DUT INFORMATIQUE Année 2 **Année 2020-2021**
IUT MONTPELLIER – SETE

RECHERCHE OPERATIONNELLE

Coloration valide des sommets d’un graphe et nombre chromatique

a) définitions

- On appelle **coloration** des sommets d'un graphe toute attribution d'une couleur à chaque sommet. Une coloration utilisant k couleurs est appelée une **k-coloration**.

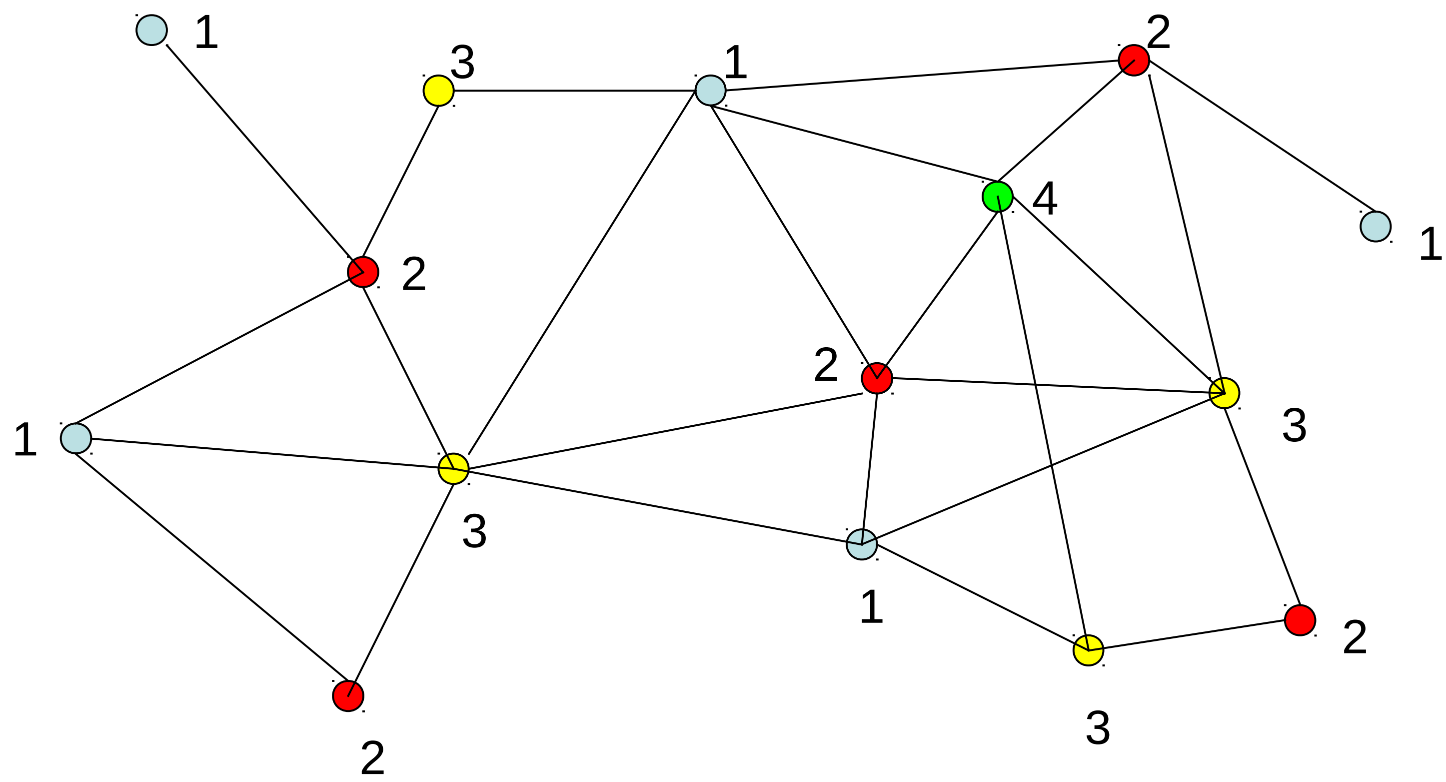
- Une coloration est **valide** lorsque 2 sommets adjacents ont toujours des couleurs différentes.

a) définitions

- Etant donné un graphe G, on appelle **nombre chromatique** de G, le plus petit nombre de couleurs nécessaires à une coloration valide de ses sommets. On note χ**(G)** ce nombre.

- Une coloration valide qui utilise χ **(G)** couleurs est **optimale.**

Cette 4-coloration est-elle optimale?



b) Encadrements du nombre chromatique

- Si **G** est d'ordre **n** on a évidemment **$X(G)<n$**
- <u>**Propriété**</u> : Soit G un graphe et $\Delta(G)$ le degré maximum d'un de ses sommets.

Alors $\boxed{\chi(G) \leq \Delta(G)+1}$.

b) Encadrements du nombre chromatique

- Pour tout graphe G connexe qui n’est ni un graphe complet, ni un cycle impair, on a :

$$\boldsymbol{\chi(G) \leq \Delta(G)}$$

b) Encadrements du nombre chromatique

- **Propriété** :
  - si on note ω**(G)** l'ordre maximum d'un sous-graphe complet de G alors

$$\boxed{\chi(G) \geq \omega(G)}$$

c) Coloration gloutonne

Donnée : un graphe **G** et un ordre total sur ses sommets noté (x_1, x_2,…, x_i, …, x_n),

Un ensemble de couleurs {1, 2, 3, …}

Résultat : une coloration **valide** de G.

Pour i allant de 1 à n **faire**

Affecter au sommet x_i la plus petite couleur non déjà affectée à ceux des sommets x_1, x_2, …, x_{i-1}, qui lui sont adjacents.

Fin Pour

Retourner l'ensemble des sommets et les couleurs qui leur sont affectées.

c) Coloration gloutonne

- Remarques:
  - La coloration gloutonne est valide mais pas nécessairement optimale.
  - Etant donné un graphe G il peut exister des colorations qui ne sont pas gloutonnes
  - Aucune coloration gloutonne ne peut utiliser plus de $\Delta(G)+1$ couleurs.
  - Il peut être pratique d'ordonner les sommets par ordre de degrés décroissants.

Exercice:

- Soit le graphe:

  G=(X={a, b, c, d, e, f, g, h}, E={ab, ac, af, ag, bg, be, bc, ch, cd, dh, ef})

  Déterminez le nombre chromatique de ce graphe.

c) Coloration gloutonne

Proposition : Etant donné un graphe G, il existe toujours au moins un ordre sur les sommets de G tel que la coloration gloutonne calculée à partir de cet ordre soit optimale.

Problème : n ! ordres possibles. Pas de solutions globales satisfaisantes.

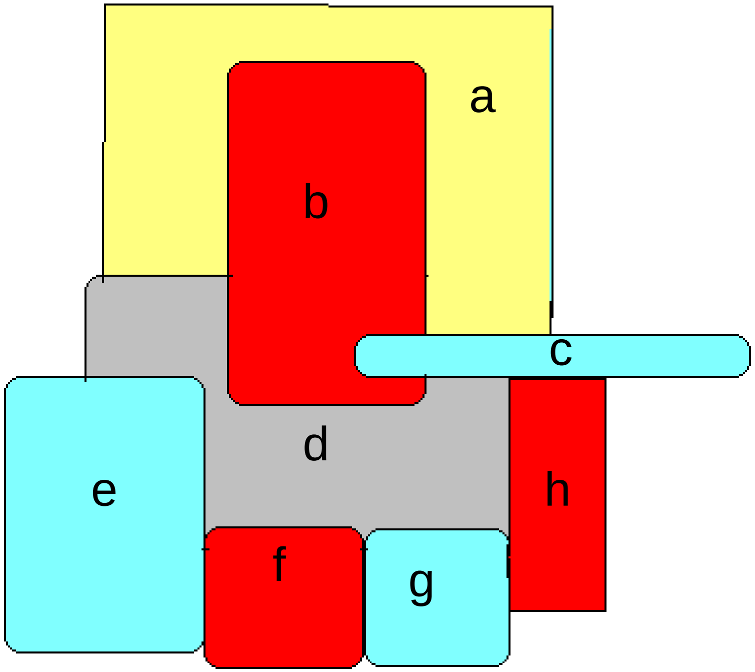
d) Coloration des arêtes

- On appelle **coloration valide des arêtes** d'un graphe toute attribution d'une couleur à chaque arête de ce graphe, de sorte que deux arêtes incidentes aient des couleurs différentes.
- Etant donné un graphe G, on appelle **indice chromatique** de G, le plus petit nombre de couleurs nécessaire à une coloration valide de ses arêtes. Ce nombre est noté $\chi'(G)$.

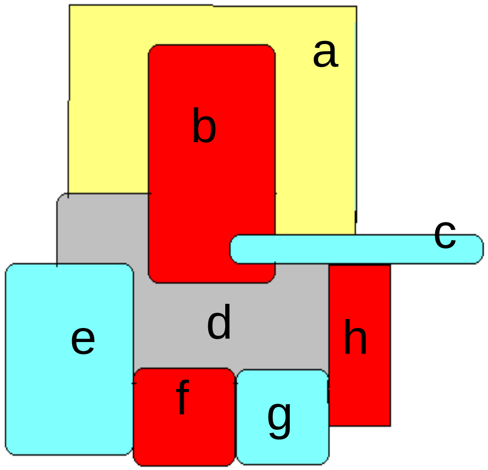
d) Coloration des arêtes

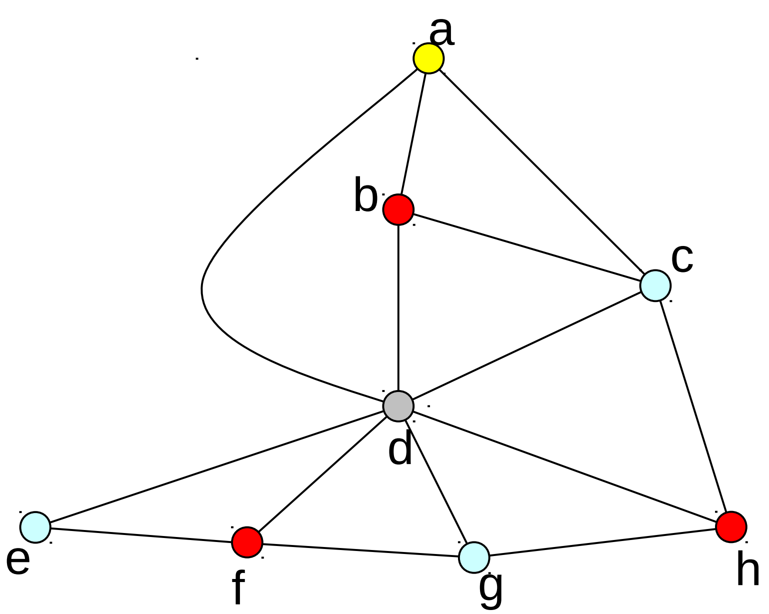
- (**théorème de Konig 1916** : Pour tout graphe biparti G, l’indice chromatique est égal au plus haut degré d’un sommet.)

e)Le problème des 4 couleurs.



Modélisation par un graphe:





Coloration

Exercice 1 :

Donnez sans justification le nombre chromatique $\chi(G)$ des graphes suivants :

1. une chaîne d'ordre $n \geq 2$.
2. Un cycle d'ordre pair $n = 2k$ avec $k \geq 2$.
3. Un cycle d'ordre impair $n = 2k + 1$ avec $k \geq 1$.
4. Un graphe R constitué d'un cycle d'ordre pair et d'un sommet supplémentaire adjacent à tous les sommets du cycle (une “roue” d'ordre impair).
5. Un graphe R' constitué d'un cycle d'ordre impair et d'un sommet supplémentaire adjacent à tous les sommets du cycle (une “roue” d'ordre pair).

Exercice 2 :

1. Démontrez par récurrence sur $n \geq 2$ qu'un arbre A d'ordre n a pour nombre chromatique $\chi(A) = 2$.
2. Montrez par un contre-exemple que la réciproque est fausse.
3. Montrez que si $\chi(G) = 2$ alors G n'admet pas de cycles impairs.

Exercice 3 :

Soient $T = \{t_1, t_2, \ldots, t_7\}$ un ensemble de 7 travaux et $M = \{m_1, m_2, \ldots, m_7\}$ un ensemble de 7 machines. Chaque travail t_i utilise un sous-ensemble M_i de machines :

$M_1 = \{m_1, m_3, m_5\}$

$M_2 = \{m_1, m_2, m_4\}$

$M_3 = \{m_2, m_3, m_5\}$

$M_4 = \{m_2, m_4, m_7\}$

$M_5 = \{m_5, m_6, m_7\}$

$M_6 = \{m_4, m_6, m_7\}$

$M_7 = \{m_5, m_6, m_7\}$

Le temps nécessaire à l'exécution de chaque travail t_i est le même (une journée) , mais deux travaux t_i et t_j ne peuvent être exécutés la même journée que s'ils utilisent des machines différentes (i.e. $M_i \cap M_j = \emptyset$).

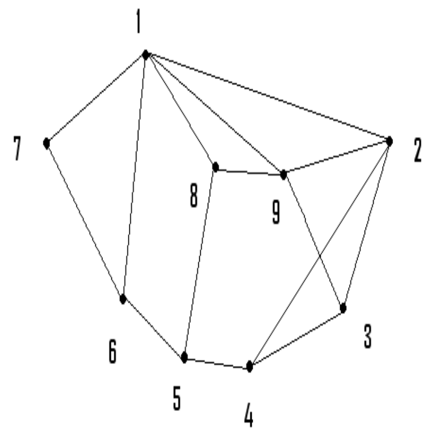
1. Représentez les contraintes de ce problème au moyen d'un graphe.
2. Combien de journées faudra-t-il au minimum pour réaliser tous ces travaux ?

Exercice 4 :

Pour chacun des graphes suivants donnez un ordre sur les sommets tel que l'algorithme de coloration gloutonne donne une coloration optimale, puis donnez si c'est possible un ordre sur les sommets tel que l'algorithme de coloration gloutonne ne donne pas une coloration optimale.

1. P_3
2. P_4
3. C_4
4. C_5
5. C_6

Exercice 5 :



1. Sans chercher à colorer le graphe G ci-dessus, donnez un encadrement de $\chi(G)$.
2. Donnez la coloration obtenue par l'algorithme de coloration gloutonne en prenant comme ordre sur les sommets $(1,2,3,4,5,6,7,8,9)$. Que peut-on en déduire pour $\chi(G)$?
3. Que vaut $\chi(G)$?

