Théorie des graphes et Optimisation

M1 Informatique

Julien Ah-Pine (julien.ah-pine@eric.univ-lyon2.fr)

Université Lyon 2 - FSEG

M1 Informatique 2010-2011

Plan du cours

Déroulement du cours

- 5 CM de 1h45
- 7 TD de 1h45 (2 groupes)
- 1 examen

Sources principales

- M. Gondran et M. Minoux, Graphes et Algorithmes (4ème édition), Lavoisier, 2009
- R. Bronson and G. Naadimuthu, Operations research (2nd édition), McGraw-Hill, 2007
- Supports de cours de l'EPFL (http ://roso.epfl.ch/teaching.html)

Motivations

Pourquoi les graphes ?

- Les graphes permettent de représenter de nombreux problèmes de manière intuitive et contribuent donc souvent à formaliser et résoudre ces derniers. ("Un bon dessin vaut mieux qu'un long discours")
- Les graphes sont rencontrés dans de nombreux domaines scientifiques : cartographie (réseaux routiers, réseaux internet,...) chimie-biologie (modélisation de molécules, ...), économie-gestion (planning de livraison, ordonnancement, ...), aide à la décision (agrégation multicritère, de préférences), recherche d'information (PageRank de Google, réseaux sociaux, ...), ...

Rappel du Sommaire

Rappel du Sommaire

Graphes : concepts orientés

Définition. (Graphe orienté)

Un graphe $G = [X, U]$ est déterminé par la donnée :

1. *d'un ensemble X dont les éléments sont appelés des* **sommets** *(ou des* **noeuds***). Si $N = |X|$ est le nombre de sommets, on dit que le graphe G est d'ordre N*
2. *d'un ensemble U dont les éléments $u \in U$ sont des couples ordonnés de sommets appelés des* **arcs***. Si $u = (i, j)$ est un arc de G, i est l'extrémité initiale et j l'extrémité terminale de u. On notera $M = |U|$ le nombre d'arcs.*

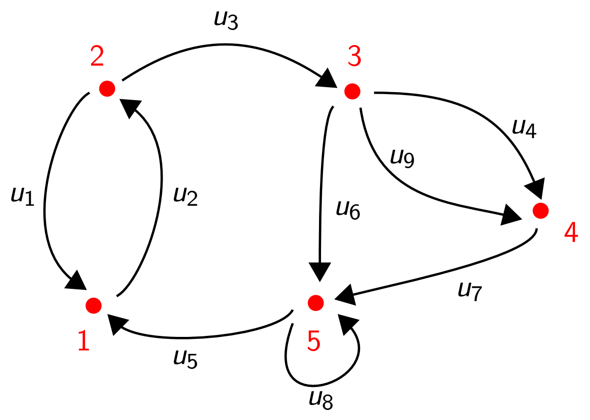
Graphes : concepts orientés

Définition. (Graphe orienté)

Un graphe $G = [X, U]$ est déterminé par la donnée :

1. *d'un ensemble X dont les éléments sont appelés des* **sommets** *(ou des* **noeuds***). Si $N = |X|$ est le nombre de sommets, on dit que le graphe G est d'ordre N*
2. *d'un ensemble U dont les éléments $u \in U$ sont des couples ordonnés de sommets appelés des* **arcs***. Si $u = (i, j)$ est un arc de G, i est l'extrémité initiale et j l'extrémité terminale de u. On notera $M = |U|$ le nombre d'arcs.*

Exemple de graphe orienté



$N = 5$
$M = 9$

Boucle et p-graphe

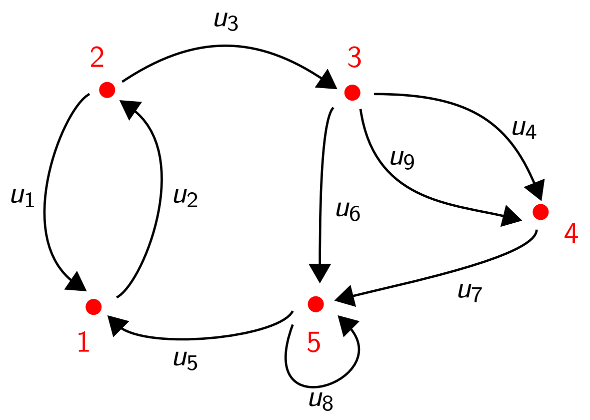
Définition. (Boucle)

Un arc $u = (i, i)$ dont les extrémités coïncident est appelé une **boucle***.*

Définition. (p-graphe)

Un **p-graphe** *est un graphe dans lequel il n'existe jamais plus de p arcs de la forme (i, j) entre deux sommets quelconques i et j, pris dans cet ordre.*

Exemple de boucle et d'un 2-graphe



$N = 5$
$M = 9$

Graphes et applications multivoques

Définition. (Ensemble de successeurs d'un sommet)

*j est un **successeur** de i s'il existe un arc de la forme (i,j). L'**ensemble des successeurs** d'un sommet $i \in X$ est noté $\Gamma(i)$.*

Définition. (Application multivoque)

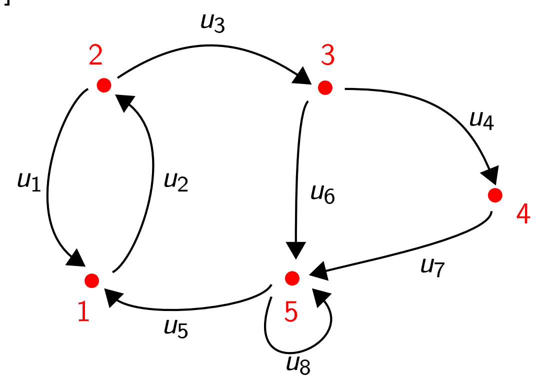
*L'application Γ qui, à tout élément de X, fait correspondre une partie de X ($\Gamma : X \to \mathcal{P}(X)$), est appelée une **application multivoque**.*

Définition. (Ensemble de prédécesseurs d'un sommet)

*j est un **prédecesseur** de i s'il existe un arc de la forme (j,i). L'**ensemble des successeurs** d'un sommet $i \in X$ est noté $\Gamma^{-1}(i)$. (Γ^{-1} est alors l'application multivoque réciproque de Γ).*

1-graphe et application multivoque

Si G est un 1-**graphe** alors il est parfaitement déterminé par la donnée de l'ensemble X et de l'application multivoque Γ. On peut donc aussi noté $G = [X, \Gamma]$.



$$\Gamma_1 = \{2\} \quad ; \quad \Gamma_2 = \{1,3\} \quad ; \quad \Gamma_3 = \{4,5\} \ldots$$

Graphes : concepts non orientés

Définition. (Graphe non orienté)

Un graphe $G = [X, U]$ dont les éléments $u \in U$ ne sont pas ordonnés est dit non orienté. Les éléments u sont alors appelés arêtes.

Il s'agit d'un graphe dont on ne s'intéresse pas à l'orientation des arcs. Dans ce cas, $(i,j) \in U$ est équivalent à $(j,i) \in U$.

Dans la suite du cours :

- L'utilisation du terme **arc** sous-entend que le graphe est **orienté**
- L'utilisation du terme *arête* sous-entend que le graphe est *non orienté*

Multigraphe et Graphe simple

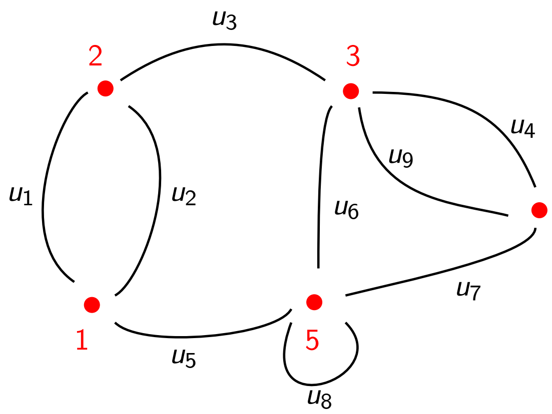
Définition. (Multigraphe)

Un multigraphe est un graphe pour lequel il peut exister plusieurs arêtes entre deux sommets i et j donnés.

Définition. (Graphe simple)

Un 1-graphe sans boucle est appelé graphe simple.

Exemple de multigraphe



$N = 5$
$M = 9$

Définitions principales

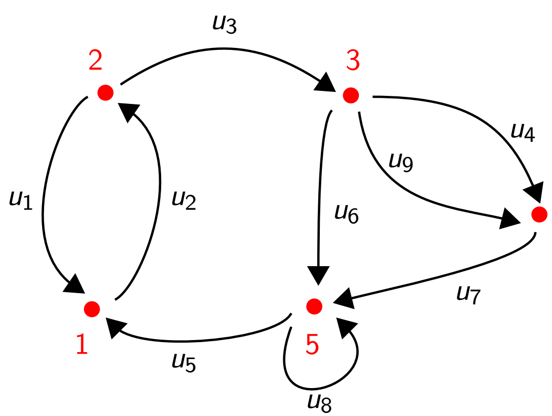
Définition. (Arcs adjacents, arêtes adjacentes)

Deux arcs (deux arêtes) sont dit ***adjacents*** *s'ils ont au moins une extrémité commune.*

Définition. (Degré et demi-degré)

- *le* ***demi-degré extérieur*** *du sommet i, noté $d_G^+(i)$ est le nombre d'arcs ayant i comme extrêmité* ***initiale***
- *le* ***demi-degré intérieur*** *du sommet i, noté $d_G^-(i)$ est le nombre d'arcs ayant i comme extrêmité* ***terminale***
- *le* ***degré*** *du sommet i, noté $d_G(i)$ est le nombre d'arcs (arêtes) ayant i comme extrémité et on a (pour les arcs) : $d_G(i) = d_G^+(i) + d_G(i)$*

Exemple



$N = 5$
$M = 9$

Graphes particuliers

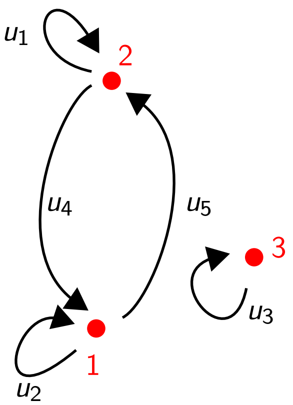
Soit G un 1-graphe orienté :

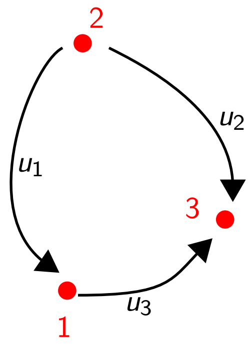
- G est **symétrique** si $\forall i, j \in X : (i,j) \in U \Rightarrow (j,i) \in U$
  (Remarque : tout graphe orienté qui est symétrique peut être vu comme un graphe non orienté et vice-versa.)
- G est **antisymétrique** si $\forall i, j \in X : (i,j) \in U \Rightarrow (j,i) \notin U$
- G est **complet** si $\forall i \neq j \in X : (i,j) \in U \vee (j,i) \in U$
- G est **transitif** si $\forall i, j, k \in X : (i,j) \in U \wedge (j,k) \in U \Rightarrow (i,k) \in U$

Soit G un graphe non orienté :

- G est symétrique
- Un graphe simple (1-graphe sans boucle) d'ordre N qui est complet est noté K_N
- un sous-ensemble $C \subset X$ tel que deux sommets quelconques de C sont reliés pas une arête est appelé une **clique**

Exemples





Sous-graphes, graphes partiels et graphes complémentaires

Définition. (Sous-graphes, graphes partiels et graphes complémentaires)

- *Le* **sous-graphe induit par** $A \subset X$ *est le graphe* G_A *dont les sommets sont les éléments de* A *dont les arcs sont les arcs de* G *ayant les deux extrémités dans* A
- *Soit* $G = [X, U]$ *et soit* $V \subset U$. *Le* **graphe partiel engendré par** V *est le graphe* $[X, V]$
- *Soit* $G = [X, U]$ *et soient* $A \subset X$, $V \subset U$. *Le* **sous-graphe partiel engendré par** A **et** V *est le graphe partiel de* G_A *engendré par* V
- *Etant donné un 1-graphe* $G = [X, U]$, *le* **graphe complémentaire de** G, *est le graphe* $[X, \overline{U}]$ *tel que :* $(i, j) \in U \Rightarrow (i, j \notin \overline{U})$ *et* $(i, j) \notin U \Rightarrow (i, j \in \overline{U})$
- *Un sous-ensemble de sommets* $C \subset X$ *dont toute paire de sommets est reliée par une arête est appelé une* **clique** *sommets quelconque*

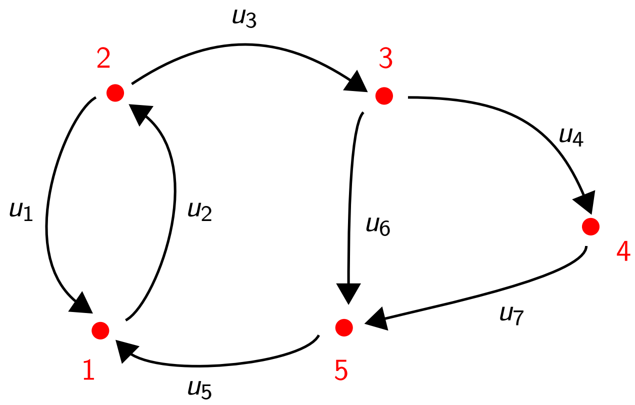
Rappel du Sommaire

Matrice d'incidence sommets-arcs

Définition. (Matrice d'incidence sommets-arcs)

Soit $G = [X, U]$ *un graphe sans boucle. La matrice d'incidence de* G *est une matrice* $A = (a_{iu})$, $i = 1, \ldots, N$, $u = 1, \ldots, M$, *à coefficients entiers dans* $\{0, 1, -1\}$ *telle que chaque colonne correspond à un arc de* G *et chaque ligne à un sommet. Si* $u = (i, j) \in U$ *alors la colonne* u *contient des termes nuls sauf pour les suivants :* $a_{iu} = 1$ *et* $a_{ju} = -1$.

Exemple de matrice d'incidence sommets-arcs



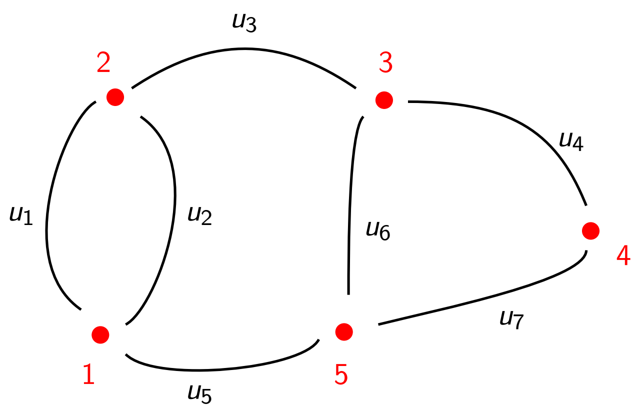
$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccccc} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 \end{array} \\ \begin{pmatrix} -1 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 \end{pmatrix} \end{array}$$

Matrice d'incidence sommets-arêtes

Définition. (Matrice d'incidence sommets-arêtes)

Soit $G = [X, U]$ un graphe non orienté sans boucle. La matrice d'incidence de G est une matrice $A = (a_{iu})$, $i = 1, \dots, N$, $u = 1, \dots, M$, à coefficients entiers dans $\{0, 1\}$ telle que chaque colonne correspond à une arête de G et chaque ligne à un sommet. Si $u = (i, j) \in U$ alors la colonne u contient des termes nuls sauf pour les suivants : $a_{iu} = 1$ et $a_{ju} = 1$.

Exemple de matrice d'incidence sommets-arêtes



$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccccc} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

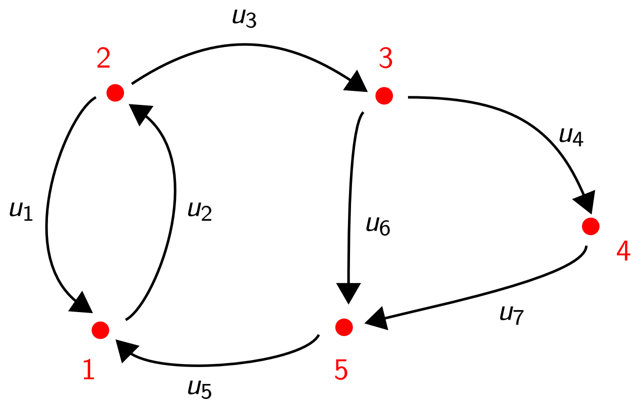
Matrice d'adjacence sommets-sommets

Définition. (Matrice d'adjacence sommets-sommets)

Soit $G = [X, U]$ un 1-graphe comportant éventuellement des boucles. La matrice d'adjacence de G est une matrice carrée $A = (a_{ij})$, $i = 1, \dots, N$, $j = 1, \dots, N$, à coefficients entiers dans $\{0, 1\}$ telle que chaque colonne correspond à un sommet de G et chaque ligne à un sommet de G et de terme général : $a_{ij} = 1 \Leftrightarrow (i, j) \in U$ ($a_{ij} = 0$ sinon).

Dans le cas non orienté, la matrice d'adjacence est symétrique.

Exemple de matrice d'adjacence



$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Rappel du Sommaire

Chaîne, chaîne élémentaire, cycle, cycle élémentaire

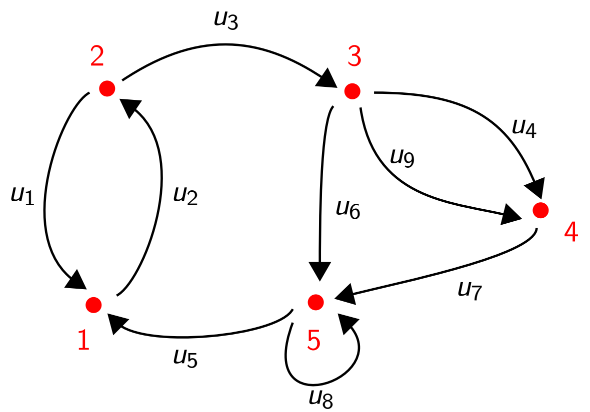
Définition. (Chaîne de longueur q et chaîne élélmentaire)

-Une chaîne de longueur q est une séquence de q arcs : $L = \{u_1, u_2, \ldots, u_q\}$ telle que chaque arc u_r de la séquence $(2 \leq r \leq q-1)$ ait une extrémité commune avec l'arc u_{r-1} $(u_{r-1} \neq u_r)$ et l'autre extrémité commune avec l'arc u_{r+1} $(u_{r+1} \neq u_r)$.
-L'extrémité i de u_1 non adjacente à u_2 et l'extrémité j de u_q non adjacente à u_{q-1} sont appelées les **extrémités de la chaîne** *L.*
-On appelle **chaîne élémentaire** *une chaîne telle qu'en la parcourant, on ne rencontre pas deux fois le même sommet.*

Définition. (Cycle et cycle élémentaire)

Un cycle est une chaîne dont les extrémités coïncident (chaîne fermée). Un **cycle élémentaire** *est un cycle minimal (pour l'inclusion) càd ne comprenant strictement aucun autre cycle. Dans un cycle élémentaire, on ne rencontre pas deux fois le même sommet (sauf l'origine).*

Exemple de matrice



$N = 5$
$M = 9$

La chaîne $L = \{u_2, u_5, u_6\}$ de longueur 3, joint le sommet 2 et 3.
La chaîne $L = \{u_1, u_5, u_6, u_3\}$ de longueur 4 est un cycle élémentaire.

