

IUT de Montpellier, département Informatique

Année universitaire 2023-2024

AIDE A LA DECISION

(BUT 3, semestre 5)

- 1. Introduction à l'optimisation linéaire – Méthode du simplexe**
- 2. Problème du sac à dos. Programmation linéaire en nombres entiers**

Introduction à l'optimisation linéaire

(ou “programmation” linéaire)



1) Modélisation :

En **recherche opérationnelle**, modéliser un problème consiste à identifier :

- ▶ les variables (i.e. les inconnues)
- ▶ les contraintes portant sur ces variables
- ▶ l'objectif

Dans un problème d'**optimisation linéaire** les contraintes et l'objectif sont des **fonctions linéaires** des variables.



Exemple avec deux variables réelles :

- ◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ 🔍 ↺

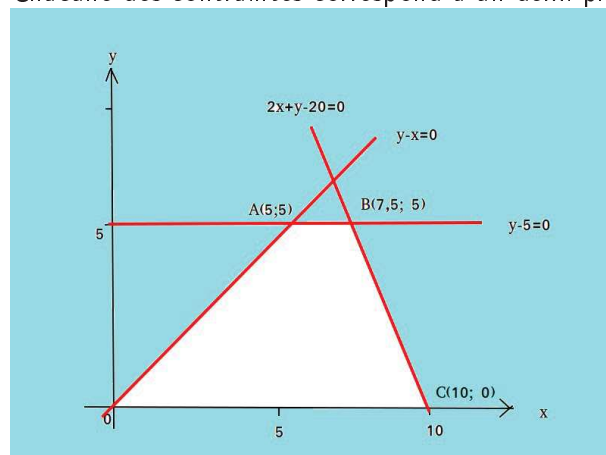
- ◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ↺ 🔍 ↻

2) Résolution graphique :

On peut résumer le problème précédent ainsi :

$$\left\{ \begin{array}{l} f(x, y) = x + y \\ 2x + y - 20 \leq 0 \\ y - 5 \leq 0 \\ y - x \leq 0 \\ x \geq 0 \\ y \geq 0 \end{array} \right.$$

on veut maximiser la fonction f

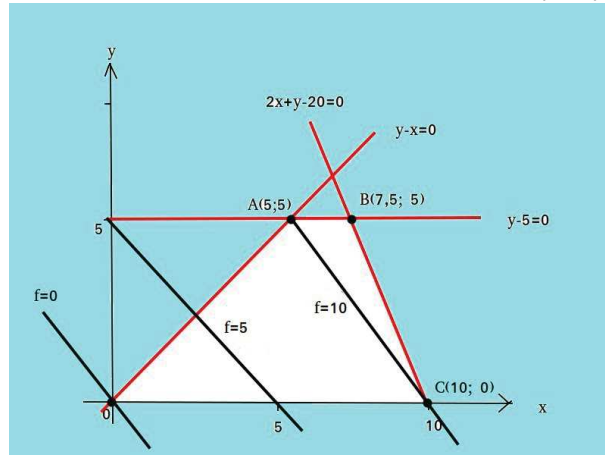


La zone blanche correspond aux points du plan vérifiant toutes les contraintes.

C'est un **polyèdre convexe** ("le polyèdre des contraintes")



La fonction $f(x, y)$ à maximiser peut être représentée par un ensemble de droites parallèles (équation : $f(x, y) = \text{constante}$)



Solution graphique : le point qui maximise f est le point B (7,5kg de carottes et 5kg de pommes de terre)

Navigation : retour, avant, après, fin, recherche, etc.

Remarque fondamentale :

Pour un problème d'optimisation linéaire à 2 variables réelles, **un au moins des sommets du polyèdre convexe est solution.**

On peut généraliser cette remarque pour 3, 4, ... variables.

Navigation : retour, avant, après, fin, recherche, etc.

3) Algorithme du simplexe à partir d'un exemple :

Cet algorithme permet de trouver un sommet du polyèdre convexe où est atteint le maximum de la fonction f .

Principe :

On part d'un sommet du polyèdre des contraintes (souvent le point O s'il fait parti du polyèdre) puis on va vers un sommet voisin qui permet d'augmenter f et ainsi de suite. Lorsqu'aucun des sommets voisins ne permet d'augmenter f l'algorithme est terminé (le maximum local est un maximum absolu).