Exercice 6 :Organisation d'un tournoi

On suppose que pour l'organisation d'un tournoi sportif (du type jeux olympiques) les horaires des épreuves sont connus à l'avance. L'objectif des organisateurs est de réaliser toutes les épreuves en utilisant le moins de gymnases possibles. Pour une journée donnée, on a le tableau suivant des épreuves avec les horaires de début et de fin pour chacune d'entre elles :

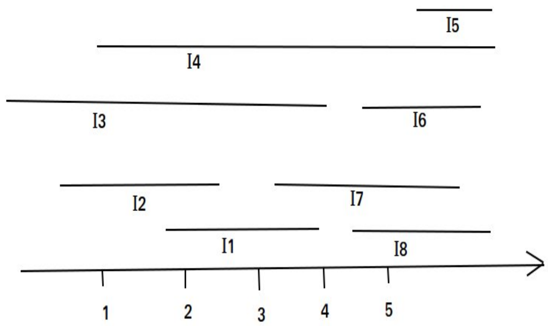
épreuves	début	fin
a	9h	10h30
b	9h30	11h30
c	10h	14h30
d	11h	12h
e	14h	16h
f	15h	17h
g	15h	18h
h	8h	10h

1. On veut résoudre ce problème en utilisant la théorie des graphes :
   (a) Modéliser la situation précédente par un graphe G permettant de résoudre le problème par une coloration optimale (le nombre de gymnases à prévoir sera le nombre chromatique du graphe).
   (b) Déterminer $\chi(G)$.
   (c) Montrer que l'algorithme glouton obtenu en ordonnant les sommets par ordre croissant des horaires de début donne une coloration optimale.

Exercice 7 : les graphes d'intervalles

Soit un ensemble $U = \{I_1, I_2, \ldots, I_n\}$ dont les n éléments sont des intervalles finis de $\mathbb{R}$. A cet ensemble on associe le graphe $G_U = (X, E)$ où $X = \{1, 2, \cdots, n\}$ et $E = \{ij, I_i \cap I_j \neq \emptyset\}$ (une arête de E relie les indices de deux intervalles d'insersection non vide). Ce graphe s'appelle le graphe d'interférence de la famille d'intervalle.

1. Représenter le graphe G correspondant à l'ensemble d'intervalles ci-dessous :



2. Déterminer $\Delta(G)$ et $\omega(G)$.
3. Démontrer que $\omega(G) \leq \chi(G) \leq \Delta(G)$
4. Donner une coloration de G en utilisant l'algorithme de coloration gloutonne du cours. On prendra comme ordre sur les sommets celui qui correspond à l'ordre croissant des premieres bornes des intervalles.
5. En déduire $\chi(G)$
6. Cas général.
   (a) Soient $(I_1, I_2, \ldots, I_n)$ n intervalles rangés dans l'ordre croissant de leur première borne. Démontrer que dans le graphe correspondant on a $deg(n) \leq \omega(G) - 1$
   (b) Démontrer par récurrence sur le nombre de sommets que pour un graphe d'intervalle G on a $\chi(G) = \omega(G)$.

Exercice 8 : Allocation des registres

Un registre est un emplacement de mémoire interne à un processeur. Il s'agit de la mémoire la plus rapide d'un ordinateur. La capacité dépasse rarement quelques dizaines d'octets. Les programmes transfèrent d'abord les données de la mémoire centrale vers des registres, puis effectuent des opérations sur ces registres et enfin transfèrent le résultat en mémoire centrale. L'objectif pour le compilateur est de minimiser le nombre de registres utilisés simultanément.

Considérons une séquence d'opérations entre plusieurs variables :

	instructions	a	b	c	d	e	f
1	$a \leftarrow 1$						
2	$e \leftarrow a + 1$	v					
3	$d \leftarrow e \times a$	v					
4	$f \leftarrow d \times 2$						
5	$c \leftarrow d + f \times e$						
6	$b \leftarrow f + 2$						
7	$a \leftarrow b + c + e$						
8	return $a \times b$	v					

On étudie la duré de vie de chaque variable (la "vitalité") c'est à dire les moments où la variable utilise un registre : tout de suite après l'affectation jusqu'à sa dernière utilisation.

Dans le tableau ci-dessus on a mis des "v" dans la colonne de la variable a lorsque celle-ci est vivante. On ne met pas de "v" au moment de l'affectation ($a \leftarrow 1$) car dans un registre l'écriture se fait en fin de cycle (et la lecture en début).

Entre la dernière utilisation d'une variable et l'affectation suivante il est inutile de conserver la valeur dans un registre. Ainsi, on peut considérer que la variable a de la ligne 7 est une nouvelle variable que l'on notera a'.

1. Compléter les autres colonnes du tableau.
2. Dresser le graphe d'interférence G où chaque variable est un sommet. Il y aura une arête entre deux variables ssi ces deux variables ne peuvent pas utiliser le même registre (c'est à dire lorqu'elles sont vivantes à un même moment).

3. Déterminer $\chi(G)$ et en déduire le nombre minimum de registres nécessaires pour l'exécution de cette séquence.

Exercice 9 :

Dans un tournoi d'échecs chaque joueur doit affronter tous les autres. Chaque partie dure 1 heure.

1. Combien y aura-t-il de parties pour n joueurs ?
2. On note G_n le graphe où les sommets représentent les joueurs (on les numérote de 1 à n) et les arêtes représentent les rencontres. Représenter G_3, G_4 et G_5 .
3. L'organisateur veut limiter la durée du tournoi. Proposer un planning des rencontres pour $n = 3, 4, 5$ et 6. On pourra utiliser la notion d'indice chromatique.

Exercice 10 :

Soit $G = (X, E)$ un graphe. On appelle graphe adjoint de G , le graphe $G' = (X', E')$ où $X' = E$ (les sommets de G' sont les arêtes de G) et $E' = \{ef, e \in E \wedge f \in E \wedge e \cap f \neq \emptyset\}$ (il y a une arête entre deux sommets de G' si et seulement s'ils correspondent à deux arêtes de G ayant une extrémité en commun).

1. Soit G=$(X = \{a, b, c, d, e, f, g, h\}, E = \{ag, af, ac, ef, ch\})$un graphe. Représenter G et G' .
2. Déterminer $\chi(G')$ et en déduire $\chi'(G)$.

Exercice 11 :

Trois jurys ont reçu la liste des candidats auxquels ils doivent faire subir une épreuve orale d'une heure :

— Jury 1 : Aurore, Charles, Denis.
— Jury 2 : Aurore, Charles, Fernand.
— Jury 3 : Aurore, Bernard, Denis.

A chaque nouvelle heure, les jurys appellent leur candidat suivant. Comme il faut bien établir une règle en cas de conflit, il est convenu que le jury 1 appelle en premier, puis le jury 2 et enfin le 3. Lorsqu'un jury appelle un candidat rendu indisponible par son interrogation dans un autre jury, la fiche du candidat passe en queue de liste pour ce jury, et ce jury appelle le candidat suivant.

1. Si les interrogations commencent à 14h, avec la règle de l'énoncé, à quelle heure se terminera la session ?
2. Proposer une meilleure organisation (en utilisant la théorie des graphes).

Exercice 12 :

Soit le graphe $G_0 = (X_0 = \{a, b\}, E_0 = \{ab\})$. A partir de ce graphe, on construit récursivement une suite de graphes $(G_n)_{n \in \mathbb{N}}$ de la façon suivante :

Pour construire G_{n+1} à partir de G_n , on recopie G_n puis pour chaque sommet x de G_n on crée un sommet x'qui a les mêmes voisins que x , et enfin on rajoute un sommet c_{n+1} qui est voisin de tous les sommets de type x' .

On notera $\chi_n = \chi(G_n)$ et $\omega_n = \omega(G_n)$.

1. Représenter successivement G_0, G_1 et G_2 . On ne cherchera pas à nommer les sommets de ces graphes.
2. Déterminer successivement ω_0, χ_0 ,ω_1, χ_1 puis ω_2, χ_2 .
3. Démontrer par récurrence les propositions suivantes :
   (a) $\forall n \in \mathbb{N}, |G_n| = 3 \times 2^n - 1$ (rappel : $|G_n|$ est l'ordre du graphe G_n c'est à dire son nombre de sommets)
   (b) $\forall n \in \mathbb{N}, \omega_n = 2$
   (c) $\forall n \in \mathbb{N}, \chi_n = n + 2$

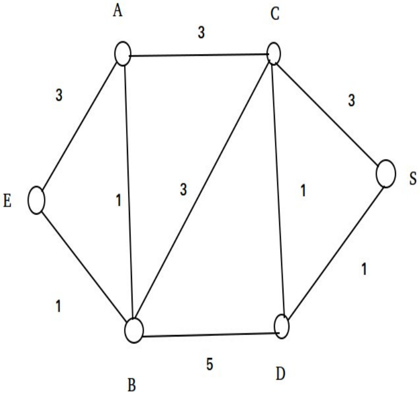
(conclusion : l'écart entre l'ordre du plus grand sous-graphe complet et le nombre chromatique peut être aussi grand que l'on veut)

Algorithme de Dijkstra

Graphes valués aux arêtes

Définitions

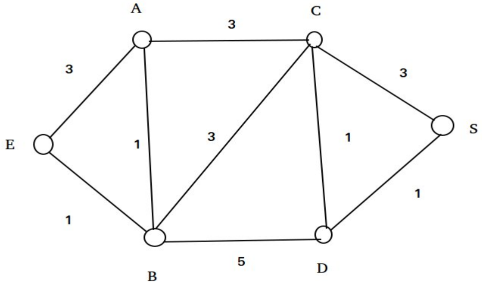
Un graphe simple non orienté $G = (X, E)$ est dit **valué aux arêtes** ssi il existe une application v de E dans $\mathbb{R}$. $v\,(xy)$ est la valuation de l'arête xy .



Définitions

La valuation d'une chaîne P**, notée** $v(P)$**,** est la somme des valuations de chacune de ses arètes.

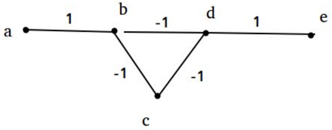
La distance entre deux sommets d'un graphe valué est la valuation minimum d'une chaîne du graphe ayant ces deux sommets pour extrémités (on notera $d_{G,v}(x,y)$ cette distance)



Pour la chaîne $P = (\{E, A, C, B\}, \{EA, AC, CB\})$ on a $v(P) = 9$.
$d_{G,v}(E, C) = 4$

Définitions

Attention avec les valuations négatives. Dans le graphe ci-dessous quelle est la distance du sommet a au sommet e ?



On se contentera par la suite de valuations positives, c'est à dire des applications de E dans $\mathbb{R}^+$.

Algorithme de Dijkstra

Algorithme : DistanceDeDijkstra
Données : un graphe G=(X,E), une valuation positive v
et un sommet a de G.
Résultat : une fonction d donnant la distance d'un sommet quelconque
au sommet a

Pour tout sommet x de G **faire**
    $d(x) \leftarrow \infty$
Fin Pour

$d(a) \leftarrow 0$

Pour tout voisin x de a **faire**
    $d(x) \leftarrow v(ax)$
Fin Pour

marquer le sommet a en rouge

TantQue il reste des sommets t non rouges et tels que $d(t) \neq \infty$ **faire**

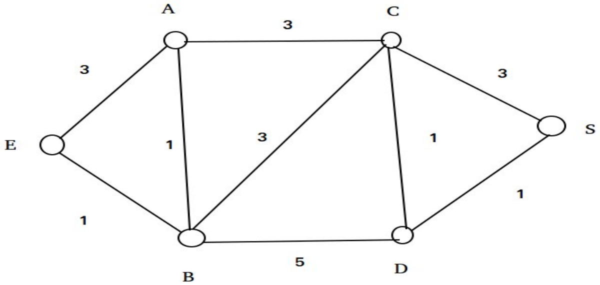
  choisir un sommet x non rouge qui minimise la fonction d

  Pour tout voisin y non rouge du sommet x **faire**
    $d(y) \leftarrow \min(d(y)\ ;\ d(x)+v(xy))$
  FinPour

  marquer x en rouge

FinTantQue

Retourner la fonction d



	E	A	B	C	D	S
E(0)	X	3E	1E	∞	∞	∞
B(1)	X	2B	X	4B	6B	∞
A(2)	X	X	X	4B	6B	∞
C(4)	X	X	X	X	5C	7C
D(5)	X	X	X	X	X	6 D

P=(E,B,C,D,S) et v(P)=6.

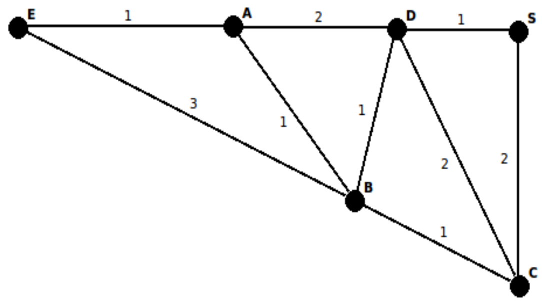
Théorème

*L'algorithme **DistanceDeDijkstra** appliqué à un graphe* $G = (X, E)$ *dont les arêtes sont valués par une fonction* v *positive ou nulle, et à un sommet* a *de* G *, retourne une fonction* d *telle que, pour tout sommet* x *de* G *on ait* $d(x) = d_{G,v}(a, x)$

Problèmes de "plus court chemin" dans un graphe et algorithme de Dijkstra

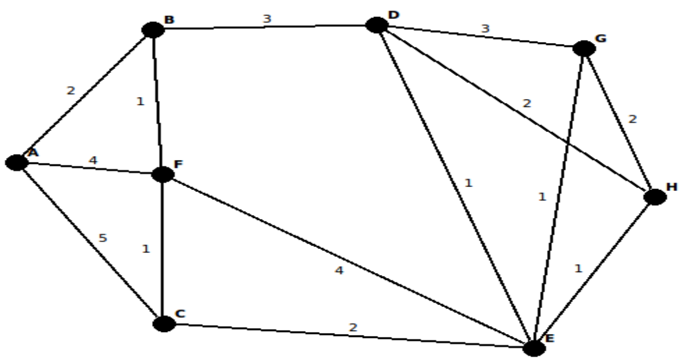
exercice 1 :

Pour le graphe valué ci dessous, déterminer la distance du sommet E au sommet S ainsi que tous les parcours réalisant cette distance.



Exercice 2 :

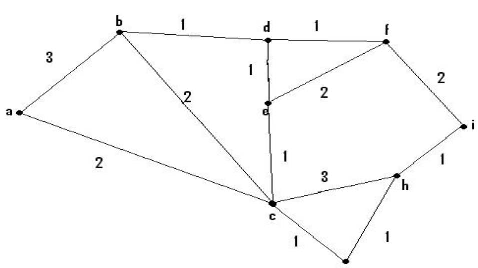
On considère le graphe G , valué aux arêtes, dont une représentation est donnée ci-dessous :



Déterminez $d_G(A, H)$ ainsi que toutes les chaînes réalisant cette distance en utilisant l'algorithme de Dijkstra.

Exercice 3 :

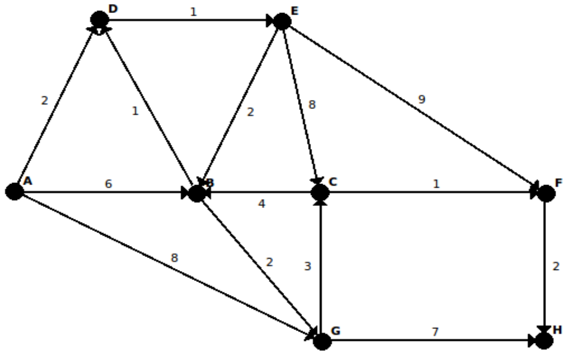
Pour le graphe valué ci dessous, déterminer la distance du sommet a au sommet i ainsi que tous les parcours réalisant cette distance.