Chemin, chemin élémentaire, circuit, circuit élémentaire

Définition. (Chemin de longueur q et chemin élélmentaire)

-Un chemin de longueur q est une séquence de q arcs :
$P = \{u_1, u_2, \ldots, u_q\}$ *avec*
$u_1 = (i_0, i_1); u_2 = (i_1, i_2); u_3 = (i_2, i_3); \ldots; u_q = (i_{q-1}, i_q)$. *(chaîne dont tous les arcs sont* **orientés dans le même sens**.*)*
-Les extrémité i_0 *et* i_q *sont respectivement appelés extrémités* **initiale** *et* **terminale** *du chemin* P.
-On appelle **chemin élémentaire** *un chemin telle qu'en la parcourant, on ne rencontre pas deux fois le même sommet.*

Définition. (Circuit et circuit élémentaire)

Un circuit est un chemin dont les extrémités coïncident (chemin fermée). Un **circuit élémentaire** *est un circuit minimal (pour l'inclusion) càd ne comprenant strictement aucun autre circuit. Dans un circuit élémentaire, on ne rencontre pas deux fois le même sommet (sauf l'origine).*

Parcours, parcours eulérien et hamiltonien

Un parcours d'un graphe G est une chaîne, un cycle, un chemin ou un circuit.

Définition. (Parcours eulérien)

Un parcours d'un graphe G est dit eulérien s'il passe une et une seule fois par chaque arc ou arête de G (il peut passer plusieurs fois par un même sommet).

Définition. (Parcours hamiltonien)

Un parcours d'un graphe G est dit hamiltonien s'il passe une et une seule fois par chaque sommet de G (et donc au plus une fois par chaque arc ou arête).

Fermeture transitive d'un graphe

Définition. (Fermeture transitive)

On appelle fermeture transitive de l'application multivoque Γ, *l'application multivoque* $\hat{\Gamma}$ *définie par :*

$$\hat{\Gamma}(i) = i \cup \Gamma(i) \cup \Gamma^2(i) \cup \Gamma^3(i) \cup \ldots \cup \Gamma^{N-1}(i)$$

où $\Gamma^2(i) = \Gamma(\Gamma(i))$.
$\Gamma^k(i)$ *reprśente l'ensemble des sommets que l'on peut atteindre à partir de* i *avec des chemins de longueur* k. $\hat{\Gamma}(i)$ *représente l'ensemble des sommets que l'on peut atteindre par un chemin à partir de* i.
- On dit que $\hat{\Gamma}(i)$ *est l'ensemble des* **descendants** *de* i.
- L'application réciproque (multivoque) $\widehat{\Gamma^{-1}}(i)$ *est l'ensemble des* **ancêtres** *de* i.

Connexité

Définition. (Graphe connexe)

Un graphe est dit connexe, si pour tout couple de sommets i *et* j, *il existe une chaîne joignant* i *et* j.

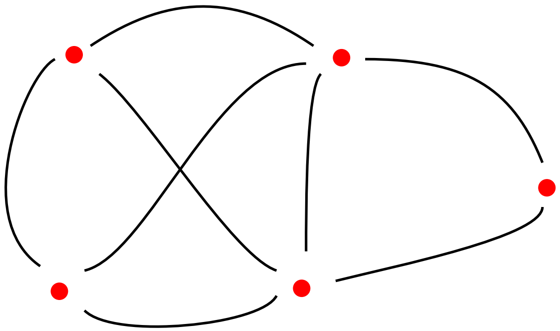
Définition. (Graphe orienté fortement connexe)

Un graphe orienté est dit fortement connexe, si pour tout couple de sommets ordonnés (i, j), *il existe un chemin joignant* i *et* j.

Théorème d'Euler

Théorème. (Théorème d'Euler)

Un mulrigraphe $G = [X, U]$ connexe admet un parcours eulérien ssi le nombre de sommets de degré impair est 0 ou 2. S'il y en a 0, alors il s'agit d'un cycle d'orignie quelconque. S'il y en a 2 alors le parcours est une chaîne reliant ces deux noeuds.



Rappel du Sommaire

Exemples d'applications

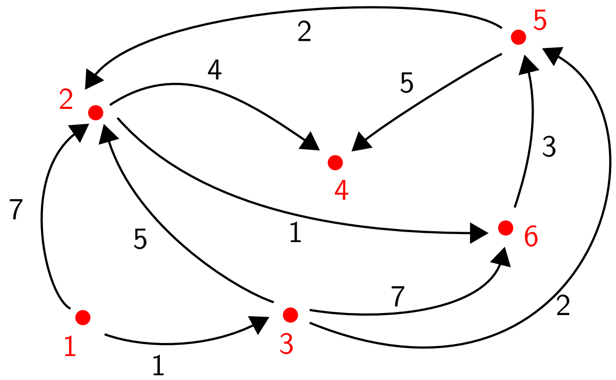
Les problèmes de cheminement dans les graphes sont parmi les problèmes les plus anciens de la théorie des graphes. Le problème du plus court chemin est, parmi ceux-ci, le plus typique et possède de nombreuses applications :

- Problèmes de tournées
- Problèmes d'optimisation de réseaux (routiers, télécommunications)
- certains problèmes d'investissements et de gestion de stocks
- certains problèmes en intelligence artificielle
- ...

Graphes valués

Définition. (Graphe valué)

Etant donné un graphe $G = [X, U]$, on associe à chaque arc u un nombre $l(u) \in \mathbb{R}$ appelé "longueur de l'arc". On dit que G est valué par les longueurs $l(u)$. Si $u = (i, j)$, on utilisera également la notation l_{ij} pour désigner la longueur de l'arc u.



Problème du plus court chemin

Définition. (Problème du plus court chemin)

Le problème du plus court chemin entre deux sommets i et j sera de trouver un chemin $\mu(i,j)$ de i à j dont la longueur totale :

$$l(\mu) = \sum_{u \in \mu(i,j)} l(u) \quad \textit{soit minimum}$$

La longueur d'un chemin est la somme des longueurs des arcs le consituant.

Algorithme de Moore-Dijkstra

Détermination du plus court chemin du sommet s aux autres sommets dans un graphe valué dont les longueurs sont positives càd $\forall u \in U : l(u) \geq 0$. Exemples de longueurs positives ou nulles :

- Temps (en minutes, en heures, ...)
- Coûts (en euros, ...), ...

Posons $X = \{1, 2, \dots, N\}$ et soit l_{ij} la longueur de l'arc $(i,j) \in U$.

Définissons $\pi^*(i)$ comme la longueur du plus court chemin de s à i. Nous avons $\pi^*(s) = 0$.

L'algorithme utilise la représentation du graphe par Γ. Il procède en $N-1$ itérations. Au début de chaque itération, l'ensemble des sommets est partitionné en deux sous-ensembles, S et $\overline{S} = X \setminus S$ avec $s \in S$.

Chaque sommet i de X est affecté d'une étiquette $\pi(i)$ qui vérifie la propriété suivante :

- si $i \in S, \pi(i) = \pi^*(i)$
- si $i \in \overline{S}, \pi(i) = \min_{j \in S, j \in \Gamma^{-1}(i)}(\pi(j) + l_{ji})$

Pseudo code de l'algorithme de Moore-Dijkstra

a) $\pi(s) \leftarrow 0$; $\overline{S} \leftarrow \{1, 2, \dots, N\} \setminus \{s\}$;
  Pour tout $(i \in \overline{S})$ faire :
    Si $(i \in \Gamma(s))$ alors
      $\pi(i) \leftarrow l_{si}$;
    Sinon
      $\pi(i) \leftarrow +\infty$;
    FSi
  FPour

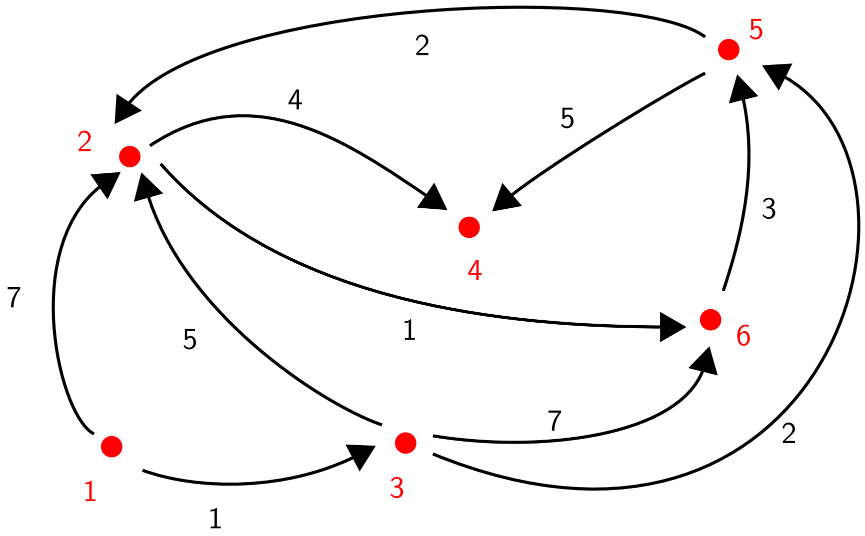
Eléments de la théorie des graphes | Problème du plus court chemin

Pseudo code de l'algorithme de Moore-Dijkstra

a) $\pi(s) \leftarrow 0$; $\overline{S} \leftarrow \{1, 2, \dots, N\} \setminus \{s\}$;
  Pour tout $(i \in \overline{S})$ faire :
    Si $(i \in \Gamma(s))$ alors
      $\pi(i) \leftarrow l_{si}$;
    Sinon
      $\pi(i) \leftarrow +\infty$;
    FSi
  FPour

b) **Tant que** $(\overline{S} \neq \emptyset)$ faire :
    Sélectionner $(i^* \in \overline{S})$ tel que $\pi(i^*) = \min_{i \in \overline{S}}\{\pi(i)\}$
    $\overline{S} \leftarrow \overline{S} \setminus \{i^*\}$
    Pour tout $(i \in \Gamma_{i^*} \cap \overline{S})$ faire :
      $\pi(i) \leftarrow \min(\pi(i), \pi(i^*) + l_{i^*i})$;
    FPour
  FTq

Exemple d'application

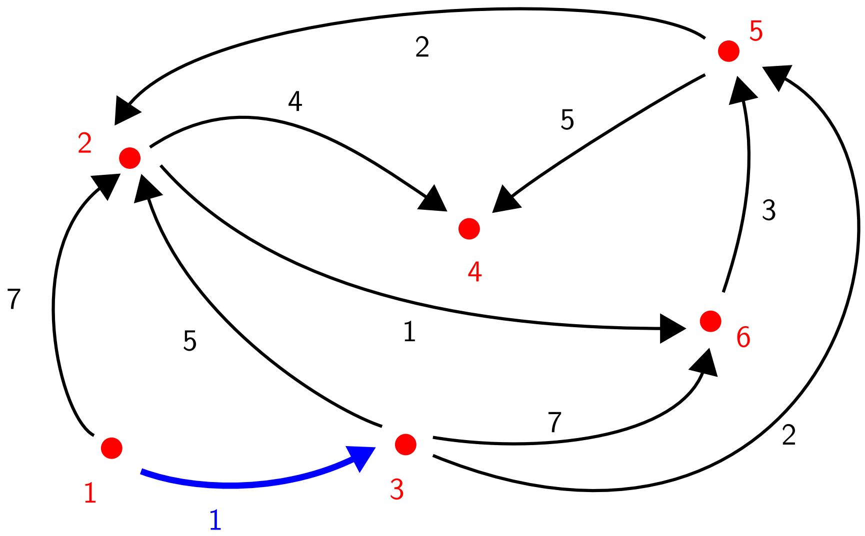


Calcul du plus court chemin entre 1 et les autres sommets.

Déroulement de l'algorithme

1. (initialisation, étape a)) $S = \{1\}$, $\pi = (0, 7, 1, +\infty, +\infty, +\infty)$

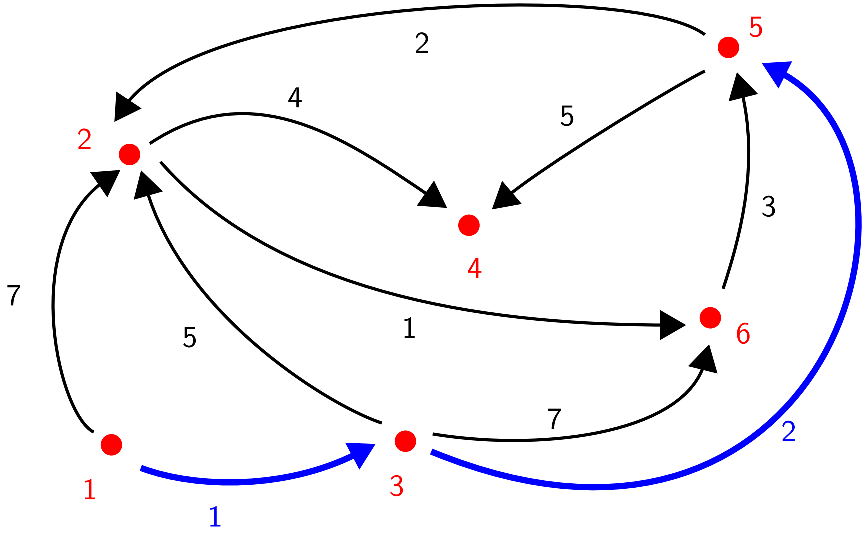
Exemple d'application : cas d'un graphe orienté



Déroulement de l'algorithme

1. (initialisation) $S = \{1\}$, $\pi = (0, 7, 1, +\infty, +\infty, +\infty)$
2. $j = 3$, $S = \{1, 3\}$, $\Gamma_3 \cap \overline{S} = \{2, 5, 6\}$, $\pi(2) = \min(7, 1 + 5) = 6$, $\pi(5) = \min(+\infty, 1 + 2) = 3$, $\pi(6) = \min(+\infty, 1 + 7) = 8$

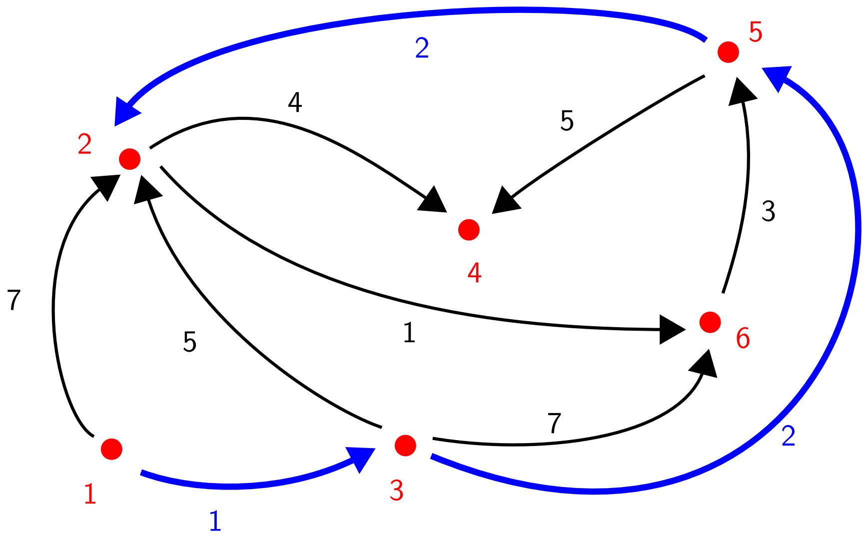
Exemple d'application : cas d'un graphe orienté



Déroulement de l'algorithme

1. (initialisation) $S = \{1\}$, $\pi = (0, 7, 1, +\infty, +\infty, +\infty)$
2. $j = 3$, $S = \{1, 3\}$, $\Gamma_3 \cap \overline{S} = \{2, 5, 6\}$, $\pi(2) = \min(7, 1 + 5) = 6$, $\pi(5) = \min(+\infty, 1 + 2) = 3$, $\pi(6) = \min(+\infty, 1 + 7) = 8$
3. $j = 5$, $S = \{1, 3, 5\}$, $\Gamma_5 \cap \overline{S} = \{2, 4\}$, $\pi(2) = \min(6, 3 + 2) = 5$, $\pi(4) = \min(+\infty, 3 + 5) = 8$

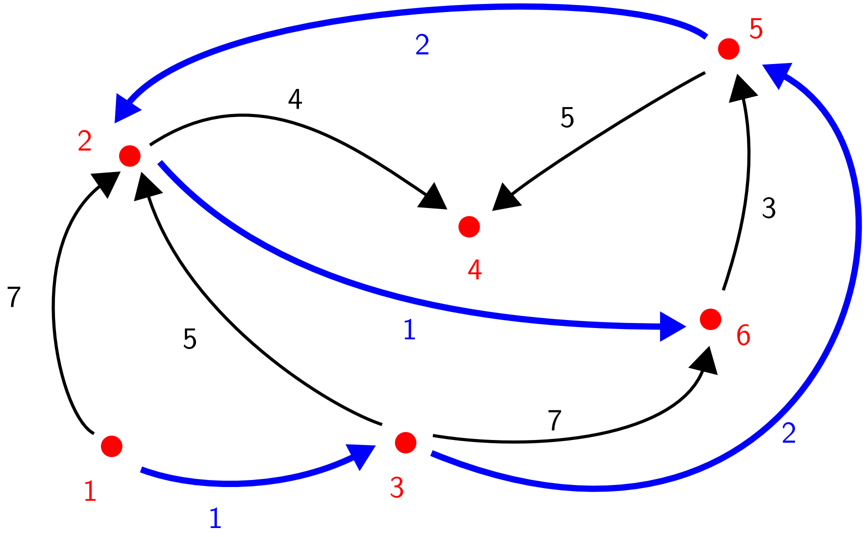
Exemple d'application : cas d'un graphe orienté



Déroulement de l'algorithme

1. (initialisation) $S = \{1\}$, $\pi = (0, 7, 1, +\infty, +\infty, +\infty)$
2. $j = 3$, $S = \{1, 3\}$, $\Gamma_3 \cap \overline{S} = \{2, 5, 6\}$, $\pi(2) = \min(7, 1 + 5) = 6$, $\pi(5) = \min(+\infty, 1 + 2) = 3$, $\pi(6) = \min(+\infty, 1 + 7) = 8$
3. $j = 5$, $S = \{1, 3, 5\}$, $\Gamma_5 \cap \overline{S} = \{2, 4\}$, $\pi(2) = \min(6, 3 + 2) = 5$, $\pi(4) = \min(+\infty, 3 + 5) = 8$
4. $j = 2$, $S = \{1, 2, 3, 5\}$, $\Gamma_2 \cap \overline{S} = \{4, 6\}$, $\pi(4) = \min(8, 5 + 4) = 8$, $\pi(6) = \min(8, 5 + 1) = 6$