Simplexe : exemple sur un cas très simple

On considère le problème suivant où la fonction f doit être maximisée :

$$\begin{cases} f(x_1, x_2) = x_1 + 2x_2 \\ x_1 + x_2 \leq 5 \\ x_1, x_2 \geq 0 \end{cases}$$

Pour pouvoir utiliser l'algorithme du simplexe le problème doit toujours être posé sous la forme précédente : une fonction f à **maximiser**, des variables **toutes positives ou nulles** et des contraintes sous la forme d'inéquations du type $ax_1 + bx_2 + cx_3 + \dots \leq \beta$.)



On introduit 1 **variable d'écart** x_3 positive ou nulle :

$$x_3 = 5 - x_1 - x_2$$

remarque : la contrainte $x_1 + x_2 \leq 5$ équivaut à $x_3 \geq 0$.



Le problème peut donc s'écrire plus simplement :

maximiser f avec les contraintes $x_1, x_2, x_3 \geq 0$.

On peut résumer les données ainsi :

Dictionnaire 1

$$x_3 = 5 - x_1 - x_2$$

$$f = x_1 + 2x_2$$

Les variables x_1, x_2 sont **hors base** tandis que la variable x_3 est dans la **base**.



Solution basique associée au dictionnaire :

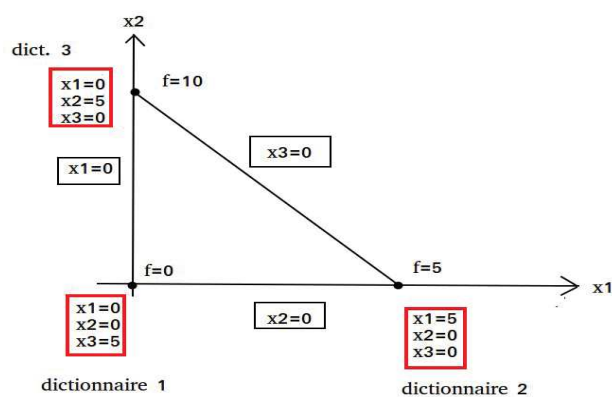
C'est la solution obtenue en donnant la valeur 0 à toutes les variables hors-base.

Dans notre exemple, la solution basique associée au **dictionnaire 1** est $x_1 = 0, x_2 = 0, x_3 = 5$.

On constate que pour cette solution toutes les variables sont positives ou nulles donc le **dictionnaire 1 est réalisable**.

Pour cette solution on a $f = 0$

Navigation icons: back, forward, search, etc.



Lorsqu'on passe d'un sommet au suivant :

1. la fonction f augmente (au sens large)
2. Une variable nulle (hors base) devient non nulle (dans la base)
3. Une variable non nulle (dans la base) devient nulle (hors base)

Navigation icons: back, forward, search, etc.

L'algorithme :

On passe successivement d'un dictionnaire à un autre par le mécanisme suivant :

1. on choisit une variable x **hors-base** dont le coefficient dans f est strictement positif (donc cette variable en augmentant fera augmenter f). Si une telle variable n'existe pas l'algorithme est fini.
2. Dans le dictionnaire en cours on fait entrer x dans la base et on fait sortir une variable y de la base. *La variable y choisie est celle qui s'annule la première lorsque x augmente.*
3. Si aucune variable y de la base ne s'annule lorsque x augmente, le problème n'a pas de solutions (f n'est pas borné)
4. on exprime les nouvelles variables de base et la fonction f en fonction des variables hors base. Puis on revient en 1.

« □ » « 🔍 » « ↺ » « ↻ » « ⌂ » « 🔍 » « ↺ » « ↻ » « ⌂ »

Dictionnaire 1
$x_3 = 5 - x_1 - x_2$
$f = x_1 + 2x_2$

La solution basique du dictionnaire 1 est
($x_1 = 0, x_2 = 0, x_3 = 5$) pour $f=0$

On fait entrer x_1 dans la base et on fait sortir x_3 :
 $x_3 \geq 0 \Rightarrow x_1 \leq 5$ (x_3 s'annule lorsque x_1 vaut 5)

Dictionnaire 2
$x_1 = 5 - x_2 - x_3$
$f = 5 + x_2 - x_3$

La solution basique du dictionnaire 2 est
($x_1 = 5, x_2 = 0, x_3 = 0$) pour $f=5$