Exercice 4 :

L'algorithme de Dijkstra s'utilise aussi avec des graphes orientés.

Déterminez, pour le graphe orienté ci-dessous, le plus court chemin du sommet a au sommet h, puis donnez la distance de a à h :



Exercice 5 :

Routage par information d'état des liens.

Dans un réseau le “sommet “ A (i.e. le routeur A) reçoit les paquets d'information d'état des liens de chaque sommet. Il connait donc les voisins de chaque sommet du graphe ainsi que les coûts associés :

A	coût
B	4
E	5

B	coût
A	4
C	2
F	6

C	coût
B	2
D	7
E	1

D	coût
C	7
F	3

E	coût
A	5
C	1
F	3

F	coût
B	6
D	3
E	3

1. Aider A à reconstruire le réseau.
2. Calculer les tables de routages de A puis celles de D (Pour un sommet donné x, la table de routage contient, pour chaque destination y, son coût total et le premier sommet de la plus courte chaîne en allant de x vers y).

Exercice 6 :

Programmation dynamique (cf. michel.minguenaud@insa-rouen.fr)

La demande d'un équipement en janvier, février et mars est de deux unités. Les deux unités sont livrées à la fin de chaque mois. Le fabricant souhaite établir le plan de production de cet équipement. Le stock ne peut pas dépasser 2 unités en février et mars et est nul en janvier et en avril. La production maximale pour un mois donné est de 4 unités. Le premier mois, seuls les coûts de productions sont imputables ; les mois suivants le stock entre en ligne de compte.

Pour un stock de i équipements et une production y , le coût mensuel vaut : $C(y,i) = f(y) + 6i$ avec $f(0) = 0$; $f(1) = 15$; $f(2) = 17$; $f(3) = 19$; $f(4) = 21$.

Résoudre ce problème en utilisant la théorie des graphes.

Recherche Opérationnelle

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* :

- On dispose de 20 euros pour acheter des légumes.
- Le maraîcher propose des carottes à 2 euros le kilo et des pommes de terre à 1 euro le kilo.
- Par goût, on veut au moins autant de carottes que de pomme de terre.
- La marmite à pomme de terre ne dépasse pas les 5 kilos de contenance.
- Pour avoir le plus à manger on veut maximiser la quantité totale (en kilo) de légumes.

- **variables** : la quantité **x** de carottes en kilo et la quantité **y** de pommes de terre.
- **contraintes :**
  - argent : $2x + y \leq 20$
  - marmite : $y \leq 5$
  - goût : $x \geq y$
  - autres : $x \geq 0, y \geq 0$
- **Objectif** : maximiser $x + y$

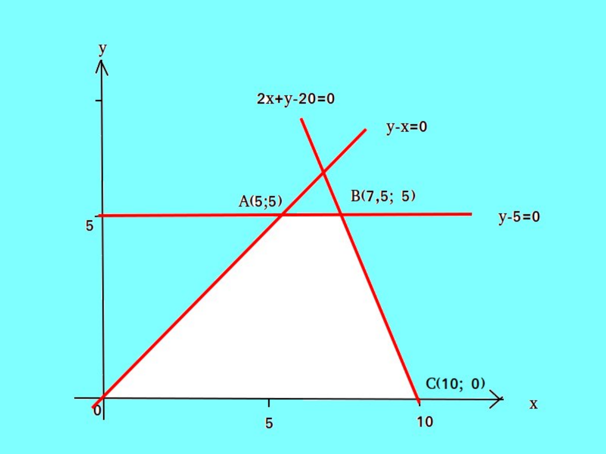
2) Résolution graphique :

On peut résumer le problème précédent ainsi :

$$\begin{cases} f(x, y) = x + y \\ 2x + y - 20 \leq 0 \\ y - 5 \leq 0 \\ y - x \leq 0 \\ x \geq 0 \\ y \geq 0 \end{cases}$$

on veut maximiser la fonction f

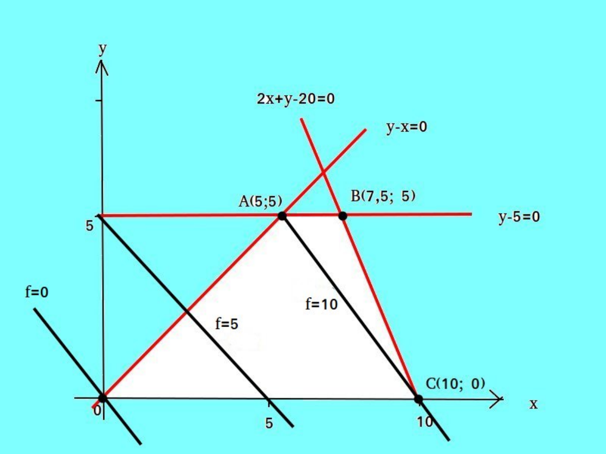
Chacune des contraintes correspond à un demi-plan :



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 : $f(x, y)$ =constante)



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

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.

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 $ax_1 + bx_2 + cx_3 + \ldots \leq \beta$.)

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

$$\boxed{x_3 = 5 - x_1 - x_2}$$

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

Le problème peut donc s'écrire plus simplement :

$$\boxed{\textbf{maximiser } f \textbf{ avec les contraintes } x_1, x_2, x_3 \geq 0 \textbf{ .}}$$

On peut résumer les données ainsi :

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

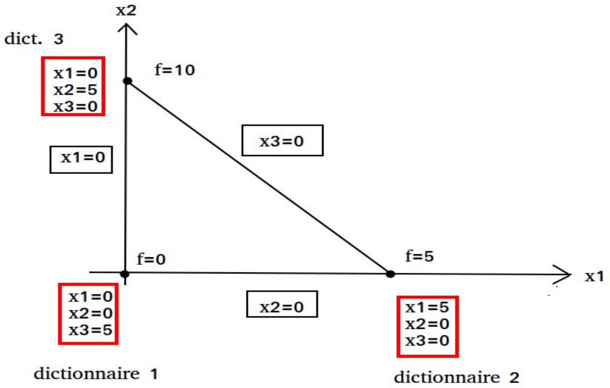
Les variables x_1, x_2 sont **hors base** tandis que la variable x_3 est dans la **base**.

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 $f = 0$



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)

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

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

On fait entrer x_2 dans la base et on fait sortir x_1 :$x_1 \geq 0 \Rightarrow x_2 \leq 5$

Dictionnaire 3
$x_2 = 5 - x_1 - x_3$
$f = 10 - x_1 - 2x_3$

La solution basique du dictionnaire 3 est $(x_1 = 0, x_2 = 5, x_3 = 0)$pour f=10

Dictionnaire 3
$x_2 = 5 - x_1 - x_3$
$f = 10 - x_1 - 2x_3$

Dans le dictionnaire 3 toutes les variables présentes dans l'expression de f ont un coefficient négatif =>

L'algorithme est fini.

La solution du problème d'optimisation est la solution basique du dernier dictionnaire :

La solution est $(x_1 = 0, x_2 = 5)$pour une valeur maximale de f=10

4)Simplexe : un exemple plus complet

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

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.

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 :

.........$x_4 \geq 0 \Rightarrow x_1 \leq 20$
.........$x_5 \geq 0 \Rightarrow x_1 \leq 5$
(pour résoudre ces 2 inéquations on suppose que les autres variables hors base x_2 et x_3 sont nulles)

Cest donc la variable x_5 qui sort de la base :

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

Dictionnaire 2

$$
\begin{aligned}
x_1 &= 5 - \tfrac{1}{2}x_2 - \tfrac{1}{2}x_5 \\
x_4 &= 15 - \tfrac{1}{2}x_2 - x_3 + \tfrac{1}{2}x_5 \\
\hline
f &= 10 + 3x_3 - x_5
\end{aligned}
$$

on fait entrer x_3 et on fait sortir x_4 (il n'y a pas le choix).

Dictionnaire 3

$$
\begin{aligned}
x_3 &= 15 - \tfrac{1}{2}x_2 - x_4 + \tfrac{1}{2}x_5 \\
x_1 &= 5 - \tfrac{1}{2}x_2 - \tfrac{1}{2}x_5 \\
\hline
f &= 55 - \tfrac{3}{2}x_2 - 3x_4 + \tfrac{1}{2}x_5
\end{aligned}
$$

on fait entrer x_5 et on sort x_1

Dictionnaire 4

$$
\begin{aligned}
x_5 &= 10 - 2x_1 - x_2 \\
x_3 &= 20 - x_1 - x_2 - x_3 \\
\hline
f &= 60 - x_1 - 2x_2 - 3x_4
\end{aligned}
$$

Fin de l'algorithme : la solution est $(x_1 = 0, x_2 = 0, x_3 = 20)$ pour $f = 60$

5) Dictionnaire dégénéré :

définition :
Une solution basique est dite **dégénérée** lorsque une ou plusieurs valeurs de la base sont à 0.

Exemple :
Avec le dictionnaire

$$\begin{array}{|c|}\hline x_5 = -2x_1 + 4x_2 + 3x_4 \\ x_3 = 1 - x_4 \\ x_6 = x_1 - 3x_2 + 2x_4 \\ \hline f = 4 + 2x_1 - x_2 - 4x_4 \\ \hline \end{array}$$

La solution basique est dégénérée car pour cette solution on a $x_5 = 0$ et $x_6 = 0$.
En faisant entrer x_1 dans la base, f n'augmentera pas.

Avec un dictionnaire dégénéré, il y a risque de cyclage de l'algorithme du simplexe.

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 polynome des contraintes : il faut parfois trouver un autre point de départ de l'algorithme (ce sujet ne sera pas traité ici).

TD d'optimisation linéaire :

Exercice 1 :

Une usine de textile fabrique 3 variétés de tissu T1, T2 et T3 à partir de 3 laines L1, L2 et L3. Le tableau suivant recense les poids (en kg) des laines intervenant dans la composition d'un mètre de tissu :

	T1	T2	T3
L1	0,375	0,125	0,1
L2	0,5	0,05	0,2
L3	0,5	0,2	0,15

On dispose d'un stock de 4000 kg de laine L1, 800 kg de laine L2 et 1500 kg de laine L3. Les métiers à tisser ne peuvent fabriquer que 8000 m de tissu. Les profits nets résultant de la vente d'un mètre de tissu sont respectivement de 2,6 euros, 4 euros et 3,6 euros pour T1, T2 et T3.

Ecrire le problème de maximisation du profit sous la forme d'un programme linéaire.

Exercice 2 :

Un constructeur de postes de télévision possède 4 modèles à son catalogue : le portatif NB (M1), le standard NB (M2), le standard couleur (M3) et le couleur de luxe (M4).

L'entreprise comporte un atelier de montage et un de tests. Les durées nécessaires pour le montage et les tests des différents modèles sont (en heures) :

	M1	M2	M3	M4
Montages	8	10	12	15
Tests	2	2	4	5

La force de travail de l'atelier de montage est de 6000 heures par mois, celle de l'atelier de tests est de 1500 heures par mois et les profits des postes M1, M2, M3 et M4 sont respectivement de 400 euros, 600 euros, 800 euros et 1000 euros. L'entreprise dispose chaque mois de 450 transformateurs et de 300 tubes cathodiques couleur. On a besoin d'un transformateur dans chaque poste (NB ou couleur). La quantité disponible de tubes cathodiques NB n'est pas limitée.

Ecrire le problème de maximisation du profit de cette entreprise sous la forme d'un programme linéaire.

Exercice 3 :

On veut maximiser la fonction linéaire à 2 variables suivantes : $f(x, y) = x + y$.

On suppose de plus qu'on a les contraintes suivantes :

$$\begin{cases} x - y \leq 2 \\ \frac{1}{2}x + y \leq 2 \\ x \geq 0 \\ y \geq 0 \end{cases}$$

On se propose de résoudre ce problème d'optimisation linéaire tout d'abord par une méthode graphique et ensuite en utilisant l'algorithme du simplexe :

1. Résolution graphique :
   (a) Dans un repère orthonormé (unité 2cm) , tracer les 4 droites d'équations respectives $x = 0$, $y = 0$, $x - y = 2$ et $\frac{1}{2}x + y = 2$.
   (b) Déterminer l'ensemble des points du plan vérifiant toutes les contraintes (c'est le polyèdre des contraintes) . Vous hachurerez les points qui ne font pas partie de cette zone. Le point $O(0, 0)$ appartient-il à cet ensemble ?
   (c) Sur le même graphique tracer les droites d'équations respectives $f(x, y) = 0$, $f(x, y) = 1$ et $f(x, y) = 2$.
   (d) Résoudre graphiquement le problème d'optimisation. On donnera la valeur de f maximale, ainsi que les valeurs de x et de y correspondantes.

2. Algorithme du simplexe :
   On considère les deux variables d'écart z et t définies par $z = 2 - x + y$ et $t = 2 - \frac{1}{2}x - y$. A l'aide de ces deux variables les contraintes $x - y \leq 2$ et $\frac{1}{2}x + y \leq 2$ peuvent se réécrire respectivement $z \geq 0$ et $t \geq 0$.

(a) Compléter les tableaux suivants :

dictionnaire 1	expressions en fonction des variables hors base
variables hors base	
variables dans la base	
f	

variable entrant dans la base	....
contraintes	
variable sortant de la base	

(b) Terminer l'algorithme et résoudre le problème d'optimisation linéaire.

(c) Donner dans l'odre les sommets du polyèdre des contraintes qui ont été évalués par l'algorithme. Pour chacun de ces sommets donner la valeur de f .

Exercice 4 :

Résoudre graphiquement puis en utilisant l'algorithme du simplexe chacun des problèmes d'optimisation linéaire suivants :

$$P_1 \begin{cases} f_1(x_1, x_2) = x_1 + 2x_2 \\ x_1 + x_2 \leq 3 \\ x_2 \leq 2 \\ x_1, x_2 \geq 0 \end{cases}$$

$$P_2 \begin{cases} f_2(x_1, x_2) = x_1 + 2x_2 \\ x_1 - x_2 \leq 2 \\ -2x_1 + x_2 \leq 2 \\ \\ x_1, x_2 \geq 0 \end{cases}$$

Exercice 5 :

Résoudre graphiquement, puis avec l'algorithme du simplexe, le problème d'optimisation linéaire suivant. Pour les dictionnaires dégénérés vous utiliserez la règle de Bland.

$$P_3 \begin{cases} f_3(x_1, x_2) = 2x_1 + x_2 \\ x_1 + x_2 \leq 3 \\ x_1 \leq 2 \\ x_2 - x_1 \leq 0 \\ x_1 - 2x_2 \leq 0 \\ x_1, x_2 \geq 0 \end{cases}$$

Exercice 5 :

Résoudre en utilisant l'algorithme du simplexe les problèmes d'optimisation linéaire suivants :

$$P_4 \begin{cases} f_4(x_1, x_2, x_3) = 3x_1 + 2x_2 + 4x_3 \\ x_1 + x_2 + 2x_3 \leq 4 \\ 2x_1 + 3x_3 \leq 5 \\ x_1, x_2, x_3 \geq 3 \end{cases}$$

$$P_5 \begin{cases} f_5(x_1, x_2, x_3) = 4x_1 - x_2 + x_3 \\ x_1 + x_2 + x_3 \leq 10 \\ x_1 - x_2 \leq 0 \\ x_1 - x_3 \leq 0 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

Exercice 6 :

Un assembleur d'ordinateurs portables propose deux modèles sur le marché : le X1 et le X2. Le X1 est équipé d'une carte wifi dont il ne possède que 600 exemplaires. De plus il a en réserve 1600 barettes mémoire dont 2 sont nécessaires pour assembler un X1 et une pour assembler un X2. Les autres composantes sont considérés comme étant en quantité illimitée. Enfin, il ne pourra pas stocker plus de 1200 ordinateurs portables une fois l'assemblage fait.