Exemple d'application : cas d'un graphe orienté



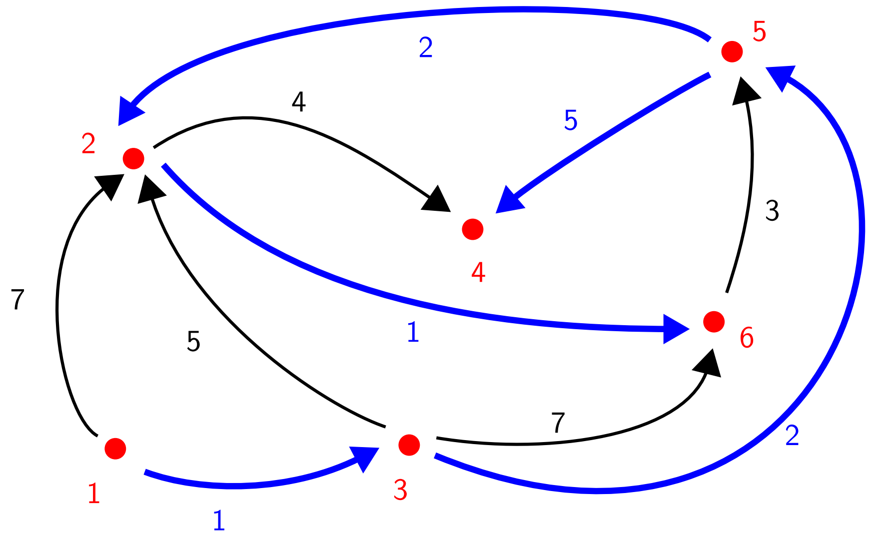
Déroulement de l'algorithme

1. (initialisation) $S = \{1\}$, $\pi = (0, 7, 1, +\infty, +\infty, +\infty)$
2. $j = 3$, $S = \{1, 3\}$, $\Gamma_3 \cap \overline{S} = \{2, 5, 6\}$, $\pi(2) = \min(7, 1 + 5) = 6$, $\pi(5) = \min(+\infty, 1 + 2) = 3$, $\pi(6) = \min(+\infty, 1 + 7) = 8$
3. $j = 5$, $S = \{1, 3, 5\}$, $\Gamma_5 \cap \overline{S} = \{2, 4\}$, $\pi(2) = \min(6, 3 + 2) = 5$, $\pi(4) = \min(+\infty, 3 + 5) = 8$
4. $j = 2$, $S = \{1, 2, 3, 5\}$, $\Gamma_2 \cap \overline{S} = \{4, 6\}$, $\pi(4) = \min(8, 5 + 4) = 8$, $\pi(6) = \min(8, 5 + 1) = 6$
5. $j = 6$, $S = \{1, 2, 3, 5, 6\}$, $\Gamma_6 \cap \overline{S} = \emptyset$,

Déroulement de l'algorithme

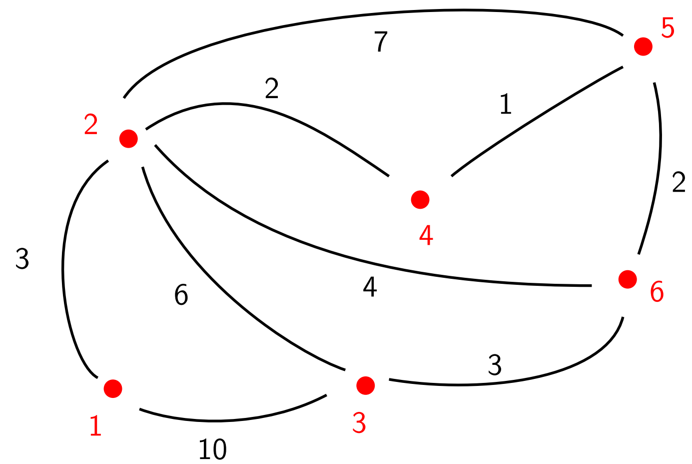
1. (initialisation) $S = \{1\}$, $\pi = (0, 7, 1, +\infty, +\infty, +\infty)$
2. $j = 3$, $S = \{1, 3\}$, $\Gamma_3 \cap \overline{S} = \{2, 5, 6\}$, $\pi(2) = \min(7, 1 + 5) = 6$, $\pi(5) = \min(+\infty, 1 + 2) = 3$, $\pi(6) = \min(+\infty, 1 + 7) = 8$
3. $j = 5$, $S = \{1, 3, 5\}$, $\Gamma_5 \cap \overline{S} = \{2, 4\}$, $\pi(2) = \min(6, 3 + 2) = 5$, $\pi(4) = \min(+\infty, 3 + 5) = 8$
4. $j = 2$, $S = \{1, 2, 3, 5\}$, $\Gamma_2 \cap \overline{S} = \{4, 6\}$, $\pi(4) = \min(8, 5 + 4) = 8$, $\pi(6) = \min(8, 5 + 1) = 6$
5. $j = 6$, $S = \{1, 2, 3, 5, 6\}$, $\Gamma_6 \cap \overline{S} = \emptyset$,
6. $j = 4$, $S = \{1, 2, 3, 4, 5, 6\}$, avec $\pi(4) = 8$.

Exemple d'application : cas d'un graphe orienté



Au final, nous obtenons : $\pi^* = (0, 5, 1, 8, 3, 6)$.

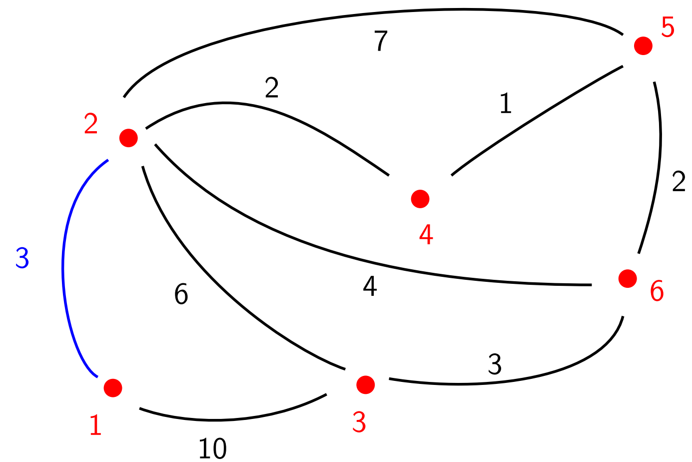
Exemple d'application : cas d'un graphe non orienté



Déroulement de l'algorithme

1. (initialisation) $S = \{1\}$, $\pi = (0, 3, 10, +\infty, +\infty, +\infty)$

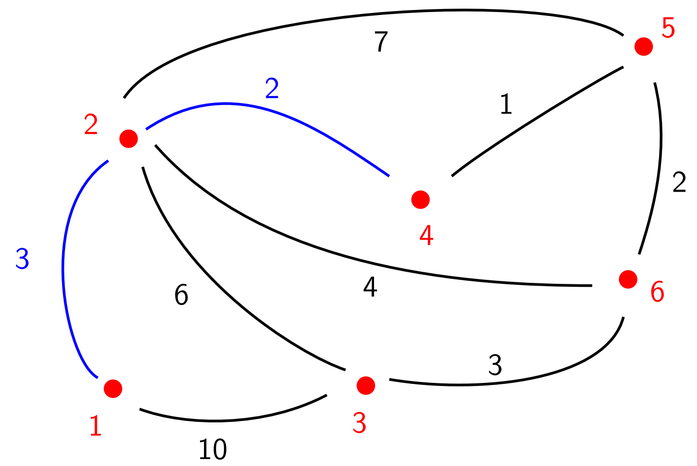
Exemple d'application : cas d'un graphe non orienté



Déroulement de l'algorithme

1. (initialisation) $S = \{1\}$, $\pi = (0, 3, 10, +\infty, +\infty, +\infty)$
2. $j = 2$, $S = \{1, 2\}$, $\Gamma_2 \cap \overline{S} = \{3, 4, 5, 6\}$, $\pi(3) = \min(10, 3 + 6) = 9$, $\pi(4) = \min(+\infty, 3 + 2) = 5$, $\pi(5) = \min(+\infty, 3 + 7) = 10$, $\pi(6) = \min(+\infty, 3 + 4) = 7$

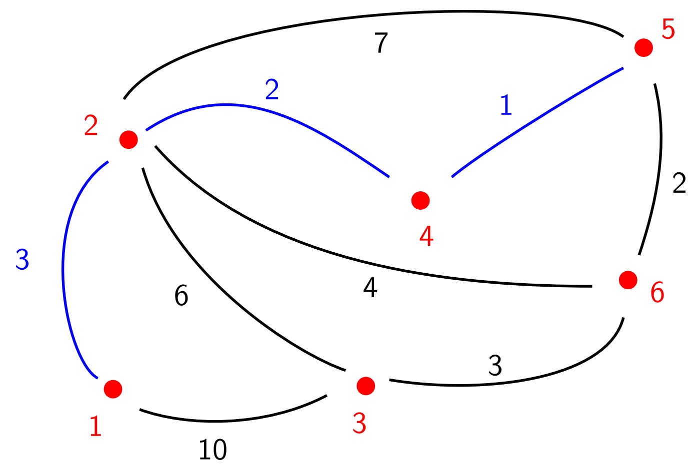
Exemple d'application : cas d'un graphe non orienté



Déroulement de l'algorithme

1. (initialisation) $S = \{1\}$, $\pi = (0, 3, 10, +\infty, +\infty, +\infty)$
2. $j = 2$, $S = \{1, 2\}$, $\Gamma_2 \cap \overline{S} = \{3, 4, 5, 6\}$, $\pi(3) = \min(10, 3 + 6) = 9$, $\pi(4) = \min(+\infty, 3 + 2) = 5$, $\pi(5) = \min(+\infty, 3 + 7) = 10$, $\pi(6) = \min(+\infty, 3 + 4) = 7$
3. $j = 4$, $S = \{1, 2, 4\}$, $\Gamma_4 \cap \overline{S} = \{5\}$, $\pi(5) = \min(10, 5 + 1) = 6$

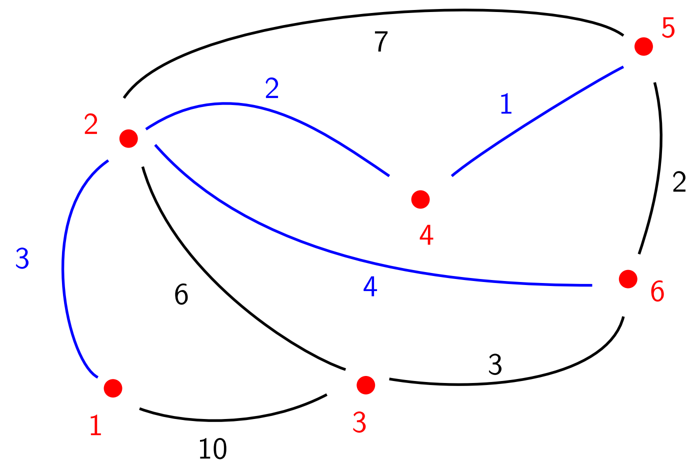
Exemple d'application : cas d'un graphe non orienté



Déroulement de l'algorithme

1. (initialisation) $S = \{1\}$, $\pi = (0, 3, 10, +\infty, +\infty, +\infty)$
2. $j = 2$, $S = \{1, 2\}$, $\Gamma_2 \cap \overline{S} = \{3, 4, 5, 6\}$, $\pi(3) = \min(10, 3 + 6) = 9$, $\pi(4) = \min(+\infty, 3 + 2) = 5$, $\pi(5) = \min(+\infty, 3 + 7) = 10$, $\pi(6) = \min(+\infty, 3 + 4) = 7$
3. $j = 4$, $S = \{1, 2, 4\}$, $\Gamma_4 \cap \overline{S} = \{5\}$, $\pi(5) = \min(10, 5 + 1) = 6$
4. $j = 5$, $S = \{1, 2, 4, 5\}$, $\Gamma_5 \cap \overline{S} = \{6\}$, $\pi(6) = \min(7, 6 + 2) = 7$

Exemple d'application : cas d'un graphe non orienté



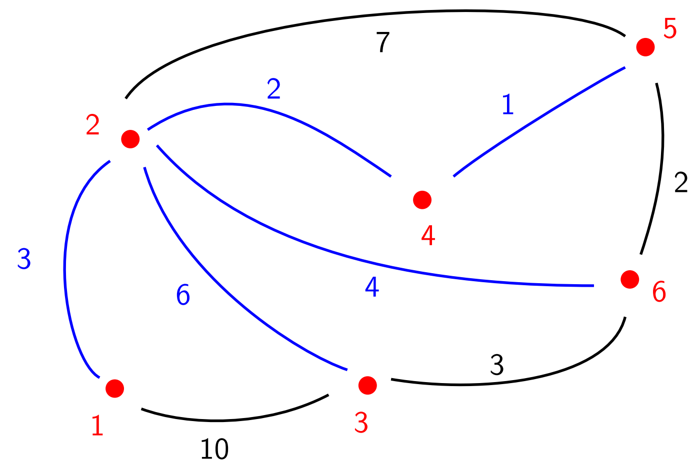
Déroulement de l'algorithme

1. (initialisation) $S = \{1\}$, $\pi = (0, 3, 10, +\infty, +\infty, +\infty)$
2. $j = 2$, $S = \{1, 2\}$, $\Gamma_2 \cap \overline{S} = \{3, 4, 5, 6\}$, $\pi(3) = \min(10, 3 + 6) = 9$, $\pi(4) = \min(+\infty, 3 + 2) = 5$, $\pi(5) = \min(+\infty, 3 + 7) = 10$, $\pi(6) = \min(+\infty, 3 + 4) = 7$
3. $j = 4$, $S = \{1, 2, 4\}$, $\Gamma_4 \cap \overline{S} = \{5\}$, $\pi(5) = \min(10, 5 + 1) = 6$
4. $j = 5$, $S = \{1, 2, 4, 5\}$, $\Gamma_5 \cap \overline{S} = \{6\}$, $\pi(6) = \min(7, 6 + 2) = 7$
5. $j = 6$, $S = \{1, 2, 4, 5, 6\}$, $\Gamma_6 \cap \overline{S} = \{3\}$, $\pi(3) = \min(9, 7 + 3) = 9$

Déroulement de l'algorithme

1. (initialisation) $S = \{1\}$, $\pi = (0, 3, 10, +\infty, +\infty, +\infty)$
2. $j = 2$, $S = \{1, 2\}$, $\Gamma_2 \cap \overline{S} = \{3, 4, 5, 6\}$, $\pi(3) = \min(10, 3 + 6) = 9$, $\pi(4) = \min(+\infty, 3 + 2) = 5$, $\pi(5) = \min(+\infty, 3 + 7) = 10$, $\pi(6) = \min(+\infty, 3 + 4) = 7$
3. $j = 4$, $S = \{1, 2, 4\}$, $\Gamma_4 \cap \overline{S} = \{5\}$, $\pi(5) = \min(10, 5 + 1) = 6$
4. $j = 5$, $S = \{1, 2, 4, 5\}$, $\Gamma_5 \cap \overline{S} = \{6\}$, $\pi(6) = \min(7, 6 + 2) = 7$
5. $j = 6$, $S = \{1, 2, 4, 5, 6\}$, $\Gamma_6 \cap \overline{S} = \{3\}$, $\pi(3) = \min(9, 7 + 3) = 9$
6. $j = 3$, $S = \{1, 2, 3, 4, 5, 6\}$.

Exemple d'application : cas d'un graphe orienté



Au final, nous obtenons : $\pi^* = (0, 3, 9, 5, 6, 7)$.

Propriétés des plus courts chemins

Propriété. (Principe d'optimalité)

Tout sous-chemin d'un plus court chemin est un court chemin.

Démonstration.

A faire. □

Bien fondé de l'algorithme de Moore-Dijkstra

Propriété.

L'algorithme calcule successivement les sommets les plus proches de s : le sommet ajouté à l'ensemble S à l'itération $k = 1, \dots, N-1$ de l'étape b), est le $k$ème sommet le plus proche de s.

Démonstration.

Notons S_k l'ensemble S à l'itération k et $\overline{S}_k = X \setminus S_k$.

- Propriété vraie pour $k = 1$ (S_1 contient le plus proche voisin de s)
- Supposons que la propriété est vraie pour $k > 1$ (S_k contient les k plus proches voisins de s) et montrons qu'elle est vraie pour $k+1$:
  - Observation : à l'issue de l'itération k, $\forall i \in \overline{S}_k, \pi_k(i) = \min_{j \in S_k, j \in \Gamma^{-1}(i)}(\pi^*(j) + l_{ji})$. $\pi_k(i)$ est le plus court chemin entre s et i **parmi** ceux passant par des sommets de S_k.
  - A l'itération $k+1$, on sélectionne i^* tel que $\pi_k(i^*) = \min_{i \in \overline{S}_k} \pi_k(i)$. Montrons que celui-ci est le $k+1$ ème plus proche voisin de s en montrant que pour tout autre chemin $\mu(s, i^*)$, $l(\mu(s, i^*)) \geq \pi_k(i^*)$.
  - Posons $\mu(s, i^*) = \{\mu(s, h), \mu(h, i^*)\}$ où h est le **premier** sommet rencontré dans $\mu(s, i^*)$ qui appartienne à $\overline{S}_k$.
  - On a $l(\mu(s, h)) \geq \pi_k(h)$ (par définition), puis $l(\mu(s, h)) + l(\mu(h, i^*)) \geq \pi_k(h)$ car $l(\mu(h, i^*)) \geq 0$ et donc $l(\mu(s, i^*)) \geq \pi_k(h)$. Or $\pi_k(h) \geq \pi_k(i^*)$ (par définition). On en déduit que $l(\mu(s, i^*)) \geq \pi_k(i^*)$.

□

Complexité de l'algorithme de Moore-Dijkstra

Soit N le nombre de sommets et M le nombre d'arcs. A chaque itération de l'étape b), il y a deux opérations : une opération de sélection et une opération de mise à jour.

- Le nombre d'opérations de sélection à l'itération $k = 1, \dots, N-1$ est $N - k$ d'où au total $\frac{N(N-1)}{2}$ opérations de sélection ($O(N^2)$).
- Le nombre d'opérations de mise à jour à l'itération k est le nombre de successeurs du sommet sélectionné d^+_{i*}. Donc au total, nous avons au plus $\sum_{i \in X} d^+_i = M$ opérations de mise à jour ($O(M)$).

Propriété.

Le temps requis par l'algorithme de Moore-Dijkstra est en $O(N^2)$.

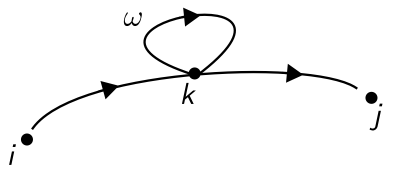
Remarque : si le graphe est peu dense (M petit), on peut utiliser des structures de données particulières pour améliorer la complexité de $O(N^2)$ à $O(M \log N)$.

Cas des graphes valués quelconques

Soit $G = [X, U]$ un graphe valué tel que $\forall u \in U : l(u) \in \mathbb{R}$.

Conditions d'existence de plus courts chemins :

Soit $\mu(i,j)$ un chemin de i à j comprenant un circuit ω comme illustré ci-dessous :



Soit $\mu'(i,j)$ un chemin de i à j ne comprenant pas le circuit ω. On a : $l(\mu) = l(\mu') + l(\omega)$. Ainsi :

- Si $l(\omega) < 0$, il n'existe pas de plus court chemin de i à j (ω est un **circuit absorbant**)
- Si $l(\omega) \geq 0$ alors $l(\mu') \leq l(\mu)$ et dans la recherche d'un plus court chemin on peut se restreindre aux **chemins élémentaires**.

Dans la suite on supposera qu'il n'y a pas de circuit de longueur négative.

Algorithme de Ford-Bellman

Détermination du plus court chemin du sommet s aux autres sommets dans un graphe valué dont les longueurs sont quelconques càd $\forall u \in U : l(u) \in \mathbb{R}$ et pour lequel il n'y a pas de circuit de longueur négative.

