimums locaux.

Dans cet exemple le domaine induit est compact, condition suffisante pour l'existence d'un minimum global.

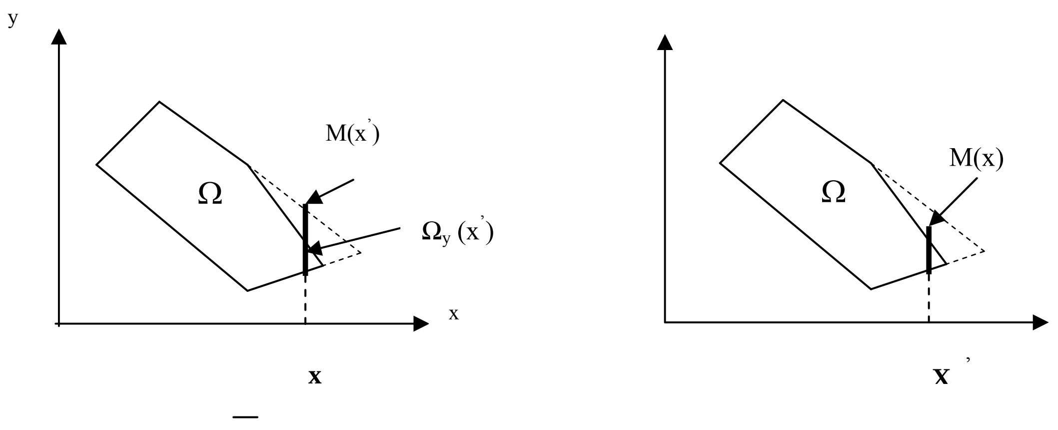
Exemple1.2 : [24]

Modèle1	Modèle 2
$\max\limits_{x,y} \; -x-4y$ $s.c\; 3x+2y \le 36$ $\max\limits_{y} \; y$ $s.c\; -x-y \le -8$ $-3x+2y \le 6$ $3x+4y \le 48$ $2x-5y \le 9$	$\max\limits_{x,y} \; -x-4y$ $s.c\; \max\limits_{y} \; y$ $s.c\; 3x+2y \le 36$ $-x-y \le -8$ $-3x+2y \le 6$ $3x+4y \le 48$ $2x-5y \le 9$



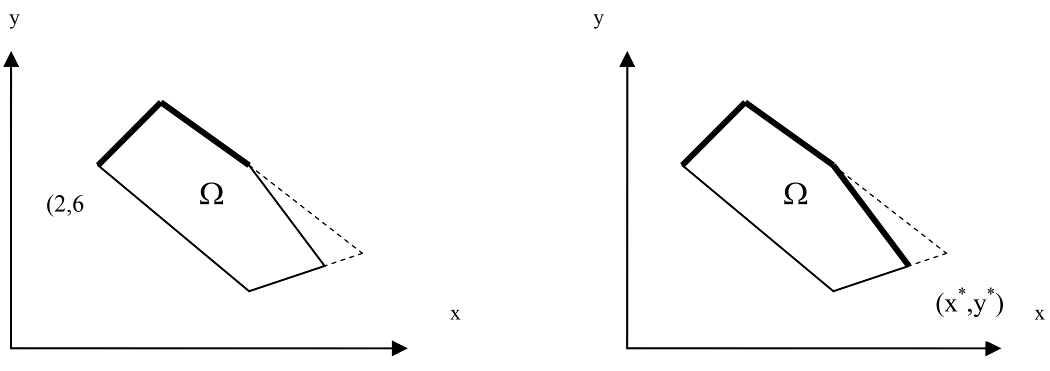
Ensemble des contraintes des modèles 1 et 2

Modèle1 Modèle2



$\Omega_y(x)$ et $M(x)$

Modèle1 Modèle2



Domaine induit et optimalité

1.2. Cas de non unicité de la solution du problème second niveau [5] :

Exemple1.3 :

Soit le problème de programmation à deux niveaux suivant :

$$\min_{x\geq 0} F = (2x_1 + 4x_2)y_1 + (3x_1 + x_2)y_2 \qquad (a)$$

$$\textit{sujet à} \quad x_1 + x_2 = 1 \qquad (b)$$

$$\min_{y\geq 0} f = (-x_1 - 3x_2)y_1 + (-4x_1 - 2x_2)y_2 \qquad (c)$$

$$y_1 + y_2 = 1 \qquad (d)$$

La solution optimale du problème second niveau ((c), (d)) en fonction de x est :

$$y(x) = \begin{cases} (1,0) & si \quad x_1 < \dfrac{1}{4}, \\ y_1 + y_2 = 1 & si \quad x_1 = \dfrac{1}{4}, \\ (0,1) & si \quad x_1 > \dfrac{1}{4}. \end{cases}$$

En remplaçant dans le problème haut niveau on obtient :

$$\begin{cases} \min_x F = \begin{cases} 2x_1 + 4x_2 & & si \quad x_1 < \dfrac{1}{4}, \\ 2y_1 + \dfrac{3}{2} & (0 \leq y_1 \leq 1) & si \quad x_1 = \dfrac{1}{4}, \\ 3x_1 + x_2 & & si \quad x_1 > \dfrac{1}{4}, \end{cases} \\ \qquad \textit{sujet à} \quad x_1 + x_2 = 1, \quad x_1 \geq 0, \quad x_2 \geq 0. \end{cases}$$

Pour $x_1 = \dfrac{1}{4}$, F n'est pas bien définie, la plus petite valeur de la fonction objectif est $F = \dfrac{3}{2}$, cette valeur est atteinte dans le cas ou le meneur choisit $(x_1, x_2) = \left(\dfrac{1}{4}, \dfrac{3}{4}\right)$ et le suiveur répondra avec $(y_1, y_2) = (0,1)$.

Exemple1.4 : [16]

Soit le problème du second niveau: $\Psi(y) = \arg\min_{x}\{-xy : \quad 0 \le x \le 1\}$

et considérons le problème biniveau suivant: $\min_{y}\left\{x^2 + y^2 : x \in \Psi(y), -1 \le y \le 1\right\}$.

Le problème dessus peut s'écrire de la manière suivante:

$$\begin{aligned}
&\min_{y} x^2 + y^2 && (1)\\
&\textit{sujet à } -1 \le y \le 1 && (2)\\
&\qquad \min_{x} -xy && (3)\\
&\qquad \textit{sujet à } \; 0 \le x \le 1 && (4)
\end{aligned}$$

(1) et (2) représente le problème haut niveau ;

(3) et (4) représente le problème second niveau.

Le domaine optimal du problème second niveau (3) et (4) est :

$$\Psi(y) = \begin{cases} \{1\} & si \quad y > 0, \\ \{0\} & si \quad y < 0, \\ [0,1] & si \quad y = 0. \end{cases}$$

En remplaçant dans la fonction objectif du problème haut niveau on aura :

$$F(x(y), y) = \begin{cases} 1 + y^2 & si \quad y > 0, \\ y^2 & si \quad y < 0, \\ \in [0,1] & si \quad y = 0. \end{cases}$$

Pour y = 0, la valeur optimale de la fonction objectif est égale à zéro (0), mais cette valeur est atteinte seulement dans le cas ou le suiveur choisit x = 0. Si ce n'est pas le cas le problème n'a pas de solution.

Notons que le meneur n'est pas autorisé à forcer le suiveur à faire un choix, donc il ne peut pas connaître la valeur de sa fonction objectif avant que le suiveur ait transmis son choix.

Dans les cas pareils le meneur peut utiliser deux approches pour prévoir la valeur de sa fonction objectif.

2.1 - Cas optimiste :

Le meneur peut utiliser cette approche dans le cas où il y a coopération avec le suiveur i.e, que ce dernier choisira une solution qui sera la meilleure pour le meneur.

Soit $\varphi_0(y) = \min_x \{F(x,y), x \in \Psi(y)\}(1.1)$ la valeur optimiste de la fonction objectif du problème haut niveau.

La position optimiste du problème de programmation à deux niveaux sera :

$\min_y \{\varphi_0, G(y) \leq 0\}(1.2)$. Le couple $(x_0(y_0), y_0)$, tel que y_0 solution de (1.2) et $x_0(y_0)$ solution de (1.1) pour $y = y_0$, est une solution optimale du problème :

$$\min_{x,y} \{F(x,y), G(y) \leq 0, x \in \Psi(y)\}.$$

Pour un problème donné : $\min\{\alpha(x):\ \beta(x) \leq 0,\ \delta(x) = 0\}$ avec des contraintes d'égalités et d'inégalités $(\alpha,\ \beta_i,\ \delta_j :\ R^n \to R, i = 1,...p, j = 1,...q)$, les contraintes de qualifications de Mangasarian- Fromowitz (MFCQ) au point x_0 s'écrivent :

1) il existe une direction d tel que: $\nabla\beta_{\mathrm{i}}(x_0)d < 0 \quad \forall i \ \ \beta_i(x_0) = 0,\ \nabla\delta_j(x_0)d = 0 \quad \forall j$

2) les gradients $\{\nabla\delta_{\mathrm{j}}(x_0), \forall j\}$ sont linéairements indépendants.

Théorème : [16]

Si l'ensemble $\{(x,y), g(x,y) \leq 0, G(y) \leq 0\ \}$ *est non vide et compact et* $\forall\, y$ *tel que* $G(y) \leq 0$ *(MFCQ) est satisfaite alors le problème (1.2) possède une solution optimale.* ***(C)***

2.2- Cas pessimiste :

Dans le cas de non coopération du suiveur et du meneur, ce dernier peut estimer les dommages dus à un mauvais choix du suiveur,

On a : $\min_y \{\varphi_p(y) : G(y) \leq 0\}\ (1.3) \quad ou \quad \varphi_p = \max_x \{F(x,y), x \in \Psi(y)\}(1.4)$

La valeur de la fonction (1.4) représente la mauvaise valeur de la fonction objectif du problème haut niveau.

Théorème [16]

Soit l'application univoque Ψ inférieurement semi continu en chaque point y tel que G(y)≤ 0 et les conditions du théorème précédent sont satisfaites alors le problème (1.3) possède une solution optimale.

Rappel :

ψ est dit inférieurement semi continu en $y = y_0$ si pour chaque ensemble ouvert A avec $\psi(y_0) \cap A \neq \emptyset$ il existe un voisinage ouvert B de y_0 avec $\psi(y) \cap A \neq \emptyset \quad , \forall y \in B$.

1.3 Ordre de la décision [5] :

Exemple 1.5

$$\min_x \left\{ F = \begin{bmatrix} x_1 \, x_2 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 4 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} : x_1 + x_2 = 1 \; ; x_{1;} \quad , x_2 \geq 0 \right\} \quad (1)$$

sujet à

$$\min_y \left\{ f = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} -1 & -4 \\ -3 & -2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} : y_1 + y_2 = 1, \quad y_1 \geq 0, \quad y_2 \geq 0 \right\} \quad (2)$$

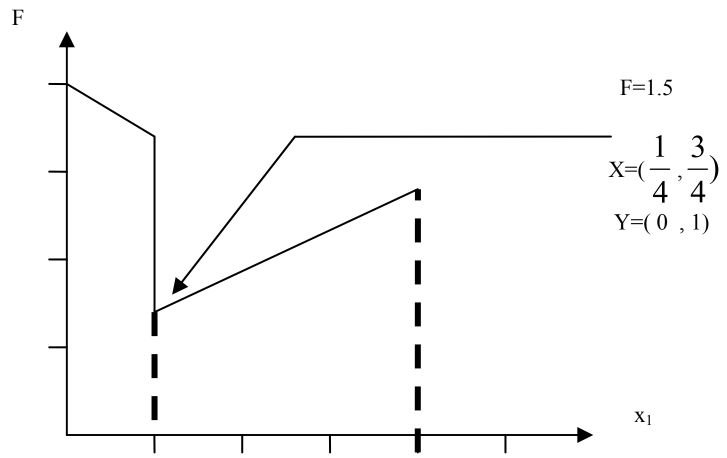
La solution du problème second niveau (2) (comme étant une fonction de x est :

$$Y(x) = \begin{cases} (1,0) & si \quad x_1 < \dfrac{1}{4}, \\ y_1 + y_2 & si \quad x_1 = \dfrac{1}{4}, \\ (0,1) & si \quad x_1 > \dfrac{1}{4}. \end{cases}$$

En remplaçant ces valeurs dans la fonction objectif du meneur on obtient :

$$\min_x F = \begin{cases} 2x_1 + 4x_2 & & si \quad x_1 < \dfrac{1}{4}, \\ 2y_1 + \dfrac{3}{2} & (0 \leq y_1 \leq 1) & si \quad x_1 = \dfrac{1}{4}, \\ 3x_1 + x_2 & & si \quad x_1 > \dfrac{1}{4}, \end{cases}$$

$$\text{sujet à} \quad x_1 + x_2 = 1, \quad x_1 \geq 0, \quad x_2 \geq 0.$$



Solution optimale du meneur

Par la suite on inverse l'exemple précédent : i.e, on permute les rôles du meneur et du suiveur.

$$\min_y f = -\left(x_1 + 3x_2\right)y_1 - \left(4x_1 + 2x_2\right)y_2 \qquad (a)$$

sujet à

$$y_1 + y_2 = 1, \quad y_1 \geq 0, \quad y_2 \geq 0 \qquad (b)$$

$$\min_x F = \left(2y_1 + 3y_2\right)x_1 + \left(4y_1 + y_2\right)x_2 \qquad (c)$$

sujet à

$$x_1 + x_2 = 1, \quad x_1 \geq 0, \quad x_2 \geq 0 \qquad (d)$$

On aura :