1. Il est possible de sous-traiter l'assemblage des portables à une société spécialisée, celle-ci acceptant à condition que le marché porte sur au moins 1000 ordinateurs. Un accord est-il possible ?
2. En supposant l'accord signé, l'assembleur table sur un bénéfice de 20 euros par modèle X1 et de 12 euros par modèle X2 assemblé. Quelle quantité de modèles X1 et X2 doit-il faire assembler pour obtenir un bénéfice maximal ? (méthode graphique).

Maths 4

Programmation linéaire en nombre entier (PLNE)
Résolution avec la méthode de séparation et évaluation

I) 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

Exemple : un problème de sac à dos

- Un sac à dos de capacité totale **21kg**
- 3 types d'objet :
  - Objet 1 : **4kg** pour un bénéfice de **38 euros**
  - Objet 2: **10kg** pour un bénéfice de **100 euros**
  - Objet 3 : **3kg** pour un bénéfice de **27 euros**.
- Les quantités x1, x2 et x3 de chaque produit sont **entières**.
- On veut maximiser le bénéfice :

  f(x1,x2,x3)=38 x1 + 100 x2 +27 x3

I)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.

I)Algorithme de séparation et évaluation (suite)

On dispose de plus d’une **fonction h** d’évaluation par excés :

Cette fonction permet d’obtenir *simplement* un **majorant du maximum de f** sur n’importe lequel des sous-ensembles de E.

Un problème de sac à dos

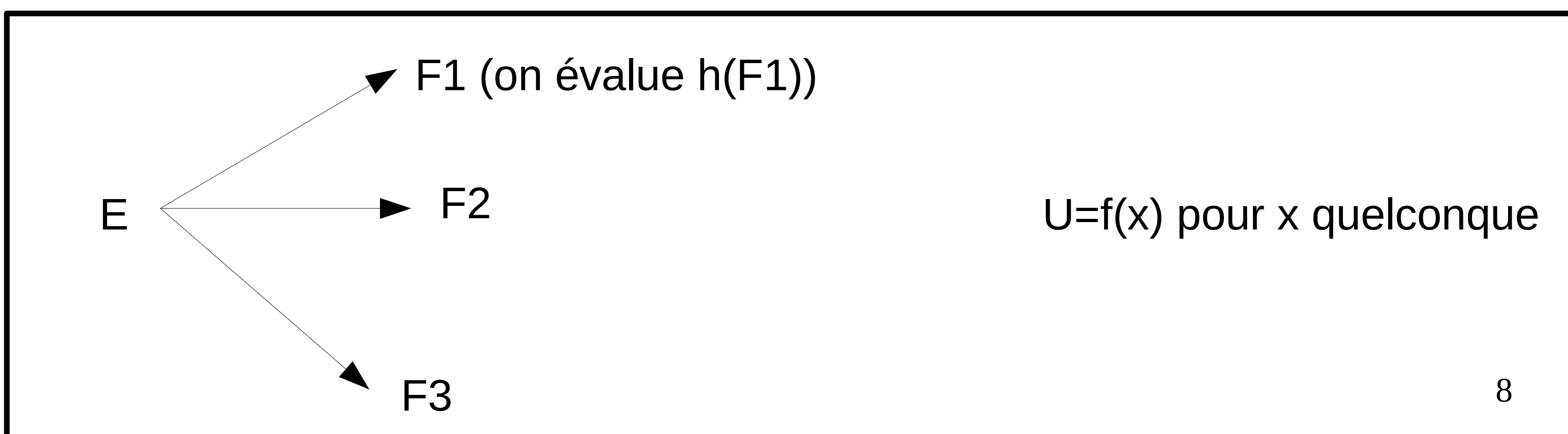
- **Fonction d’évaluation par excés :h**
  - On n’impose pas aux quantités xi d’être entières.
  - On calcule un majorant du bénéfice en remplissant le sac avec **uniquement le produit qui rapporte le plus par unité de poids.**

Un problème de sac à dos

- Le produit 1 rapporte 38/4=**9,5 euros par kg**
- Le produit 2 rapporte 100/10=**10 euros par kg**
- Le produit 3 rapporte 27/3=**9 euros par kg.**
- On peut mettre une quantité maximale **x2=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)=2,1X100=210 euros**

I)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 F1,F2,….,Fn (qui forment une partition de E).
  - On évalue h sur un des Fi



I) Algorithme de séparation et évaluation(suite)

- **Première étape (suite)** :
  - Si **h(Fi)≤U** => on élague Fi.
  - Si **h(Fi)=f(x)** (pour un x dans Fi) => on élague Fi et on réévalue U (U=Max(h(Fi),U)).
  - Si **Fi est un singleton x** => on élague Fi et on réévalue U (U=Max(U, f(x))

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

Un problème de sac à dos

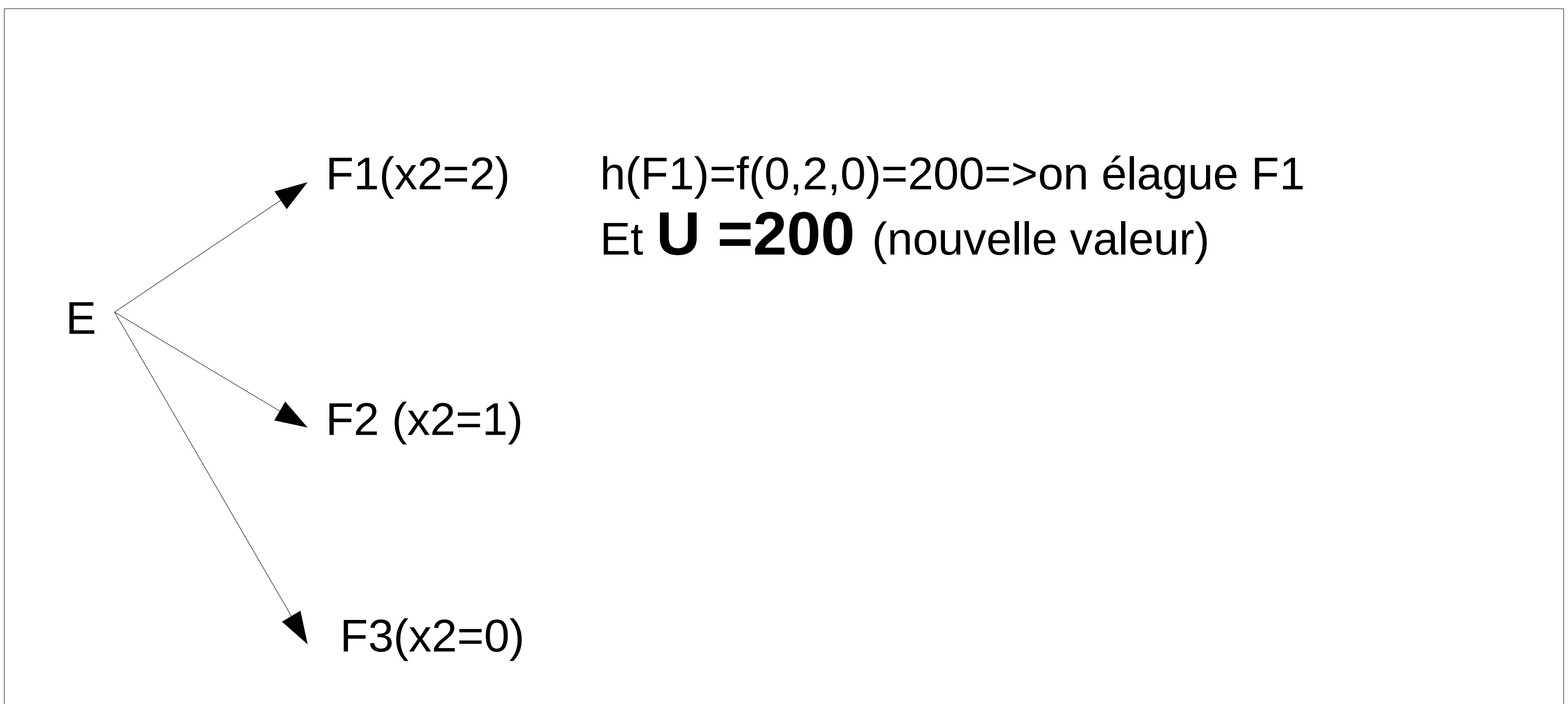
- Initialisation de U : U=0

- Séparation de E :
  - F1 : les sacs à dos tels que x2=2
  - F2: les sacs à dos tels que x2=1
  - F3 : les sacs à dos tels que x2=0

- (le calcul de h(E) montre que x2≤2)

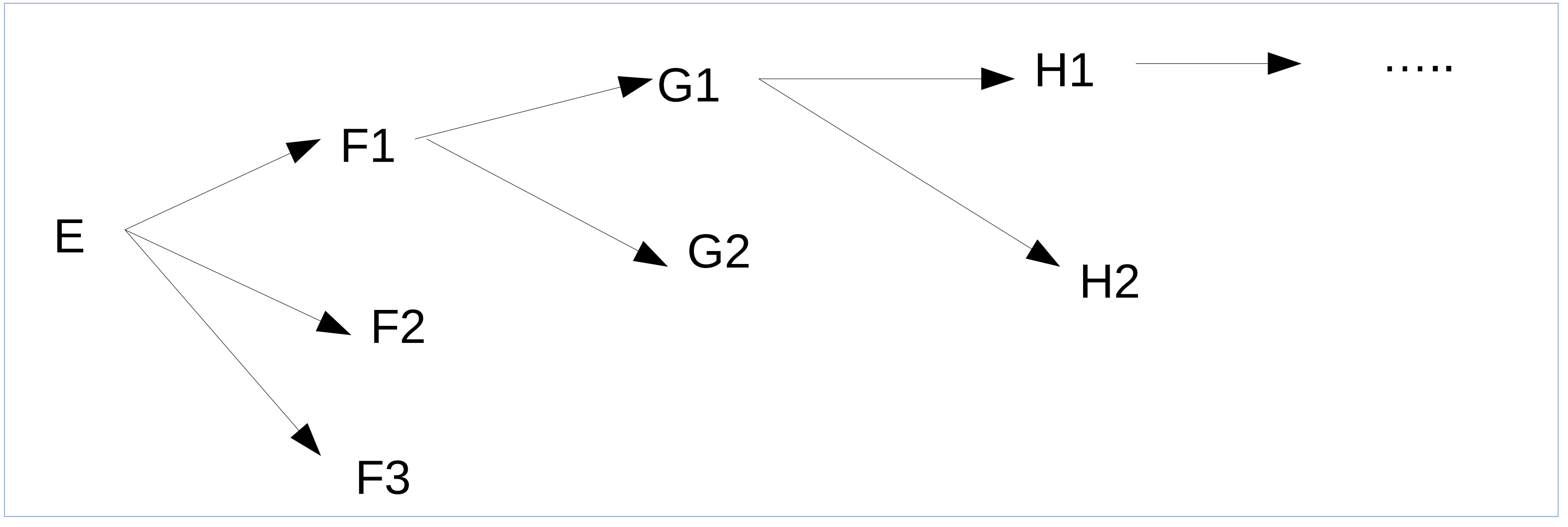
Un problème de sac à dos

- **Évaluation de F1 :**
  - **h(F1)** : il reste 1kg disponible donc un seul sac à dos possible : x1=0, x2=2 et x3=0 donc **h(F1)=200**



I)Algorithme de séparation et évaluation(suite)

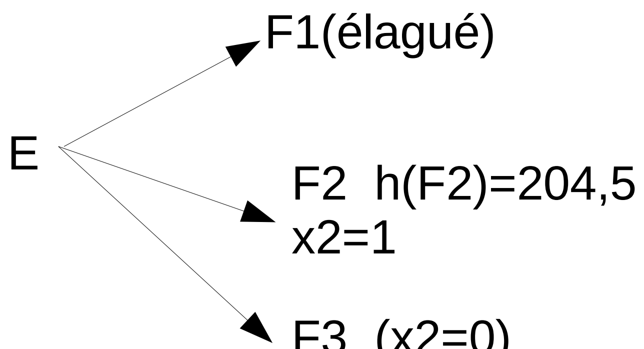
- **Deuxième étape :**
  - On recommence l'opération de séparation et évaluation sur un des sommets Fi non élagués.
- On obtient ainsi un arbre qu'il est préférable de parcourir **en profondeur** :



Un problème de sac à dos

- **Evaluation de F2 :**
  - **h(F2)** : il reste 11kg disponible que l'on remplit avec le produit 1 (plus rentable que 3) : x1=11/4=2,75 => **h(F2)=1X100+2,75X38=204,5**

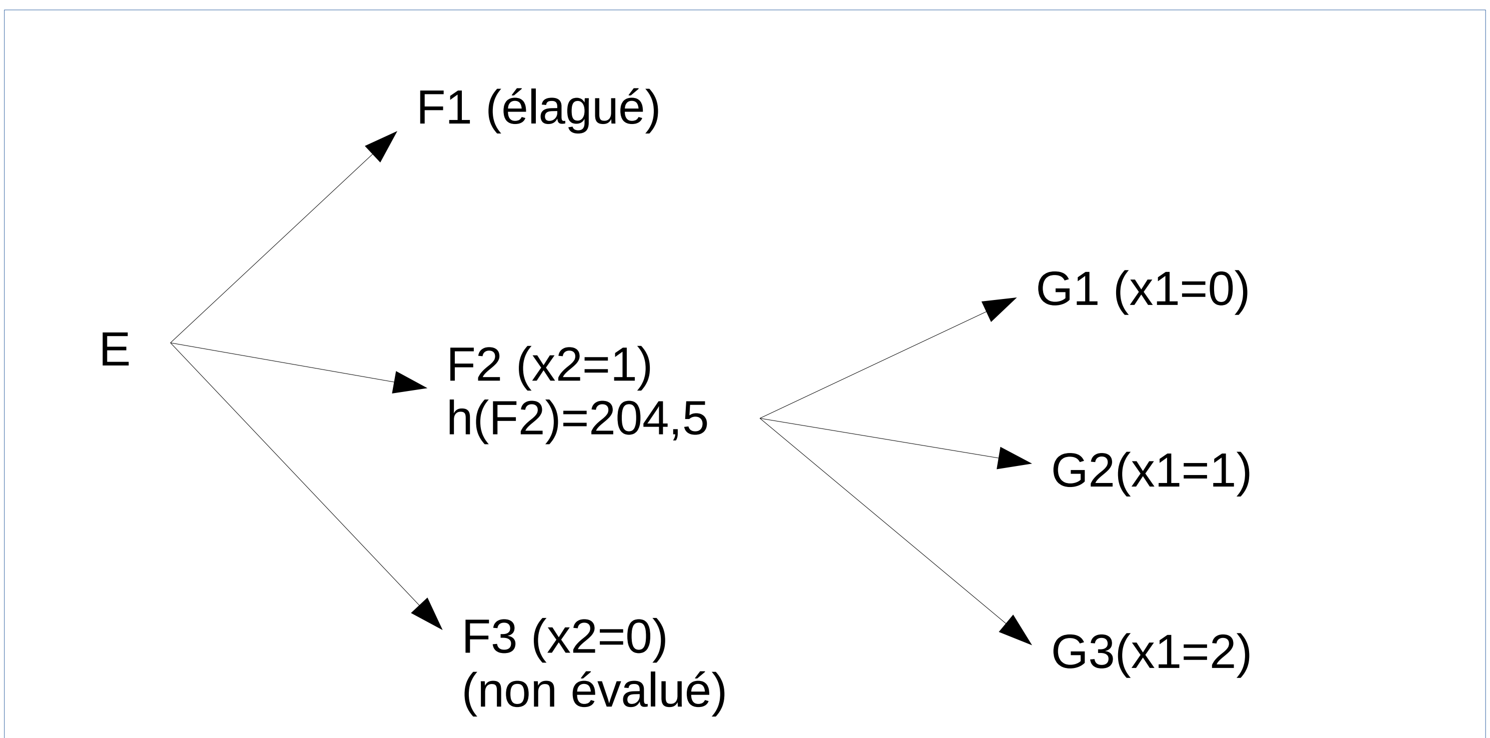
h(F2)>U (car U=200) =>on n'élague pas F2.