Comme préédemment, $\pi^*(i)$ est la longueur du plus court chemin de s à i et nous avons $\pi^*(s) = 0$.

L'algorithme utilise la représentation du graphe par Γ^{-1}. L'algorithme affine successivement une borne supérieure de la longueur du plus court chemine entre s et tous les autres sommets jusqu'à atteindre la longueur minimale.

Pseudo code de l'algorithme de Ford-Bellman

a) $\pi(s) \leftarrow 0$;
Pour tout $(i \in \{1, 2, \dots, N\} \setminus \{s\})$ faire :
  $\pi(i) \leftarrow +\infty$;
FPour

Pseudo code de l'algorithme de Ford-Bellman

a) $\pi(s) \leftarrow 0$;
Pour tout $(i \in \{1, 2, \dots, N\} \setminus \{s\})$ faire :
  $\pi(i) \leftarrow +\infty$;
FPour

b) **Répéter** :
  Pour tout $(i \in \{1, 2, \dots, N\} \setminus \{s\})$ faire :
    $\pi(i) \leftarrow \min(\pi(i), \min_{j \in \Gamma^{-1}(i)} \pi(j) + l_{ji})$;
  FPour
Tant que (une des valeurs $\pi(i)$ change dans la boucle Pour)

Complexité de l'algorithme de Ford-Bellman

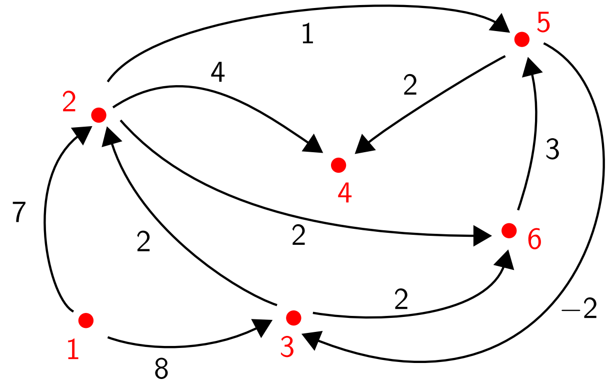
Soit N le nombre de sommets et M le nombre d'arcs. L'étape b) demande chaque fois M opérations d'additions et de comparaisons. Par ailleurs, il y a au plus N itérations de l'étape b) (car les chemins élémentaires sont de longueur inférieure ou égale à N)

Propriété.

Le temps requis par l'algorithme de Ford-Bellman est en $O(NM)$.

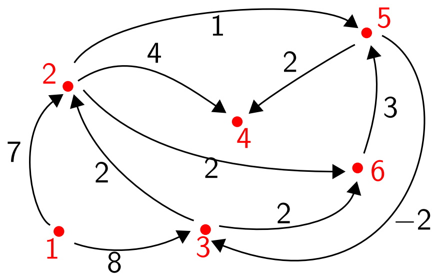
Remarque : L'ordre dans lequel les sommets sont parcourus à l'étape b) est important puisqu'il influence le nombre de fois que cette étape est itérée (avant convergence).

Exemple d'application : cas d'un graphe valué de longueur quelconque



Calcul du plus court chemin entre 1 et les autres sommets.

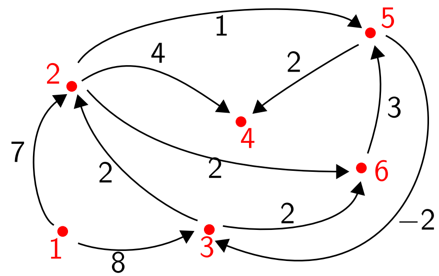
Déroulement de l'algorithme



Avec l'ordre suivant : $2, 3, 4, 5, 6$.

- Initialisation a) $\pi = (0, +\infty, +\infty, +\infty, +\infty, +\infty)$
- Itération 1 de b) $\begin{cases} \pi(2) = \min(\pi(1) + l_{12}, \pi(3) + l_{32}) = 7 \\ \pi(3) = \min(\pi(1) + l_{13}, \pi(5) + l_{53}) = 8 \\ \pi(4) = \min(\pi(2) + l_{24}, \pi(5) + l_{54}) = +\infty \\ \pi(5) = \min(\pi(2) + l_{25}, \pi(6) + l_{65}) = +\infty \\ \pi(6) = \min(\pi(2) + l_{26}, \pi(3) + l_{36}) = +\infty \end{cases}$

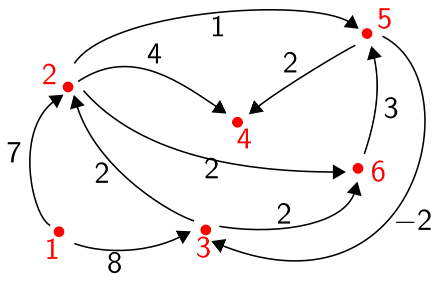
Déroulement de l'algorithme



Avec l'ordre suivant : $2, 3, 4, 5, 6$.

- Valeur courante $\pi = (0, 7, 8, +\infty, +\infty, +\infty)$
- Itération 2 de b) $\begin{cases} \pi(2) = \min(\pi(1) + l_{12}, \pi(3) + l_{32}) = 7 \\ \pi(3) = \min(\pi(1) + l_{13}, \pi(5) + l_{53}) = 8 \\ \pi(4) = \min(\pi(2) + l_{24}, \pi(5) + l_{54}) = 11 \\ \pi(5) = \min(\pi(2) + l_{25}, \pi(6) + l_{65}) = 8 \\ \pi(6) = \min(\pi(2) + l_{26}, \pi(3) + l_{36}) = 9 \end{cases}$

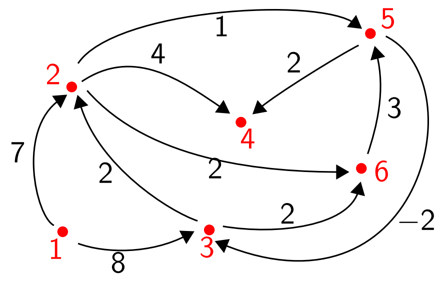
Déroulement de l'algorithme



Avec l'ordre suivant : $2, 3, 4, 5, 6$.

- Valeur courante $\pi = (0, 7, 8, 11, 8, 9)$
- Itération 3 de b) $\begin{cases} \pi(2) = \min(\pi(1) + l_{12}, \pi(3) + l_{32}) = 7 \\ \pi(3) = \min(\pi(1) + l_{13}, \pi(5) + l_{53}) = 8 \\ \pi(4) = \min(\pi(2) + l_{24}, \pi(5) + l_{54}) = 10 \\ \pi(5) = \min(\pi(2) + l_{25}, \pi(6) + l_{65}) = 8 \\ \pi(6) = \min(\pi(2) + l_{26}, \pi(3) + l_{36}) = 9 \end{cases}$

Déroulement de l'algorithme



Avec l'ordre suivant : $2, 3, 4, 5, 6$.

- Valeur courante $\pi = (0, 7, 8, 10, 8, 9)$
- Itération 4 de b) $\begin{cases} \pi(2) = \min(\pi(1) + l_{12}, \pi(3) + l_{32}) = 7 \\ \pi(3) = \min(\pi(1) + l_{13}, \pi(5) + l_{53}) = 8 \\ \pi(4) = \min(\pi(2) + l_{24}, \pi(5) + l_{54}) = 10 \\ \pi(5) = \min(\pi(2) + l_{25}, \pi(6) + l_{65}) = 8 \\ \pi(6) = \min(\pi(2) + l_{26}, \pi(3) + l_{36}) = 9 \end{cases}$

Identification d'un chemin de longueur minimale

Etant donné un graphe valué $G = [X, U]$ et le vecteur, π^*, des longueurs des plus courts chemins entre s et les autres sommets. Comment déterminer les chemins correspondants à ces longueurs minimales ?

Pseudo code de l'identification des chemins de longueurs minimales

Identification d'un chemin de longueur minimale entre s et i étant donné π^* et les longueurs des arcs du graphe valué G.
On procède de l'extrémité finale i et on remonte progressivement vers s.

a) $k \leftarrow i$;
$\mu \leftarrow \{\}$;

b) **Tant que** $k \neq i$ faire :
    Rechercher j tel que $\pi^*(j) = \pi^*(k) - l_{jk}$;
    $\mu \leftarrow (j, k) \cup \mu$;
    $k \leftarrow j$;
FTq

Rappel du Sommaire

Motivations

Pourquoi l'optimisation et la programmation linéaire ?

- Beaucoup de problèmes théoriques et pratiques (et dans toute discipline scientifique) se modélisent sous forme de problèmes d'optimisation.
- La programmation linéaire concerne la modélisation et la résolution d'un certain type de problème d'optimisation. En particulier il s'agit de problèmes "faciles" en optimisation. Pour autant ce sont des problèmes très largement rencontrés en pratique.
- La programmation linéaire touche beaucoup de domaines : énergie (pétrole, gaz, ...), transports (aériens, routiers, ...), planification, management, économie, finance, ...

Rappel du Sommaire

Problèmes d'optimisation

Dans un problème d'optimisation, on cherche à maximiser ou à minimiser une quantité donnée par une **fonction objectif**. Celle-ci dépend d'un nombre fini de variables. Ces variables peuvent être indépendantes ou être reliées entre elles par des **contraintes**. Formellement, nous avons de manière générale le **programme mathématique** suivant :

$$\begin{array}{lccc} \text{Optimiser :} & f(x_1, x_2, \ldots, x_n) & & \\ \text{Sous les contraintes :} & \left.\begin{array}{c} g_1(x_1, x_2, \ldots, x_n) \\ g_2(x_1, x_2, \ldots, x_n) \\ \ldots \\ \ldots \\ g_m(x_1, x_2, \ldots, x_n) \end{array}\right\} & \begin{array}{c} \leq \\ = \\ \geq \end{array} & \left\{\begin{array}{c} b_1 \\ b_2 \\ \ldots \\ \ldots \\ b_m \end{array}\right. \end{array} \quad (1)$$

$f, g_1, \ldots, g_m$ sont des fonctions mathématiques et $b_1, \ldots, b_m$ sont des constantes réelles. Chacune des m contraintes est associée à un des signes parmi $\leq, =, \geq$.

Différents types de problèmes

- Problèmes linéaires : problème (1) où la fonction objectif est linéaire en x ainsi que les contraintes. Exemple :

$$f(x_1,\ldots,x_n)=\sum_{j=1}^{n} c_j x_j \text{ et } \forall i=1,\ldots,m : g_i(x_1,\ldots,x_n)=\sum_{j=1}^{n} a_{ij}x_j \quad (2)$$

où les c_i et les a_{ij} sont des réels.
- Problèmes en nombres entiers : problème (1) où en plus des contraintes, on impose à l'ensemble des variables à prendre des valeurs entières.
- Problèmes quadratiques : problème (1) où la fonction objectif est quadratique en x mais les contraintes sont linéaires en x. Exemple :

$$f(x_1,\ldots,x_n)=\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_i x_j$$

Modélisation d'un problème

Traduire un problème d'optimisation qui s'exprime initialement de manière littérale par un programme mathématique qu'on arrivera à résoudre par des algorithmes :

1. Déterminer la quantité à optimiser et l'exprimer comme une fonction mathématique et donc identifier les variables dont dépend la fonction objectif.
2. Identifier toutes les conditions requises exprimées, les restrictions et limitations de chaque variable et les exprimer par des fonctions mathématiques : définir les contraintes du modèle.
3. Exprimer les conditions cachées qui ne sont pas exprimées de manière explicite. En générale ces conditions concernent la nature des variables à déterminer : celles-ci peuvent être non-négatives, entières par exemple.

L'objet de ce cours

Dans ce cours, nous aborderons plus particulièrement :

- Les problèmes ou programmes linéaires (PL)
- La méthode du Simplexe (algorithme de résolution exacte de PL)
- Les concepts de dualité en PL
- Les problèmes ou programmes linéaires en nombres entiers

Exemple d'un programme linéaire

Une entreprise fabrique deux biens (des pièces mécaniques par exemple). Ces fabrications nécessitent l'utilisation de deux ateliers dont les capacités de production exprimées en heures d'usinage sont de 12. Supposons que :

- Chaque unité du 1er produit nécessite 2h d'usinage dans l'atelier 1 et 1h dans l'atelier 2.
- Chaque unité du 2ème produit nécessite 1h d'usinage dans l'atelier 1 et 2h dans l'atelier 2.

Sachant que la marge sur le 1er produit est $p_1 = 4$ et que celle sur le 2ème produit est de $p_2 = 3$, déterminer un programme mathématique qui modélise le problème de l'optimisation de la marge de l'entreprise sous les contraintes de production décrites précédemment.

Exemple d'un programme linéaire

Dénotons x_1 et x_2 les quantités produites des deux biens.

- Déterminer la fonction objectif :

Exemple d'un programme linéaire

Dénotons x_1 et x_2 les quantités produites des deux biens.

- Déterminer la fonction objectif :

$$\text{Maximiser :} \quad f(x_1, x_2) = 4x_1 + 3x_2$$

- Ecrire les contraintes de production :

Exemple d'un programme linéaire

Dénotons x_1 et x_2 les quantités produites des deux biens.

- Déterminer la fonction objectif :

$$\text{Maximiser :} \quad f(x_1, x_2) = 4x_1 + 3x_2$$

- Ecrire les contraintes de production :

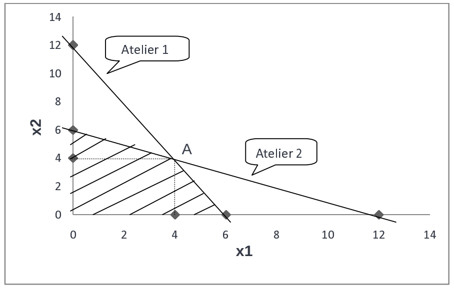
$$\text{Sous les contraintes :} \quad \begin{cases} 2x_1 + x_2 \leq 12 \\ x_1 + 2x_2 \leq 12 \\ x_1 \geq 0\ ,\ x_2 \geq 0. \end{cases}$$

Résolution graphique (domaine des solutions)

- Représenter le domaine de solutions réalisables formé sur le plan de production $\mathbf{X} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$:

Résolution graphique (domaine des solutions)

- Représenter le domaine de solutions réalisables formé sur le plan de production $\mathbf{X} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$:

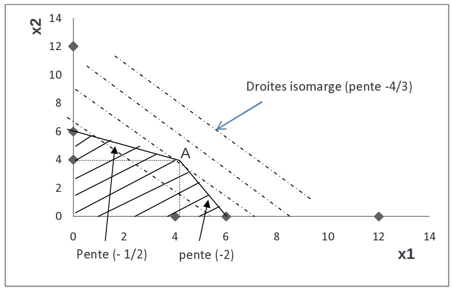


Résolution graphique (droites isomarges)

- Droites isomarges : droites des plans de production conduisant à la même marge. Elles sont données par : $4x_1 + 3x_2 = z$ avec $z \geq 0$

Résolution graphique (droites isomarges)

- Droites isomarges : droites des plans de production conduisant à la même marge. Elles sont données par : $4x_1 + 3x_2 = z$ avec $z \geq 0$
- La solution optimale est donnée par $\mathbf{X}^* = \begin{pmatrix} 4 \\ 4 \end{pmatrix}$ et correspond à une marge de $z^* = 28$.



Rappel du Sommaire

Conditions de non-négativité des contraintes

Les contraintes linéaires sont de la forme :

$$\sum_{j=1}^{n} a_{ij}x_j \sim b_i$$

où $\sim$ représente l'un des signes suivants $\leq, =, \geq$. Nous supposerons dans la suite que les constantes b_i sont **non-négatives**, $\forall i = 1, \ldots, m$.

Pour se ramener à ce cas, il suffit de multiplier la contrainte par -1 si la constante b correspondante est négative.

Exemple : $2x_1 - 3x_2 \leq -5 \Leftrightarrow -2x_1 + 3x_2 \geq 5$.

Variables d'écart et variables de surplus

Une contrainte linéaire de la forme $\sum a_{ij}x_j \leq b_i$ peut être convertie en une contrainte d'égalité en **ajoutant** une nouvelle variable non-négative au membre de gauche de l'inégalité. Une telle variable est appelée **variable d'écart** et est égale à la différence entre le membre de droite et celui de gauche.

Exemple : $4x_1 + 3x_3 + 5x_4 \leq 300 \Leftrightarrow 4x_1 + 3x_3 + 5x_4 + x_5 = 300$.

De manière symétrique, une contrainte linéaire de la forme $\sum a_{ij}x_j \geq b_i$ peut être convertie en une contrainte d'égalité en **retranchant** une nouvelle variable non-négative au membre de gauche de l'inégalité. Une telle variable est appelée **variable de surplus** et est égale à la différence entre le membre de gauche et celui de droite.

Exemple : $4x_1 + 6x_2 + x_3 \geq 54 \Leftrightarrow 4x_1 + 6x_2 + x_3 - x_4 = 54$.

Variables artificielles - Générer une solution (de base réalisable) initiale

Après avoir transformer chaque contrainte linéaire en contrainte d'égalité en introduisant soit des variables d'écart soit des variables de surplus, il s'agit maintenant d'**ajouter** une nouvelle variable, dite **variable artificielle**, aux membres de gauche de chaque contrainte qui **ne contient pas de variable d'écart.**

De cette manière, une **solution initiale non-négative** à ce nouvel ensemble de contraintes est obtenue en attribuant à chaque variable d'écart et à chaque variable artificielle, la valeur du membre de droite correspondant et en attribuant la valeur nulle à toutes les autres variables incluant les variables de surplus.

Générer une solution (de base réalisable) initiale (exemple)

$$\left\{\begin{array}{llcl} x_1 & +2x_2 & \leq & 3 \\ 4x_1 & +5x_2 & \geq & 6 \\ 7x_1 & +8x_2 & = & 15 \end{array}\right.$$

Générer une solution (de base réalisable) initiale (exemple)

$$\begin{cases} x_1 & +2x_2 & \leq & 3 \\ 4x_1 & +5x_2 & \geq & 6 \\ 7x_1 & +8x_2 & = & 15 \end{cases}$$

$$\begin{cases} x_1 & +2x_2 & +x_3 & & = & 3 \\ 4x_1 & +5x_2 & & -x_4 & = & 6 \\ 7x_1 & +8x_2 & & & = & 15 \end{cases}$$

Générer une solution (de base réalisable) initiale (exemple)

$$\begin{cases} x_1 & +2x_2 & \leq & 3 \\ 4x_1 & +5x_2 & \geq & 6 \\ 7x_1 & +8x_2 & = & 15 \end{cases}$$

$$\begin{cases} x_1 & +2x_2 & +x_3 & & = & 3 \\ 4x_1 & +5x_2 & & -x_4 & = & 6 \\ 7x_1 & +8x_2 & & & = & 15 \end{cases}$$

$$\begin{cases} x_1 & +2x_2 & +x_3 & & & & = & 3 \\ 4x_1 & +5x_2 & & -x_4 & +x_5 & & = & 6 \\ 7x_1 & +8x_2 & & & & +x_6 & = & 15 \end{cases}$$

Générer une solution (de base réalisable) initiale (exemple)

$$\begin{cases} x_1 & +2x_2 & \leq & 3 \\ 4x_1 & +5x_2 & \geq & 6 \\ 7x_1 & +8x_2 & = & 15 \end{cases}$$

$$\begin{cases} x_1 & +2x_2 & +x_3 & & = & 3 \\ 4x_1 & +5x_2 & & -x_4 & = & 6 \\ 7x_1 & +8x_2 & & & = & 15 \end{cases}$$

$$\begin{cases} x_1 & +2x_2 & +x_3 & & & & = & 3 \\ 4x_1 & +5x_2 & & -x_4 & +x_5 & & = & 6 \\ 7x_1 & +8x_2 & & & & +x_6 & = & 15 \end{cases}$$

Une solution non-négative (de base réalisable) est alors
$x_3 = 3, x_5 = 6, x_6 = 15$ et $x_1 = x_2 = x_4 = 0$.