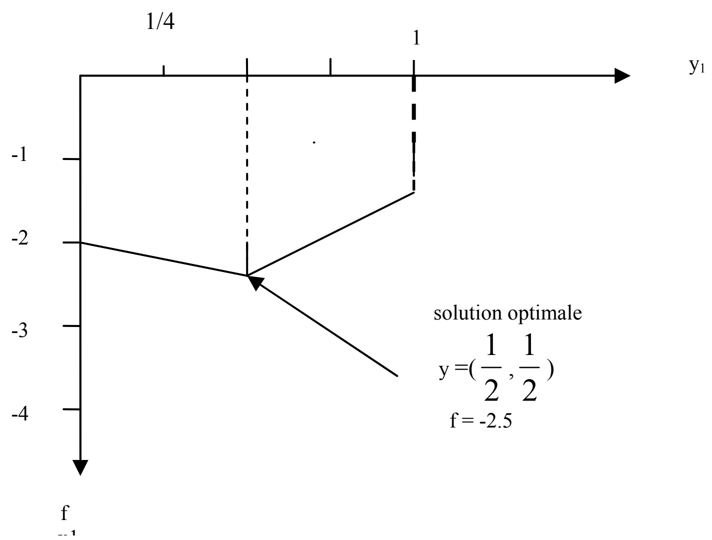
La solution optimale du problème second niveau (en tant que fonction de y) est :

$$X\left(y\right) = \begin{cases} (1,0) & si \quad y_1 > \dfrac{1}{2}, \\ x_1 + x_2 = 1 & si \quad y_1 = \dfrac{1}{2}, \\ (0,1) & si \quad y_1 < \dfrac{1}{2}. \end{cases}$$

En remplaçant ces valeurs dans le problème haut niveau on obtient :

$$\min_y f = \begin{cases} -y_1 - 4y_2 \quad , & y_1 > \dfrac{1}{2} \ , \\ -(3-2x_1)y_1 - (2x_1+2)y_2 \ , & y_1 = \dfrac{1}{2} \ , \\ -3y_1 - 2y_2 \quad , & y_1 < \dfrac{1}{2} \ , \end{cases}$$

$$\text{sujet à} \quad y_1 + y_2 = 1, \quad y_1 \geq 0, \quad y_2 \geq 0 \ .$$



solution optimale du suiveur

	Example1	Example2
Solution(x)	(1/4 , 3/4)	$x_1 + x_2 = 1$
Coût(F)	1,5	2,5
Solution(y)	(0,1)	(1/2 ,1/2)
Coût (f)	-2,5	-2,5

On remarque que les résultats des deux exemples sont différents.

Donc :

- L'ordre dans lequel la décision est prise est important.
- Les rôles du meneur et du suiveur ne sont pas interchangeables.

CHAPITRE 2

"La science sans conscience n'est que ruine de l'âme"

LA PROGRAMMATION A DEUX NIVEAUX LINEAIRE EN VARIABLES CONTINUES

2.1 Introduction

La majorité des travaux de recherche concernant le problème (BPP) était consacré au cas linéaire. Plusieurs méthodologies d'approches ont vu le jour dont certaines sont devenues célèbres par l'efficacité de leurs algorithmes de résolution. Dans ce chapitre, nous présentons les algorithmes les plus connus.

La programmation à deux niveaux linéaire (BLPP) (Bilevel Linear Programming Problem) peut être formulée de la manière suivante :

$$
(2.1)\quad
\begin{array}{ll}
\min\limits_{x\in X} F(x,y) = c_1 x + d_1 y & \text{a} \\
\text{sujet à} \quad A_1 x + B_1 y \le b_1 & \text{b} \\
\qquad \min\limits_{y\in Y} f(x,y) = c_2 x + d_2 y & \text{c} \\
\qquad \text{sujet à} \quad A_2 x + B_2 y \le b_2 & \text{d}
\end{array}
$$

$x \in X \subset IR^n, y \in Y \subset IR^m, f, F : X \times Y \to IR, A_1 \in IR^{P\times n}, A_2 \in IR^{q\times n}, B_1 \in IR^{p\times m}$, $B_2 \in IR^{q\times m}, c_1, c_2 \in IR^n, d_1, d_2 \in IR^m, b_1 \in IR^p, b_2 \in IR^q$.

Remarque :

Quand le meneur choisit un vecteur *x*, le 1er terme de la fonction objectif du problème second niveau devient une constante. Donc, on peut remplacer *f(x, y)* par *f(y)*.

Définitions

a - *ensemble de contraintes d'un problème de programmation à deux niveaux linéaire (BLPP)*

$$\Omega = \{(x,y), \quad x \in X,\ y \in Y,\ A_1x + B_1y \le b_1, \quad A_2x + B_2y \le b_2\}.$$

b - *ensemble réalisable du problème du second niveau pour chaque* $x \in X$ *:*

$$\Omega(x) = \{y \in Y, \quad B_2y \le b_2 - A_2x\}.$$

c- *projection de* Ω *sur l'espace de décision du meneur :*

$$\Omega(X) = \{x \in X, \exists y \in Y, \quad A_1x + B_1y \le b_1, \quad A_2x + B_2y \le b_2\}..$$

d- *ensemble de réactions rationnelles du suiveur pour chaque* $x \in X$

$$M(x) = \left\{y \in Y, \quad y \in \arg\min\left[f\left(x,\hat{y}\right), \hat{y} \in \Omega(x)\right]\right\}.$$

e- ensemble induit :

$$DI = \{(x,y), \quad (x,y) \in \Omega,\ y \in M(x)\}.$$

Pour s'assurer que le problème (2.1) est bien posé ; on suppose que Ω est non vide et compact. De plus, pour chaque décision prise du meneur, le suiveur aura toujours une réponse i.e, $M(x)$ est non vide.

Exemple 2.1.1 :

$$\begin{aligned}
&\min_{x\ge 0} F(x,y) = x - 4y \\
&\textit{sujet à} \quad \min_{y\ge 0} f(y) = y \\
&\qquad\qquad \textit{sujet à} \quad -x - y \le -3 \\
&\qquad\qquad\qquad\qquad -2x + y \le 0 \\
&\qquad\qquad\qquad\qquad 2x + y \le 12 \\
&\qquad\qquad\qquad\qquad -3x + 2y \ge -4
\end{aligned}$$

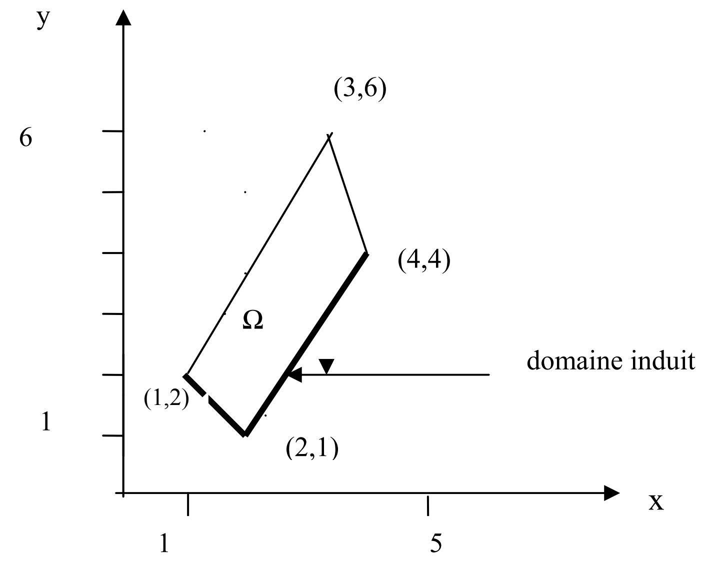


Figure 2.1.1

- Le polyèdre de la figure (2.1.1) représente l'ensemble des contraintes de l'exemple (2.1.1)
- Pour un x fixé tel que $x \in \Omega(X) = \{x, \quad 1 \leq x \leq 4\}$, l'ensemble réalisable du suiveur, $\Omega(x)$ est l'ensemble des points en ligne verticale contenues dans le polyèdre.

- Comme la fonction objectif du problème second niveau de cet exemple est de minimiser y, alors l'ensemble des réactions rationnelles du suiveur au point x est la valeur la plus petite parmi ces points.
- On a : $\max\left\{3 - x, \dfrac{3x-4}{2}\right\} \leq y \leq \min\{2x, 12-2x\}$

  Donc : $M(x) = \max\left\{3 - x,\ 3x - \dfrac{4}{2}\right\}$
- Le domaine induit est la partie en gras du graphe représenté par la figure 2.1.1
- La solution optimale de cet exemple est le point $(x^*, y^*) = (4,4), F^* = -12, f^* = 4$.

Le meneur possède une meilleure solution au point (3,6) mais ce point n'appartient pas

au domaine induit i .e , si le meneur choisit, x =3 le suiveur répondra par y =2.5
et on aura $F\left(3,2.5\right)$ = - 7 et $f\left(2.5\right)=2.5$.

C'est une meilleure solution pour le suiveur mais pas pour le meneur.

Remarques :

1- La solution optimale n'est pas Pareto optimale. Savard et Marcotte [28] ont montré que dans le cas ou les seconds termes des fonctions objectifs du meneur et du suiveur sont colinéaires ($d_1 = \alpha d_2$, $\alpha > 0$) ; alors il n'y a pas de garanties que la solution soit optimale.

2- La solution optimale est un point extrême du domaine induit qui est aussi un point extrême du domaine Ω.

3- Le domaine induit est composé de contraintes linéaires par morceau qui constituent aussi les faces de **Ω**.

2.2 Propriétés théoriques

Jeroslow [21] était le premier a démontré que la programmation linéaire à deux niveaux est un problème NP – dur. Bard [3] le démontre aussi en réduisant le problème de maximisation d'une fonction convexe quadratique sur un polyèdre à un problème de max-min.
Ce résultat a été renforcé par Hansen et al. [20] par la proposition suivante :

Proposition 2.2.1 [5] : *Le problème linéaire max-min est NP-dur.*

Leurs démonstration est basée sur le problème de Kernel prouvé par Chvatal [20] que c'est un problème NP –dur.

Corollaire2.2.1 [5]: *Le problème de programmation à deux niveaux linéaire est NP- dur.*

Corollaire 2.2.2 [5] :
Le problème de programmation linéaire à deux niveaux (2.1) est équivalent à minimiser F sur un ensemble réalisable constitué de contraintes d'égalité linéaire par morceau.

En général, comme on minimise une fonction linéaire $F = c_1x + d_1y$ sur DI et comme F est bornée sur $\Omega\left(\min\left\{c_1x + d_1y,\ (x, y) \in \Omega\right\}\right)$, alors on peut donner le corollaire suivant :

Corollaire2.2.3 [5] :

La solution d'un problème de programmation linéaire à deux niveaux est un point extrême du domaine induit noté DI

Ce résultat a été prouvé par Bialas et Karwan .

Théorème 2.2.2 [5]:

La solution (x, y*) d'un problème de programmation à deux niveaux linéaire est un point extrême de* Ω.

Corollaire 2.2.4 [5]

Si x est un point extrême de DI alors c'est un point extrême de Ω.

Remarque :

D'après le corollaire (2.2.2) et comme le domaine induit est non convexe en général, l'ensemble des solutions optimales du problème (2.1), dans le cas de non unicité de la solution du problème second niveau, n'est pas nécessairement convexe ; on peut le constater en remplaçant la fonction objectif du meneur de l'exemple précédent (2.2.1) par $F = 2x - 3y$ et on aura (1,2) et (4,4) comme solutions optimales mais aucun point appartenant à la ligne qui les joignent n'est dans le domaine induit.

Soient $u \in R^q, v \in R^m$ les variables duales associées aux contraintes (2.1d) et $y \geq 0$.

Proposition 2.2.2 [5] :

Une condition nécessaire pour que $\left(x^*, y^*\right)$ soit une solution du programme linéaire à deux niveaux (2.1) est l'existence d'un vecteur u^* et v^* *tel que : $\left(x^*, y^*, u^*, v^*\right)$ est une solution du programme :*

$$
\textbf{(2.2)} \quad
\begin{array}{ll}
\min\limits_{x} c_1 x + d_1 y & (a) \\
\text{sujet à } \; A_1 x + B_1 y \leq b_1 & (b) \\
\qquad uB_2 \quad - v = -d_2 & (c) \\
\qquad u\left(b - A_2 x - B_2 y\right) + vy = 0 & (d) \\
\qquad A_2 x + B_2 y \leq b_2 & (e) \\
\qquad x \geq 0, y \geq 0, u \geq 0, v \geq 0. & (f)
\end{array}
$$

Ce programme est obtenu en remplaçant le problème second niveau par les conditions de *Kuhn et Tuker*. Cette formulation a jouée un rôle déterminant dans le développement des algorithmes de recherche des solutions optimales.

Rappel [16] :

Soit le problème de programmation à deux niveaux suivant :

$$
\begin{array}{l}
\min\limits_{x} F\left(x, y\right) \\
\textit{sujet à } G\left(x, y\right) \leq 0 \\
\qquad \min\limits_{y} f\left(x, y\right) \\
\qquad \textit{sujet à } \; g\left(x, y\right) \leq 0
\end{array}
$$

Les conditions de Kuhn et Tucker du problème second niveau sont :

$$
\begin{array}{l}
\nabla_y f\left(x, y\right) + \lambda^T \nabla_y g\left(x, y\right) = 0 \\
g\left(x, y\right) \leq 0, \;\; \lambda \leq 0, \quad \lambda^T g\left(x, y\right) = 0
\end{array}
$$

Les conditions de Kuhn et Tucker sont des conditions nécessaires d'optimalité et sont aussi des conditions suffisantes dans le cas d'un problème convexe.

2.2 Algorithmes pour la programmation à deux niveaux linéaire [5]:

Plusieurs algorithmes ont été développés pour la résolution d'un programme linéaire à deux niveaux. Dans ce chapitre nous présentons les plus importants.