« □ » « 🔍 » « ↺ » « ↻ » « ⌂ » « 🔍 » « ↺ » « ↻ » « ⌂ »

4) Simplexe : un exemple plus complet

$$\begin{cases} f(x_1, x_2, x_3) = 2x_1 + x_2 + 3x_3 \\ x_1 + x_2 + x_3 \leq 20 \\ 2x_1 + x_2 \leq 10 \end{cases} \quad (x_1, x_2, x_3 \geq 0)$$

Dictionnaire 1
$x_4 = 20 - x_1 - x_2 - x_3$
$x_5 = 10 - 2x_1 - x_2$
$f = 2x_1 + x_2 + 3x_3$

La solution basique du dictionnaire 1 est
 $(x_1 = 0, x_2 = 0, x_3 = 0, x_4 = 20, x_5 = 10)$ pour $f = 0$

Remarque : il y a deux variables dans la base : à chaque itération il faudra choisir celle qu'on enlève.

Navigation icons

Dictionnaire 1
$x_4 = 20 - x_1 - x_2 - x_3$
$x_5 = 10 - 2x_1 - x_2$
$f = 2x_1 + x_2 + 3x_3$

On fait entrer x_1 dans la base. On enlève de la base la variable la plus contraignante :

$$\begin{aligned} \dots\dots\dots x_4 \geq 0 &\Rightarrow x_1 \leq 20 \\ \dots\dots\dots x_5 \geq 0 &\Rightarrow x_1 \leq 5 \end{aligned}$$

(pour résoudre ces 2 inéquations on suppose que les autres variables hors base x_2 et x_3 sont nulles)

C'est donc la variable x_5 qui sort de la base :

Dictionnaire 2
$x_1 = 5 - \frac{1}{2}x_2 - \frac{1}{2}x_5$
$x_4 = 15 - \frac{1}{2}x_2 - x_3 + \frac{1}{2}x_5$
$f = 10 + 3x_3 - x_5$

Navigation icons

Théorème de Bland :

Il ne peut y avoir cyclage lorsque, à toute itération effectuée à partir d'un dictionnaire dégénéré, on choisit les variables entrante et sortante comme celles du plus petit indice parmi les candidats possibles.

Remarque :

Le point O n'appartient pas forcément au polynôme des contraintes : il faut parfois trouver un autre point de départ de l'algorithme (ce sujet ne sera pas traité ici).

Programmation linéaire en nombre entier (PLNE)

Résolution avec la méthode de
séparation et évaluation

1

1

1. Algorithme de séparation et évaluation

- On veut maximiser une fonction f définie sur un ensemble E discret (fini en général)

$$\begin{cases} \text{Max } f(x) \\ x \in E \end{cases}$$

- On veut évaluer ce maximum sans calculer $f(x)$ sur tous les éléments de E

2

2

Exemple : un problème de sac à dos

- Un sac à dos de capacité totale **21kg**
- 3 types d'objet :
 - Objet 1 : **4 kg** pour un bénéfice de **38 euros**
 - Objet 2: **10 kg** pour un bénéfice de **100 euros**
 - Objet 3 : **3 kg** pour un bénéfice de **27 euros**.
 - Les quantités x_1 , x_2 et x_3 de chaque produit sont **entières**.
- On veut maximiser le bénéfice :

$$f(x_1, x_2, x_3) = 38x_1 + 100x_2 + 27x_3$$

3

3

1. Algorithme de séparation et évaluation (suite)

On notera f_{\max} le maximum recherché.

Au cours de l'algorithme on notera **U** la meilleure solution en cours.

On dispose de plus d'une **fonction h** d'évaluation par excès. Cette fonction permet d'obtenir un **majorant du maximum de f** sur n'importe lequel des sous-ensembles de E.

4

4

Exemple 1 : un problème de sac à dos

. Fonction d'évaluation par excès : h

On n'impose pas aux quantités x_i d'être entières.