Coûts de pénalité dans la fonction objectif liés aux variables artificielles

Les variables d'écart ou de surplus ne changent pas la nature des contraintes ni de la fonction objectif. Par contre, ce n'est pas le cas des variables artificielles. Le nouveau système de contraintes est équivalent à l'initial uniquement si les variables artificielles sont nulles.

Ainsi pour garantir cette solution, les variables artificielles sont ajoutées dans la fonction objectif soit avec un coefficient très grand s'il s'agit d'un problème de minimisation ; soit d'un coefficient très petit s'il s'agit au contraire d'un problème de maximisation. Ces coefficients seront respectivement dénotés M et $-M$ en supposant que $M > 0$.

Forme standard d'un PL

Définition.

Un programme linéaire est mis sous sa forme standard si ses contraintes sont toutes modélisées en tant qu'égalités et si une solution initiale est connue. En notation matricielle, la forme standard s'écrit :

$$\begin{aligned} \textit{Optimiser :} \quad & z(\mathbf{X}) = \mathbf{C}'\mathbf{X} \\ \textit{Sous les contraintes :} \quad & \mathbf{AX} = \mathbf{B} \\ \textit{Avec :} \quad & \mathbf{X} \geq \mathbf{0} \end{aligned}$$

où $\mathbf{X}$ *est le vecteur des inconnues incluant les variables d'écart, de surplus et artificielles ;* $\mathbf{C}'$ *est le vecteur ligne correspondant à la la transposée de* $\mathbf{C}$ *qui est le vecteur des coûts ;* $\mathbf{A}$ *est la matrice des coefficients des contraintes et* $\mathbf{B}$ *est le vecteur colonne des constantes.*

Si $\mathbf{X}_0$ désigne le vecteur des variables d'écart et artificielles, alors une solution initiale est donnée par $\mathbf{X}_0 = \mathbf{B}$ et toute autre cellule de $\mathbf{X}$ n'étant pas incluse dans $\mathbf{X}_0$ étant nulle.

Rappels en algèbre linéaire : Dépendance linéaire entre vecteurs

Définition.

Soit $\{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_n\}$ *un ensemble de n vecteurs de dimension m. Ces vecteurs sont* **linéairement dépendants** *s'il existe des réels non nuls* $\alpha_1, \alpha_2, \ldots, \alpha_n$ *tels que :*

$$\alpha_1 \mathbf{P}_1 + \alpha_2 \mathbf{P}_2 + \ldots + \alpha_n \mathbf{P}_n = \mathbf{0} \tag{3}$$

Les vecteurs seront dits **linéairement indépendants** si l'équation (3) est vérifiée uniquement pour la solution $\alpha_1 = \alpha_2 = \ldots = \alpha_n = 0$.

Rappels en algèbre linéaire : Exemple de vecteurs linéairement dépendants

Soit l'ensemble de vecteurs suivant :
$\{(1,2,0,0,0)', (1,0,0,0,0)', (0,0,1,1,0)', (0,1,0,0,0)'\}$.

Ils sont linéairement dépendants :

$$? \begin{pmatrix}1\\2\\0\\0\\0\end{pmatrix} +? \begin{pmatrix}1\\0\\0\\0\\0\end{pmatrix} +? \begin{pmatrix}0\\0\\1\\1\\0\end{pmatrix} +? \begin{pmatrix}0\\1\\0\\0\\0\end{pmatrix} = \begin{pmatrix}0\\0\\0\\0\\0\end{pmatrix}$$

Rappels en algèbre linéaire : Exemple de vecteurs linéairement dépendants

Soit l'ensemble de vecteurs suivant :
$\{(1,2,0,0,0)', (1,0,0,0,0)', (0,0,1,1,0)', (0,1,0,0,0)'\}$.

Ils sont linéairement dépendants :

$$-1 \begin{pmatrix}1\\2\\0\\0\\0\end{pmatrix} + 1 \begin{pmatrix}1\\0\\0\\0\\0\end{pmatrix} + 0 \begin{pmatrix}0\\0\\1\\1\\0\end{pmatrix} + 2 \begin{pmatrix}0\\1\\0\\0\\0\end{pmatrix} = \begin{pmatrix}0\\0\\0\\0\\0\end{pmatrix}$$

Rappels en algèbre linéaire : Théorème

Théorème.

Un ensemble de $m+1$ (ou plus) vecteurs de dimension m est linéairement dépendant.

Rappels en algèbre linéaire : Combinaison convexe

Définition.

Un vecteur $\mathbf{P}$ *de dimension* m *est une combinaison convexe des* n *vecteurs* $\{\mathbf{P}_1, \ldots, \mathbf{P}_n\}$ *de dimension* m*, s'il existe des* **réels non-négatifs de somme 1**, $\beta_1, \beta_2, \ldots, \beta_n$*, tels que :*

$$\mathbf{P} = \beta_1 \mathbf{P}_1 + \beta_2 \mathbf{P}_2 + \ldots + \beta_n \mathbf{P}_n \quad (4)$$

Définition.

Etant donné deux vecteurs $\mathbf{P}_1$ *et* $\mathbf{P}_2$ *de dimension* m*, on appelle l'ensemble de toutes les combinaisons convexes de* $\mathbf{P}_1$ *et* $\mathbf{P}_2$*, le* **segment** *entre ces deux vecteurs.*

Rappels en algèbre linéaire : Exemple de combinaison convexe de vecteurs

Soit le vecteur $\mathbf{P} = (5/3, 5/6)'$ et soient les vecteurs suivants : $\mathbf{P}_1 = (1,1)', \mathbf{P}_2 = (3,0)', \mathbf{P}_3 = (1,2)'$.

$\mathbf{P}$ est une combinaison convexe de $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$:

$$\begin{pmatrix} 5/3 \\ 5/6 \end{pmatrix} =? \begin{pmatrix} 1 \\ 1 \end{pmatrix} +? \begin{pmatrix} 3 \\ 0 \end{pmatrix} +? \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Rappels en algèbre linéaire : Exemple de combinaison convexe de vecteurs

Soit le vecteur $\mathbf{P} = (5/3, 5/6)'$ et soient les vecteurs suivants : $\mathbf{P}_1 = (1,1)', \mathbf{P}_2 = (3,0)', \mathbf{P}_3 = (1,2)'$.

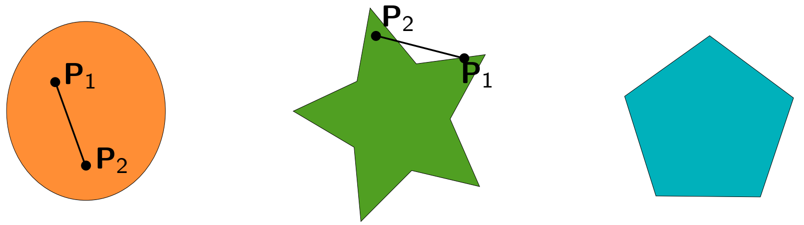
$\mathbf{P}$ est une combinaison convexe de $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$:

$$\begin{pmatrix} 5/3 \\ 5/6 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \frac{1}{3} \begin{pmatrix} 3 \\ 0 \end{pmatrix} + \frac{1}{6} \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Rappels en algèbre linéaire : Ensemble convexe

Définition.
Un ensemble de vecteurs de dimension m forme un **ensemble convexe** *si pour deux vecteurs appartenant à cet ensemble leur segment appartient également à l'ensemble.*

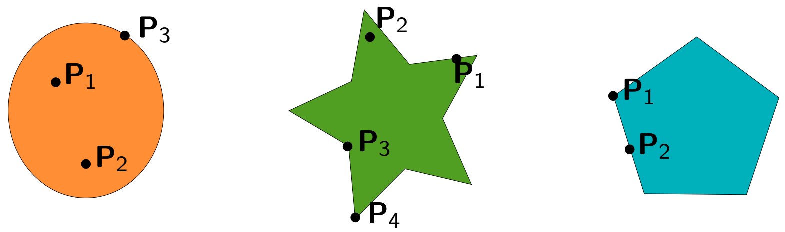
Exemples :



Points extrêmes d'ensembles convexes

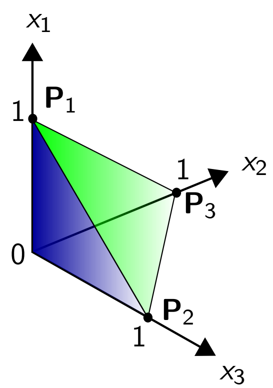
Définition.
Un vecteur **P** *est un* **point extrême** *d'un ensemble convexe s'il ne peut être exprimé comme une combinaison convexe de deux autres vecteurs de l'ensemble. En d'autres termes, un* **point extrême** *ne peut se trouver sur le segment formé par deux autres vecteurs de l'ensemble.*

Exemples :



Points extrêmes d'ensembles convexes

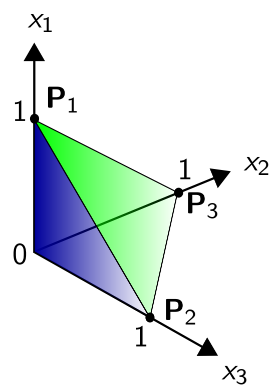
Théorème.
L'espace de solutions d'un ensemble d'équations linéaires est un ensemble convexe comprenant un ensemble fini de points extrêmes.



- Soit, dans $\mathbb{R}^3$, l'ensemble convexe défini par : $x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0,\ x_1 + x_2 + x_3 \leq 1$.
- Sur la figure la face en vert est formée des points vérifiant :
- La face en bleue est formée des points vérifiant :
- L'arête $\mathbf{P}_1\mathbf{P}_2$ est la frontière formée des points vérifiant à la fois :
- Le sommet $\mathbf{P}_3$ est tel que :

Points extrêmes d'ensembles convexes

Théorème.
L'espace de solutions d'un ensemble d'équations linéaires est un ensemble convexe comprenant un ensemble fini de points extrêmes.



- Soit, dans $\mathbb{R}^3$, l'ensemble convexe défini par : $x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0,\ x_1 + x_2 + x_3 \leq 1$.
- Sur la figure la face en vert est formée des points vérifiant : $x_1 + x_2 + x_3 = 1$
- La face en bleue est formée des points vérifiant : $x_2 = 0$
- L'arête $\mathbf{P}_1\mathbf{P}_2$ est la frontière formée des points vérifiant à la fois : $x_1 + x_2 + x_3 = 1$ et $x_2 = 0$
- Le sommet $\mathbf{P}_3$ est tel que : $x_1 + x_2 + x_3 = 1$, $x_1 = 0$ et $x_3 = 0$

Points extrêmes d'ensembles convexes et solutions d'un programme linéaire

Nous avons les résultats suivants :

- Soit $\mathbb{S}$ l'ensemble des solutions vérifiant les contraintes linéaires d'un programme linéaire mis sous forme standard. D'après le théorème précédent et du fait que l'intersection d'ensembles convexes est un ensemble convexe, il en découle que $\mathbb{S}$ **est convexe comprenant un nombre fini de points extrêmes**.
- Si **l'optimum (maximum ou minimum) d'une fonction objectif** existe alors il **est atteint en un point extrême de** $\mathbb{S}$.
- Si $\mathbf{A}$ est de dimension $m \times n$ avec $m \leq n$, alors les points extrêmes de $\mathbb{S}$ comportent au moins $n - m$ composantes nulles.

Solution de base réalisable

Dénotons les colonnes de la matrice des coefficients des contraintes $\mathbf{A}$, de taille $m \times n$, par $\mathbf{A}_1, \ldots, \mathbf{A}_n$. Le système de contraintes d'égalité $\mathbf{AX} = \mathbf{B}$ peut alors s'écrire de manière suivante :

$$x_1\mathbf{A}_1 + x_2\mathbf{A}_2 + \ldots + x_n\mathbf{A}_n = \mathbf{B}$$

Les vecteurs colonnes $\mathbf{A}_j; j = 1, \ldots, n$ et le vecteur $\mathbf{B}$ sont de dimension m et sont les données du problème. L'objectif est de trouver des solutions non-négatives au système précédent. On supposera que $m \leq n$ et que le rang de $\mathbf{A}$ est m ce qui implique qu'il existe au moins une collection de m vecteurs dans $\{\mathbf{A}_j\}_{j=1,\ldots,n}$ qui soient linéairement indépendants.

Solution de base réalisable (suite)

Une **solution de base réalisable** est obtenue en attribuant à $n - m$ variables parmi $\{x_j\}_{j=1,\ldots,n}$ une valeur nulle à condition que les m vecteurs $\mathbf{A}_j$ correspondant aux variables x_j distinctes de 0 soient linéairement indépendants. Ces variables dont les valeurs sont distinctes de 0 sont appelées **variables de base**.
Si une (ou plusieurs) variables de base est nulle alors on dit que la solution est **dégénérée**. Au contraire si toutes les variables de base sont positives alors on dit que la solution est **non dégénérée** (cas traités dans ce cours).

Nous avons les résultats suivants :

- La fonction objectif atteint son **optimum en une solution de base réalisable**
- **Les points extrêmes** de $\mathbb{S}$ **sont** précisément, **les solutions de base réalisables**.

$\rightarrow$ Un PL sous forme standard peut être résolu en cherchant parmi les solutions de base réalisables celle(s) qui optimise(nt) la fonction objectif.

Solution de base réalisable (suite)

- $\mathbf{X} = [\mathbf{X}_b, \mathbf{X}_{hb}]$ où $\mathbf{X}_b$ (resp $\mathbf{X}_{hb}$) est le vecteur de taille $(m \times 1)$ (resp $(n - m \times 1)$) issu de $\mathbf{X}$ dont les composantes sont celles des m variables de bases (resp $n - m$ variables hors base)
- $\mathbf{C} = [\mathbf{C}_b, \mathbf{C}_{hb}]$ où $\mathbf{C}_b$ (resp $\mathbf{C}_{hb}$) est le vecteur de taille $(m \times 1)$ issu de $\mathbf{C}$ (resp $(n - m \times 1)$) dont les coûts sont ceux relatifs aux m variables de bases (resp $n - m$ variables hors base)
- $\mathbf{A} = [\mathbf{A}_b, \mathbf{A}_{hb}]$ où $\mathbf{A}_b$ (resp $\mathbf{A}_{hb}$) est la sous-matrice de taille $(m \times m)$ (resp $(m \times n - m)$) issue de $\mathbf{A}$ dont les composantes sont les colonnes relatives aux m variables de bases (resp $n - m$ variables hors base). $\mathbf{A}_b$ est de rang m.

Nous avons : $\mathbf{AX} = \mathbf{B} \Leftrightarrow \mathbf{A}_b\mathbf{X}_b + \mathbf{A}_{hb}\mathbf{X}_{hb} = \mathbf{B}$.

Expression des variables de base en fonction des variables hors base :

$$\mathbf{X}_b = \mathbf{A}_b^{-1}(\mathbf{B} - \mathbf{A}_{hb}\mathbf{X}_{hb})$$

Condition d'optimalité d'une solution de base réalisable

Expression de la fonction objectif :

$$\begin{aligned} z(\mathbf{X}) = \mathbf{C}'\mathbf{X} &= \mathbf{C}'_b\mathbf{X}_b + \mathbf{C}'_{hb}\mathbf{X}_{hb} \\ &= \mathbf{C}'_b\left(\mathbf{A}_b^{-1}(\mathbf{B} - \mathbf{A}_{hb}\mathbf{X}_{hb})\right) + \mathbf{C}'_{hb}\mathbf{X}_{hb} \\ &= \mathbf{C}'_b\mathbf{A}_b^{-1}\mathbf{B} + \underbrace{\left(\mathbf{C}'_{hb} - \mathbf{C}'_b\mathbf{A}_b^{-1}\mathbf{A}_{hb}\right)}_{\overline{\mathbf{C}}_{hb}}\mathbf{X}_{hb} \end{aligned}$$

Les quantités $\overline{\mathbf{C}}_{hb}$ sont appelées **"coûts" réduits des variables hors base**. On peut toujours trouver une base permettant d'améliorer la fonction objectif tant qu'il existe une composante de $\overline{\mathbf{C}}_{hb}$ qui est positive. Autrement dit une **condition nécessaire et suffisante d'optimalité** d'une solution de base réalisable est :

$$\mathbf{C}'_{hb} - \mathbf{C}'_b\mathbf{A}_b^{-1}\mathbf{A}_{hb} \leq \mathbf{0} \tag{5}$$

Passage d'une solution de base réalisable à une autre

Pour la solution de base réalisable courante $\mathbf{X} = [\mathbf{X}_b, \mathbf{X}_{hb}]$ et $z(\mathbf{X}) = \mathbf{C}'_b\mathbf{A}_b^{-1}\mathbf{B} + \underbrace{\left(\mathbf{C}'_{hb} - \mathbf{C}'_b\mathbf{A}_b^{-1}\mathbf{A}_{hb}\right)}_{\overline{\mathbf{C}}_{hb}}\mathbf{X}_{hb}$.

Le passage d'une solution de base réalisable à une autre le long d'une arête permettant d'améliorer la fonction objectif s'effectue en faisant entrer une nouvelle variable hors base dans la base et en faisant sortir de la base une autre variable :

- On cherche dans $\overline{\mathbf{C}}_{hb}$ la variable hors base $x_s = \mathrm{Argmax}_{i \in \mathbf{X}_{hb}}\{\overline{\mathbf{C}}_{hb}(i)\}$ où $\overline{\mathbf{C}}_{hb}(s)$ est le gain le plus grand que l'on peut avoir pour la fonction objectif.
- On définit la colonne de travail comme étant $\mathbf{A}_s$, le vecteur colonne de $\mathbf{A}_{hb}$ relatif à x_s.
- On calcule $\overline{\mathbf{B}} = \mathbf{A}_b^{-1}\mathbf{B}$ et $x_t = \mathrm{Argmin}_{i \in \mathbf{X}_b : \mathbf{A}_s(i) > 0}\{\frac{\overline{\mathbf{B}}(i)}{\mathbf{A}_s(i)}\}$. x_t est la variable qui sort de la base. $\frac{\overline{\mathbf{B}}(t)}{\mathbf{A}_s(t)}$ est la quantité maximale que l'on peut allouer à x_s sans violer les contraintes.

Passage d'une solution de base réalisable à une autre (suite)

Le changement de base se traduit par l'expression des variables de base en fonction des variables hors base :

- on peut exprimer les variables de base et z en fonction des seules variables hors base
- les colonnes de la matrice des contraintes correspondant aux variables de base forment une matrice unité (à une permutation près)

D'un point de vue algèbrique cela revient à éliminer la variable entrante du sytème $\mathbf{AX} = \mathbf{B}$. Nous pouvons effectuer ceci en appliquant les formules de l'élimination de Gauss-Jordan (méthode du pivot).