En général, il existe quatre approches pour résoudre (2.1) :

1) La première approche est basée sur le corollaire (2.1.1) et le théorème (2.2.2). Candler et Townsely [12] étaient les premiers à développer un algorithme de recherche d'une solution optimale. Bialas et Karwan [8] ont proposé aussi un algorithme que nous présenterons dans ce chapitre.
2) La deuxième est connue par l'approche de Khun et Tucker basée sur (2.2) et qui utilise la méthode de séparation et évaluation après suppression de la contrainte (2.2d) ; les différentes méthodes proposées utilisent de divers techniques pour s'assurer que la contrainte (2.2d) est satisfaite [4, 18, 21, 23].
3) La troisième approche se base sur la méthode des pénalités. Aiyoshi et Shimizu [27] transforment le problème second niveau en utilisant la méthode barrière, ensuite le problème haut niveau muni de quelques conditions est résolu pour des valeurs (paramètre barrière) décroissantes. Pour garantir la convergence, la fonction objectif du suiveur doit être strictement convexe.
4) La quatrième catégorie d'algorithmes essaye d'obtenir des informations sur les gradients du second niveau dans le but de calculer les dérivées directionnelles. Toutes les procédures supposent que : *y(x)* est localement unique pour chaque x ; *y(x)* est continûment différentiable.

2.3.1 *K*th - Best Algorithm [5]

Dans le cas d'un domaine induit borné et d'un ensemble de solutions optimales du second niveau univoque ; l'algorithme permet le calcul d'une solution optimale globale du problème :

$$\begin{aligned} &\min_{x\in X} F(x,y) = c_1 x + d_1 y \\ &\text{sujet à} \quad A_1 x + B_1 y \le b_1 \\ &\qquad\qquad \min_{y\in Y} f(x,y) = c_2 x + d_2 y \qquad\qquad (2.1) \\ &\qquad\qquad \text{sujet à} \quad A_2 x + B_2 y \le b_2 \end{aligned}$$

Soient $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, des solutions réalisables ordonnées du problème d'optimisation linéaire suivant : $\min\{c_1 x + d_1 y, \quad (x,y)\in\Omega\}$ (2.3) tel que :

$$c_1 x_i + d_1 y_i \le c_1 x_{i+1} + d_1 y_{i+1} \quad i = 1,\ldots,n-1$$

Donc, résoudre le problème (2.1) est équivalent à trouver l'indice i dans K^* :

$K^* = \{i \in \{1,...,n\}, (x_i, y_i) \in DI\}$. IL s'agit de trouver le k^{ème} meilleur point extrême du problème (2.1) et cela se fera par la procédure suivante :

Algorithme :

Etape1 : Poser $i := 1$. Rrésoudre le problème (2.3) avec la méthode du simplexe pour obtenir la solution optimale (x_1, y_1).

Soit $W := \{(x_1, y_1)\}$ et $T := \varnothing$. Aller à l'étape 2

Etape2 : résoudre le problème du second niveau $\min\{d_2 y,\ y \in M(x_i)\}$ (2.4) avec la méthode du simplexe borné afin de vérifier si la solution appartient à l'ensemble $M(x_i)$

Soit $\tilde{y}$ une solution optimale de (2.4) :

1. Si $\tilde{y} = y_i$, Stop. (x_i, y_i) est la solution optimale globale de (2.1)

2. Sinon, aller à l'étape 3.

Etape3 : Soit W_i l'ensemble de sommets adjacents à (x_i, y_i) tel que $(x, y) \in W_i$

$c_1 x + d_1 y \geq c_1 x_i + d_1 y_i$.

Poser $T := T \cup \{(x_i, y_i)\}$ et $W := (W \cup W_i) / T$. Aller à l'étape4

Etape4 : Faire $i := i + 1$. Choisir (x_i, y_i) tel que :

$c_1 x_i + d_1 y_i = \min\{c_1 x_i + d_1 y_i : (x, y) \in W\}$. Aller à l'étape 2.

Exemple2.3.1 :

$$\min_{x\geq 0} F(x,y) = x - 4y$$
$$\text{sujet à } \min_{y\geq 0} f(y) = y$$
$$\text{sujet à } \quad -x - y \leq -3$$
$$-2x + y \leq 0$$
$$2x + y \leq 12$$
$$-3x + 2y \geq -4$$

1 – résoudre le problème suivant :

$$\min_{x\geq 0} F(x,y) = x - 4y$$
$$\text{sujet à } \quad -x - y \leq -3$$
$$2x + y \leq 12$$
$$-3x + 2y \geq -4$$
$$-2x + y \leq 0$$

Le tableau initial est le suivant :

T_1	x	y	t_1	t_2	t_3	t_4	v_1	v_2	b
t_1	1	1	-1	0	0	0	1	0	3
t_2	-2	1	0	1	0	0	0	0	0
t_3	2	1	0	0	1	0	0	0	12
v_2	3	-2	0	0	0	- 1	0	1	4
$-F$	1	-4	0	0	0	0	0	0	0

Le tableau optimal est le suivant :

T5	x	y	t_1	t_2	t_3	t_4	b
y	0	1	0	0	3/7	3/7	4
t_2	0	0	0	1	1/7	-5/2	4
t_1	0	0	1	0	5/7	1/7	5
x	1	0	0	0	2/7	$\frac{-43}{100}$	4
$-F$	0	0	0	0	$\frac{10}{9}$	$\frac{9}{7}$	12

La solution optimale est : $(x, y) = (4, 4)$.

2 – résoudre le problème second niveau pour $x = 4$:

$$\min_{y\geq 0} y$$

sujet à

$$y \geq -1$$
$$y \leq 8$$
$$y = 4$$

Le tableau initial est le suivant :

T1	y	t_0	t_1	v	b
t_0	-1	1	0	0	1
t_1	1	0	1	0	8
v	1	0	0	1	4
$-F$	1	0	0	0	0

Le tableau optimal est le suivant :

T2	*y*	t_0	t_1	*b*
t_0	0	1	0	5
t_1	0	0	1	4
y	1	0	0	4
-F	0	0	0	-4

La solution optimale est : $\tilde{y} = 4$

$\tilde{y} = y,$ alors $y \in M(4)$

Donc, la solution optimale de l'exemple (2.3.1) est : $(x, y) = (4, 4)$

2.3.2 Approche de Kuhn et Tucker:

Bard et Moore [4] développèrent un algorithme dont l'idée principale est de résoudre le programme (2.2) sans la contrainte (2.2d) ; à chaque itération, l'algorithme vérifie si cette dernière est satisfaite, si oui le point appartient au domaine induit ; sinon la méthode de séparation et évaluation est utilisée.

Avant de présenter l'algorithme ; nous donnons quelques notations :

$w = \{1, \ldots, q+m\}$ est l'ensemble des indices des variables $\in$ à la contrainte (2.2d).

F_0 valeur initiale de la fonction objectif égale à l'infini.

Au k^{eme} niveau (séparation et évaluation), on définit un sous ensemble d'indices $w_k \subset w$, et un chemin p_k correspondant à une affectation soit de u_i*=0* ou g_i *=0* $(g_i = b_i - A_i x - B_i y)$ pour $i \in w_k$ et on a :

$$S_K^+ = \{i, \quad i \in w_k \ et \quad u_i = 0\},$$
$$S_K^- = \{i, \quad i \in w_k \ et \quad g_i = 0\},$$
$$S_K^0 = \{i, \quad i \notin w_k\}.$$

Algorithme:

Etape 0 Poser $k := 0, S_k^+ = \varnothing,\quad S_k^0 := \{1,...,q+m\}$ et $F_0 := \infty$.

Etape1 (Itération k) poser $u_i := 0$ pour $i \in S_k^+$ et $g_i := 0$ pour $i \in S_k^-$.

Résoudre le problème (2.2) sans la contrainte (2.2d),

1. Si le programme obtenu est non réalisable aller à **(5)**,
2. Sinon, faire $k := k+1$ et noter la solution $\left(x^k, y^k, u^k\right)$.

Etape2 Si $F\left(x^k, y^k\right) \geq F_0$, aller en **(5).**

Etape3

1. Si $u_i^k g_i\left(x^k, y^k\right) = 0, i = 1,...,q+m$ aller en **(4).**
2. Sinon choisir i pour lequel $u_i g_i\left(x^k, y^k\right)$ est plus grand et noter le i_1

Faire $S_k^+ := S_k^+ \cup \{i_1\},\; S_k^0 := S_k^0 \setminus \{i_1\}, S_k^- = S_k^-$ ajouter i_1 à P_k aller à (1)

Etape 4 $F_0 := F\left(x^k, y^k\right)$.

Etape 5

1. S'il n'y a pas de nœuds. Aller à(6).
2. Sinon séparer au niveau des sommets restants ; écrire S_k^+, S_k^-, S_k^0 et P_k

tels citer en haut, aller à (1) .

Etape6

1. Si $F_0 = \infty$, pas de solution réalisable pour le problème (2.1) ;
2. Sinon, déclarer la solution réalisable associée à F_0 comme solution optimale.

Exemple 2.3.2 :

$$\min_{x \geq 0} F(x,y) = -8x_1 - 4x_2 + 4y_1 - 40y_2 - 4y_3$$

$$\text{sujet à} \quad \min_{y \geq 0} f(x,y) = x_1 + 2x_2 + y_1 + y_2 + 2y_3$$

$$\text{sujet à} \quad -y_1 + y_2 + y_3 \leq 1$$

$$2x_1 - y_1 + 2y_2 - \frac{1}{2}y_3 \leq 1$$

$$2x_2 + 2y_1 - y_2 - \frac{1}{2}y_3 \leq 1$$

En appliquant (2,2), on obtient :

$$
\begin{aligned}
&\min_{x\geq 0} F(x,y) = -8x_1 - 4x_2 + 4y_1 - 40y_2 - 4y_3 \\
&\text{sujet à} \quad -y_1 + y_2 + y_3 + v_1 = 1 \\
&\qquad 2x_1 - y_1 + 2y_2 - \frac{1}{2}y_3 + v_2 = 1 \\
&\qquad 2x_2 + 2y_1 - y_2 - \frac{1}{2}y_3 + v_3 = 1 \\
&\qquad u_1 + u_2 - 2u_3 + u_4 + v_4 = 1 \\
&\qquad -u_1 - 2u_2 + u_3 + u_5 + v_5 = 1 \\
&\qquad -u_1 + \frac{1}{2}u_2 + \frac{1}{2}u_3 + u_6 + v_6 = 2
\end{aligned}
$$

Le tableau initial est le suivant:

T1	x_1	x_2	y_1	y_2	y_3	u_1	u_2	u_3	u_4	u_5	u_6	v_1	v_2	v_3	v_4	v_5	v_6	b
v_1	0	0	-1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	1
v_2	2	0	-1	2	$-\frac{1}{2}$	0	0	0	0	0	0	0	1	0	0	0	0	1
v_3	0	2	2	-1	$-\frac{1}{2}$	0	0	0	0	0	0	0	0	1	0	0	0	1
v_4	0	0	0	0	0	1	1	-2	1	0	0	0	0	0	1	0	0	1
v_5	0	0	0	0	0	-1	-2	1	0	1	0	0	0	0	0	1	0	1
v_6	0	0	0	0	0	-1	$\frac{1}{2}$	$\frac{1}{2}$	0	0	1	0	0	0	0	0	1	2
$-F$	-8	-4	4	-40	-4	0	0	0	0	0	0	0	0	0	0	0	0	0

Le tableau optimal est le suivant :

T	x_1	x_2	y_1	y_2	y_3	u_1	u_2	u_3	u_4	u_5	u_6	b
y_1	$\frac{-2}{3}$	$\frac{2}{3}$	0	0	1	0	0	0	0	0	0	1
y_2	1	1	0	1	0	0	0	0	0	0	0	3/2
$y3$	$\frac{1}{3}$	$\frac{5}{3}$	1	0	0	0	0	0	0	0	0	3/2
u_4	0	0	0	0	0	1	1	-2	1	0	0	1
u_5	0	0	0	0	0	-1	-2	1	0	1	0	1
u_6	0	0	0	0	0	-1	$\frac{1}{2}$	$\frac{1}{2}$	0	0	1	2
$-F$	84/3	96/3	0	0	0	0	0	0	0	0	0	-58

La solution optimale est $x^1 = (0,0)$, $y^1 = (3/2, 3/2, 1)$, $u = (0,0,0,1,1,2)$ avec $F = 58$. Cette solution ne vérifie pas la contrainte (2.2d), alors une variable de séparation est choisie (u_6) et on pose $S_1^- = \varnothing, S_1^0 = \{1,2,3,4,5\}, P_1 = (6)$.

Dans les deux itérations suivantes, on sépare au niveau de la variable (u_5) et (u_4) et on passe à l'étape5 (backtrack) ; on aura $S_3^+ = \{5,6\}$, $S_3^- = \{4\}, S_3^0 = \{1,2,3\}, P_3 = (6,5,4)$, aller à l'étape1.

Une solution réalisable est trouvée. Aller à l'étape2, ensuite à l'étape3, la contrainte (2.2d) est vérifiée, alors aller à l'étape4 et on pose : $F_0 = -29.2$. Ensuite, l'algorithme passe à l'étape5 et on aura $S_4^+ = \{6\}$, $S_4^- = \{5\}, S_4^0 = \{1,2,3,4\}$, $P_4 = (6,5)$.

Aller à l'étape1, une autre solution réalisable est trouvée, aller à l'étape 2 on a :
la valeur de la fonction objectif du meneur est supérieure à F_0, alors l'algorithme passe à l'étape5 (backtrack) et on pose $S_5^+ = \varnothing, S_5^- = \{6\}, S_5^0 = \{1,2,3,4,5\}$ et $P_4 = (6)$.

La procédure se poursuit jusqu'à épuisement de tous les sommets. La solution optimale est $x^* = (0,0,5), y = (0,0,6,0,4), u^* = (0,1,3,6,0,0)$ et $F* = -29.2$.