On calcule un majorant du bénéfice en remplissant le sac avec **uniquement le produit qui rapporte le plus par unité de poids.**

5

5

Exemple 1 : problème de sac à dos

- . Le produit 1 rapporte $38/4 = \underline{9,5 \text{ euros par kg}}$
Le produit 2 rapporte $100/10 = \underline{10 \text{ euros par kg}}$
Le produit 3 rapporte $27/3 = \underline{9 \text{ euros par kg.}}$
- . On peut mettre une quantité maximale
 $x_2 = 21/10 = 2,1$ de produit 2 (le plus rentable)
- . On obtient l'évaluation du majorant h sur
l'ensemble E de tous les sacs à dos possibles : $h(E) = 2,1 \times 100 = 210 \text{ euros}$

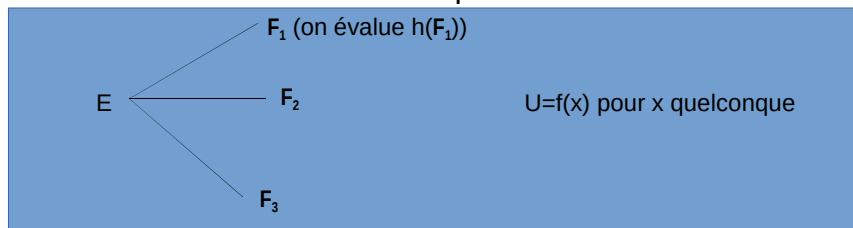
6

6

1. Algorithme de séparation et évaluation (suite)

. Première étape :

- on initialise U avec une valeur quelconque de f (ou alors $U=0$ si f est positive)
- On sépare E en plusieurs sous-ensembles F_1, F_2, \dots, F_n (qui forment une partition de E).
- On évalue h sur un des F_i



7

1. Algorithme de séparation et évaluation (suite)

. Première étape (suite) :

- Si $h(F_i) \leq U$, alors on élague F_i .
- Si $h(F_i) = f(x)$ (pour un x dans F_i), alors on élague F_i et on réévalue U ($U=\text{Max}(h(F_i), U)$).
- Si F_i est un singleton x , alors on élague F_i et on réévalue U ($U=\text{Max}(U, f(x))$)

Élagage de F_i :

on arrête de rechercher le maximum de f sur F_i .
Ce sous-ensemble ne sera plus « séparé ».

8

8

Exemple 1 : problème de sac à dos

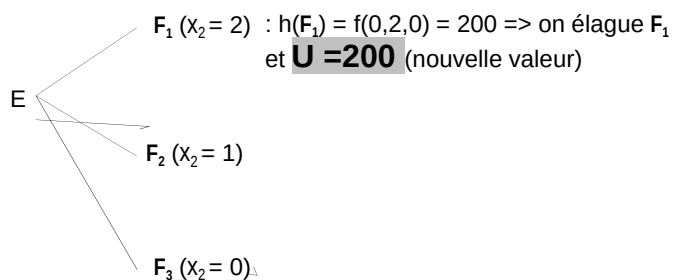
- Initialisation de U : $U = 0$
- Séparation de E :
 - F_1 : les sacs à dos tels que $x_2 = 2$
 - F_2 : les sacs à dos tels que $x_2 = 1$
 - F_3 : les sacs à dos tels que $x_2 = 0$
- (le calcul de $h(E)$ montre que $x_2 \leq 2$)

9

9

Exemple 1 : problème de sac à dos

- Évaluation de F_1 :
 - $h(F_1)$: il reste 1kg disponible donc un seul sac à dos possible : $x_1 = 0$, $x_2 = 2$ et $x_3 = 0$ donc **$h(F_1) = 200$**



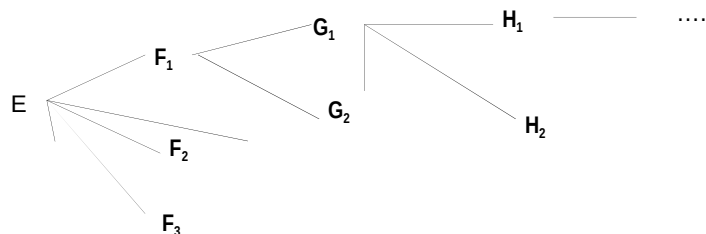
10

10

1. Algorithme de séparation et évaluation (suite)

• Deuxième étape :

- On recommence l'opération de séparation et évaluation sur un des sommets F_i non élagués.
- On obtient ainsi un arbre qu'il est préférable de parcourir **en profondeur** :



11

11

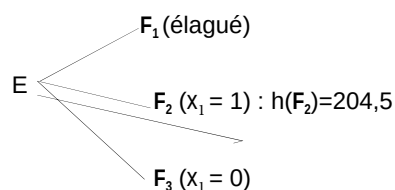
Exemple 1 : Problème de sac à dos

• Évaluation de F_2 :

- $h(F_2)$: il reste 11kg disponible que l'on remplit avec le produit 1 (plus rentable que 3) :

$$x_1 = 11/4 = 2,75 \Rightarrow h(F_2) = 1 \times 100 + 2,75 \times 38 = 204,5$$

- $h(F_2) > U$ (car $U=200$) \Rightarrow on n'élague pas F_2 .