U est inchangé (U=200).
Prochaine étape :
On sépare F2 avant de passer à F3 (parcours en profondeur)

Un problème de sac à dos

- En évaluant h(F2) on a vu que les sacs à dos de F2 vérifient : $x1 \leq 2$.
- On en déduit la séparation suivante de F2:



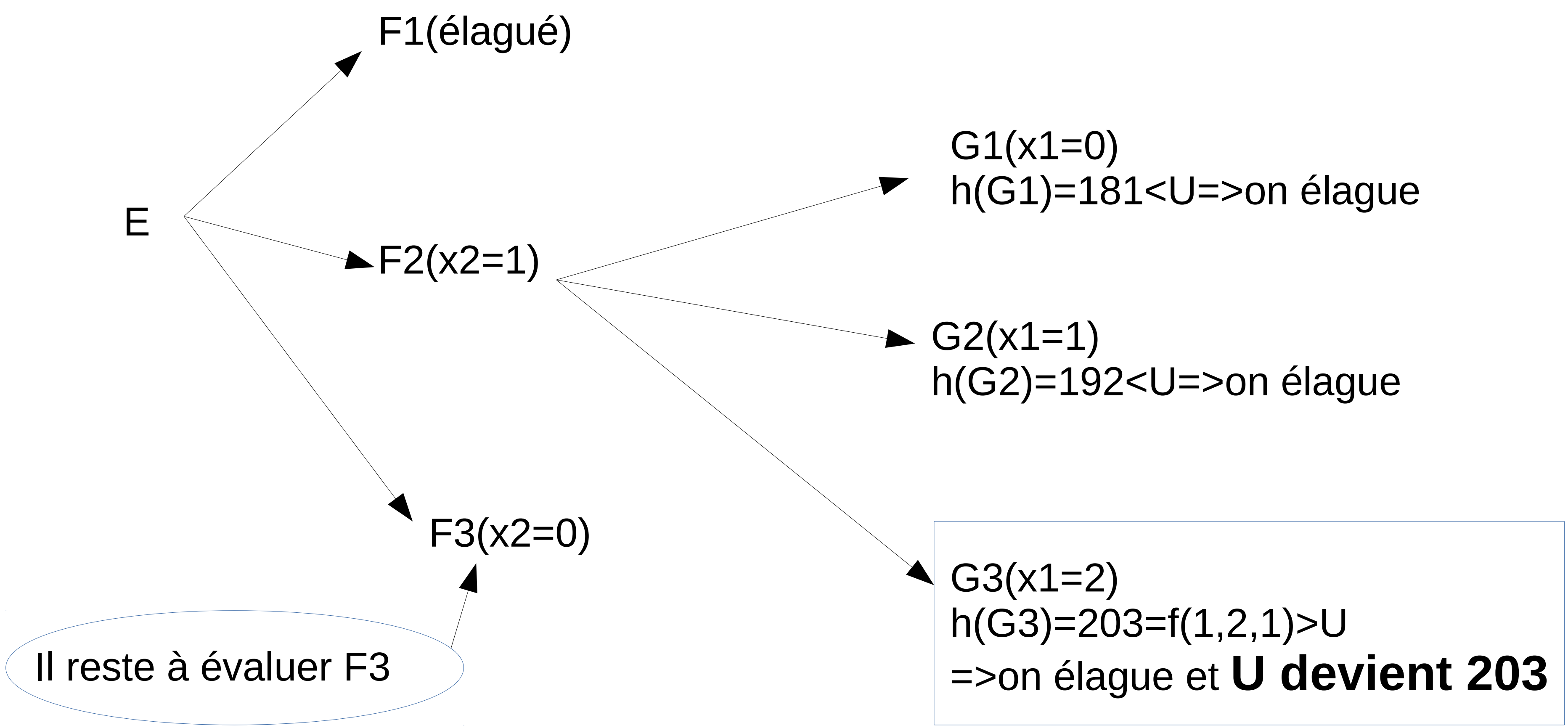
U=200

Un problème de sac à dos

- U=200
- On évalue G1 (x2=1 et x1=0) : h(G1)=181<U=>on élague G1
- On évalue G2 (x2=1 et x1=1) : h(G2)=192<U=>on élague G2
- On évalue G3 (x2=1 et x1=2) : **h(G3)=203=f(1,2,1)>U=> on élague G3 et U devient 203**

Un problème de sac à dos

- U=200



Un problème de sac à dos :

- **Evaluation de F3 :**
  - **h(F3)** : il reste 21 kg disponible que l'on remplit avec le produit 1 (plus rentable que 3) : x1=21/4=5,25 => **h(F2)=5,25X38=199,5**
  - **h(F3)<U=>on élague F3**
- **La solution est donc fmax=203 pour x1=1, x2=2 et x3=1**

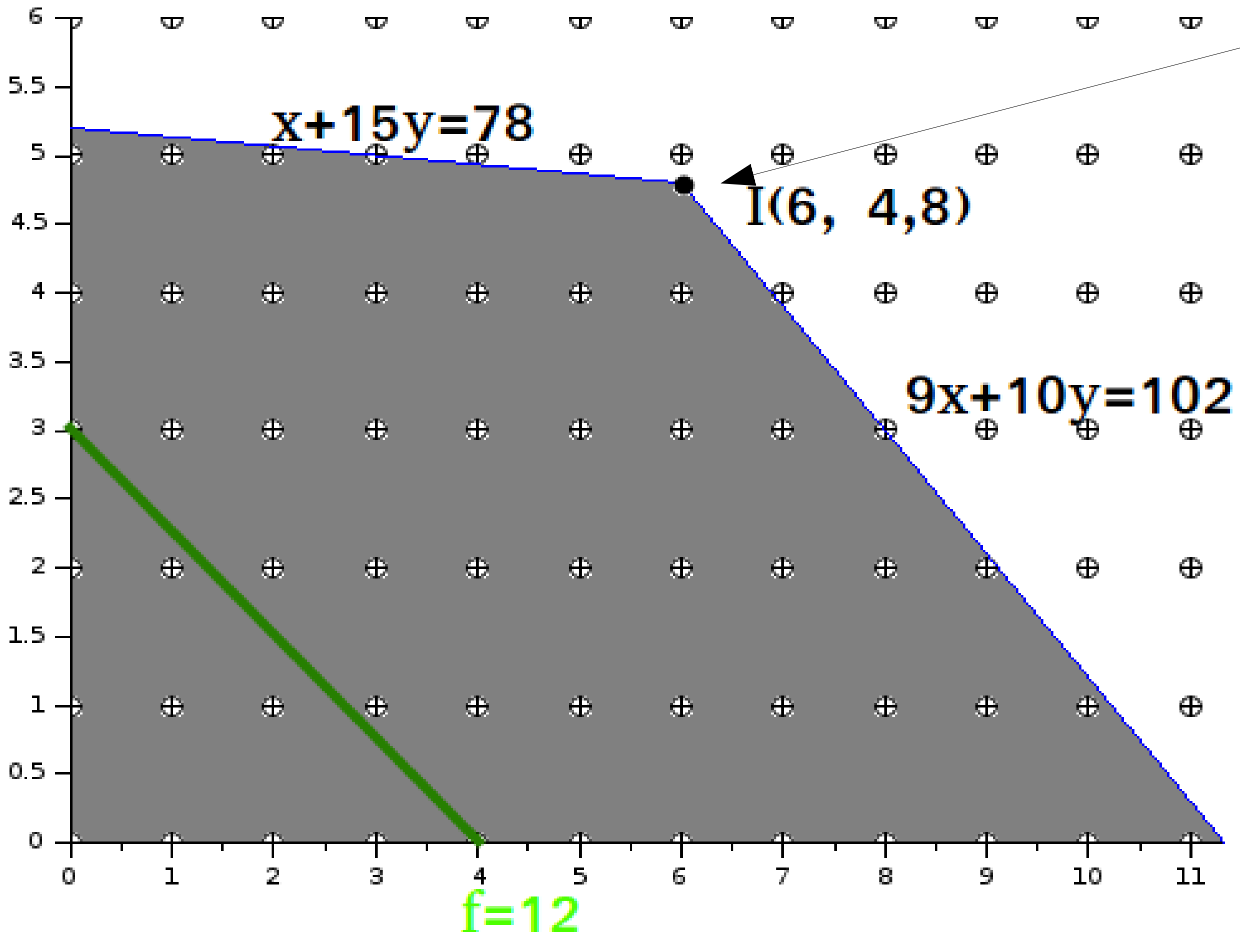
II) Programmation linéaire en nombre entier

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

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

Un exemple de programmation linéaire en nombre entier

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



Le point I correspond à la Solution si on enlève les Contraintes d'intégrité

Fmax est obtenue sur un des Points à coordonnées entières Du polygone admissible

II) Programmation linéaire en nombre entier (suite)

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

- La fonction **h** d'évaluation par excés est la solution du PLNE ***sans les contraintes d'intégrité***

  =>

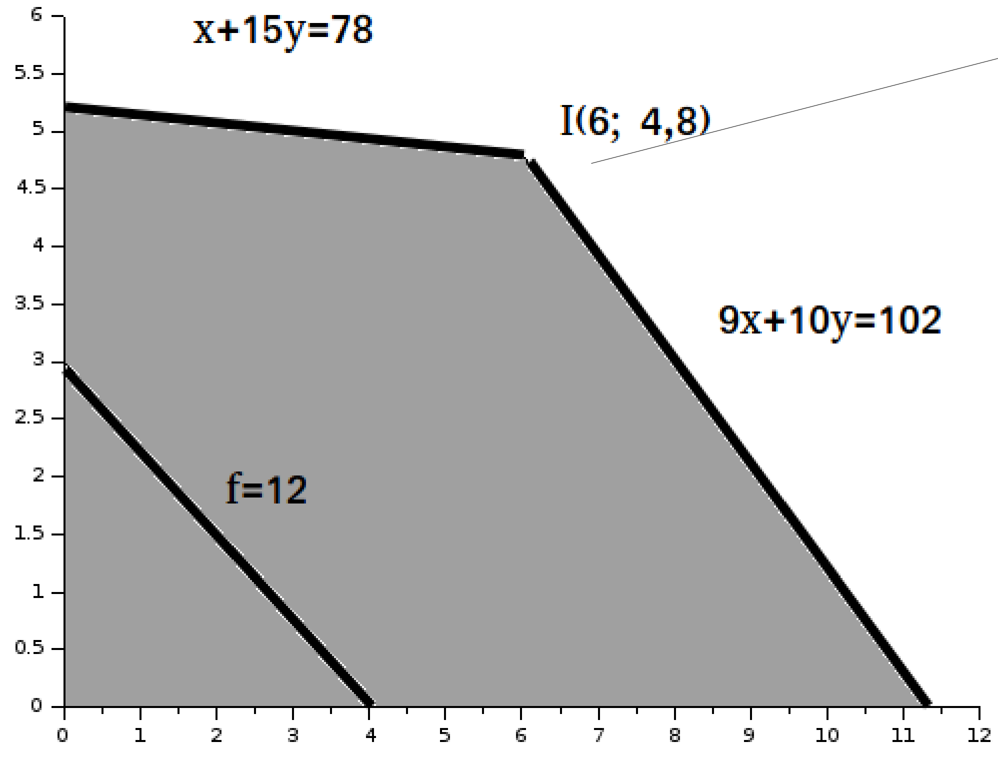
  on peut calculer h par l'algorithme du simplexe

  ou graphiquement .