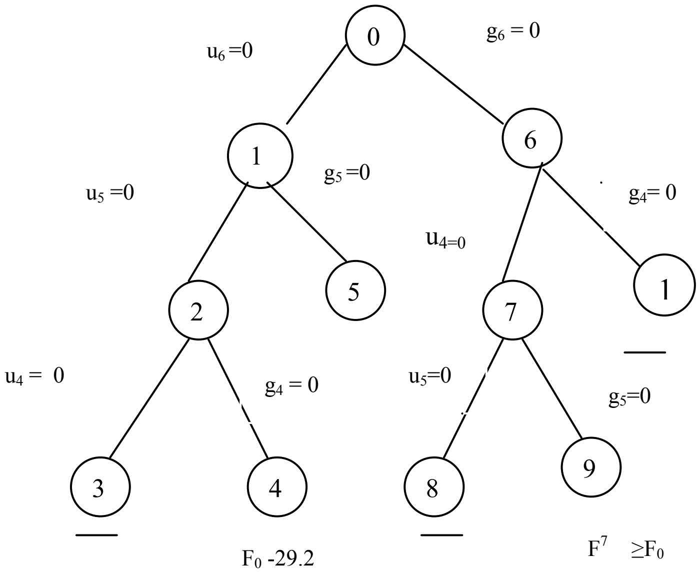


Figure de l'exemple 2.3.2

CHAPITRE 3

LA PROGRAMMATION A DEUX NIVEAUX LINEAIRE EN VARIABLES DISCRETES

Introduction

Dans beaucoup de problèmes d'optimisation un sous ensemble de variables prend des valeurs entières, ce qui peut parfois compliquer le problème.

Dans ce chapitre nous présentons les différents types de la programmation à deux niveaux linéaire en variables entières.

3.2. Les différents types d'un problème de programmation à deux niveaux linéaire en variables discrètes [15] :

Soit le problème de programmation à deux niveaux linéaire :

$$(3.1) \quad \begin{cases} \min\limits_{x \in X} F(x,y) = c_1 x + d_1 y & (a) \\ \text{sujet à } A_1 x + B_1 y \leq b_1 & (b) \\ \qquad \min\limits_{y \in Y} f(x,y) = c_2 x + d_2 y & (c) \\ \qquad \text{sujet à } A_2 x + B_2 y \leq b_2 & (d) \end{cases}$$

$x \in X \subset IR^{n}, y \in Y \subset IR^{m}, f, F : IR^{n} \times IR^{m} \rightarrow R, c_1, c_2 \in IR^{n}, d_1, d_2 \in IR^{m}$,
$A_1 \in IR^{p \times n}, A_2 \in IR^{q \times n}, B_1 \in IR^{p \times m}, B_2 \in IR^{q \times m}$.

Un problème de programmation à deux niveaux linéaire en variables discrètes est obtenu si : L'ensemble X ou l'ensemble Y ou les deux à la fois prennent des valeurs discrètes ; d'une autre manière c'est dans le cas ou le problème du haut niveau ou le problème du second niveau ou les deux à la fois sont discrets.

Exemple3.1 :

Soit le problème de programmation à deux niveaux linéaire suivant:

$$\langle c,x\rangle + \langle d,y\rangle \to \min_{x,y} \qquad (1)$$

sujet à

$$By \le b, \quad y = (y_1, y_2) \;\; ,x \in \Psi(y) \qquad (2)$$

ou

$$\Psi(y) = \arg\max_x \left\{ \langle y_1, x\rangle / \, Ax + Cy_2 \le a, x \in X \right\}$$

posons: $q = 0$(i.e il n'y a pas de contraintes $By \le b$)

$$A = \begin{pmatrix} 3 \\ 1 \\ -1 \\ -3 \end{pmatrix} \qquad C = \begin{pmatrix} -1 \\ 1 \\ 0 \\ 1 \end{pmatrix} \qquad a = \begin{pmatrix} 6 \\ 6 \\ -1 \\ 0 \end{pmatrix} \quad c = -5, d = (0,1)^T$$

1^er^cas : $X = [0,4] \qquad Y = \left\{ (y_1, y_2), y_1 = -1, y_2 \in [0,4] \right\} \subset IR^2$

Le problème second niveau (2) peut $\hat{\text{e}}$tre écrit :

$$\max_{x \in X} \langle y_1, x\rangle$$

$$\text{sujet à } \; Ax + Cy_2 \le a$$

L'ensemble réalisable du problème second niveau (2) au point 'y':

$$\Omega(y) = \left\{ x, \;\; Ax \le a - Cy_2 \right\}.$$

$$= \left\{ x, x \ge 1, x \ge \frac{y_2}{3}, x \le \frac{6 + y_2}{3}, x \le 6 - y_2 \right\}$$

$$= \left\{ x, \quad \max\left\{ 1, \frac{y_2}{3} \right\} \le x \le \min\left\{ \frac{6 + y_2}{3}, 6 - y_2 \right\} \right\}.$$

Notons F la fonction objectif du problème haut niveau .

L'ensemble des solutions optimales du problème (2) est :

$$\Psi(y)=\begin{cases} 1 & si \quad 0 \le y_2 \le 3 \\ \dfrac{y_2}{3} & si \quad 3 \le y_2 \le 4 \end{cases} \quad , \; F\left(x(y),y\right)=\begin{cases} -5+y_2 & si \quad 0 \le y_2 \le 3 \\ -5\dfrac{y_2}{3}+y_2 & si \quad 3 \le y_2 \le 4 \end{cases}$$

il est clair que *(x,y) = (1,-1, 0)* est la solution optimale globale , ($\frac{4}{3}$, -1 , 4) est une solution optimale locale.

2eme cas :

$$X=[0,4]$$
$$Y=\left\{(-1,0)^T,(-1,1)^T,(-1,2)^T,(-1,3)^T,(-1,4)^T\right\}$$

L'ensemble des solutions optimales du problème haut niveau est :

$$\Psi(y)=\begin{cases} 1 & si \quad y_2 \in \{0,1,2,3\} \\ \dfrac{y_2}{3} & si \quad y_2=4 \end{cases} \quad ,$$

Donc : $$F\left(x(y),y\right)=\begin{cases} -5+y_2 & si \;\; y_2 \in \{0,1,2,3\} \\ -2\dfrac{y_2}{3} & si \quad y_2=4 \end{cases}$$

L'ensemble de solutions réalisables du problème haut niveau est :

$$DI=\left\{ \; (1,-1,0) \; , \; (1,-1,1) \; , \; (1,-1,2) \; , \; (1,-1,3) \; , \; (\frac{4}{3},-1,4) \; \right\}.$$

La solution optimale globale est (1, -1 ,0) et $\left(\frac{4}{3},-1,4\right)$ est une solution optimale locale .

3ème cas : $X=\{0,1,2,3,4\}$

$$Y=\{-1\}\times[0,4]$$

On aura : $\Psi(y)=\begin{cases} 1 & 0 \le y_2 \le 3 \\ 2 & 3 < y_2 \le 4 \end{cases}$

La valeur la plus petite de la fonction objectif du problème haut niveau est -7,
mais pas de solution optimale globale.

4$^{\text{ème}}$ cas : $X = \{0,1,2,3,4\}$

$Y = \{0,1,2,3,4\}$

On aura: $\Psi(y) = \begin{cases} 1 & si \;\; y_2 \in \{0,1,2,3\} \\ 2 & si \;\; y_2 = 4 \end{cases}$

L'ensemble de solutions réalisables du problème haut niveau est :

$$DI = \{(1,-1,0)\ ,\ (1,-1,1)\ ,\ (1,-1,2)\ ,\ (1,-1,3)\ ,\ (2,-1,4)\}$$

Donc :

$$F(x(y),y) = \begin{cases} -5 + y_2 & y_2 \in \{0,1,2,3\} \\ -10 + y_2 & y_2 = 4 \end{cases}$$

La solution optimale globale est $(x^*, y^*) = (2,-1,4)$ et $F^* = -6$.

$(1,-1,0)$ est une solution optimale locale ; notons que cette solution est une solution optimale globale du problème relaxé pour les deux niveaux .

Théorème3.1. [15] : *Soit l'ensemble*
$M := \{ (x, y)^T : By \leq b \;, y \in Y \;, Ax + Cy \leq a \;, x \in X \}$; non vide et compact et soit X un polyèdre convexe. Supposons que $Y = R^m$ ou $Y = Z^m$; alors le problème de programmation à deux niveaux linéaire discret possède une solution optimale.

Théorème3.2. [15] : *Considérons le problème (1), (2) où $X = Z^n$, $Y = Z^m$;*
Si l'ensemble M (défini dans le théorème 3.1) est non vide et compact, alors le problème de programmation à deux niveaux linéaire discret possède une solution optimale.

3.3 Propriétés d'un problème de programmation à deux niveaux linéaire en variables bivalentes [5] :

On exposera les propriétés du problème de PDN linéaire quand quelques ou toutes les variables prennent des valeurs binaires.
Nous commencerons par étudier la géométrie du domaine réalisable et discuter l'existence de solution optimale.

Pour chaque $x \in X$ on suppose que la solution optimale du problème second niveau est unique, et on s'intéresse aux trois modèles suivant :

1. Problème de programmation à deux niveaux linéaire discret (DL – BLPP)

i.e $X = \{0,1\}^n$ et $Y = \{0,1\}^m$.

2. Programmation à deux niveaux linéaire discrète continue (DCL – BLPP)

où $X = \{0,1\}^n$ et $Y = IR^m$

3. Programmation à deux niveaux continue – discrète (CDL – BLPP)

où $X = IR^n$ *et* $Y = \{0,1\}^m$.

Propriété 3.2.1 [5]
Soit l'ensemble $S_U = \{(x, y),\ A_1 x + B_1 y \le b_1\}$.
Si $S_U = IR^{n+m}$ alors le domaine induit DI est non vide si $\Omega \neq \varnothing$.
Si $S_U \neq IR^{n+m}$ alors DI est non vide s'il existe $\bar{x} \in X$ tel que $(\bar{x}, \bar{y}) \in S_U$.

Propriété3.2.2 [5]
Les domaines induits des problèmes DCL – BLPP et DL – BLPP sont respectivement inclus dans les domaines induits des problèmes L – BLPP et CDL - BLPP.

Exemple3.2.1 :

$$n = m = 1, \quad q = 4, \quad S_U = R^2, \quad d_2 = 1$$

$$A_2 = \begin{bmatrix} 1 \\ -1 \\ 5 \\ -5 \end{bmatrix} \qquad B_2 = \begin{bmatrix} 1 \\ 1 \\ -4 \\ -4 \end{bmatrix} \qquad \mathrm{b}_2 = \begin{bmatrix} 2 \\ 2 \\ 10 \\ 10 \end{bmatrix}$$

soit le problème de programmation à deux niveaux :

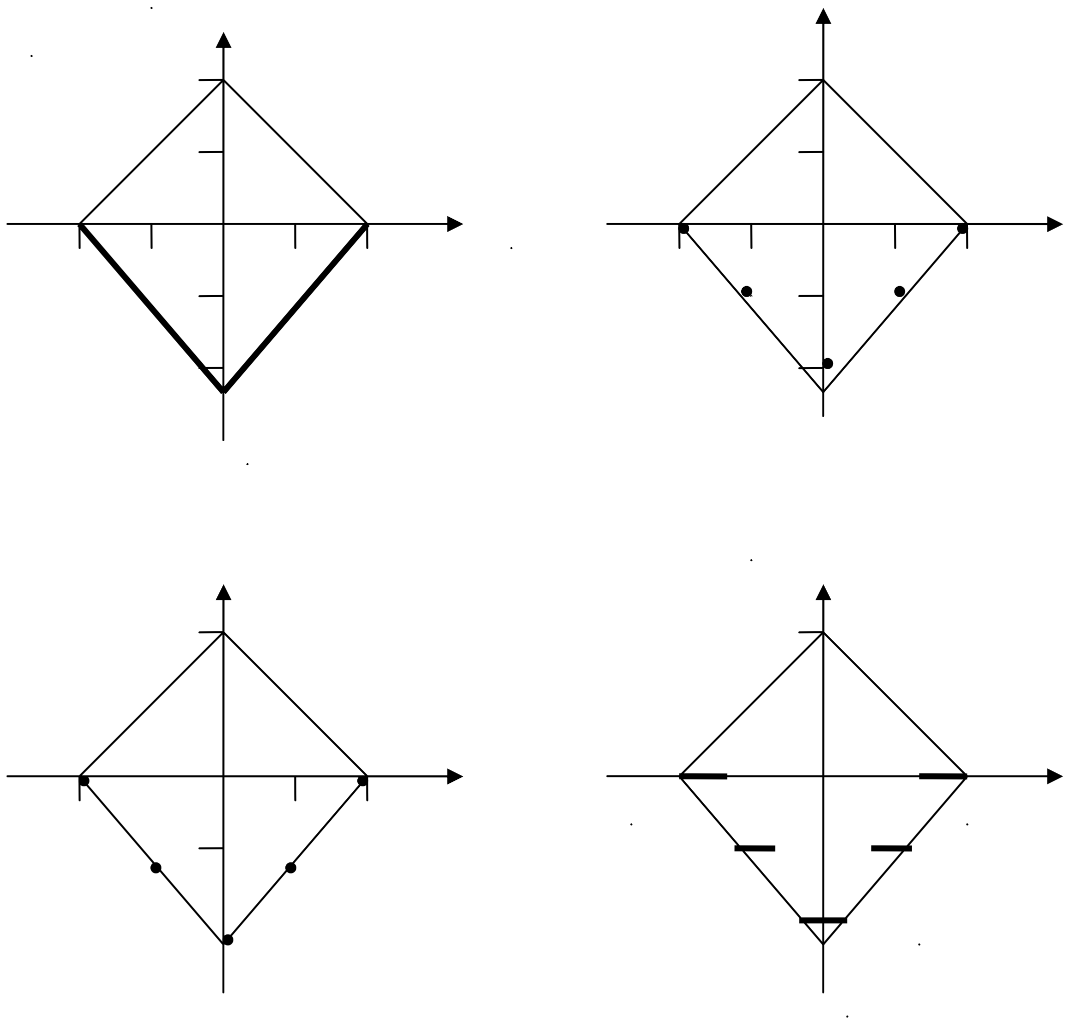
$$\begin{aligned} &\min_x F(x, y) = c_1 x + d_1 y \\ &\quad \textit{sujet à } A_1 x + B_1 y \le b_1 \\ &\quad \min_y f(y) = d_2 y \\ &\quad \textit{sujet à } A_2 x + B_2 y \le b_2 \end{aligned} \qquad (3)$$

Le problème second niveau est :

$$\begin{aligned} &\min y \\ &\textit{sujet à} \quad x + y \le 2 \\ &\qquad\qquad -x + y \le 2 \\ &\qquad\qquad 5x - 4y \le 10 \\ &\qquad\qquad -5x - 4y \le 10 \end{aligned}$$

La figure 2.1 ci dessous représente les domaines induits des quatre problèmes cités précédemment.