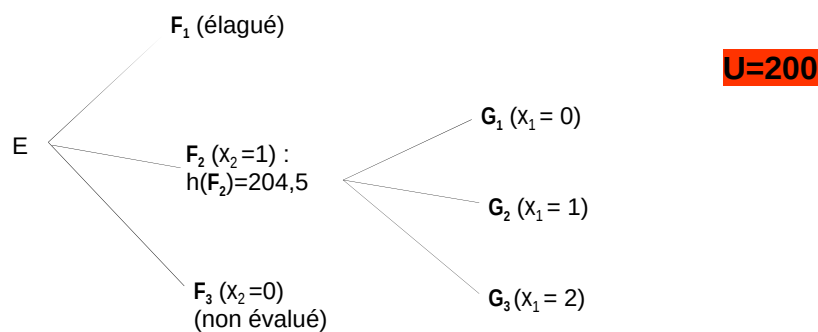
U est inchangé ($U=200$).
Prochaine étape :
 On sépare F_2 avant de passer à F_3
 (parcours en profondeur)

12

12

Exemple 1 : Problème de sac à dos

- En évaluant $h(F_2)$ on a vu que les sacs à dos de F_2 vérifient : $x_1 \leq 2$.
- On en déduit la séparation suivante de F_2 :



13

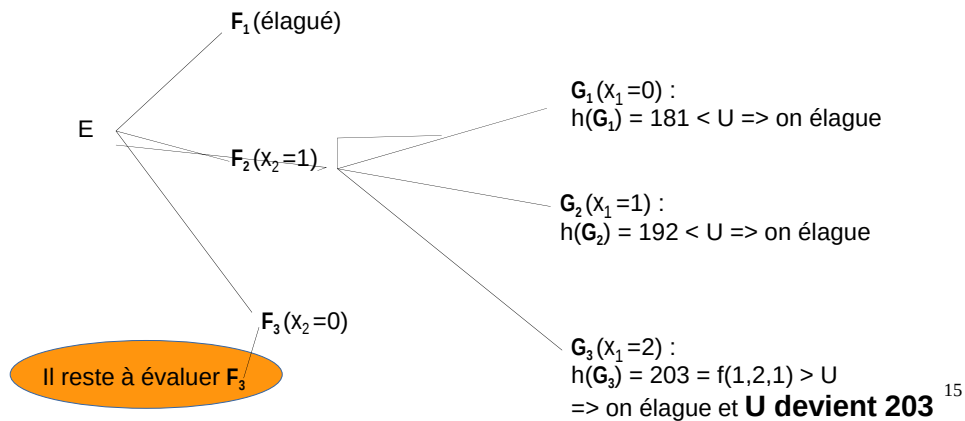
13

Exemple 1 : Problème de sac à dos

- $U=200$
- On évalue G_1 ($x_2 = 1$ et $x_1 = 0$) :
 $h(G_1) = 181 < U \Rightarrow$ on élague G_1
- On évalue G_2 ($x_2 = 1$ et $x_1 = 0$) :
 $h(G_2) = 192 < U \Rightarrow$ on élague G_2
- On évalue G_3 ($x_2 = 1$ et $x_1 = 2$) :
 $h(G_3) = 203 = f(1, 2, 1) > U \Rightarrow$ on élague G_3
et U devient 203

14

Exemple 1 : Problème de sac à dos



15

Exemple 1 : Problème de sac à dos

. Évaluation de F_3 :