Calcul de h(E) sur l'exemple :

Début de l'algorithme : U=0

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



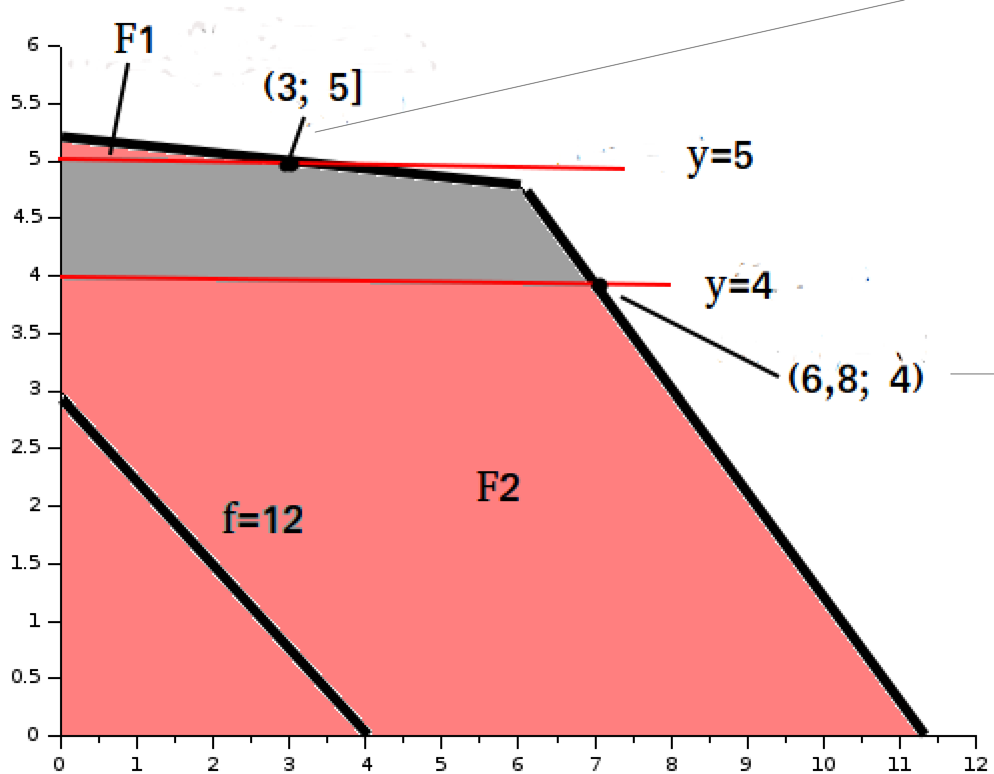
h(E)=3X6+4X4,8=37,2

On sépare E en 2 :
F1=E∩{y≤4}
F2=E∩{y≥5}

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

U=O

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



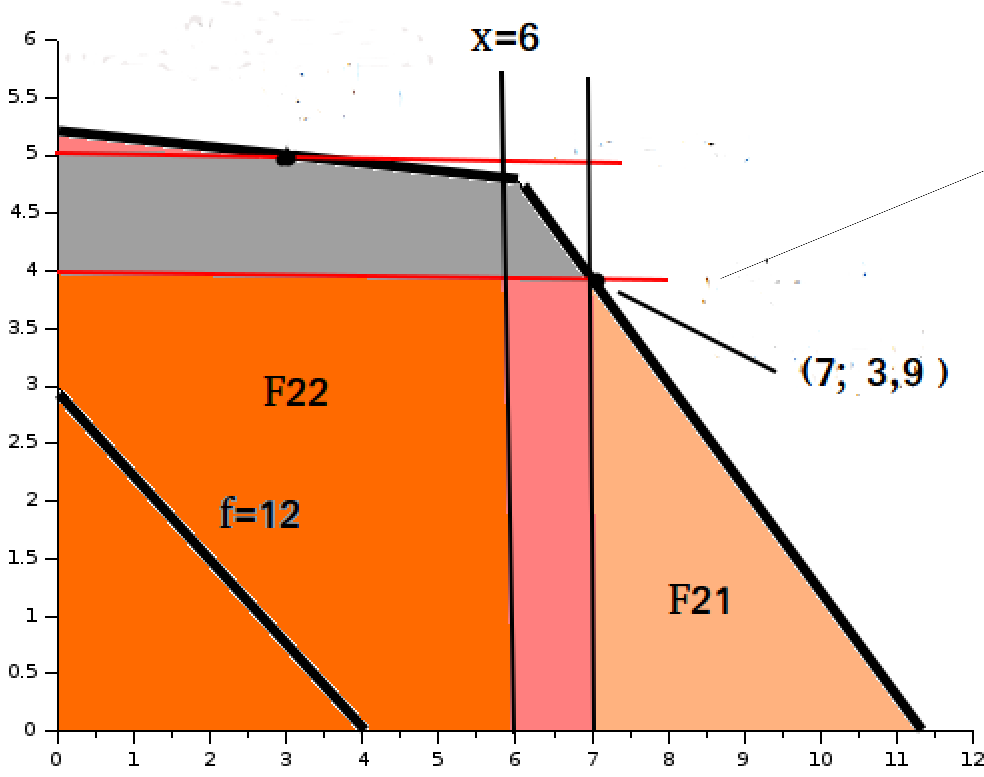
Sur F1 : max en (3,5)
=>h(F1)=3X3+4X5=29
=>on élague F1
Et U devient 29

Sur F2 : max en (6,8 ; 4)
=>h(F2)=3X6,8+4X4=36,4>U
=> On sépare F2 en 2 :
F21=F2∩{x≥7}
F22=F2∩{x≤6}

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

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

U=29



Sur F21 :

h(F21)=3X7+4X3,9=36,6>U

=> On sépare F21 en 2 :

F211=F21∩{y≤3}

F212=F21∩{y≥4}=Ø (on élague)

Sur F22 :

Avant d'évaluer F22 on

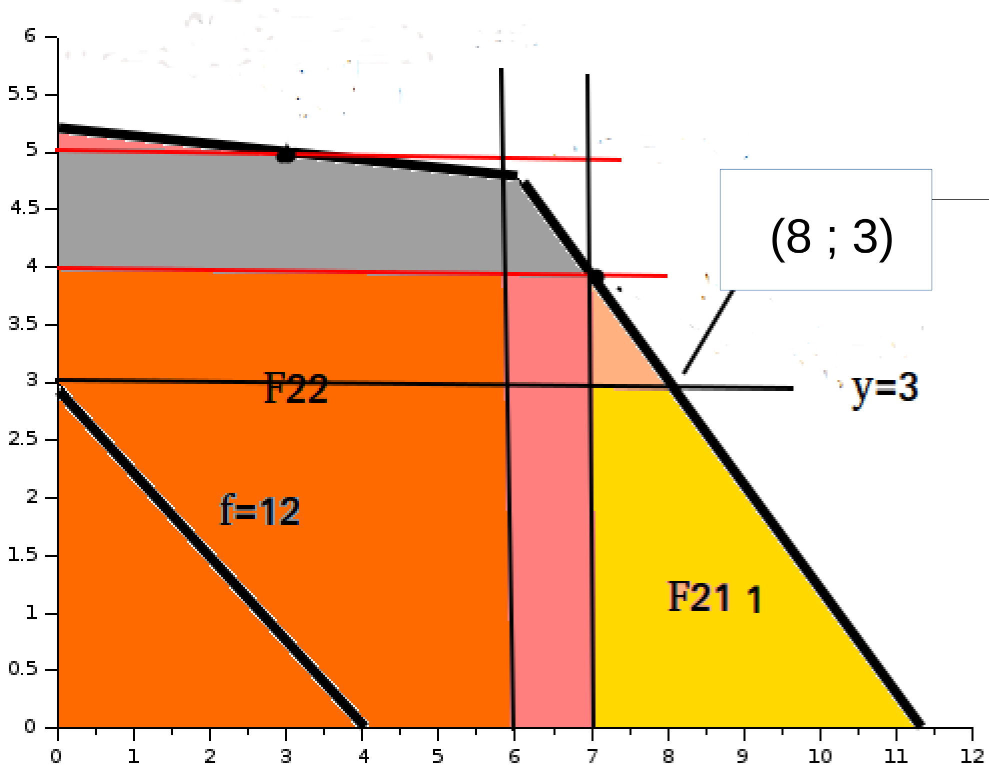
Étudie F211

(parcours en profondeur)

Suite de l’algorithme de séparation et évaluation sur l’exemple

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

U=29



Sur F211 : max sur (8,3)
h(F211)=3X8+4X3=36>U
=>
On élague F211
Et **U devient 36**

F22 : max en (6;4) :
h(F22)=3X6+4X4=34<U
=> On élague F22

La solution est donc fmax=36 pour x=8 et y=3

Exercices de programmation linéaire en nombres entiers

Exercice 1 :

On désire mettre dans un sac à dos deux types de produits :
— le produit 1 qui pèse 5 kg et qui rapporte 4 euros par unité.
— le produit 2 qui pèse 7kg et qui rapporte 6 euros par unité.
On peut mettre dans ce sac à dos un maximum de 20 kg. Chaque produit doit être mis en quantité entière.
On notera x l'inconnue représentant la quantité de produit 1 et y celle de produit 2.

1. Quel est le produit qui rapporte le plus par unité de poids ?
2. Si on n'imposait pas à x et y d'être des entiers, quelle serait la solution du problème ?
3. Donnez un encadrement de x puis de y .
4. Ecrivez le bénéfice total du sac à dos en fonction de x et de y .
5. Ecrivez la contrainte sur le poids sous la forme d'une inéquation en x et y .
6. Résolvez ce problème de sac à dos en utilisant un algorithme de séparation et évaluation.
7. Comment pouvait-on résoudre ce problème graphiquement ?

Exercice 2 :

On considère le même sac à dos que dans l'exercice précédent mais on rajoute un troisième produit. Les données sont consignées dans le tableau suivant :

$P_{tot} = 20kg$

produit i	1	2	3
bénéfice b_i	4	6	7
poids p_i	5	7	8

On note z la quantité du troisième produit. Les autres notations sont inchangées.

1. Quel est le produit qui rapporte le plus par unité de poids ?
2. Si on n'imposait pas à x , y et z d'être des entiers, quelle serait la solution du problème ?
3. On considère l'ensemble F des sacs à dos pour lesquels $z = 1$. Comment obtenir un majorant simple du bénéfice sur l'ensemble F. On notera $h(F)$ ce majorant.
4. Donnez un encadrement de x , de y puis de z .
5. Ecrivez le bénéfice total du sac à dos en fonction de x , y et z.
6. Ecrivez la contrainte sur le poids sous la forme d'une inéquation en x , y et z.
7. Résolvez ce problème de sac à dos en utilisant un algorithme de séparation et évaluation.

Exercice 3 :

Résolvez en utilisant la méthode de séparation et évaluation les 2 problèmes de sac à dos suivants :

1. $P_{tot} = 22kg$

produit i	1	2	3
bénéfice b_i	4	6	7
poids p_i	5	7	8

2. $P_{tot} = 20kg$

produit i	1	2	3
bénéfice b_i	5	6	7
poids p_i	5	7	6

Exercice 4 :

Résolvez le problème de sac à dos suivant en utilisant la méthode de séparation et évaluation :

$P_{Tot} = 26$ kg

produit	A	B	C	D
bénéfice	3	4	5	6
poids	2	3	4	5

Exercice 5 :

On considère le problème de programmation linéaire en nombres entiers suivant :

$$P \begin{cases} f(x,y) = 3x + 5y \\ x + 2y \leq 3 \\ 6x + 8y \leq 15 \\ x \geq 0;\ y \geq 0 \\ x \in \mathbb{N},\ y \in \mathbb{N} \end{cases}$$

1. On considère la relaxation linéaire RP du problème précédent.
   (a) Représentez graphiquement le polyèdre des contraintes du problème RP .
   (b) Donnez en extension l'ensemble E des points à coordonnées entières du polyèdre des contraintes.
   (c) Résolvez graphiquement le problème RP .
   (d) Déduisez-en un majorant de $f_{max}(E)$ (maximum de la fonction f sur l'ensemble E).
2. Résolvez le problème P initial en utilisant la méthode de séparation et évaluation du cours. Vous résoudrez graphiquement chacun des problèmes de relaxation.

Exercice 6 :

Résolvez les problèmes de programmation linéaire en nombres entiers suivants en utilisant la méthode de séparation et évaluation. Pour chacun des problèmes de relaxation linéaire vous utiliserez la méthode graphique.

1. $P_1 \begin{cases} f_1(x,y) = x + 5y \\ x + 10y \leq 20 \\ x \leq 2 \\ x, y \geq 0 \\ x, y \in \mathbb{N} \end{cases}$
2. $P_2 \begin{cases} f_2(x,y) = 3x + y \\ 8x + 3y \leq 20 \\ 3x + 3y \leq 16 \\ x, y \geq 0 \\ x, y \in \mathbb{N} \end{cases}$
3. $P_3 \begin{cases} f_3(x,y) = 8x + 5y \\ x + y \leq 6 \\ 9x + 5y \leq 45 \\ x, y \geq 0 \\ x, y \in \mathbb{N} \end{cases}$

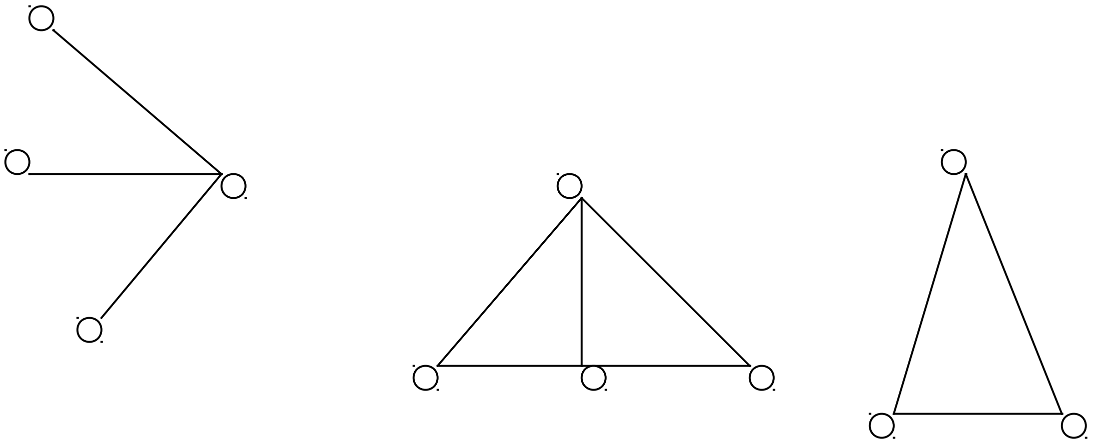
Graphes Eulériens
et
Graphes hamiltoniens

I)Graphes Eulériens

- On appelle parcours eulérien d’un graphe G tout parcours qui contient ***une fois et une seule chaque arête*** du graphe.

- On appelle graphe eulérien un graphe admettant un parcours eulérien ***fermé***.