L - BLPP DL - BLPP



DCL - BLPP CDL -BLPP

Remarques :

1. La solution optimale de DL – BLPP est un point intérieur de l'ensemble Ω .

2. Le domaine induit du problème CDL – BLPP est non compact alors le problème n'aura pas de solution optimale même si DI est non vide. On donne un exemple illustratif.

Exemple 3.2.2 :

Soit le problème de programmation à deux niveaux linéaire(3) dans le cas continu discret avec :

$n = m = 1, \quad S_U = R^2, \quad q = 3, \quad c_1 = d_1 = 1, d_2 = -1$

$$A_2 = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}, \qquad b_2 = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$$

On aura :

$$\mathrm{DI} = \left\{ (0,2) \ \left\{ \left] 0, \frac{1}{2} \right], 1 \right\}, \ \left\{ \left] \frac{1}{2}, 1 \right], 0 \right\} \right\}.$$

Le meneur doit choisir un x très petit mais supérieur à $\frac{1}{2}$ afin d'obliger le suiveur à choisir $y = 0.$

Dans le cas où le meneur choisit $x = \frac{1}{2}$, le suiveur répondra avec $y = 1$ produisant une dégradation de la valeur de la fonction objectif du meneur. Alors, s'il n'y a pas coopération le problème n'aura pas de solution.

Propriété3.2.3 [5]

Soit Ω un ensemble borné i.e .Si $S_U = IR^{n+m}$ alors le problème de programmation à deux niveaux discret (DL – BLPP) et le problème de programmation à deux niveaux discret-continu (DCL – BLPP) possèdent une solution optimale si Ω est non vide.

si $S_U \neq IR^{n+m}$ alors DL – BLPP et DCL – BLPP et L - BLPP possèdent une solution s'il existe $\bar{x} \in X, \left(\bar{x}, \bar{y}\right) \in S_U$.

Cette propriété confirme que si Ω est borné alors une condition nécessaire pour l'existence d'une solution optimale coïncide avec la condition qui rend le domaine induit non vide.

3.4. Algorithme de résolution d'un problème de programmation à deux niveaux linéaire discret [5]

On va présenter un algorithme développé par [Bard ,Moore] pour la résolution d'un problème de programmation à deux niveaux en variables binaires.

Le problème est formulé de la manière suivante :

$$(3.2)\quad \begin{array}{ll} \min\limits_{x} F(x,y) = c_1 x + d_1 y & (\mathrm{a}) \\ \textit{sujet à } A_1 x + B_1 y \le b_1 & (\mathrm{b}) \\ x \in X = \{0,1\}^{\mathrm{n}} & (\mathrm{c}) \\ \min\limits_{y} f(y) = d_2 y & (d) \\ \textit{sujet à } A_2 x + B_2 y \le b_2 & (e) \\ y \in Y = \{0,1\}^{\mathrm{m}} & (3.2f) \end{array}$$

$$c_1 \in IR^n, d_1, d_2 \in IR^m, b_1 \in IR^p, b_2 \in IR^q, A_1 \in IR^{p\times n}, B_1 \in IR^{p\times m}, A_2 \in IR^{q\times n}, B_2 \in R^{q\times m}$$

Une fois x est choisi, le problème du second niveau (3.2d,f) devient un programme linéaire binaire en fonction de y seulement.

L'idée principale sur laquelle se base l'algorithme est le fait que la solution optimale (x, y) d'un problème de PDN linéaire en variables discrètes binaires possède un y appartenant à l'ensemble des réactions rationnelles du suiveur *M(x)*. En limitant la recherche sur cette ensemble, il sera possible de découvrir un meilleur point appartenant au domaine induit.

Ceci est achevé par la formulation et la résolution répétée de quelques instances du programme suivant obtenu à partir de (3.2).

$$\begin{array}{ll} \min f(y) = d_2 y & \\ \textit{sujet à} & A_1 x + B_1 y \le b_1 \\ & A_2 x + B_2 y \le b_2 \\ & F(x,y) = c_1 x + d_1 y \le \alpha \\ & \sum_{j=1}^{n} x_j \ge \beta \\ & x \in X, y \in Y \end{array}$$

Soit $W = \{1, 2, ..., n\}$

A la k[ième] itération, on définit un chemin p_k de longueur $l = |w_k|$ $(w_k \subset W)$ correspondant à une affectation $x_j = 0$ *ou* $x_j = 1$ pour $j \in w_k$

Par exemple $x_3 = 0$ *et* $x_2 = 1$ (choisi dans cet ordre) à l'instant k, alors :

$$w_k = \{3, 2\}, l = 2, p_k = (\underline{3}, 2)$$

On défini par la suite:

$$S_K^+ = \{j, j \in w_k, x_j = 1\}$$
$$S_k^- = \{j, j \in w_k, x_j = 0\}$$
$$S_K^0 = \{j, j \notin w_k\}$$

Algorithme

Etape0 (Initialisation)

faire $k := 0, S_k^+ := \varnothing, S_k^- := \varnothing, S_k^0 := \{1, ..., n\}, \alpha := 0, \beta := 0$ et $\overline{F} = \infty$.

Etape1 : (itération générale)

Poser $x_j = 1$ pour $j \in S_k^+$ et $x_j = 0$ pour $j \in S_k^-$, pour les valeurs courantes de α et β, trouver une solution réalisable pour le problème (3.2a) _(3.2f).

1. Si oui faire k : = k+1 noter la solution (x^k, y^k) , aller à l'etape 2.

2. Sinon, aller à l'étape6.

Etape2 :(Bounding) fixer $x = x^k$, relaxer $(3.2d)$ et $(3.2e)$ et résoudre $(3.2a, b, c, f)$

pour avoir un point $(x^k, y^k) \in DI$. Calculer $F(x^k, y^k)$ poser

$\overline{F} := \min\{\overline{F}, F(x^k, y^k)\}$.

Etape3 : $J = \{j \in S_{k-1}^0, x_j^k = 1\}$

1. Si $J = \varnothing$, poser $S_k^+ := S_{k-1}^+$, $S_k^- := S_{k-1}^-, S_k^0 := S_{k-1}^0, P_k := P_{k-1}$ et aller à l'étape 5.

2. Si non aller à l'etape 4.

Etape4 : (Branching) créer $|J|$ nouveaux sommets de la manière suivante :

associer $j \in J$ à P_{k-1} suivant un ordre un par un afin d'obtenir les sommets restants et le chemin P_k.

Poser $S_k^+ = S_{k-1}^+ \cup J, S_k^0 = S_{k-1}^0 / J$ et $S_k^- = S_{k-1}^-$.

Le chemin précédent est allongé par une longueur $|J|$

Etape5 : Faire $\alpha = \overline{F} - 1, \beta = 1 + \left|S_k^+\right|$, aller à (1).

Etape6 : (Backtracking)

1. S'il n'y a plus de nœuds, aller à (7)

2. Sinon backtrack à partir du nœud courant (index de sa variable est noté j') brancher en posant $x_{j'}^k = 0$.

Donner S_k^+, S_k^-, S_k^0 et P_k comme indiqué au dessus.

Poser $\beta := 0$, aller à (1).

Etape7 : (fin)

1. Si $\overline{F} = \infty$, pas de solution .

2. Sinon déclarer la solution réalisable associé à $\overline{F}$ comme solution optimale.

Exemple3 :

$$
\begin{aligned}
&\min_{x\geq 0} F(x,y) = x + y \\
&\min_{y\geq 0} f(y) = -5x - y \\
&\textit{sujet à} \quad -x - \frac{y}{2} \leq -2 \\
&\qquad\qquad -\frac{x}{4} + y \leq 2 \\
&\qquad\qquad x + \frac{y}{2} \leq 8 \\
&\qquad\qquad x - 2y \leq 4
\end{aligned}
$$

La solution optimale de ce problème est $(x,y) = \left(\frac{8}{9}, \frac{20}{9}\right)$ avec $\overline{F} = \frac{28}{9}$.

D'après le graphe ci-dessous on a $x < 8$ et $y < 4$.

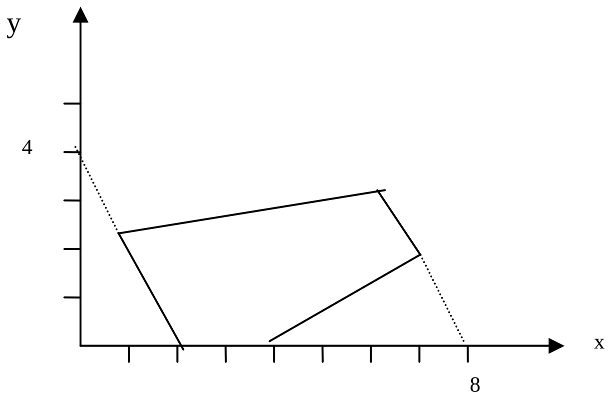


figure de l'exemple3

Pour transformer ce problème à un problème binaire , on pose : $x = x_1 + 2x_2 + 2^2 x_3$

$y = y_1 + 2y_2$.

$x_i \quad , y_i \in \{0,1\}$.

L'exemple 3 devient :

$$\min_x F(x,y) = x_1 + 2x_2 + 4x_3 + y_1 + 2y_2$$
$$\min_y f(x,y) = -5x_1 - 10x_2 - 20x_3 - y_1 - 2y_2$$
$$\begin{aligned} \textit{sujet à} \quad & -2x_1 - 4x_2 - 8x_3 - y_1 - 2y_2 \le -4 \\ & -x_1 - 2x_2 - 4x_3 + 4y_1 + 8y_2 \le 8 \\ & 2x_1 + 4x_2 + 8x_3 + y_1 + 2y_2 \le 16 \\ & x_1 + 2x_2 + 4x_3 - 2y_1 - 4y_2 \le \quad 4 \\ & x_i, y_i \in \{0,1\}. \end{aligned}$$

En appliquant (3.2), on obtient :

$$\min_y f(y) = -y_1 - 2y_2$$
$$\begin{aligned} \textit{sujet à} \quad & -2x_1 - 4x_2 - 8x_3 - y_1 - 2y_2 \le -4 \\ & -x_1 - 2x_2 - 4x_3 + 4y_1 + 8y_2 \le 8 \\ & 2x_1 + 4x_2 + 8x_3 + y_1 + 2y_2 \le 16 \\ & x_1 + 2x_2 + 4x_3 - 2y_1 - 4y_2 \le 4 \\ F(x,y) = & \, x_1 + 2x_2 + 4x_3 + y_1 + 2y_2 \le \alpha \\ & x_1 + x_2 + x_3 \ge \beta \\ & x_i, y_i \in \{0,1\}. \end{aligned}$$

Après initialisation, on trouve (4,3) comme solution (étape1), l'étape2 confirme que ce point appartient au domaine induit, on aura :

$\overline{F} = 7, S_1^+ = \{3\}, S_1^- = \emptyset, S_1^0 = \{1,2\}, P_1 = (3), \alpha = 6, \beta = 2.$

Résoudre à nouveau le problème (3.2a) – (3.2f) (étape1), on trouve le point (5,1) comme solution, à l'étape2, $\left(x^2, \hat{y}^2\right) = (5,3)$ et $F\left(x^2, \hat{y}^2\right) = 8,$ donc $\overline{F} = 7$.

$S_2^+ = \{1,3\}, S_2^0 = \{2\}, P_2 = (3,1), \alpha = 6, \beta = 3.$

Passant à l'étape 1, pas de solution réalisable, l'algorithme passe à l'étape6 et on obtient : $S_2^+ = \{3\}, S_2^- = \{1\}, S_2^0 = \{2\}, P_2 = (3,1), \alpha = 6, \beta = 0.$

Retournant à l'étape1 une solution réalisable est trouvée qui est (4,0).

A l'étape2 on a : $\left(x^3, \hat{y}^3\right) = (4,3) avec F = 7.$

à l'étape 3, comme on a $S_0^2 = \{2\}$ et $x_2^3 = 0$ alors $J = \emptyset,$ et on aura : $S_3^+ = S_2^+ = \{3\}.$

donc : $\beta = 1 + \left|S_3^+\right| = 2$, ensuite on passe à l'étape1.

Le sous problème obtenu est non réalisable, l'algorithme passe à l'étape6 et backtracks.

Les calculs continuent, la solution optimale de ce problème est $\left(x^*, y^*\right) = (1,2)$ avec $F^* = 3.$

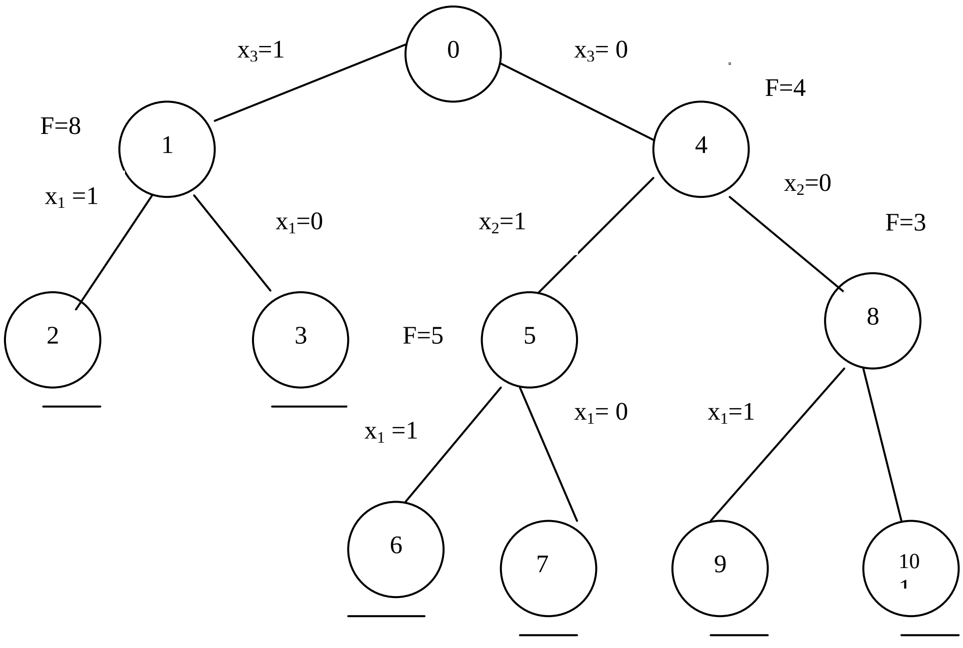


Figure de l'exemple 3

Remarque :

A l'étape1 de l'algorithme précédent une solution réalisable est trouvée $\left(x^k, y^k\right)$, un sous programme est résolu à l'étape2 un sous programme est résolu pour trouver une solution $\left(x, \hat{y}^k\right)$ appartenant au domaine induit, il sera donc intéressant de s'avoir dans quels cas on aura $\hat{y}^k = y^k$ afin d'éviter cette étape.