Rappel du Sommaire

L'algorithme du simplexe

Il s'agit d'un algorithme permettant de résoudre **efficacement** les PL mis sous leur forme standard :

$$\begin{aligned} \text{Optimiser :} \quad & z(\mathbf{X}) = \mathbf{C}'\mathbf{X} \\ \text{Sous les contraintes :} \quad & \mathbf{AX} = \mathbf{B} \\ \text{Avec :} \quad & \mathbf{X} \geq \mathbf{0} \end{aligned}$$

où $\mathbf{B} \geq \mathbf{O}$ et une solution de base intiale $\mathbf{X}$ est connue.

Conçue initialement par G. Dantzig en 1947, la procédure revient à parcourir les solutions de base réalisables afin d'améliorer la valeur de la fonction objectif jusqu'à obtenir un optimum.

Méthode des tableaux

On suppose un problème de maximisation mis sous la forme standard. La méthode des tableaux est une façon de présenter les données d'un PL et de mettre en oeuvre l'algorithme du simplexe pour résoudre ce dernier. Le "tableau simplexe" est une représentation tabulaire regroupant les données calculées à chaque itération de l'algorithme du simplexe. Il est de la forme suivante où chaque entité est màj à chaque itération :

		$\mathbf{X}'$ $\mathbf{C}'$	
$\mathbf{X}_b$	$\mathbf{C}_b$	$\mathbf{A}$	$\mathbf{B}$
		$\mathbf{C}' - \mathbf{C}_b'\mathbf{A}$	$\mathbf{C}_b'\mathbf{B}$

Pour un problème de minimisation, il faut changer la dernière ligne par son opposée.

Méthode du simplexe

1. Déterminer le **nombre positif le plus grand** de la dernière ligne du tableau (en excluant la dernière colonne). La colonne dans laquelle ce nombre apparaît sera appelé **colonne de travail**.
2. Calculer des ratios en divisant pour chaque **nombre positif** de la colonne de travail (en excluant la dernière ligne), le nombre de la dernière colonne correspondant par ce nombre positif. Déterminer ensuite l'élément de la colonne de travail qui donne **le plus petit ratio**. Cet élément sera désigné par **élément pivot**. Si aucun élément de la colonne de travail est positif alors le PL n'a pas de solution.
3. Modifier les lignes du tableau en utilisant des opérations élémentaires (multiplication par des constantes, addition ou soustraction) entre les lignes de sorte d'une part à ce que **l'élément pivot soit égal à** 1 et d'autre part, à ce que tous **les autres éléments de la colonne de travail valent** 0 (formule de Gauss-Jordan).
4. Remplacer la variable de base x_t (1ère colonne) sur la ligne correspondante à l'élément pivot (**variable sortante**) par la variable hors base x_s correspondant à la colonne de l'élément pivot (**variable entrante**).

Méthode du simplexe (suite)

5. Répéter les étapes 1 à 4 jusqu'à ce qu'il n'y ait plus de termes strictement positifs dans la dernière ligne du tableau (en excluant la dernière colonne).
6. La solution optimale est obtenue en attribuant à chaque variable de la première colonne la valeur correspondante de la dernière colonne du tableau. Toutes les autres variables sont nulles. La valeur optimale de la fonction objectif est le nombre obtenu dans la cellule correspondant à la dernière ligne et dernière colonne du tableau pour un problème de maximisation (l'opposé de ce nombre pour un problème de minimisation).

Bien fondé de l'algorithme du simplexe

Théorème.

Soit $\mathbf{X}$ *une solution de base réalisable quelconque et soit le vecteur des "coûts réduits" des variables hors base :* $(\mathbf{C}'_{hb} - \mathbf{C}'_b \mathbf{A}_b^{-1} \mathbf{A}_{hb})$
S'il existe une variable hors base x_s *telle que son coût réduit soit strictement positif alors :*

- *ou bien on peut augmenter indéfiniment la valeur de* x_s *sans sortir de l'ensemble des solutions réalisables et dans ce cas z est non borné*
- *ou bien on peut déterminer une autre solution de base réalisable* $\hat{\mathbf{X}}$ *tel que* $z(\hat{\mathbf{X}})$ *soit meilleure que* $z(\mathbf{X})$

Théorème.

Sous l'hypothèse de non-dégénérescence, l'algorithme du simplexe converge en un nombre fini d'itérations.

Exemple d'application

Problème de maximisation initial :

$$\begin{array}{rl} \text{Maximiser :} & f(x_1,x_2) = 4x_1 + 3x_2 \\ \text{Sous les contraintes :} & \left\{\begin{array}{ccccc} 2x_1 & + & x_2 & \leq & 12 \\ x_1 & + & 2x_2 & \leq & 12 \end{array}\right. \\ \text{Avec :} & x_1 \geq 0 \ , \ x_2 \geq 0 \end{array}$$

Ajout des variables d'écart :

$$\begin{array}{rl} \text{Maximiser :} & f(x_1,x_2) = 4x_1 + 3x_2 \\ \text{Sous les contraintes :} & \left\{\begin{array}{ccccccccc} 2x_1 & + & x_2 & + & x_3 & & & = & 12 \\ x_1 & + & 2x_2 & & & + & x_4 & = & 12 \end{array}\right. \\ \text{Avec :} & x_1 \geq 0 \ , \ x_2 \geq 0 \ , \ x_3 \geq 0 \ , \ x_4 \geq 0 \end{array}$$

PL mis sous forme standard :

$$\mathbf{A} = \begin{pmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{pmatrix} \quad ; \quad \mathbf{B} = \begin{pmatrix} 12 \\ 12 \end{pmatrix} \quad ; \quad \mathbf{C} = \begin{pmatrix} 4 \\ 3 \end{pmatrix}$$

Exemple d'application (méthode du simplexe)

Tableau simplexe initial :

		x_1	x_2	x_3	x_4	
		4	3	0	0	
x_3	0	2	1	1	0	12
x_4	0	1	2	0	1	12
		4	3	0	0	0

Solution de base réalisable initiale :

- la solution de base réalisable initiale $\mathbf{X} = (0,0,12,12)'$ où les variables de base sont $\{x_3, x_4\}$.
- l'expression des variables de base en fonction des variables hors-base :
$$\left\{\begin{array}{l} 2x_1 + x_2 + x_3 = 12 \\ x_1 + 2x_2 + x_4 = 12 \end{array}\right. \Leftrightarrow \left\{\begin{array}{l} x_3 = 12 - 2x_1 - x_3 \\ x_4 = 12 - x_1 - 2x_2 \end{array}\right.$$
- le vecteur des coûts réduits vaut $\overline{\mathbf{C}}_{hb} = (\underbrace{4}_{x_1}, \underbrace{3}_{x_2})'$
- la fonction objectif vaut $z(\mathbf{X}) = 4x_1 + 3x_2 = 0$

Exemple d'application (méthode du simplexe)

Tableau simplexe initial :

		x_1	x_2	x_3	x_4	
		4	3	0	0	
x_3	0	2	1	1	0	12
x_4	0	1	2	0	1	12
		4	3	0	0	0

Exemple d'application (méthode du simplexe)

Tableau simplexe initial :

		x_1	x_2	x_3	x_4	
		4	3	0	0	
x_3	0	2	1	1	0	12
x_4	0	1	2	0	1	12
		4	3	0	0	0

1ère itération :

Etape 1 (coûts réduits - variable à faire entrer dans la base)
4 est le plus grand nombre positif de la dernière ligne (coûts réduits des variables hors base) et correspond à x_1 (variable hors base qui permettrait d'augmenter le plus z).

$$\begin{cases} x_3 = 12 - 2x_1 \\ x_4 = 12 - x_1 \end{cases}$$

De combien peut-on augmenter x_1 sans pour autant violer les contraintes ?

Exemple d'application (méthode du simplexe)

Tableau simplexe initial :

		x_1	x_2	x_3	x_4	
		4	3	0	0	
x_3	0	2	1	1	0	12
x_4	0	1	2	0	1	12
		4	3	0	0	0

1ère itération :

Etape 2 (ratios - variable à faire sortir de la base)

$$\begin{cases} x_3 = 12 - 2x_1 \geq 0 \\ x_4 = 12 - x_1 \geq 0 \end{cases} \Leftrightarrow \begin{cases} x_1 \leq 12/2 \\ x_1 \leq 12/1 \end{cases}$$

Ces quantités correspondent aux ratios $12/2 = 6$ (1ère ligne) et $12/1 = 12$ (2ème ligne).
Le plus petit ratio correspond à la 1ère ligne : l'élément pivot est 2.

Exemple d'application (méthode du simplexe)

Tableau simplexe initial :

		x_1	x_2	x_3	x_4	
		4	3	0	0	
x_3	0	2	1	1	0	12
x_4	0	1	2	0	1	12
		4	3	0	0	0

1ère itération :

Etapes 3-4 (changement de base)
On fait entrer x_1 dans la base et on fait sortir x_3. L'expression des variables de base en fonction des variables hors-base devient :

$$\begin{cases} 2x_1 + x_2 + x_3 = 12 \\ x_1 + 2x_2 + x_4 = 12 \end{cases} \Leftrightarrow \begin{cases} x_1 = 6 - \frac{1}{2}x_2 - \frac{1}{2}x_3 \\ x_4 = 6 - \frac{3}{2}x_2 + \frac{1}{2}x_3 \end{cases}.$$

Ceci revient à appliquer les transformations lignessuivantes $l_1' = l_1/2$ et $l_2' = l_2 - l_1/2$.

Exemple d'application (méthode du simplexe)

Tableau simplexe après les étapes 3 et 4 de la 1ère itération :

		x_1	x_2	x_3	x_4	
		4	3	0	0	
x_1	4	1	1/2	1/2	0	6
x_4	0	0	3/2	−1/2	1	6
		0	1	−2	0	24

Solution de base réalisable à l'issue de la 1ère itération :

- la solution de base réalisable $\mathbf{X} = (6, 0, 0, 6)'$ où les variables de base sont $\{x_1, x_4\}$.
- l'expression des variables de base en fonction des variables hors-base :
$$\begin{cases} 2x_1 + x_2 + x_3 = 12 \\ x_1 + 2x_2 + x_4 = 12 \end{cases} \Leftrightarrow \begin{cases} x_1 = 6 - \frac{1}{2}x_2 - \frac{1}{2}x_3 \\ x_4 = 6 - \frac{3}{2}x_2 + \frac{1}{2}x_3 \end{cases}$$
- le vecteur des coûts réduits vaut $\overline{\mathbf{C}}_{hb} = (\underbrace{1}_{x_2}, \underbrace{-2}_{x_3})'$
- la fonction objectif vaut $z(\mathbf{X}) = 4x_1 + 3x_2 = 24 + x_2 - 2x_3 = 24$

Exemple d'application (méthode du simplexe)

Tableau simplexe après la 1ère itération :

		x_1	x_2	x_3	x_4	
		4	3	0	0	
x_1	4	1	$1/2$	$1/2$	0	6
x_4	0	0	$3/2$	$-1/2$	1	6
		0	1	-2	0	24

Exemple d'application (méthode du simplexe)

Tableau simplexe après la 1ère itération :

		x_1	x_2	x_3	x_4	
		4	3	0	0	
x_1	4	1	$1/2$	$1/2$	0	6
x_4	0	0	$3/2$	$-1/2$	1	6
		0	1	-2	0	24

2ème itération :

Etape 1 (coûts réduits - variable à faire entrer dans la base)
1 est le plus grand nombre positif de la dernière ligne (coûts réduits des variables hors base) et correspond à x_2 (variable hors base qui permettrait d'augmenter le plus z).

$$\begin{cases} x_1 = 6 - \frac{1}{2}x_2 \\ x_4 = 6 - \frac{3}{2}x_2 \end{cases}$$

De combien peut-on augmenter x_2 sans pour autant violer les contraintes ?

Exemple d'application (méthode du simplexe)

Tableau simplexe après la 1ère itération :

		x_1	x_2	x_3	x_4	
		4	3	0	0	
x_1	4	1	$1/2$	$1/2$	0	6
x_4	0	0	$3/2$	$-1/2$	1	6
		0	1	-2	0	24

2ème itération :

Etape 2 (ratios - variable à faire sortir de la base)

$$\begin{cases} x_1 = 6 - \frac{1}{2}x_2 \geq 0 \\ x_4 = 6 - \frac{3}{2}x_2 \geq 0 \end{cases} \Leftrightarrow \begin{cases} x_2 \leq 12 \\ x_2 \leq 4 \end{cases}$$

Ces quantités correspondent aux ratios $6/(1/2) = 12$ (1ère ligne) et $6/(3/2) = 4$ (2ème ligne).
Le plus petit ratio correspond à la 2ème ligne : l'élément pivot est $3/2$.

Exemple d'application (méthode du simplexe)

Tableau simplexe après la 1ère itération :

		x_1	x_2	x_3	x_4	
		4	3	0	0	
x_1	4	1	$1/2$	$1/2$	0	6
x_4	0	0	$3/2$	$-1/2$	1	6
		0	1	-2	0	24

2ème itération :

Etapes 3-4 (changement de base)
On fait entrer x_2 dans la base et on fait sortir x_4. L'expression des variables de base en fonction des variables hors-base devient :

$$\begin{cases} x_1 = 6 - \frac{1}{2}x_2 - \frac{1}{2}x_3 \\ x_4 = 6 - \frac{3}{2}x_2 + \frac{1}{2}x_3 \end{cases} \Leftrightarrow \begin{cases} x_2 = 4 + \frac{1}{3}x_3 - \frac{2}{3}x_4 \\ x_1 = 4 - \frac{2}{3}x_3 + \frac{1}{3}x_4 \end{cases}.$$

Ceci revient à appliquer les transformations lignes suivantes $l'_2 = l_2/(3/2)$ et $l'_1 = l_1 - l_2/3$.

Exemple d'application (méthode du simplexe)

Tableau simplexe après les étapes 3 et 4 de la 2ème itération :

		x_1	x_2	x_3	x_4	
		4	3	0	0	
x_1	4	1	0	$2/3$	$-1/3$	4
x_2	3	0	1	$-1/3$	$2/3$	4
		0	0	$-5/3$	$-2/3$	28

Solution de base réalisable à l'issue de la 2ème itération :

- la solution de base réalisable $\mathbf{X} = (4, 4, 0, 0)'$ où les variables de base sont $\{x_1, x_2\}$.
- l'expression des variables de base en fonction des variables hors-base :
$$\begin{cases} x_2 = 4 + \frac{1}{3}x_3 - \frac{2}{3}x_4 \\ x_1 = 4 - \frac{2}{3}x_3 + \frac{1}{3}x_4 \end{cases}$$
- le vecteur des coûts réduits vaut $\overline{\mathbf{C}}_{hb} = (\underbrace{-5/3}_{x_3}, \underbrace{-2/3}_{x_4})'$
- la fonction objectif vaut $z(\mathbf{X}) = 4x_1 + 3x_2 = 28 - \frac{5}{3}x_3 - \frac{2}{3}x_4 = 28$

Exemple d'application (méthode du simplexe)

Tableau simplexe après la 2ème itération :

		x_1	x_2	x_3	x_4	
		4	3	0	0	
x_1	4	1	0	$2/3$	$-1/3$	4
x_2	3	0	1	$-1/3$	$2/3$	4
		0	0	$-5/3$	$-2/3$	28

Exemple d'application (méthode du simplexe)

Tableau simplexe après la 2ème itération :

		x_1	x_2	x_3	x_4	
		4	3	0	0	
x_1	4	1	0	$2/3$	$-1/3$	4
x_2	3	0	1	$-1/3$	$2/3$	4
		0	0	$-5/3$	$-2/3$	28

Fin de l'algorithme :

Etape 5 (non-vérification de la condition d'optimalité)
Il n'y a plus de coût réduit de variables hors base qui soit positif donc on ne peut plus améliorer la fonction objectif

Exemple d'application (méthode du simplexe)

Tableau simplexe après la 2ème itération :

		x_1	x_2	x_3	x_4	
		4	3	0	0	
x_1	4	1	0	$2/3$	$-1/3$	4
x_2	3	0	1	$-1/3$	$2/3$	4
		0	0	$-5/3$	$-2/3$	28

Fin de l'algorithme :

Etape 6 (détermination de la solution optimale)
La solution de base réalisable $\mathbf{X}^* = (4, 4, 0, 0)'$ où les variables de base sont $\{x_1, x_2\}$ est la solution optimale et $z(\mathbf{X}^*) = 28$.

Rappel du Sommaire

Rappel du Sommaire

Combinaisons linéaires d'équations linéaires

De manière générale, toute **combinaison linéaire d'équations linéaires** donne une équation linéaire **valide**.

Exemple :

$$
\begin{array}{lcccccccc}
l_1 : & 1 & x_1 & + & 2 & x_2 & \leq & 4 \\
l_2 : & 3 & x_1 & - & 1 & x_2 & = & 5 \\
l_3 : & 1 & x_1 & + & 1 & x_2 & \geq & 2
\end{array}
$$

Combinaisons linéaires d'équations linéaires

De manière générale, toute **combinaison linéaire d'équations linéaires** donne une équation linéaire **valide**.

Exemple :

$$
\begin{array}{lccccccccc}
l_1 : & 1 & x_1 & + & 2 & x_2 & \leq & 4 & (\times\ 4) \\
l_2 : & 3 & x_1 & - & 1 & x_2 & = & 5 & (\times\ -1) \\
l_3 : & 1 & x_1 & + & 1 & x_2 & \geq & 2 & (\times\ -2)
\end{array}
$$

Combinaisons linéaires d'équations linéaires

De manière générale, toute **combinaison linéaire d'équations linéaires** donne une équation linéaire **valide**.

Exemple :

$$\begin{array}{rrlrlcrl}
l_1 : & 1 & x_1 & + & 2 & x_2 & \leq & 4 & (\times 4)\\
l_2 : & 3 & x_1 & - & 1 & x_2 & = & 5 & (\times -1)\\
l_3 : & 1 & x_1 & + & 1 & x_2 & \geq & 2 & (\times -2)\\
& & & & & & & & \\
l_4 = 4l_1 - l_2 - 2l_3 : & -1 & x_1 & + & 7 & x_2 & \leq & 7 &
\end{array}$$

Exemple introductif

Reprenons le problème de maximisation suivant :

$$\begin{array}{rl}
\text{Maximiser :} & f(x_1, x_2) = z(\mathbf{X}) = 4x_1 + 3x_2\\
\text{Sous les contraintes :} & \left\{\begin{array}{lrcrcr} l_1 : & 2x_1 & + & x_2 & \leq & 12\\ l_2 : & x_1 & + & 2x_2 & \leq & 12\end{array}\right.\\
\text{Avec :} & x_1 \geq 0\ ,\ x_2 \geq 0
\end{array}$$

Exemple introductif

Reprenons le problème de maximisation suivant :

$$\begin{array}{rl}
\text{Maximiser :} & f(x_1, x_2) = z(\mathbf{X}) = 4x_1 + 3x_2\\
\text{Sous les contraintes :} & \left\{\begin{array}{lrcrcr} l_1 : & 2x_1 & + & x_2 & \leq & 12\\ l_2 : & x_1 & + & 2x_2 & \leq & 12\end{array}\right.\\
\text{Avec :} & x_1 \geq 0\ ,\ x_2 \geq 0
\end{array}$$

Multiplions par 3 la première inégalité :

$3l_1 : \quad 6x_1 \quad + \quad 3x_2 \quad \leq \quad 36$

Exemple introductif

Reprenons le problème de maximisation suivant :

$$\begin{array}{rl}
\text{Maximiser :} & f(x_1, x_2) = z(\mathbf{X}) = 4x_1 + 3x_2\\
\text{Sous les contraintes :} & \left\{\begin{array}{lrcrcr} l_1 : & 2x_1 & + & x_2 & \leq & 12\\ l_2 : & x_1 & + & 2x_2 & \leq & 12\end{array}\right.\\
\text{Avec :} & x_1 \geq 0\ ,\ x_2 \geq 0
\end{array}$$

Multiplions par 3 la première inégalité :

$3l_1 : \quad 6x_1 \quad + \quad 3x_2 \quad \leq \quad 36$

Si on compare cette inégalité avec la fonction objectif, on peut voir que : $z(\mathbf{X}) = 4x_1 + 3x_2 \leq 6x_1 + 3x_2 \leq \mathbf{36}$.