- Parmi ces trois graphes un seul est Eulérien:



théorème d'Euler

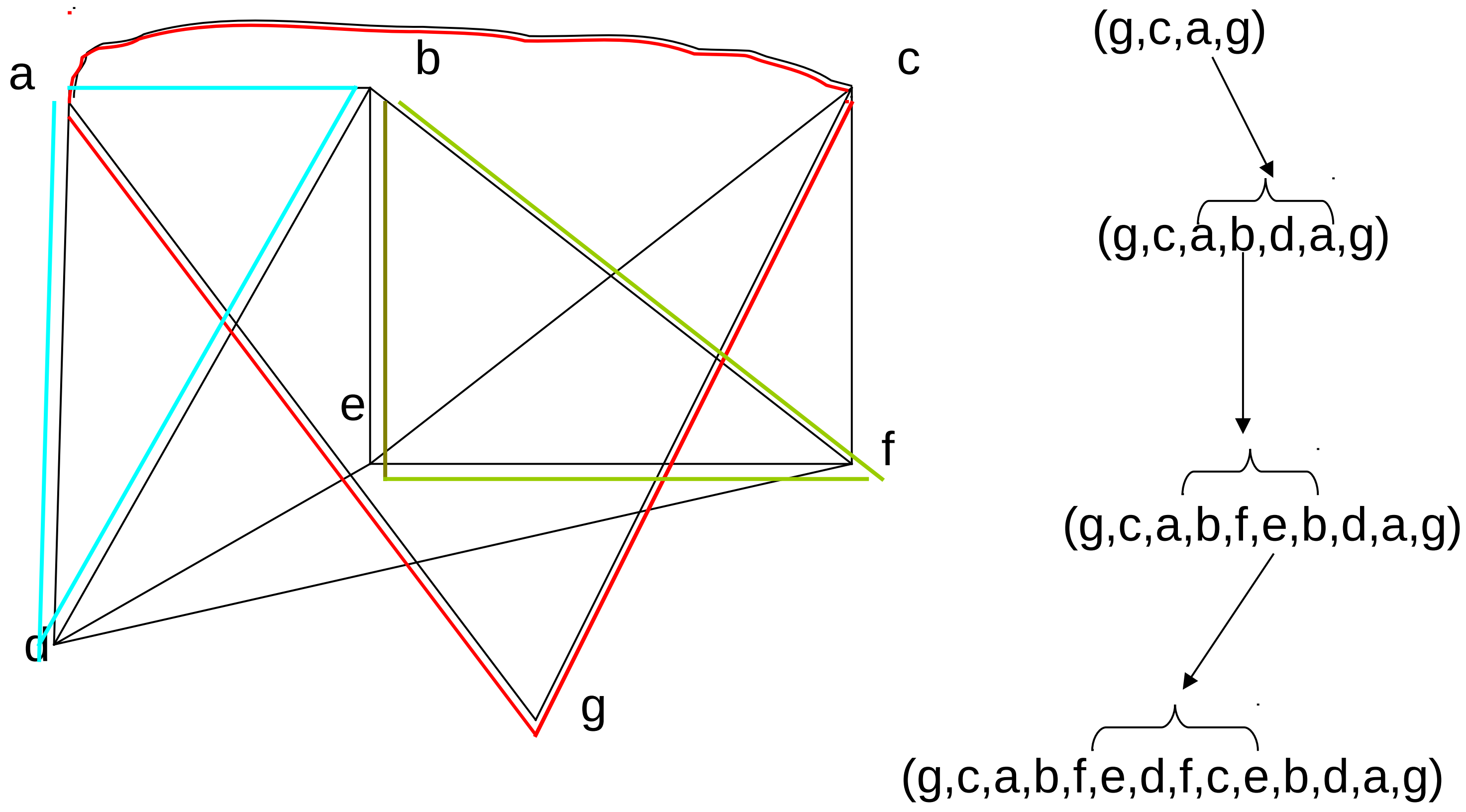
- Soit G =(X,E) un graphe **connexe**.

  G admet un **parcours eulérien fermé (i.e.** un *parcours fermé* qui passe une et une seule fois par chaque arête)

  $\Leftrightarrow$

  $\forall x \in X$, deg(x) est **pair**.

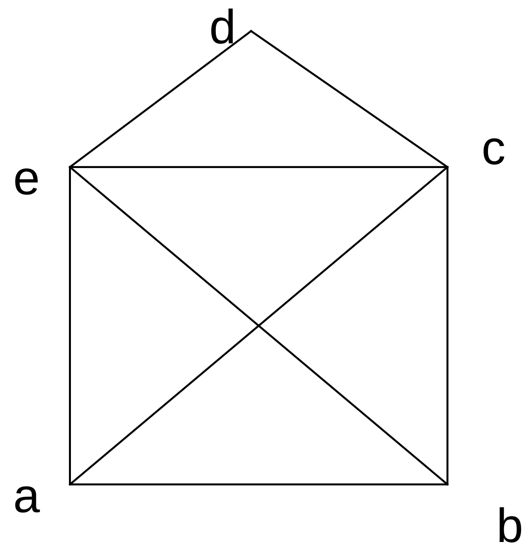
Idée de l’algorithme:



- **Théorème sur les parcours eulériens ouverts** :
- *Une condition nécessaire et suffisante* pour qu'un graphe connexe possède un parcours *eulérien ouvert* est que tous ses sommets, à l'exception de 2 d'entre eux, soient de degré pair.
- Lorsque la condition est satisfaite, tout parcours eulérien ouvert du graphe admet ces deux sommets de degré impair comme extrémités.

II) Graphes hamiltoniens

- *Un parcours hamiltonien* d'un graphe G est un parcours fermé dans lequel chaque sommet du graphe n'a qu'une seule occurrence.



Π=(a, b, c, d, e, a)

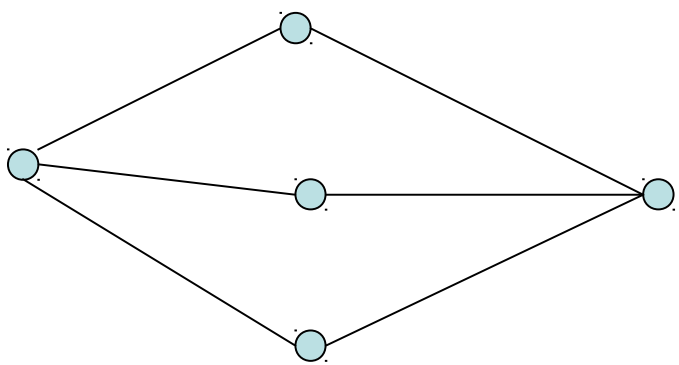
Graphes et cycles hamiltoniens

- Un **cycle hamiltonien** d’un graphe G d’ordre n est un sous-graphe de G qui est un cycle d’ordre n.
- Un **graphe hamiltonien** est un graphe admettant un cycle hamiltonien.(donc aussi un parcours hamiltonien)

Quelques remarques:

- 1) *Si* **G** *hamiltonien* alors $n \geq 3$.
- 2) Tout graphe complet avec $n \geq 3$ est hamiltonien.
- 3) *Si* **G** *est hamiltonien* alors **G** n'a pas de sommets de degré 1.
- 4) *Si* **G** *est hamiltonien* alors **G** est connexe.
- 5) *Si* **G** *est hamiltonien* alors **G** n'a pas de points d'articulation.

Les conditions précédentes ne sont pas suffisantes:

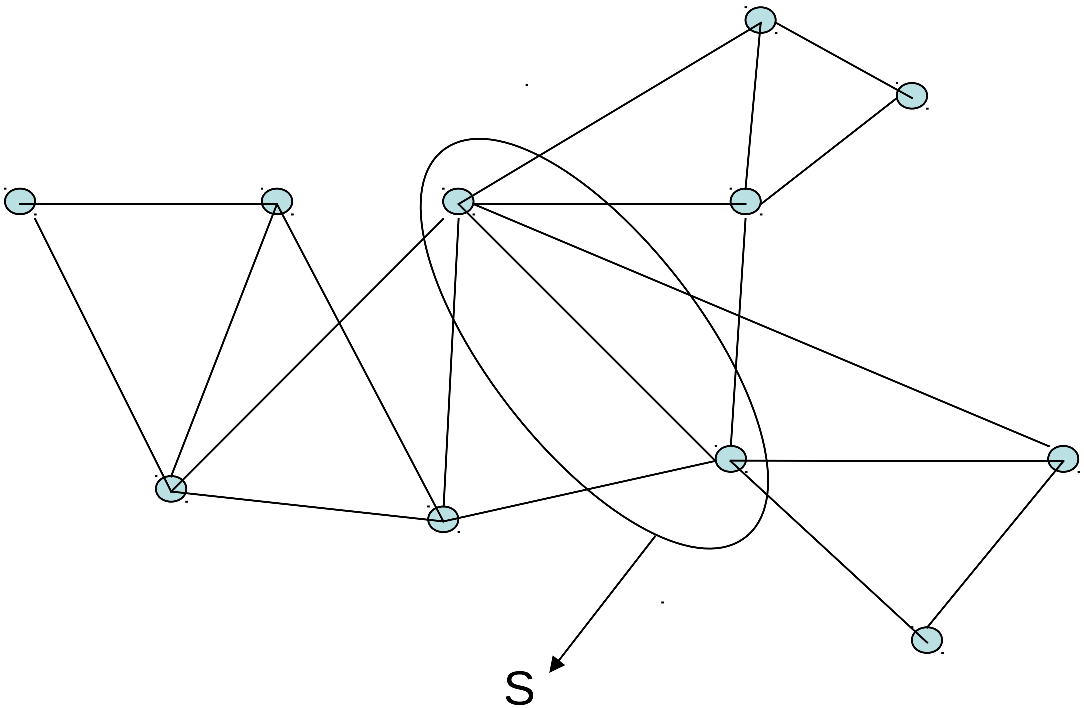


Une autre condition nécessaire mais pas suffisante

- Si <u>un graphe **G** est hamiltonien</u> alors:

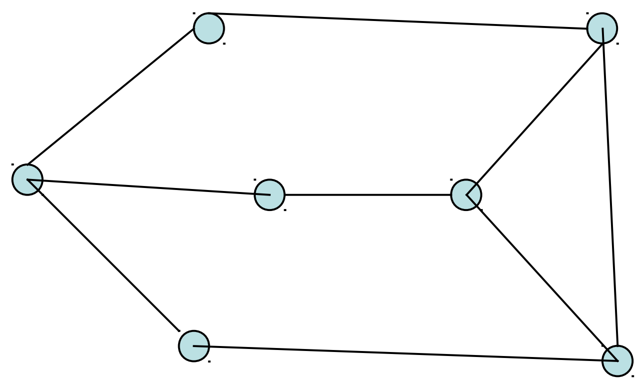
  Pour **tout sous-ensemble S non vide** de ses sommets, le nombre de composantes connexes de **G-S** doit être inférieur ou égal à **|S|**

G est-il hamiltonien?



G-S a 3 composantes connexes alors que **S** n’a que deux sommets.
Donc ce graphe **n’est pas hamiltonien**.

Mais cette condition n'est toujours pas suffisante!



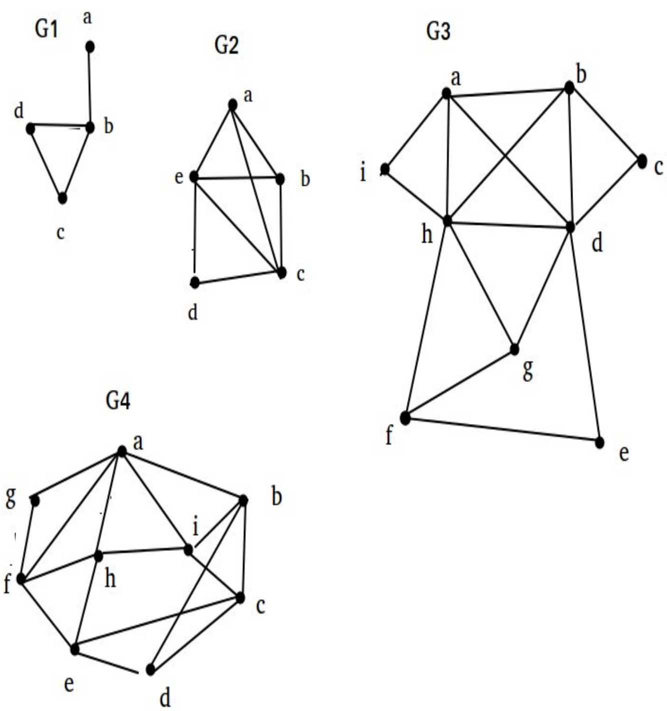
Une condition suffisante mais pas nécessaire:

- Si dans un graphe **G** d'ordre n>2 tous les sommets sont de degré au moins **n/2**, ou si, de façon plus générale, pour *toute paire* **x** et **y** de sommets de **G** on a **deg(x) +deg(y)** $\geq$ **n**, alors le graphe est hamiltonien.

Les graphes eulériens et les graphes hamiltoniens

Exercice 1 :

Pour chacun des graphes suivants, précisez s'il admet un parcours eulérien ou un parcours eulérien fermé ou ni l'un ni l'autre. Déterminez lorsqu'ils existent ces parcours.



Exercice 2 :

Déterminez, à un isomorphisme près, tous les graphes eulériens d'ordre 3, 4 et 5 .

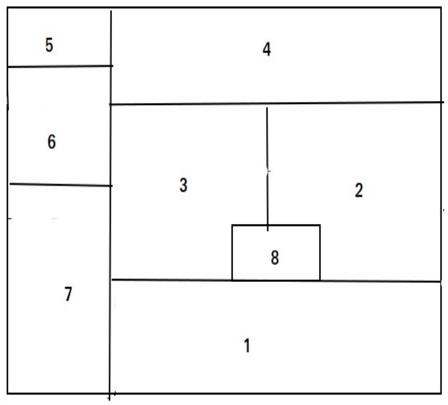
Exercice 3 :

On considère un jeu de dominos utilisant les chiffres 0,1,2,3 et 4, tel que sur chaque domino figurent deux chiffres distincts.

1. On vous propose de résoudre le problème P : "Est-il possible d'aligner tous les dominos de sorte que, lorsque deux pièces se touchent, les chiffres en contact soient identiques".
2. Reprenez le problème P avec des chiffres allant de 0 à 5, puis de 0 à 6.
3. Reprenez le problème P en supposant que les chiffres sur chaque domino ne sont pas forcément distincts.

Exercice 4 :

Huit pays sont représentés ci-dessous avec leurs frontières :



1. Représentez cette situation par un graphe G où les pays sont les sommets du graphe.
2. Est-il possible de partir d'un pays et d'y revenir après avoir franchi chaque frontière une fois et une seule ?
3. Est-il possible de partir d'un pays, de franchir chaque frontière une fois et une seule et de revenir dans un autre pays ?
4. On veut colorier la carte de façon à ce que deux pays ayant une frontière commune ne soient pas de la même couleur. Combien faut-il de couleurs au minimum ?

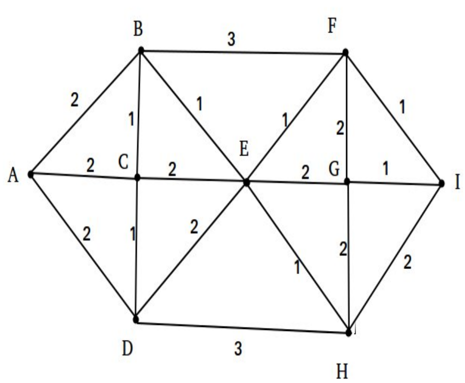
Exercice 5 :

Soit le graphe $G = (X, E)$ avec $X = \{a, b, c, d, e, f, g\}$ et $E = \{ab, af, bc, bf, cd, cg, de, dg, ef, eg\}$.