Si le problème $(3.2a)-(3.2f)$ est résolu à l'optimalité alors la proposition suivante permet de vérifier si la solution trouvée appartient au domaine induit .

Proposition 3.2.4 [6]:

Soit $\left(\overline{x}, \overline{y}\right)$ une solution du problème $(3.2a)-(3.2f)$, alors $\left(\overline{x}, \overline{y}\right) \in DI$ si $c_1\overline{x} + \sum_{j=1}^{m}\left(d_{1j} - \min\left\{d_{1j}, 0\right\}\right) \le \alpha$.

Algorithme modifié :

Etape1a (itération générale)

Poser $x_j = 1$ pour $j \in S_+^k$ et $x_j = 0$ pour $j \in S_k^-$, pour les valeurs courantes de α et β . Trouver une solution optimale du problème $(2.3a)-(2.3f)$.

1. Si oui faire k =k+1, noter la solution $\left(x^k, y^k\right)$, aller à l'étape2a.

2. Sinon aller à l'étape 6.

Etape2a (Bounding)

Utiliser la proposition pour vérifier si $\left(x^k, y^k\right)$ appartient au domaine induit.

1. Si oui aller à l'étape 2b

2. Sinon, fixer $x = x^k$, relaxer $(3.2d)$ et $(3.2e)$ et résoudre le problème $(3.2a)-(3.2f)$ pour avoir un point $\left(x^k, \hat{y}^k\right) \in DI$.

Etape2b Calculer $F\left(x^k, \hat{y}^k\right)$ et faire $\overline{F} = \min\left\{\overline{F}, F\left(x^k, \hat{y}^k\right)\right\}$.

CHAPITRE 4

UNE METHODE POUR LA RESOLUTION D'UN PROBLEME DE PROGRAMMATION A DEUX NIVEAUX LINEAIRE DISCRET

1. Introduction :

Dans ce chapitre nous proposons un algorithme pour la résolution d'un problème de programmation à deux niveaux en nombre entiers.

La formulation d'un problème de programmation à deux niveaux discret est la suivante :

$$(P_1) \quad \begin{cases} \max\limits_{x_2} F(x_1, x_2) = c_{21}x_1 + c_{22}x_2 = c_2 x \\ \textit{sujet à} \\ \quad \max\limits_{x_1} f(x_1, x_2) = c_{11}x_1 + c_{12}x_2 = c_1 x \\ \quad \textit{sujet à} \\ \qquad A_1 x_1 + A_2 x_2 \le b_2 \\ \qquad x_1, x_2 \ge 0 \ , \quad x \ge 0 \\ \qquad \textit{entiers} \end{cases}$$

$c_{11}, c_{12} \in IR^{n_1}, \quad c_{22}, c_{22} \in IR^{n_2}, c_1 = (c_{11}, c_{12}), c_2 = (c_{21}, c_{22}), A_1 \in IR^{m\times n_1}, A_2 \in IR^{m\times n_2}, \quad A = (A_1, A_2), b \in R^m$

Soit le problème relaxé suivant :

$$(P_2)\quad \begin{aligned} &\min_{x_2} F(x_1,x_2) = c_{21}x_1 + c_{22}x_2 \\ &\textit{sujet à} \\ &A_1x_1 + A_2x_2 \le b \\ &x_1, x_2 \ge 0 \end{aligned}$$

Le problème second niveaux relaxé s'écrit :

$$(P_3)\quad \begin{aligned} &\min_{x_1} f(x_1,x_2) = c_{11}x_1 + c_{12}\bar{x}_2 \\ &\textit{sujet à} \\ &A_1x_1 \le b - A_2\bar{x}_2 \\ &x_1 \ge 0 ,\ \bar{x}_2 \ \textit{entier} \end{aligned}$$

La solution optimale entière du problème sera obtenue en parcourant les solutions réalisables entières du problème **(*P₂*)** suivant l'ordre décroissant de la valeur de la fonction objectif du problème haut niveau, en utilisant la technique développée dans [29].

1. Notations :

$S_0 = \{(x_1,x_2), A_1x_1 + A_2x_2 \le b, x_1 \ge 0, x_2 \ge 0\}$

Ensemble de solutions réalisables du problème relaxé (P_2).

$S_1 = \{(x_1,x_2), A_1x_1 + A_2x_2 \le b, x_1 \ge 0, x_2 \ge 0,\ x_1, x_2 \ \textit{entiers}\}$

$x^k = (x_1^k, x_2^k)$ est la k[ième] meilleure solution entière du problème (P_2) .

x^k sera un point extrême du domaine réalisable du problème relaxé après des troncatures successives sur le domaine réalisable.

B^k : base correspondante à x^k.

$I(k) = \{i, x_i^k \ \textit{est la}\ i^{\grave{e}me}\ \text{var}\textit{iable de base correspendante à la}\ k^{\grave{e}me} \textit{meilleur solution entière}\ x^k\}$

$\bar{I}(k) = \{\text{indices de variables hors base}\}$

$Z^k = c_{21}x_1^k + c_{22}x_2^k = c_2 x^k$

$y_j^k = \left\{ y_{ij}^k \right\} = \left(B^k \right)^{-1} .a_{ij}$

$Z_j = \sum_{i \in I(k)} c_{2i} y_{ij}^k$

$\delta_j^k = c_{2j} - z_j .$

$J(k) = \left\{ j,\ j \in \overline{I}(k), \delta_j^k = 0 \right\}.$

$T(k) = \overline{I}(k) - J(k).$

L'ensemble réalisable du problème second niveau au point $x_2 = \overline{x}_2$ est donnée par :

$S_2 = \left\{ x_1, A_1 x_1 \leq b - A_2 \overline{x}_2 \ , x_2 \geq 0 \right\}$

L'ensemble de solutions optimales du problème second niveau au point $\overline{x}_2$ est

$\psi\left(\overline{x}_2\right) = \left\{ x_1 ; x_1 \in [\arg\max\ f\left(\tilde{x}_1, \overline{x}_2\right), \tilde{x}_1 \in S_2 ; \tilde{x}_1 entier] \right\}.$

Définition1 :

Si $\overline{x}_1 \in \psi\left(\overline{x}_2\right)$ *alors* $\left(\overline{x}_1, \overline{x}_2\right)$ *est une solution réalisable du problème du programmation à deux niveaux.*

Définition2 :

$\left(x_1^*, x_2^*\right)$ est une solution optimale d'un problème de programmation à deux niveaux si :

a- $\left(x_1^*, x_2^*\right)$ est réalisable.

b- Pour chaque solution bi niveau réalisable $\left(x_1, x_2\right)$; $F\left(x_1^*, x_2^*\right) \leq F\left(x_1, x_2\right)$.

On suppose :

(i) - S_1 est non vide.

(ii) – Pour chaque décision prise par le meneur , le suiveur possède une réponse i.e, $S_2 \neq \emptyset$

(iii) - $\psi\left(\overline{x}_2\right)$ est un singleton .

a- Si la proposition (ii) est non vérifiée i.e, S_2 est vide, alors $\Psi\left(\overline{x}_2\right)$ sera vide . pas de solutions réalisables .Donc le problème est non réalisable.

b- Si (iii) est non satisfaite i.e, $\Psi\left(\overline{x}_2\right)$ est non singulier ; le suiveur aura plusieurs possibilités pour une même décision prise par le meneur. S'il n'y a pas coopération pas de solution car le meneur ne peut pas forcer le suiveur à faire un choix.

Théorème 3 :

Toutes les solutions entières de (P₂) dont la valeur de la fonction objectif est inférieure à $F\left(x^k\right)$ *appartiennent au demi espace fermé* $\sum_{j \in T(k)} x_j \geq 1$.

Preuve :

Soit X^k une solution optimale entière du problème (P_2) dont la valeur de la fonction objectif est $F\left(X^k\right)$

alors, $AX^k = b$

i.e , $$\sum_{i \in I(k)} a_i x_i^k = b \qquad (1)$$

$x_i^k, i \in I\left(k\right)$, sont les variables or base non nulles.

$a_i, i \in I\left(k\right)$ sont les colonnes de A associés à la base.

L'ajout et la suppression de $\phi_j a_j$ à (1) avec $\phi_j > 0$ donne :

$$\sum_{i \in I(k)} a_i x_i^k - \phi_j a_j + \phi_j a_j = b$$

Comme $$a_j = \sum_{i \in I(k)} a_i y_{ij}^k$$

Donc, $$\sum_{i \in I(k)} a_i x_i^k - \phi_j \left(\sum_{i \in I(k)} a_i y_{ij}^k \right) + \phi_j a_j = b$$

$$\sum_{i \in I(k)} a_i \left(x_i^k - \phi_j y_{ij}^k \right) + \phi_j a_j = b$$

il est réalisable si :

$$x_j^k - \phi_j y_{ij}^k \geq 0$$

$$\phi_j > 0$$

i.e $$0 < \phi_j \leq \frac{x_j^k}{y_{ij}^k} \quad \text{si} \quad y_{ij}^k > 0$$

A partir d'une solution optimale entière X^k, sur S_1, à l'étape k une solution $\bar{X}^k$ est crée en diminuant x_j^k, $j \in \bar{I}(k) - J(k)$

$$\bar{X}^k = \begin{cases} x_i^k = x_i^k - \phi_j y_{ij}^k, & i \in I(k), \\ \bar{x}_j^k = \phi_j, & j \in J(k), \\ \bar{x}_v^k = 0, & v \in \bar{I}(k) - \{j(k)\}. \end{cases}$$

$\bar{X}^k$ est une solution réalisable entière

$(\text{i})\ \phi_j \leq \min_{i \in I(k)} \{x_j^k / y_{ij}^k, \quad y_{ij}^k > 0\}$,

$(\text{ii})\ \phi_j$ est un entier positif,

$(\text{iii})\ \phi_j y_{ij}^k$ est un entier pour tout $i \in I(k)$.

Considérons :

$$F(\bar{X}^k) - F(x^k) = c_2 \bar{X}^k - c_2 X^k$$

$$= \sum_{i \in I(k)} c_{2i} \bar{x}_i^k + c_{2j} \bar{x}_j^k + \sum_{v \in \bar{I}(k) - \{j\}} c_{2v} \bar{x}_v^k - \sum_{i \in I(k)} c_{2i} x_i^k$$

$$= \sum_{i \in I(k)} c_{2i} \left(x_i^k - \phi_j y_{ij}^k\right) + c_{2j} \phi_j - \sum_{i \in I(k)} c_{2i} x_i^k$$

$$= \sum_{i \in I(k)} c_{2i} x_i^k - \phi_j \sum_{i \in I(k)} c_{2i} y_{ij}^k + c_{2j} \phi_j - \sum_{i \in I(k)} c_{2i} x_i^k$$

$$= \phi_j \left(c_{2j} - \sum_{i \in I(k)} c_{2i} y_{ij}^k \right)$$

$$= \phi_j \left(c_{2j} - z_j\right) < 0 \ \text{ pour } j \in T(k) = \bar{I}(k) - J(k)$$

Donc : $$F(\bar{X}^k) \leq F(X^k).$$

Comme ϕ_j est un entier positif, $\bar{x}^k$ satisfait $\sum_{j \in T(k)} x_j \geq 1$.

3. Algorithme

Etape1 : Résoudre le problème (P_2). Soit (x_1^*, x_2^*) la solution optimale,

1. Si elle est entière aller à l'etape2.

2. Sinon appliquer la technique coupe de Gomory pour trouver la solution entière, soit $\bar{x} = (\bar{x}_1, \bar{x}_2)$ une telle solution entière.

Etape2 : Résoudre le problème **(P3)** pour $x_2 = \bar{x}_2$. Soit $(\tilde{x}_1, \tilde{x}_2) = (\tilde{x}_1, \bar{x}_2)$ est telle solution.

1. Si $\tilde{x}_1 = \bar{x}_1$ alors $(\bar{x}_1, \bar{x}_2)$ est la solution optimale du problème bi niveau. Terminer.

2. Sinon aller à l'etape3**.**

Etape 3 : Commençons par le tableau optimal correspondant à la solution $X^1 = (\bar{x}_1, \bar{x}_2)$ du problème **(P2).** Trouver la 2ème meilleure solution entière en introduisant la contrainte $\sum_{j \in T(1)} x_j \geq 1$. Résoudre ce problème par le dual simplex.

1. Si la solution est entière alors c'est la deuxième meilleure solution, noter la x^2.

2. Sinon ajouter la coupe de Gomory pour trouver la solution. Soit $x^2 = (\bar{x}_1, \bar{x}_2)$ la 2ème meilleure solution entière. Aller à l'etape 2.

Etape Générale : commence par la kième meilleure solution x^k et introduire la coupe : $\sum_{j \in T(K)} x_j \geq 1.$

En utilisant le dual simplex et la méthode coupe de Gomory, trouver la (k+1)ème meilleure solution, aller à l'étape2 jusqu`à l'obtention d'une solution pour le problème de programmation à deux niveaux.