On obtient ainsi une **borne supérieure de** $z(\mathbf{X}^*)$

Exemple introductif

Reprenons le problème de maximisation suivant :

$$\begin{array}{rl} \text{Maximiser :} & f(x_1,x_2) = z(\mathbf{X}) = 4x_1 + 3x_2 \\ \text{Sous les contraintes :} & \left\{ \begin{array}{lcccccc} l_1 : & 2x_1 & + & x_2 & \leq & 12 \\ l_2 : & x_1 & + & 2x_2 & \leq & 12 \end{array} \right. \\ \text{Avec :} & x_1 \geq 0 \ , \ x_2 \geq 0 \end{array}$$

On peut faire mieux en prenant :

$2l_1 + \frac{1}{2}l_2 : \quad 4.5x_1 \quad + \quad 3x_2 \quad \leq \quad 30$

Exemple introductif

Reprenons le problème de maximisation suivant :

$$\begin{array}{rl} \text{Maximiser :} & f(x_1,x_2) = z(\mathbf{X}) = 4x_1 + 3x_2 \\ \text{Sous les contraintes :} & \left\{ \begin{array}{lcccccc} l_1 : & 2x_1 & + & x_2 & \leq & 12 \\ l_2 : & x_1 & + & 2x_2 & \leq & 12 \end{array} \right. \\ \text{Avec :} & x_1 \geq 0 \ , \ x_2 \geq 0 \end{array}$$

On peut faire mieux en prenant :

$2l_1 + \frac{1}{2}l_2 : \quad 4.5x_1 \quad + \quad 3x_2 \quad \leq \quad 30$

On a donc :

$z(\mathbf{X}) = 4x_1 + 3x_2 \leq 4.5x_1 + 3x_2 \leq \mathbf{30}$.

On obtient une meilleure borne supérieure de $z(\mathbf{X}^*)$

Exemple introductif

Reprenons le problème de maximisation suivant :

$$\begin{array}{rl} \text{Maximiser :} & f(x_1,x_2) = z(\mathbf{X}) = 4x_1 + 3x_2 \\ \text{Sous les contraintes :} & \left\{ \begin{array}{lcccccc} l_1 : & 2x_1 & + & x_2 & \leq & 12 \\ l_2 : & x_1 & + & 2x_2 & \leq & 12 \end{array} \right. \\ \text{Avec :} & x_1 \geq 0 \ , \ x_2 \geq 0 \end{array}$$

On peut faire encore mieux :

$\frac{5}{3}l_1 + \frac{2}{3}l_2 : \quad 4x_1 \quad + \quad 3x_2 \quad \leq \quad 28$

Exemple introductif

Reprenons le problème de maximisation suivant :

$$\begin{array}{rl} \text{Maximiser :} & f(x_1,x_2) = z(\mathbf{X}) = 4x_1 + 3x_2 \\ \text{Sous les contraintes :} & \left\{ \begin{array}{lcccccc} l_1 : & 2x_1 & + & x_2 & \leq & 12 \\ l_2 : & x_1 & + & 2x_2 & \leq & 12 \end{array} \right. \\ \text{Avec :} & x_1 \geq 0 \ , \ x_2 \geq 0 \end{array}$$

On peut faire encore mieux :

$\frac{5}{3}l_1 + \frac{2}{3}l_2 : \quad 4x_1 \quad + \quad 3x_2 \quad \leq \quad 28$

On a dans ce cas :

$z(\mathbf{X}) = 4x_1 + 3x_2 \leq 28$.

On obtient une meilleure borne supérieure de $z(\mathbf{X}^*)$ (qui s'avère ici être la solution optimale).

Généralisation de l'exemple précédent

Considérons un PL **non standard** avec des contraintes d'**inégalités** $\leq$ **uniquement**.

$$\begin{array}{lcccccc}
l_1 : & a_{11}x_1 & + & \dots & + & a_{1n}x_n & \leq & b_1 \\
l_2 : & a_{21}x_1 & + & \dots & + & a_{2n}x_n & \leq & b_2 \\
 & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots \\
l_m : & a_{m1}x_1 & + & \dots & + & a_{mn}x_n & \leq & b_m
\end{array}$$

Généralisation de l'exemple précédent

Considérons un PL **non standard** avec des contraintes d'**inégalités** $\leq$ **uniquement**.

$$\begin{array}{lccccccc}
l_1 : & a_{11}x_1 & + & \dots & + & a_{1n}x_n & \leq & b_1 & (\times\ y_1 \geq 0) \\
l_2 : & a_{21}x_1 & + & \dots & + & a_{2n}x_n & \leq & b_2 & (\times\ y_2 \geq 0) \\
 & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots & \vdots \\
l_m : & a_{m1}x_1 & + & \dots & + & a_{mn}x_n & \leq & b_m & (\times\ y_m \geq 0)
\end{array}$$

Généralisation de l'exemple précédent

Considérons un PL **non standard** avec des contraintes d'**inégalités** $\leq$ **uniquement**.

$$\begin{array}{lccccccc}
l_1 : & a_{11}x_1 & + & \dots & + & a_{1n}x_n & \leq & b_1 & (\times\ y_1 \geq 0) \\
l_2 : & a_{21}x_1 & + & \dots & + & a_{2n}x_n & \leq & b_2 & (\times\ y_2 \geq 0) \\
 & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots & \vdots \\
l_m : & a_{m1}x_1 & + & \dots & + & a_{mn}x_n & \leq & b_m & (\times\ y_m \geq 0)
\end{array}$$

$$\sum_i y_i l_i : \quad \sum_i y_i a_{i1} x_1 \quad + \quad \dots \quad + \quad \sum_i y_i a_{in} x_n \quad \leq \quad \sum_i y_i b_i$$

Généralisation de l'exemple précédent

Considérons un PL **non standard** avec des contraintes d'**inégalités** $\leq$ **uniquement**.

$$\begin{array}{lccccccc}
l_1 : & a_{11}x_1 & + & \dots & + & a_{1n}x_n & \leq & b_1 & (\times\ y_1 \geq 0) \\
l_2 : & a_{21}x_1 & + & \dots & + & a_{2n}x_n & \leq & b_2 & (\times\ y_2 \geq 0) \\
 & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots & \vdots \\
l_m : & a_{m1}x_1 & + & \dots & + & a_{mn}x_n & \leq & b_m & (\times\ y_m \geq 0)
\end{array}$$

$$\sum_i y_i l_i : \quad \sum_i y_i a_{i1} x_1 \quad + \quad \dots \quad + \quad \sum_i y_i a_{in} x_n \quad \leq \quad \sum_i y_i b_i$$

De plus, $\forall x_1, \dots, x_m \geq 0$, si $c_1 \leq \sum_{i=1}^m y_i a_{i1}, \dots, c_n \leq \sum_{i=1}^m y_i a_{in}$ alors :

$$z(\mathbf{X}) = \mathbf{C}'\mathbf{X} = c_1 x_1 + \dots + c_n x_n \leq \sum_{i=1}^m y_i a_{i1} x_1 + \dots + \sum_{i=1}^m y_i a_{in} x_n \leq \sum_{i=1}^n y_i b_i$$

Programme dual du PL de l'exemple (cas particulier)

Pour déterminer la **plus petite borne supérieure possible** de la fonction objectif il faut résoudre le **PL** suivant :

$$\begin{array}{rl} \text{Minimiser :} & w(\mathbf{Y}) = \sum_{i=1}^{n} b_i y_i \\ \text{Sous les contraintes :} & \left\{ \begin{array}{lcl} \sum_{i=1}^{m} y_i a_{i1} & \geq & c_1 \\ & \vdots & \\ \sum_{i=1}^{m} y_i a_{in} & \geq & c_n \end{array} \right. \\ \text{Avec :} & y_1 \geq 0, \ldots, y_n \geq 0 \end{array}$$

Programme dual du PL de l'exemple (cas particulier) (suite)

Dans le cas particulier où les contraintes linéaires sont toutes du type $\leq$ nous avons le formalisme matriciel suivant :

$$\underbrace{\begin{array}{rl} \text{Maximiser :} & z(\mathbf{X}) = \mathbf{C}'\mathbf{X} \\ \text{s.l.c :} & \mathbf{AX} \leq \mathbf{B} \\ \text{avec :} & \mathbf{X} \geq \mathbf{0} \end{array}}_{\text{Primal (PLP)}} \quad \text{et} \quad \underbrace{\begin{array}{rl} \text{Minimiser :} & w(\mathbf{Y}) = \mathbf{B}'\mathbf{Y} \\ \text{s.l.c :} & \mathbf{A}'\mathbf{Y} \geq \mathbf{C} \\ \text{avec :} & \mathbf{Y} \geq \mathbf{0} \end{array}}_{\text{Dual (PLD)}}$$

où $\mathbf{A}'$ est la tranposée de $\mathbf{A}$.

Rappel du Sommaire

Programme dual d'un PL quelconque (cas général)

A tout **PL quelconque** on peut lui associer un **PL dual, noté PLD**. Le PL initial est alors dénommé le PL primal et sera noté PLP. Le tableau suivant résume les correspondances entre primal et dual.

Primal (PLP)		Dual (PLD)
Problème de maximisation	$\leftrightarrow$	Problème de minimisation
A matrice des coefficients des contraintes	$\leftrightarrow$	$\mathbf{A}'$ matrice des coefficients des contraintes
Variable $x_j \geq 0$	$\leftrightarrow$	$j$ème contrainte de type $\geq$
Variable $x_j \in \mathbb{R}$	$\leftrightarrow$	$j$ème contrainte de type $=$
Variable $x_j \leq 0$	$\leftrightarrow$	$j$ème contrainte de type $\leq$
$i$ème contrainte de type $\leq$	$\leftrightarrow$	Variable $y_i \geq 0$
$i$ème contrainte de type $=$	$\leftrightarrow$	Variable $y_i \in \mathbb{R}$
$i$ème contrainte de type $\geq$	$\leftrightarrow$	Variable $y_i \leq 0$

On a la propriété que **le dual du dual est le primal**.

Programme dual d'un PL quelconque (cas général)

A tout **PL quelconque** on peut lui associer un **PL dual, noté PLD**. Le PL initial est alors dénommé le PL primal et sera noté PLP. Le tableau suivant résume les correspondances entre primal et dual.

Primal (PLP)		Dual (PLD)
Problème de maximisation	$\leftrightarrow$	Problème de minimisation
$\mathbf{A}$ matrice des coefficients des contraintes	$\leftrightarrow$	$\mathbf{A}'$ matrice des coefficients des contraintes
Variable $x_j \geq 0$	$\leftrightarrow$	$j$ème contrainte de type $\geq$
Variable $x_j \in \mathbb{R}$	$\leftrightarrow$	$j$ème contrainte de type $=$
Variable $x_j \leq 0$	$\leftrightarrow$	$j$ème contrainte de type $\leq$
$i$ème contrainte de type $\leq$	$\leftrightarrow$	Variable $y_i \geq 0$
$i$ème contrainte de type $=$	$\leftrightarrow$	Variable $y_i \in \mathbb{R}$
$i$ème contrainte de type $\geq$	$\leftrightarrow$	Variable $y_i \leq 0$

On a la propriété que **le dual du dual est le primal**.

Programme dual du PL de l'exemple (cas particulier)

$$\underbrace{\begin{array}{rl} \text{Minimiser :} & w(\mathbf{Y}) = \mathbf{B}'\mathbf{Y} \\ \text{s.l.c :} & \mathbf{A}'\mathbf{Y} \geq \mathbf{C} \\ \text{avec :} & \mathbf{Y} \geq \mathbf{0} \end{array}}_{\text{Primal (PLP)}} \quad \text{et} \quad \underbrace{\begin{array}{rl} \text{Maximiser :} & z(\mathbf{X}) = \mathbf{C}'\mathbf{X} \\ \text{s.l.c :} & \mathbf{A}\mathbf{X} \leq \mathbf{B} \\ \text{avec :} & \mathbf{X} \geq \mathbf{0} \end{array}}_{\text{Dual (PLD)}}$$

Le dual du dual est le primal (ici on passe d'un problème de minimisation à un problème de maximisation).

Exemple de dualisation dans le cas général

$$\underbrace{\begin{array}{rl} \text{Maximiser :} & 2x_1 - 3x_2 \\ \text{s.l.c :} & \left\{ \begin{array}{lcl} x_1 - x_2 & \leq & 1 \\ 2x_1 + 3x_2 & \geq & 4 \\ x_1 + x_2 & = & 3 \end{array} \right. \\ \text{avec :} & x_1 \geq 0, x_2 \geq 0 \end{array}}_{\text{Primal}}$$

Le dual de ce PL est le suivant :

Exemple de dualisation dans le cas général

$$\underbrace{\begin{array}{rl} \text{Maximiser :} & 2x_1 - 3x_2 \\ \text{s.l.c :} & \left\{ \begin{array}{lcl} x_1 - x_2 & \leq & 1 \\ 2x_1 + 3x_2 & \geq & 4 \\ x_1 + x_2 & = & 3 \end{array} \right. \\ \text{avec :} & x_1 \geq 0, x_2 \geq 0 \end{array}}_{\text{Primal}}$$

Le dual de ce PL est le suivant :

$$\underbrace{\begin{array}{rl} \text{Minimiser :} & y_1 + 4y_2 + 3y_3 \\ \text{s.l.c :} & \left\{ \begin{array}{lcl} y_1 + 2y_2 + y_3 & \geq & 2 \\ -y_1 + 3y_2 + y_3 & \geq & 2 \end{array} \right. \\ \text{avec :} & y_1 \geq 0, y_2 \leq 0, y_3 \in \mathbb{R} \end{array}}_{\text{Dual}}$$

Programme dual d'un PL standard

On suppose un PL de maximisation mis **sous forme standard**. Le dual dans ce cas est le suivant :

$$\underbrace{\begin{array}{rl} \text{Maximiser :} & z(\mathbf{X}) = \mathbf{C}'\mathbf{X} \\ \text{s.l.c :} & \mathbf{AX} = \mathbf{B} \\ \text{avec :} & \mathbf{X} \geq \mathbf{0} \end{array}}_{\text{Primal (PLP)}} \quad \text{et} \quad \underbrace{\begin{array}{rl} \text{Minimiser :} & w(\mathbf{Y}) = \mathbf{B}'\mathbf{Y} \\ \text{s.l.c :} & \mathbf{A}'\mathbf{Y} \geq \mathbf{C} \\ \text{avec :} & \mathbf{Y} \in \mathbb{R}^n \end{array}}_{\text{Dual (PLD)}}$$

Théorèmes sur la dualité

Lemme.

Soit $\hat{\mathbf{X}}$ et $\hat{\mathbf{Y}}$ deux solutions réalisables du primal (mis sous forme standard) et du dual associé alors :

$$\underbrace{z(\hat{\mathbf{X}}) = \mathbf{C}'\hat{\mathbf{X}}} \leq \underbrace{\mathbf{B}'\hat{\mathbf{Y}} = w(\hat{\mathbf{Y}})}$$

Dualité en programmation linéaire | Programme primal et dual

Théorèmes sur la dualité

Lemme.

Soit $\hat{\mathbf{X}}$ et $\hat{\mathbf{Y}}$ deux solutions réalisables du primal (mis sous forme standard) et du dual associé alors :

$$\underbrace{z(\hat{\mathbf{X}}) = \mathbf{C}'\hat{\mathbf{X}}} \leq \underbrace{\mathbf{B}'\hat{\mathbf{Y}} = w(\hat{\mathbf{Y}})}$$

Démonstration.

$(\mathbf{A}\hat{\mathbf{X}} = \mathbf{B}) \Rightarrow (\hat{\mathbf{Y}}'\mathbf{A}\hat{\mathbf{X}} = \hat{\mathbf{Y}}'\mathbf{B})$.

Comme de plus, $\hat{\mathbf{X}} \geq \mathbf{0}, \mathbf{A}'\hat{\mathbf{Y}} \geq \mathbf{C}$ nous avons donc $\left(\mathbf{A}'\hat{\mathbf{Y}}\right)'\hat{\mathbf{X}} \geq \mathbf{C}'\hat{\mathbf{X}}$. □

Théorèmes sur la dualité (suite)

Corollaire.

Si $\mathbf{X}^*$ et $\mathbf{Y}^*$ sont deux solutions réalisables du primal et du dual associé tels que $\mathbf{C}'\mathbf{X}^* = \mathbf{B}'\mathbf{Y}^*$ alors $\mathbf{X}^*$ est optimum du primal et $\mathbf{Y}^*$ optimum du dual.

Théorème.

Etant donné un PL primal PLP et le PL dual associé PLD :

- *Si PLP et PLD ont des solutions réalisables alors chacun d'eux a une solution optimale $\mathbf{X}^*$ et $\mathbf{Y}^*$ tel que :*

$$z^* = \mathbf{C}'\mathbf{X}^* = \mathbf{B}'\mathbf{Y}^* = w^*$$

- *Si l'un d'eux a un optimum non borné, l'autre n'a pas de solution réalisable*

Applications des propriétés de dualité

- Si le primal a beaucoup de contraintes et moins de variables alors la méthode du simplexe sera plus efficace sur le dual

Dualité en programmation linéaire | Programme primal et dual

Applications des propriétés de dualité

- Si le primal a beaucoup de contraintes et moins de variables alors la méthode du simplexe sera plus efficace sur le dual
- A l'optimum, le tableau du simplexe fournit à la fois la solution optimale pour le primal et pour le dual

J. Ah-Pine (Univ-Lyon 2) | Théorie des graphes et Optimisation | M1 Informatique 2010-2011 / 142 - 173

Dualité en programmation linéaire | Programme primal et dual

Applications des propriétés de dualité

- Si le primal a beaucoup de contraintes et moins de variables alors la méthode du simplexe sera plus efficace sur le dual
- A l'optimum, le tableau du simplexe fournit à la fois la solution optimale pour le primal et pour le dual
- La solution optimale du dual peut s'interprèter d'un point de vue économique. De $z^* = \underbrace{\sum_j c_j x_j^*}_{\mathbf{C'X^*}} = \underbrace{\sum_i b_i y_i^*}_{\mathbf{B'Y^*}} = w^*$ nous en déduisons :

$$\frac{\partial}{\partial b_i} z^* = y_i^*$$

  y_i^* représente le **prix marginal** de la ressource i à l'optimum. Autrement dit, y_i^* représente l'augmentation potentielle de la valeur optimale du problème si la ressource i, actuellement limitée à b_i, se voyait augmenter d'une unité.

J. Ah-Pine (Univ-Lyon 2) | Théorie des graphes et Optimisation | M1 Informatique 2010-2011 / 142 - 173

Dualité en programmation linéaire | Programme primal et dual

Exemple

Une entreprise fabrique deux biens (des pièces mécaniques par exemple). Ces fabrications nécessitent l'utilisation de deux ateliers dont les capacités de production exprimées en heures d'usinage sont de 12. Supposons que :

- Chaque unité du 1er produit nécessite 2h d'usinage dans l'atelier 1 et 1h dans l'atelier 2.
- Chaque unité du 2ème produit nécessite 1h d'usinage dans l'atelier 1 et 2h dans l'atelier 2.