1. G admet-il un parcours eulérien fermé ? ouvert ?
2. Dessinez un graphe H_1 admettant un parcours eulérien fermé et obtenu à partir de G en rajoutant un nombre minimum d'arêtes.
3. Dessinez un graphe H_2 admettant un parcours eulérien ouvert et obtenu à partir de G en rajoutant un nombre minimum d'arêtes.
4. Les arêtes du graphe G représentent les galeries d'un musée tandis que les sommets sont les points de rencontre de ces galeries. Chacune de ces galeries fait 50 mètres de long. Un gardien de nuit posté en a doit effectuer une ronde toutes les heures. Au cours d'une ronde il doit parcourir entièrement au moins une fois chaque galerie et revenir à son poste.
   Donnez le parcours d'une ronde de longueur minimale

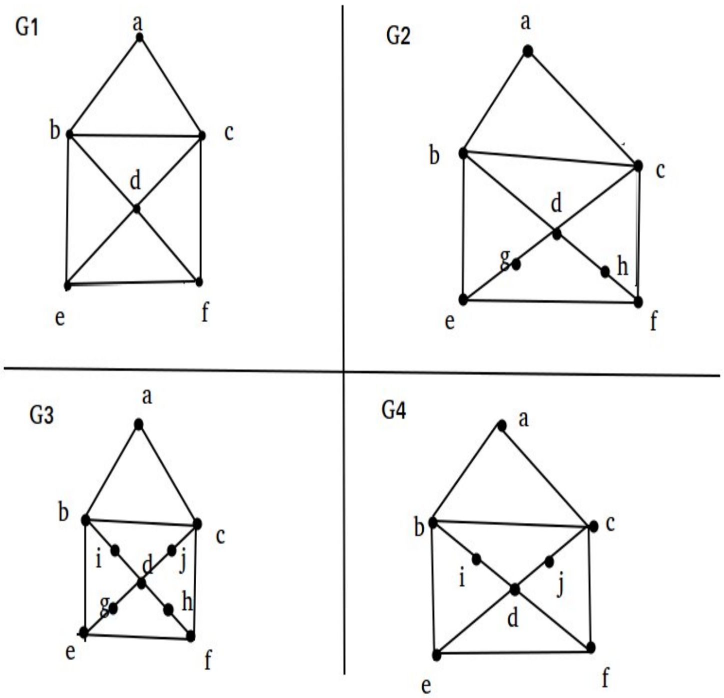
Exercice 6 :

Trouvez un parcours fermé, de longueur minimale et passant au moins une fois par chacune des arêtes.



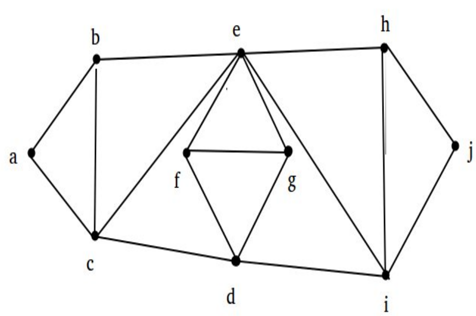
Exercice 7 :

Les graphes suivants sont-ils hamiltoniens ? Justifiez vos réponses.



Exercice 8 :

Démontrez, en utilisant les théorèmes du cours, que le graphe suivant n'est pas hamiltonien :



Exercice 9 :

On considère les séquences de degrés suivantes :

$$S_1 = (1, 1, 1, 1)$$

$$S_2 = (2, 2, 2, 2, 3, 3)$$

$$S_3 = (3, 3, 3, 3, 3)$$

$$S_4 = (3, 3, 3, 3, 3, 3)$$

1. Parmi ces séquences, une seule ne peut pas être celle d'un graphe simple. Laquelle ? Justifiez votre réponse.
2. Parmi les séquences restantes, une seule ne peut pas être celle d'un graphe connexe. Laquelle ?
3. Parmi ces séquences, une seule est celle d'un graphe nécessairement hamiltonien. Laquelle ? Justifiez votre réponse. Démontrez que pour les autres séquences il existe des graphes non hamiltoniens.

Exercice 10 :

1. Dans un club il y a 5 inscrits. Ils déjeunent régulièrement autour d'une table ronde (chaque membre a donc deux voisins de table). Combien ce club peut-il organiser de déjeuners de façon à ce que tous les inscrits n'aient jamais le même voisin ?
2. Même question avec 6 inscrits ? avec 7 inscrits ?

Contrôle de Recherche Opérationnelle 2017

Exercice 1:

On considère le problème d'optimisation linéaire suivant:

$$\begin{cases} f(x_1, x_2) = 2x_1 + 3x_2 \\ x_1 + x_2 \leq 5 \\ x_1 + 2x_2 \leq 7 \\ x_2 \leq 3 \\ x_1 \leq 4 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{cases}$$

Dans ce problème x_1 et x_2 sont deux inconnues réelles et il faut maximiser la fonction f .

1. Résolvez graphiquement le problème précédent.
   Vous représenterez précisément sur la page 3 de ce sujet le polygone des contraintes puis les droites d'équations respectives $f(x_1, x_2) = 6$ et $f(x_1, x_2) = f_{max}$ (où f_{max} est la valeur maximale de la fonction f sur le polygone des contraintes).

2. Résolvez numériquement le problème précédent en utilisant l'algorithme du simplexe tel qu'il vous a été présenté en cours. Vous exposerez en détail chacun des dictionnaires.

Exercice 2:

On veut remplir un sac à dos **de capacité maximale 26 kg** avec 3 types de produits afin de maximiser le bénéfice lors de la revente de ces produits. Chacun de ces produits doit être présent un nombre entier de fois dans le sac (0, 1, 2, 3 ....) et la somme des poids doit rester inférieure ou égale à 26 kg. On notera x_i la variable représentant la quantité d'objets i dans le sac à dos. Voici un tableau donnant pour chaque type de produit son bénéfice par unité et son poids par unité:

produits	1	2	3
bénéfice	9	4	1,4
poids	10	5	2

Résolvez ce problème en utilisant la méthode de séparation et évaluation.

Vous représenterez l'arbre de recherche et vous numéroterez chacune des étapes de l'algorithme. Pour chacune de ces étapes vous préciserez le sommet de l'arbre correspondant, la valeur de la fonction h d'évaluation par excés ainsi que la valeur de la variable U représentant la meilleure solution en cours.

Exercice 3:

On considère le problème d'optimisation linéaire en nombre entier suivant:

$$\begin{cases} f(x, y) = 2,5x + y \\ 2x + 7y \leq 28 \\ 4x + y \leq 17 \\ x \in \mathbb{N} \\ y \in \mathbb{N} \end{cases}$$

1. Tracez précisément sur la page 4 le polygone des contraintes ainsi que la droite d'équation $f(x, y) = 5$

2. Vérifiez que les points $A\begin{pmatrix} 4 \\ 1 \end{pmatrix}, B\begin{pmatrix} 0 \\ 4 \end{pmatrix}$ et $C\begin{pmatrix} 3 \\ 3 \end{pmatrix}$ sont dans le polygone des contraintes et placez-les sur le graphique.

3. Calculez les coordonnez du point d'intersection D des droites d'équations respectives $2x + 7y = 28$ et $4x + y = 17$ et placez ce point sur la figure.

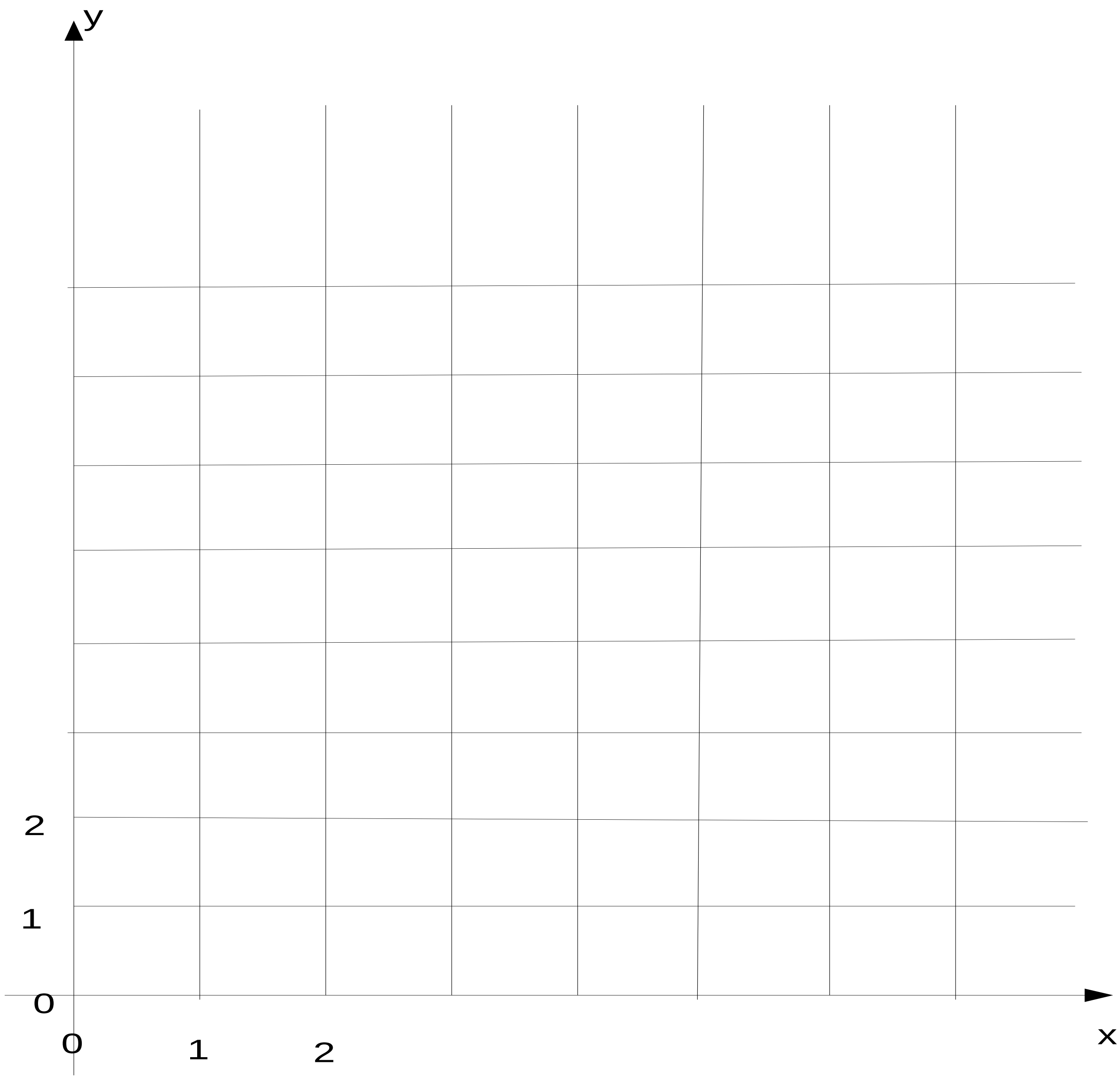
4. Utilisez la méthode de séparation et évaluation pour calculer f_{max} (i.e. le maximum recherché). Vous utiliserez la méthode graphique pour calculer la fonction h d'évaluation par excés (rappel: cette fonction est obtenue en enlevant les contraintes d'intégrité). Vous détaillerez chacune des étapes de la recherche.

Exercice 4:

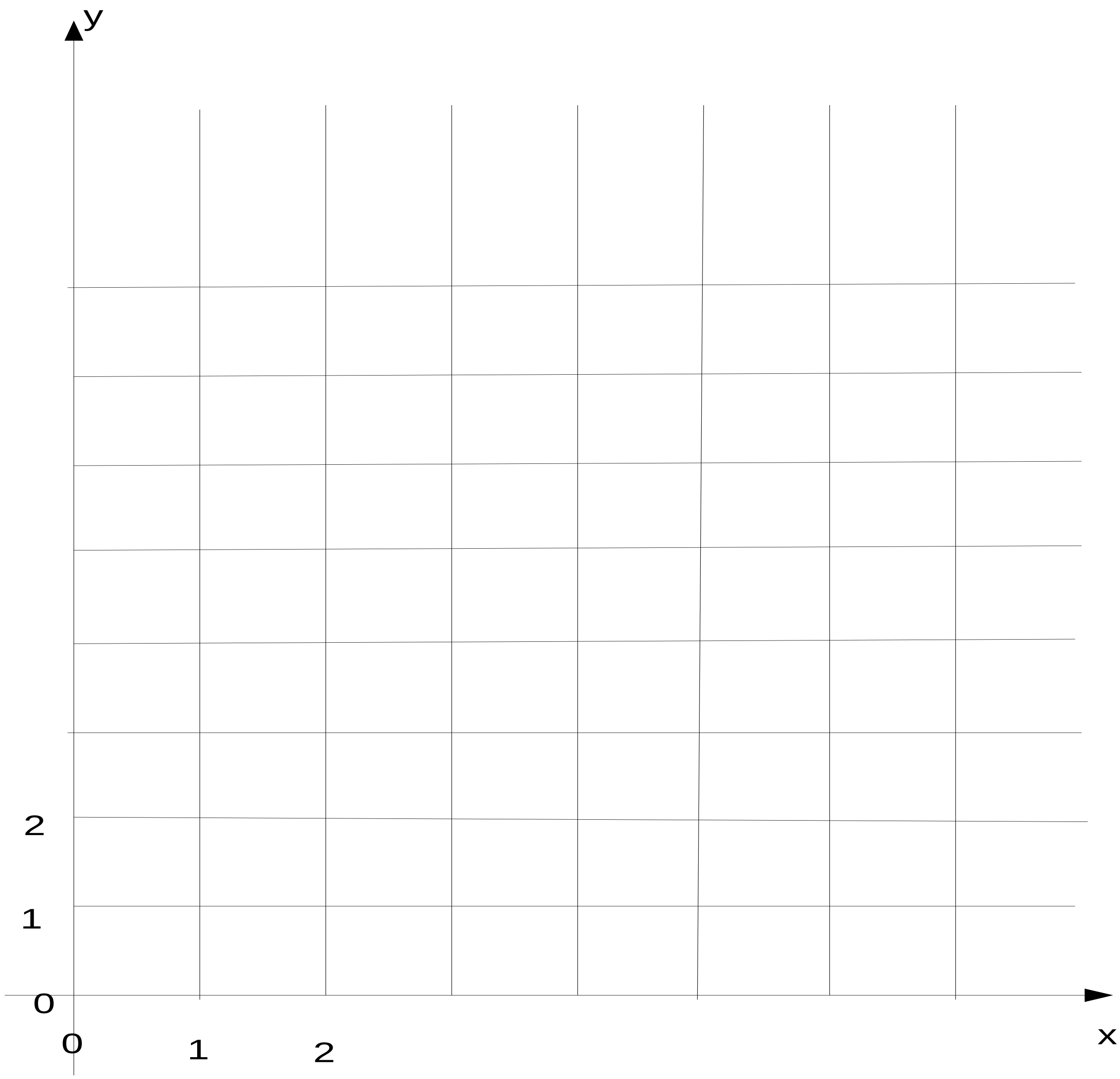
On considère le graphe défini par $G = (X = \{a, b, c, d, e, f, g, h\}, E = \{ab, ac, be, cd, ce, cg, de, ef, eg, fg, fh, gh\})$

1. Ce graphe admet-il un parcours eulérien ouvert? Si oui donnez le parcours, si non justifiez votre réponse.
2. Ce graphe admet-il un parcours eulérien fermé? Si oui donnez le parcours, si non justifiez votre réponse.
3. Ce graphe admet-il un cycle hamiltonien? Si oui donnez le cycle, si non justifiez.

Graphique de l'exercice 1:



Graphique de l'exercice 3:



Brouillon 1

Brouillon 2