Théorème4 :

L'algorithme proposé trouve une solution optimale entière pour le problème (P_1) en un nombre fini d'étapes dans le cas ou la solution existe.

Preuve :

Comme l'ensemble des contraintes est borné et tronqué par des applications répétées d'une coupe correspondante à la solution trouvée, suivies éventuellement par l'application de la coupe Gomory, le domaine S tout entier est balayé de telle manière que les points tronqués ne puisse pas réapparaître, menant à la convergence de la méthode en un nombre fini d'étapes.

L'efficacité de cet algorithme dépend de la méthode de Gomory.

Exemple illustratif 4.1

$$
(\mathrm{P}) \quad
\begin{array}{l}
\min\limits_{x_2} F(x_1, x_2) = -x_1 - 2x_2 \\
\quad \text{sujet à} \quad \min\limits_{x_1} f(x_1, x_2) = 4x_1 - x_2 \\
\qquad \qquad \text{sujet à} \quad x_1 + x_2 \le 7 \\
\qquad \qquad \qquad 2x_1 \le 11 \\
\qquad \qquad \qquad 2x_2 \le 7 \\
\qquad x_1, x_2 \ge 0, x_1, x_2, \textit{entiers}
\end{array}
$$

On considère le problème suivant :

$$
\begin{array}{l}
\min\limits_{x_2} F(x_1, x_2) = -x_1 - 2x_2 \\
\quad \text{sujet à} \quad x_1 + x_2 \le 7 \\
\qquad 2x_1 \le 11 \\
\qquad 2x_2 \le 7, \qquad x_1, x_2 \ge 0, \textit{entiers.}
\end{array}
$$

La résolution de ce problème donne la solution $(\bar{x}_1, \bar{x}_2) = (4,3)$, la solution est entière, alors on passe à l'étape 2 de l'algorithme, i .e,

Soit le problème

$$
\begin{array}{l}
\min\limits_{x_1} f(x_1, x_2) = 4x_1 - x_2 \\
\quad \text{sujet à} \quad x_1 + x_2 \le 7 \\
\qquad 2x_1 \le 11 \\
\qquad 2x_2 \le 7
\end{array}
$$

On pose $x_2 = 3$, le problème devient :

$$\min_{x_1} f(x_1, x_2) = 4x_1 - 3$$
$$\text{sujet à} \quad x_1 \le 4$$
$$2x_1 \le 11 \qquad x_1, x_2 \ge 0, entiers$$

La résolution du ce problème donne la solution $(\tilde{x}_1, \tilde{x}_2) = (0, 3)$.

Comme $\tilde{x}_1 \ne \bar{x}_1$, alors$(4,3)$ n'est pas solution optimale du problème. On passe à l'étape3 du problème, i.e, on ajoute la contrainte $\sum_{j \in T(k)} x_j \ge 1$ au tableau optimale correspondant à la solution $(4,3)$.une $2^{ème}$ solution est obtenue $(\bar{x}_1, \bar{x}_2) = (3,3)$, on passe à l'étape 2 de l'algorithme, on pose $x_2 = 3$ et on obtient le problème suivant :

$$\min_{x_1} f = 4x_1 - 3$$
$$x_1 \le 4$$
$$2x_1 \le 11$$
$$x_1 \ge 0, x_2 \ge 0, entiers.$$

La résolution du ce problème donne $(0,3)$; $\tilde{x}_1 \ne 3$, aller à l'étape3 du problème.

On obtient la solution $(\bar{x}_1, \bar{x}_2) = (5,2)$, on passe à l'étape 2,

On pose : $\bar{x}_2 = 2$, on obtient, le problème second niveau suivant :

$$\min_{x_1} f(x_1, x_2) = 4x_1 - 2$$
$$x_1 \le 5$$
$$2x_1 \le 11$$

dont la solution est $(\tilde{x}_1, \tilde{x}_2) = (0, 2)$.

Donc $\bar{x}_1 \neq \tilde{x}_1$ .On passe à l'étape 3 de l'algorithme. le balayage du domaine se continu, après 5 itérations, on obtient la solution $(\bar{x}_1, \bar{x}_2) = (0, 3)$. Allert à l'étape 2,

on pose $x_2 = 3$, on aura le problème suivant :

$$\min_{x_1} f(x_1, x_2) = 4x_1 - 3$$
$$x_1 \leq 4$$
$$2x_1 \leq 11$$

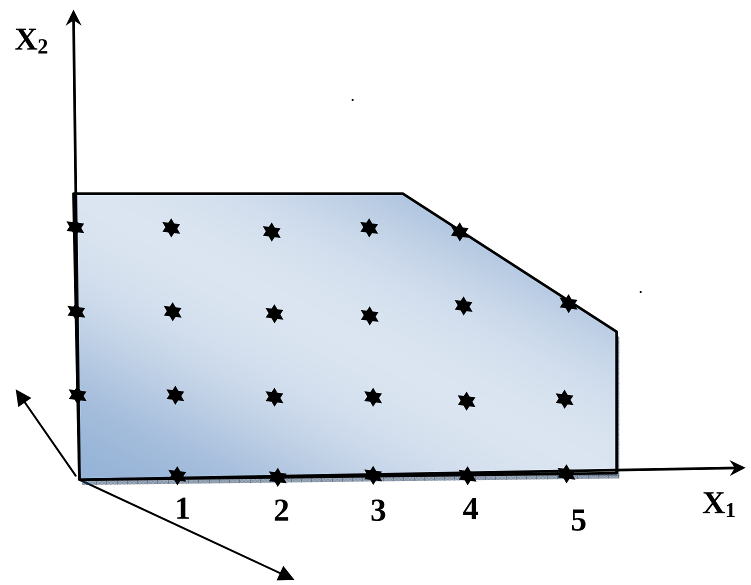
La résolution de ce problème donne $(\tilde{x}_1, \tilde{x}_2) = (0, 3)$

On a : $\tilde{x}_1 = \bar{x}_1$

Donc : $(0, 3)$ la solution optimale du problème.

Remarque : cette méthode peut être appliquée dans le cas d'un problème de programmation à deux niveaux fractionnaire linéaire discret.

<u>Résolution graphique de l'exemple **4.1 :**</u>



L'ensemble des solutions réalisables du problème second niveau au point x_2 est :

$$\Omega(x_2) = \left\{ x_1, \ x_1 \leq 7 - x_2, x_1 \leq \frac{11}{2}, \quad x_1 \geq 0, entier \right\}$$

$$= \{x_1, x_1 \leq \min\{7 - x_2, 5\}\}$$

$$= \begin{cases} x_1 \leq 5 & si \ x_2 = 0,1,2 \\ x_1 \leq 7 - x_2 & si \ x_2 = 3 \end{cases}$$

L'ensemble de solution optimale du problème second niveau au point x_2 :

$$M(x_2) = \{x_1, \quad x_1 = 0\}$$

Le domaine induit :

$$DI = \{(0,0),(0,1),(0,2),(0,3)\}$$

Donc la solution optimale est $(0,3)$, $F^* = -6$.

CONCLUSION GENERALE

La programmation à deux niveaux (BPP) possède un champ vaste d'applications. De grands efforts ont été déployés pour la mise au point de plusieurs méthodologies d'approche. Le problème PDN peut être linéaire, fractionnaire linéaire, quadratique ou mixte. En présence de variables continues, une panoplie d'algorithmes pour la résolution du problème existent dans la littérature, alors que le cas discret n'a pas eu l'attention nécessaire de la part des chercheurs ressentie à travers les recherches bibliographiques.

Dans une première étape, nous avons étudié les concepts fondamentaux de la PDN générale. Nous nous sommes intéressés par la suite au problème PDN linéaire en variables continues et discrètes. On a développé un algorithme pour la résolution d'un problème PDN linéaire en nombres entiers. La méthode consiste à trouver une solution optimale du problème constitué de la fonction objectif du meneur et des contraintes du premier et second niveau. Si la solution appartient au domaine induit alors elle est solution optimale du problème de la PDN. Dans le cas contraire, l'algorithme passe à la recherche de la deuxième meilleure solution du problème en introduisant une coupe au niveau du tableau optimal précédent. Si la solution trouvée appartient au domaine induit, elle est optimale pour le problème de la PDN. Sinon l'algorithme passe à la recherche de la troisième meilleure solution et vérifier son appartenance au domaine induit. La procédure se poursuit jusqu`à l`obtention d'un point qui optimise le problème du premier et second niveau à la fois. Des extensions possibles de la méthode au cas non linéaire pourront être les perspectives de cette thèse à savoir le problème de la programmation à deux niveaux fractionnaire linéaire, quadratique.

BIBLIOGRAPHIE

[1] M. Abbas and M. Moulaï (1999), Solving multiple objective integer linear programming, *Ricerca Operativa*, Vol. 29 n° 89, pp 15-38.

[2] J. Bard (1983), Coordination of multidivisional organization through two levels of management, *OMEGA*, **11** pp 457–468,.

[3] F. Bard (1984), Optimality conditions for the bilevel programming problem. *IEEE ansf, system men and cybernetics* vol smc 14, no 5, pp 711-717.

[4] F. Bard and J. Moore (1990), A branch and bound algorithm for the bilevel programming problem, *SIAM journal of scientific and statistic computing*, vol**11**, n° 02, pp 281-292.

[5] F. Bard (1998), Practical bilevel optimization algorithm and applications, Kulwer Academic Publishers, vol **30**.

[6] O. Ben-Ayed, D. Boyce and C. Blair (1988), A general Bilevel Linear Programming formulation of the net work design problem, *Transportation Research*, **22B**, pp 311-318.

[7] O. Ben Ayed, C. Blair, D. Boyce and L. LeBlanc (1992), Construction of a real – world bilevel programming problem. *Annals of operations Research*, **34**, pp 219-254.

[8] W. Bialas, M. Karwan and J. Shaw, A parametric complementary pivot approach for two level linear programming

[9] W. F. Bialas and M. H. Karwan (1982), Two level optimization, *IEEE transformation automatic*, vol AC **27**, no 1, pp 211-214.

[10] J. Bracken and Mc Gill (1973), Mathematical programs with optimization problems in the constraints, *Operation Research*, 21: 37-44.

[11] W. Candler and R. Norton (1977), Multilevel Programming. *Technical Report* **20**, world Bank Development Research center, Washington D.C.

[12] W. Candler and R. Tounsley (1982), A linear two level programming problem, *Computers and Operation research*, vol **9**, no1, pp 59-76.

[13] S. Dempe (1996), A bundle algorithm applied to bilevel programming problems with non- unique lower level solution, *Computational optimization and* applications, **6**, pp 227-249.

[14] S. Dempe (1999), Generalized PC^1- functions , *Optimization*, 46, 311- 326

[15] S. Dempe(1995), Discrete bilevel optimization problems, *working paper*, TU chemnitz.

[16] S. Dempe (2005), Bilevel programming. Inc;Audet, P.Hansen; and G. savard(Eds): Essays and surveys in global optimization Springer 165-193.

[17] J.Fortuny-Amat and B-Mc Carl (1981), a representation and economic interpretation of two level programming problems. *Journal of operational research society*, vol 32, pp 783-792.

[18] Carmen Galé, Herminia and I. Calvete (1999), The bilevel linear/linear fractional programming, *European journal of operational research* letters **114**, 188-197.

[19] Carmen Galé, Herminia and I. Calvete (2004), Solving linear fractional bilevel programs, *Operation Research letters*, **32**, 143-151.

[20] M.R.Garey and D.S.Johnson (1979),Computers and Intractability:A Guide to the Theorey of NP-Completeness,W.H.Freeman and Company,New York.

[21] P. Hansen, B. Jaumard and G. Savard (1992), New branch and bound rules for linear bilevel. *SIAM journal of scientific and statistical computer*, vol **13**, no 5, pp 1194–1217.

[22] R.G. Joroslow (1985), The polynomial Hierarchy and a simple method for competitive, Mathematical programming, vol. **32**, pp 146 -164.

[23] J. Judice and A. M. Faustino (1992), A Sequential LCP Method for bilevel linear programming, *Annals of Operations Research*, vol. 34, no.1-4, pp 89-106.

[24] Rachid Kihel, Ivan Arciniegas, programmation mathématique à deux niveaux.

[25] P.Marcotte and G.Savard (1991), A note on the Pareto Optimality of Solutions to the Linear Bilevel programming Problem, “ Computers and Operations Research,Vol18,No.4,pp.355-359.

[26] H. Stacklberg, the therory of the market economy, Oxford University Press, 1952.

[27] Luis. Vicente (1997), Bilevel programming introduction, History and overview.

[28] D.J .White, (1984),A Branch and Bound Method for Multi-Objective Boolean Problem, *European Journal of Operational Research*, **15**, pp 126–130.

[29] E. Aiyoshi and K. Shimizu (1984) ; Asolution method for the static constrained Stackelberg problem via penalty method. Transaction on automatic control ,vol 29 no 12 pp 1111- 1114.

ANNEXE

LA PROGRAMMATION A DEUX NIVEAUX APPLIQUEE AU PROBLEME D'ORDONNANCEMENT FLOW SHOP

Définition :

Ordonnancer c'est programmer l'exécution d'un ensemble de taches en attribuant des ressources aux taches et en fixant leurs dates d'exécution.

Les taches peuvent représenter des pièces, projets, des programmes ou des activités.

Les ressources peuvent représenter des machines industrielles, des processeurs ou des personnes.

Dans un system flow shop les taches doivent être traités par toutes les machines dans un même ordre. Dans cet exemple, le problème flow shop est un peu modifié : la prise de décision est hiérarchique et il existe m opérateurs.

Le propriétaire représente le meneur, il affecte les opérateurs aux machines dans le but de minimiser son objectif qui est le temps flot total (somme du temps d'attente et du temps de traitement).

Le suiveur est le client, il possède n taches qui doivent être traités, il choisit l'ordre de traitement dans le but de minimiser son objectif qui est la longueur de l'ordonnancement. Donc le problème peut être formulé comme un problème de programmation à deux niveaux.

soient

$mk_{i,j}$ = longueur de l'ordonnancement associé à l'ordonnancement i des opérateurs et l'ordonnancement j des taches.