- $h(F_3)$: il reste 21 kg disponible que l'on remplit avec le produit 1 (plus rentable que 3) :

$$x_1 = 21/4 = 5,25 \Rightarrow h(F_3) = 5,25 \times 38 = 199,5$$

- $h(F_3) < U \Rightarrow$ on élague F_3

. La solution est donc

$$f_{\max} = 203 \text{ pour } x_1 = 1, x_2 = 2 \text{ et } x_3 = 1$$

16

16

2. Programmation linéaire en nombre entier

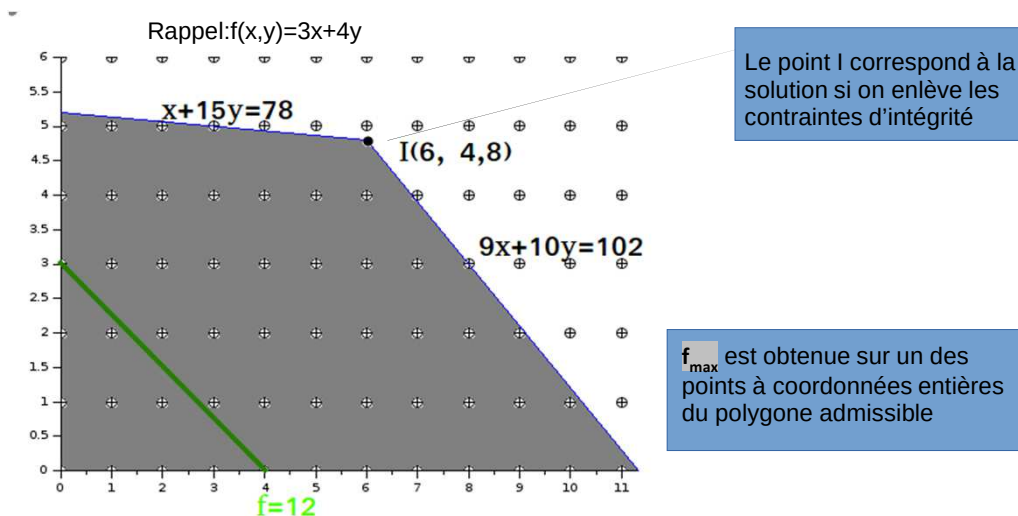
- Notation : **PLNE**
- Une PLNE se présente comme un problème de programmation linéaire classique avec en plus **des contraintes d'intégrité**.
- **Exemple 2 :**

$$P \begin{cases} f(x, y) = 3x + 4y \\ 9x + 10y \leq 102 \\ x + 15y \leq 78 \\ x \geq 0; y \geq 0 \\ x \in \mathbb{N}; y \in \mathbb{N} \end{cases}$$

17

17

Un exemple de programmation linéaire en nombre entier



18

18

2. PLNE (suite)

L'ensemble **E** est l'ensemble des points à coordonnées entières du polygone admissible.

On appellera **relaxation linéaire** du PLNE P (et on notera RP), le problème de programmation linéaire obtenu à partir de P, en supprimant les contraintes d'intégrité.

On notera Rf_{max} , le maximum de la fonction f pour le problème RP. On a donc : $Rf_{max} \geq f_{max}$

On peut calculer Rf_{max} par l'algorithme du simplexe ou graphiquement.

19

19

Pour trouver f_{max} et un point de coordonnées entières réalisant f_{max} , nous allons utiliser une méthode de « séparation et évaluation ».

L'algorithme de séparation et évaluation consiste à séparer l'ensemble E (c'est-à-dire à trouver une partition $E = F_1 \cup \dots \cup F_n$) puis à évaluer $Rf_{max}(F_i)$ sur chacune des parties F_i .

On continue à séparer les sous-ensembles puis à les évaluer tant que c'est nécessaire (cf. **élagage**).

Au cours de l'algorithme on maintient une valeur U égale à la meilleure solution en cours. On peut l'initialiser à 0 ou avec $f(x)$, x étant un élément quelconque de E .