$fl_{i,j}$ = temps flot associé à l'ordonnancement i des opérateurs.

$$k_i = \begin{cases} 1 & \text{si l'ordonnancement } i \text{ des opérateurs est choisi} \\ 0 & \text{si non.} \end{cases}$$

$$r_j = \begin{cases} 1 & \text{si l'ordonnancement } j \text{ des taches est choisi} \\ 0 & \text{si non.} \end{cases}$$

considérons n taches et m opérateurs, donc le suiveur possède n ! possibilités pour arranger les taches. et m ! ordonnancements des opérateurs.

La formulation du problème est la suivante :

$$\min \sum_{i,j} fl_{ij} r_j k_i$$

$$\min \sum_{i,j} mk_{ij} r_j k_i$$

$$\text{sujet à} \quad \sum_{j=1}^{n!} r_j = 1$$

$$\sum_{i=1}^{m!} k_i = 1$$

$$k_i \in \{0,1\}, \quad i = 1,.....m!$$

$$r_j \in \{0,1\}, \quad j = 1,....n!$$

Exemple :

Pour comprendre la procédure, un simple exemple est présenté.

Considérons trois machines, trois opérateurs et trois taches. Chaque ligne du tableau ci-dessous représente la longueur d'ordonnancement optimale et le temps flot correspondant à l'ordre des taches (représentés en tête de colonne)

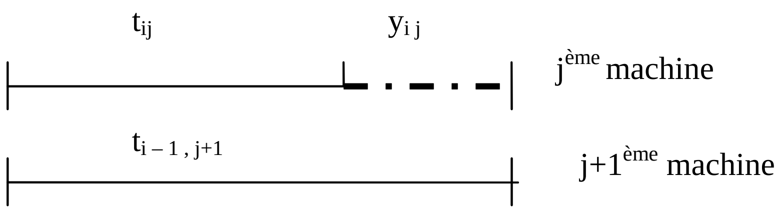
	123		132		213		231		312		321	
	mk	*fl*	*mk*	*fl*	*mk*	*fl*	*mk*	*fl*	*mk*	*fl*	*mk*	*fl*
abc	49	84	49	89	51	83	51	86	**46**	**81**	4 8	86
acb	48	83	50	91	51	86	53	89	**47**	**83**	49	86
bac	**46**	**93**	50	100	47	79	52	101	49	97	49	97
bca	47	82	47	88	50	82	48	82	**44**	**82**	47	90
cab	45	76	47	89	48	83	47	82	**43**	**82**	46	91
cba	45	81	49	89	<u>**44**</u>	<u>**80**</u>	49	88	47	85	46	86

ALGORITHME:

Définitions:

Soit t_{ij} le temps du traitement d'une tache en $i^{ème}$ position sur la $j^{ème}$ machine.

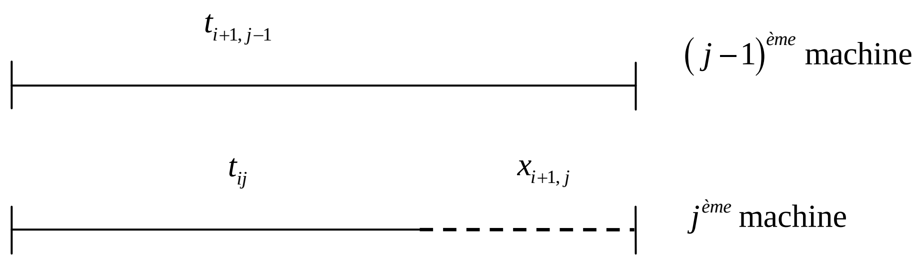
Si le traitement de la tache en $i^{ème}$ position est terminé sur la machine j , et que la $\left(j+1\right)^{ème}$ machine soit toujours occupée , alors cette tache attendra sur la $j^{ème}$ machine jusqu'à la $\left(j+1\right)^{ème}$ soit libre.



l'intervalle du temps y_{ij} est dit temps libre de la tâche . y_{ij} est le temps libre de la tâche en $i^{ème}$ position , entre la fin de son traitement sur la machine j et le début de son traitement sur la machine $j+1$.

Comme il n'y a pas du temps libre des taches sur la dernière machine , alors :
$y_{im} = 0, \quad$ pour $1 \leq i \leq m$.

Dans le cas ou le traitement de la tache en $i^{ème}$ position est terminé sur la $j^{ème}$ machine, cette
dernière passe à la machine $j+1$. $\left(\text{si } j = m \text{ alors son traitement est achevé.}\right)$; tan disque la tâche en position $i+1$ occupe toujours la machine $j-1$; donc on aura un temps machine libre.



La $j^{ème}$ machine reste libre jusqu`a ce que la tâche en$(i+1)^{ème}$ position finisse son traitement sur la machine $(j-1)$. L'intervalle du temps , noté $x_{i+1, j}$ est dit temps machine libre. $x_{i+1, j}$ est un temps libre sur la machine $j$ , depuis la fin du traitement de la tâche en position $i$ sur la machine $j$ , jusqu` a le début de la tâche en position $i+1$ sur la machine j .

Sur la première ma chine, le temps machine libre n'existe pas , donc $x_{i1} = 0, i = 1, ..., n.$
Le diagramme dessous représente l'ordonnancement de traitement de 6tâches sur 4 machines. (pour illustrer x_{ij} *et* y_{ij}).

makespan (longueur d'ordonnancement **) :** c'est la durée depuis le début de traitement de la première tâche sur la première machine jusqu `a le traitement de la dernière tâche sur la dernière machine.

à partir de la figure 1 on peut constater que

$$mk = \sum_{i=1}^{n}\left(x_{im} + t_{im}\right) = \sum_{i=1}^{n} x_{im} + \sum_{i=1}^{n} t_{im} \ .$$

pour minimiser la longueur d'ordonnancement (makespan) il suffit de minimiser $\sum_{i=1}^{n} x_{im}$

car $\sum_{i=1}^{n} t_{im}$ est constante (indépendante de l'ordre de traitement des tâches).

Flowtime (temps flot) de chaque tâche **:** c'est la durée depuis le début de traitement de la tâche sur la première machine jusqu `a la fin de son traitement sur la dernière machine.

Le flow time total est la somme des flow time des tâches,on le note *fl*.

d'après la figure 1 on peut voir :

- Le temps flot de la tâche en première position est :

$$fl_1 = x_{1m} + t_{1m.}$$

- le temps flot de la tache en deuxième position est :

$$fl_2 = x_{1m} + t_{1m} + x_{2m} + t_{2m}$$

- Le temps flot de la tâche en n^ième^ position est :

$$fl_n = x_{1m} + t_{1m} + x_{2m} + t_{2m} + ...+ x_{nm} + t_{nm}$$

donc :
$$\begin{aligned} fl &= fl_1 + fl_2 + ...+ fl_n \\ &= n\left(x_{1m} + t_{1m}\right) + \left(n-1\right)\left(x_{2m} + t_{2m}\right) + ...+ x_{nm} + t_{nm} \\ &= \sum_{i=1}^{n}\left(n+1-i\right)\left(x_{im} + t_{im}\right) \end{aligned}$$

Algorithme 1 **:** calcul de x_{ij} et y_{ij}

1. Le temps machine libre sur la première machine n'existe pas sur la première machine.

$x_{i1} = 0 \quad i = 1,..,n.$

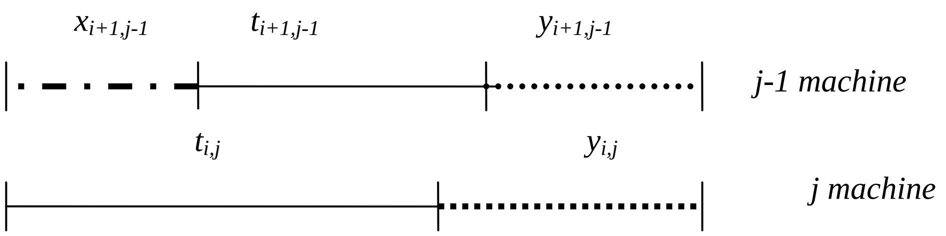
2. Le temps tâche libre sur la dernière machine n'existe pas. $y_{im} = 0, \quad i = 1,....,n.$

3. Le temps tâche libre de la tâche en 1[ère] position est nul , $y_{1j} = 0, \quad j = 1,......,m.$

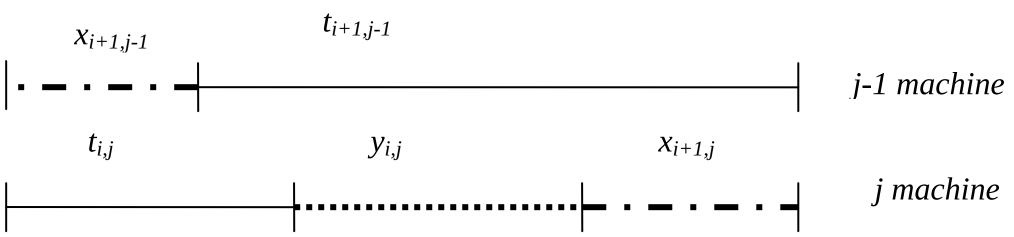
4. $x_{1j} = \sum_{i=1}^{i=j-1} t_{1i} \qquad j = 2,....,m.$

5. Si $t_{ij} + y_{ij} \geq x_{i+1,j-1} + t_{i+1,j-1}$

$$y_{i+1,j-1} = \left(t_{ij} + y_{ij}\right) - \left(x_{i+1,j-1} + t_{i+1,j-1}\right).$$



6. Si $t_{ij} + y_{ij} < x_{i+1,j-1} + t_{i+1,j-1}$ alors $x_{i+1,j} = \left(x_{i+1,j-1} + t_{i+1,j-1}\right) - \left(t_{ij} + y_{ij}\right)$



Algorithme1 :

Etape1 : $x_{ij} = 0, y_{ij} = 0$ pour $i = 1,....,n$, j =1,....,m

Etape2 : $x_{1j} = \sum_{k=1}^{j-1} t_{1k} \quad j = 2,.....,m$ **.**

Etape3 : Faire $i = 1, \quad j = 1.$

Etape4 : si

$$t_{i,j+1} + y_{i,j+1} - t_{i+1,j} - x_{i+1,j} \geq 0, \text{ alors}$$
$$y_{i+1,j} = t_{i,j+1} + y_{i,j+1} - t_{i+1j} - x_{i+1,j}$$
$$\text{si non} \quad x_{i+1,j+1} = t_{i+1,j} + x_{i+1,j} - t_{i,j+1} - y_{i,j+1}.$$

Etape5 : si $j < m$, poser $j = j + 1$ aller à l'etape 4.

Etape6 : si $i < n$, poser $i = i + 1, \quad j = 1$ aller à l'etape4

Etape 7 :stop.

***<u>Algorithme2</u>* :** minimiser la durée d'ordonnancement .

Pour un ordonnancement d'opérateurs donné et *n* tâches à traiter il existe *n* ! façons d'arranger ces tâches. Notre objectif est de trouver un ordre dont la longueur d'ordonnancement est minimale. Une méthode séparation et évaluation est proposée.

Algorithme 2 :

Initialisation : poser ***k*** égal au *1^er^ ordre* des tâches. Calculer tous les x_{ij} et y_{ij} *en* utilisant l'algorithme 1. et calculer aussi mk_k , fl_k .

Etape1 : Si k dépasse le dernier ordre , stop.

Etape2 : Faire $i = 1,n$.

Appliquer l'algorithme 1 pour trouver x_{im}.

Poser

$$mk_i = \sum_{p=1}^{i} x_{pm} + \sum_{p=1}^{m} t_{pm}$$

Si $mk_i > mk_c$ then $k = k+1$. aller à l'etape 1.

Faire,

$mk_k = mk_n$.

Etape3 :trouver fl_k ;

Si $mk_k < mk_c$ alors

$mk_c = mk_k$, $fl_c = fl_k$,

$K = k; k = k+1$, aller à l'etape1.

Si $mk_k = mk_c$ et $fl_k < fl_c$ alors

$mk_c = mk_k$, $fl_c = fl_k$,

$K = k; k = k+1$, aller à l'etape 1.

Algorithme3 **:** Solution du problème de la PDN.

l'ensemble de solution réalisable du meneur est l'ensemble des couples constitués du meilleur makespan et du flowtime, correspondant à un ordonnancement d'opérateurs donné. la solution optimale du problème est le couple dont le temps flot est le plus petit.

Rappelons que $fl = \sum_{i=1}^{n} (n+1-i).(x_{im} + t_{im})$. supposons les tâches sont ordonnées on position 1,2,…..j.

On définit $s(j,i)$ le $i^{ème}$ plus petit temps de traitement sur la machine m des tâches restantes.

Soit $\tilde{fl}_j = \sum_{i=1}^{j}(n+1-i)(x_{im}+t_{im}) + \sum_{i=j+1}^{n}(n+1-i)s(j,i-j)$

Algorithme3 :

Initialisation : Poser $l =$ le premier ordonnancement des opérateurs. Trouver la meilleure durée d'ordonnancement en utilisant l'algorithme2, calculer le temps flot correspondant.

Poser $(\overline{mk}, \overline{fl}) = (mk_c, fl_c)$.

Etape1 : poser $l = l+1$ et $k =$ la première sequence des taches .

Etape 2 : Faire $j = 1, n.$

Appliquer l'algorithme i pour calculer x_{jm} .

supposer $k =$ la séquence $\{k_1, k_2, ..., k_n\}$.

Si $\tilde{fl} > \overline{fl}$ alors

supprimer toutes les branches qui commence avec $\{k_1, k_2, ...k_j\}$.

Poser $k =$ à la séquence de tache suivante. Si $k$est supérieure à la dernière séquence aller à l'etape1;Si non aller à l'etape2.

Fin si.

Fin do.

Etape3 : Si $\tilde{fl}_n < \overline{fl}$ appliquer l'algorithme2 pour cet ordonnancement d'opérateurs.