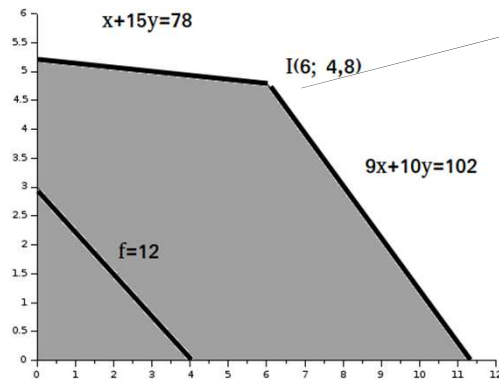
20

20

Exemple 2

Début de l'algorithme : $U=0$

(rappel : $f(x)=3x+4y$)



$$Rf_{\max} = 3 \times 6 + 4 \times 4,8 = 37,2$$



On sépare E en 2 :
 $F_2 = E \cap \{y \leq 4\}$
 $F_1 = E \cap \{y \geq 5\}$

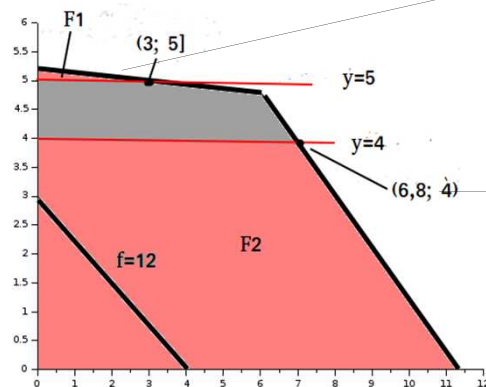
21

21

Suite de l'algorithme de séparation et évaluation sur l'exemple 2 :

(rappel : $f(x,y)=3x+4y$)

$U=0$



Sur F_1 : max en (3,5)
 $\Rightarrow Rf_{\max}(F_1) = 3 \times 3 + 4 \times 5 = 29$
 \Rightarrow on élague F_1
 et U devient 29



Sur F_2 : max en (6,9; 4)
 $\Rightarrow h(F_2) = 3 \times 6,8 + 4 \times 4 = 36,7 > U$
 \Rightarrow On sépare F_2 en 2 :
 $F_{21} = F_2 \cap \{x \geq 7\}$
 $F_{22} = F_2 \cap \{x \leq 6\}$

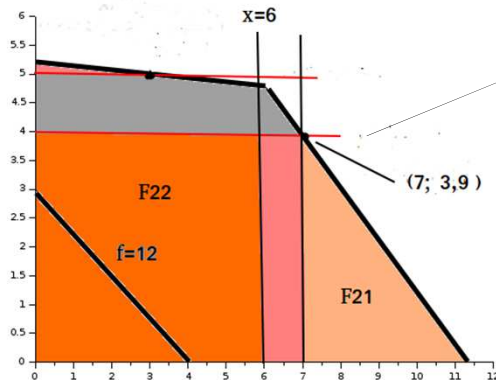
22

22

Suite de l'algorithme de séparation et évaluation sur l'exemple 2

Rappel : $f(x,y)=3x+4y$

$U=29$



Sur F_{21} :

$h(F_{21})=3 \times 7 + 4 \times 3,9 = 36,6 > U$

\Rightarrow On sépare F_{21} en 2 :

$F_{211} = F_{21} \cap \{y \leq 3\}$

$F_{212} = F_{21} \cap \{y \geq 4\} = \emptyset$ (on élague)

Sur F_{22} :

Avant d'évaluer F_{22} on

étudie F_{211}

(parcours en profondeur)

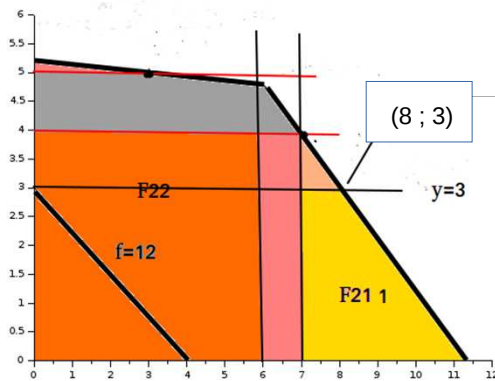
23

23

Suite de l'algorithme de séparation et évaluation sur l'exemple 2

Rappel : $f(x,y)=3x+4y$

$U=29$



Sur F_{211} : max sur (8,3)

$h(F_{211})=3 \times 8 + 4 \times 3 = 36 > U$

\Rightarrow On élague F_{211}

et U devient 36

Sur F_{22} : max en (6,4) :

$h(F_{22}) = 3 \times 6 + 4 \times 4 = 34 < U$

\Rightarrow On élague F_{22}

La solution est donc $f_{\max} = 36$ pour $x = 8$ et $y = 3$

24

24