Sachant que la marge sur le 1er produit est $p_1 = 4$ et que celle sur le 2ème produit est de $p_2 = 3$, déterminer un programme mathématique qui modélise le problème de l'optimisation de la marge de l'entreprise sous les contraintes de production décrites précédemment.

Exemple (suite)

$$\underbrace{\begin{array}{rl} \text{Maximiser :} & z(x_1,x_2) = 4x_1 + 3x_2 \\ \text{s.l.c :} & \begin{cases} 2x_1 + x_2 \leq 12 \\ x_1 + 2x_2 \leq 12 \end{cases} \\ \text{Avec :} & x_1 \geq 0 \;,\; x_2 \geq 0 \end{array}}_{\text{Primal (PLP)}}$$

Exemple (suite)

$$\underbrace{\begin{array}{rl} \text{Maximiser :} & z(x_1,x_2) = 4x_1 + 3x_2 \\ \text{s.l.c :} & \begin{cases} 2x_1 + x_2 \leq 12 \\ x_1 + 2x_2 \leq 12 \end{cases} \\ \text{Avec :} & x_1 \geq 0 \;,\; x_2 \geq 0 \end{array}}_{\text{Primal (PLP)}}$$

Le dual est le suivant :

$$\underbrace{\begin{array}{rl} \text{Minimiser :} & w(y_1,y_2) = 12y_1 + 12y_2 \\ \text{s.l.c. :} & \begin{cases} 2y_1 + y_2 \geq 4 \\ y_1 + 2y_2 \geq 3 \end{cases} \\ \text{Avec :} & y_1 \geq 0 \;,\; y_2 \geq 0 \end{array}}_{\text{Dual (PLD)}}$$

Exemple (suite)

Le tableau à l'optimum est :

		x_1	x_2	x_3	x_4	
		4	3	0	0	
x_1	4	1	0	$2/3$	$-1/3$	4
x_2	3	0	1	$-1/3$	$2/3$	4
		0	0	$-5/3$	$-2/3$	28

Exemple (suite)

Le tableau à l'optimum est :

		x_1	x_2	x_3	x_4	
		4	3	0	0	
x_1	4	1	0	$2/3$	$-1/3$	4
x_2	3	0	1	$-1/3$	$2/3$	4
		0	0	$-5/3$	$-2/3$	28

Sur sa dernière ligne l'opposé des "coûts" réduits correspond à la solution optimale du dual : $\mathbf{Y}^* = (5/3, 2/3)$.

$y_1^* = 5/3$ est le "prix" marginal de l'atelier 1 tandis que $y_2^* = 2/3$ est le "prix" marginal de l'atelier 2.

Rappel du Sommaire

Rappel du Sommaire

Programme linéaire en nombres entiers

- Un **programme linéaire en nombres entiers** est un PL dont les variables sont contraintes à ne prendre que des valeurs entières
- Lorsqu'elles sont contraintes à prendre les valeurs 0 ou 1 on parlera alors d'un **programme binaire**
- Un **programme mixte en nombres entiers** est un programme où uniquement certaines variables sont contraintes de prendre des valeurs entières

Approche naïve

1. On résoud le PL en ignorant les contraintes d'intégralité
2. Une fois la solution non entière trouvée, on arrondi

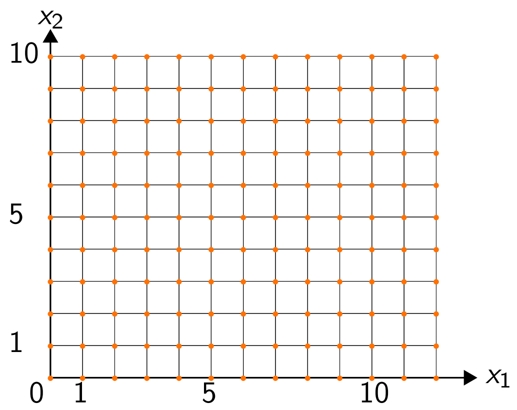
Approche naïve

1. On résoud le PL en ignorant les contraintes d'intégralité
2. Une fois la solution non entière trouvée, on arrondi

Cette approche **ne marche pas** en générale !

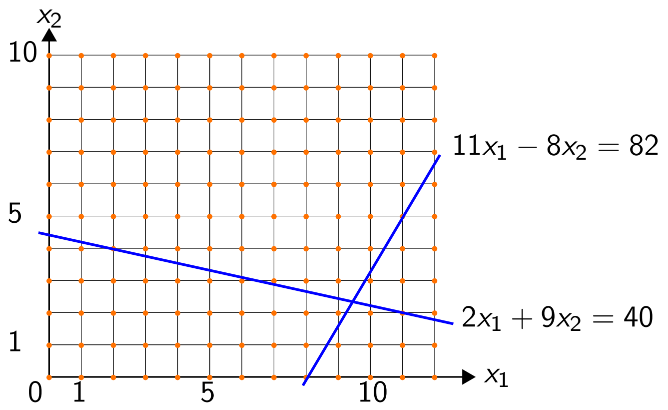
Contre exemple

$$\begin{array}{rl} \text{Maximiser :} & 3x_1 + 13x_2 \\ \text{s.l.c. :} & \left\{\begin{array}{rcrcl} 2x_1 & + & 9x_2 & \leq & 40 \\ 11x_1 & - & 8x_2 & \leq & 82 \end{array}\right. \\ \text{Avec :} & x_1, x_2 \geq 0 \ , \ x_1, x_2 \in \mathbb{N} \text{ (ensemble des entiers naturels)} \end{array}$$



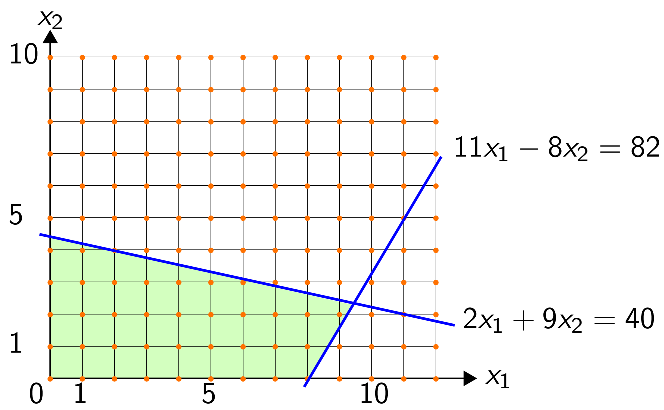
Contre exemple

$$\begin{array}{rl} \text{Maximiser :} & 3x_1 + 13x_2 \\ \text{s.l.c. :} & \left\{\begin{array}{rcrcl} 2x_1 & + & 9x_2 & \leq & 40 \\ 11x_1 & - & 8x_2 & \leq & 82 \end{array}\right. \\ \text{Avec :} & x_1, x_2 \geq 0 \ , \ x_1, x_2 \in \mathbb{N} \text{ (ensemble des entiers naturels)} \end{array}$$



Contre exemple

$$\begin{array}{rl} \text{Maximiser :} & 3x_1 + 13x_2 \\ \text{s.l.c. :} & \left\{\begin{array}{rcrcl} 2x_1 & + & 9x_2 & \leq & 40 \\ 11x_1 & - & 8x_2 & \leq & 82 \end{array}\right. \\ \text{Avec :} & x_1, x_2 \geq 0 \ , \ x_1, x_2 \in \mathbb{N} \text{ (ensemble des entiers naturels)} \end{array}$$

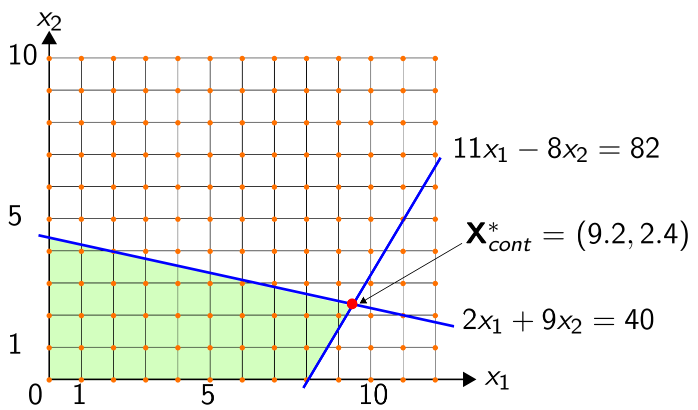


Contre exemple

Maximiser : $3x_1 + 13x_2$

$$\text{s.l.c. :} \quad \begin{cases} 2x_1 & + & 9x_2 & \leq & 40 \\ 11x_1 & - & 8x_2 & \leq & 82 \end{cases}$$

Avec : $x_1, x_2 \geq 0$, $x_1, x_2 \in \mathbb{N}$ (ensemble des entiers naturels)

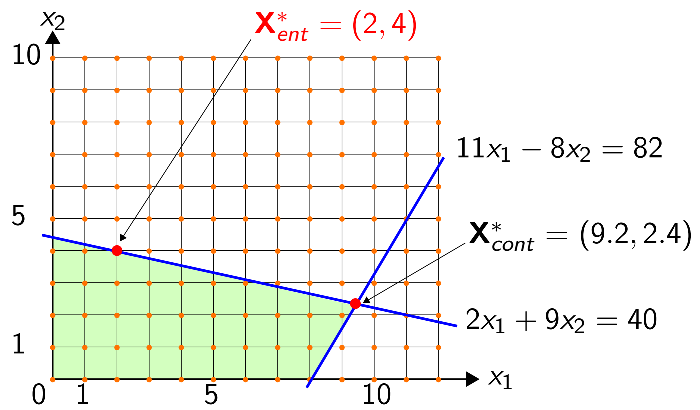


Contre exemple

Maximiser : $3x_1 + 13x_2$

$$\text{s.l.c. :} \quad \begin{cases} 2x_1 & + & 9x_2 & \leq & 40 \\ 11x_1 & - & 8x_2 & \leq & 82 \end{cases}$$

Avec : $x_1, x_2 \geq 0$, $x_1, x_2 \in \mathbb{N}$ (ensemble des entiers naturels)



Rappel du Sommaire

Principe de l'appoche Séparation- Evaluation

Il s'agit d'un algorithme exacte (càd qui permet de déterminer la solution optimale). Le principe général repose sur :

- Diviser pour mieux conquérir
- Utilisation de bornes sur le coût optimal afin d'éviter d'explorer certaines parties de l'ensemble des solutions admissibles

Principe de l'appoche Séparation- Evaluation (suite)

- Séparation ("Branch") : Soit $\mathbb{S}$ l'ensemble des solutions réalisables d'un PL : $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}$.

Programmation linéaire en nombres entiers | La méthode Séparation-Evaluation ("Branch and Bound")

Principe de l'appoche Séparation- Evaluation (suite)

- Séparation ("Branch") : Soit $\mathbb{S}$ l'ensemble des solutions réalisables d'un PL : $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}$.
  - On **partitionne** $\mathbb{S}$ en une collection finie de sous-ensembles $\mathbb{S} = \{\mathbb{S}_i\}$.

J. Ah-Pine (Univ-Lyon 2) | Théorie des graphes et Optimisation | M1 Informatique 2010-2011 / 153 - 173

Programmation linéaire en nombres entiers | La méthode Séparation-Evaluation ("Branch and Bound")

Principe de l'appoche Séparation- Evaluation (suite)

- Séparation ("Branch") : Soit $\mathbb{S}$ l'ensemble des solutions réalisables d'un PL : $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}$.
  - On **partitionne** $\mathbb{S}$ en une collection finie de sous-ensembles $\mathbb{S} = \{\mathbb{S}_i\}$.
  - On résoud séparemment les sous-problèmes $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_i$

J. Ah-Pine (Univ-Lyon 2) | Théorie des graphes et Optimisation | M1 Informatique 2010-2011 / 153 - 173

Programmation linéaire en nombres entiers | La méthode Séparation-Evaluation ("Branch and Bound")

Principe de l'appoche Séparation- Evaluation (suite)

- Séparation ("Branch") : Soit $\mathbb{S}$ l'ensemble des solutions réalisables d'un PL : $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}$.
  - On **partitionne** $\mathbb{S}$ en une collection finie de sous-ensembles $\mathbb{S} = \{\mathbb{S}_i\}$.
  - On résoud séparemment les sous-problèmes $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_i$
- Evaluation ("Bound") :
  - On suppose que l'on peut pour chaque sous-problème $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_i$, **calculer efficacement une borne supérieure** sur la valeur optimale de la fonction objectif : $\forall \mathbf{X} \in \mathbb{S}_i : \mathbf{C}'\mathbf{X} \leq B_i$

J. Ah-Pine (Univ-Lyon 2) | Théorie des graphes et Optimisation | M1 Informatique 2010-2011 / 153 - 173

Principe de l'appoche Séparation- Evaluation (suite)

- Séparation ("Branch") : Soit $\mathbb{S}$ l'ensemble des solutions réalisables d'un PL : $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}$.
  - On **partitionne** $\mathbb{S}$ en une collection finie de sous-ensembles $\mathbb{S} = \{\mathbb{S}_i\}$.
  - On résoud séparemment les sous-problèmes $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_i$
- Evaluation ("Bound") :
  - On suppose que l'on peut pour chaque sous-problème $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_i$, **calculer efficacement une borne supérieure** sur la valeur optimale de la fonction objectif : $\forall \mathbf{X} \in \mathbb{S}_i : \mathbf{C}'\mathbf{X} \leq B_i$
  - On utilise ces bornes afin de ne pas résoudre des sous-problèmes qui conduisent à des solutions non-optimales par rapport au problème initial.

Rappel du Sommaire

Exemple

Une entreprise fabrique des armoires et des tables :

- Une armoire nécessite 1h de travail et 9 m2 de bois
- Une table nécessite 1h de travail et 5 m2 de bois
- On dispose de 6h de travail et de 45 m2 de bois
- Chaque armoire génère un profit de 8 Euros, et chaque table un profit de 5 Euros
- Formuler et **résoudre en entiers**

Exemple

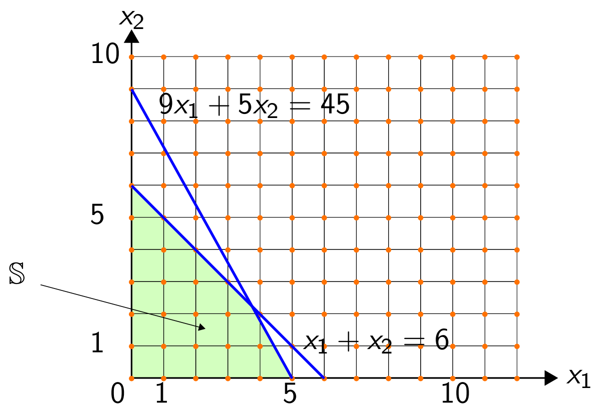
$$\begin{array}{rl} \text{Maximiser :} & 8x_1 + 5x_2 \\ \text{s.l.c. :} & \left\{ \begin{array}{rcrcr} x_1 & + & x_2 & \leq & 6 \\ 9x_1 & + & 5x_2 & \leq & 45 \end{array} \right. \\ \text{Avec :} & x_1, x_2 \geq 0 \ , \ x_1, x_2 \in \mathbb{N} \end{array}$$

Exemple

Maximiser : $8x_1 + 5x_2$

s.l.c. : $\begin{cases} x_1 & + & x_2 & \leq & 6 \\ 9x_1 & + & 5x_2 & \leq & 45 \end{cases}$

Avec : $x_1, x_2 \geq 0$, $x_1, x_2 \in \mathbb{N}$



Exemple (suite)

On commence par résoudre le PL en relâchant les contraintes d'intégrités (**relaxation linéaire du problème en nombres entiers**) càd : max $\mathbf{C'X}$ s.l.c $\mathbf{X} \in \mathbb{S}$ et $\mathbf{X} \in \mathbb{R}^2$.

Exemple (suite)

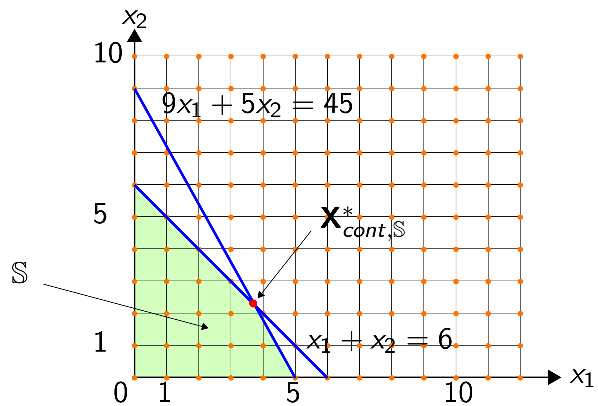
On commence par résoudre le PL en relâchant les contraintes d'intégrités (**relaxation linéaire du problème en nombres entiers**) càd : max $\mathbf{C'X}$ s.l.c $\mathbf{X} \in \mathbb{S}$ et $\mathbf{X} \in \mathbb{R}^2$.

On a la solution $z^*_{cont,\mathbb{S}} = \frac{165}{4} = 41.25$ et $\mathbf{X}^*_{cont,\mathbb{S}} = \begin{pmatrix} 15/4 \\ 9/4 \end{pmatrix} = \begin{pmatrix} 3.75 \\ 2.25 \end{pmatrix}$.

Exemple (suite)

On commence par résoudre le PL en relâchant les contraintes d'intégrités (**relaxation linéaire du problème en nombres entiers**) càd : max $\mathbf{C'X}$ s.l.c $\mathbf{X} \in \mathbb{S}$ et $\mathbf{X} \in \mathbb{R}^2$.

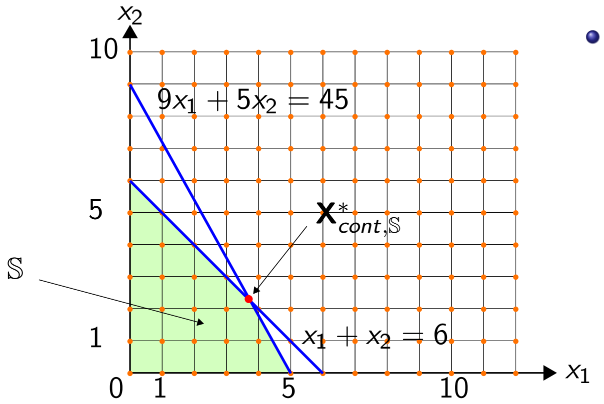
On a la solution $z^*_{cont,\mathbb{S}} = \frac{165}{4} = 41.25$ et $\mathbf{X}^*_{cont,\mathbb{S}} = \begin{pmatrix} 15/4 \\ 9/4 \end{pmatrix} = \begin{pmatrix} 3.75 \\ 2.25 \end{pmatrix}$.



Exemple (suite)

On commence par résoudre le PL en relâchant les contraintes d'intégrités (**relaxation linéaire du problème en nombres entiers**) càd : $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}$ et $\mathbf{X} \in \mathbb{R}^2$.

On a la solution $z^*_{cont,\mathbb{S}} = \frac{165}{4} = 41.25$ et $\mathbf{X}^*_{cont,\mathbb{S}} = \begin{pmatrix} 15/4 \\ 9/4 \end{pmatrix} = \begin{pmatrix} 3.75 \\ 2.25 \end{pmatrix}$.

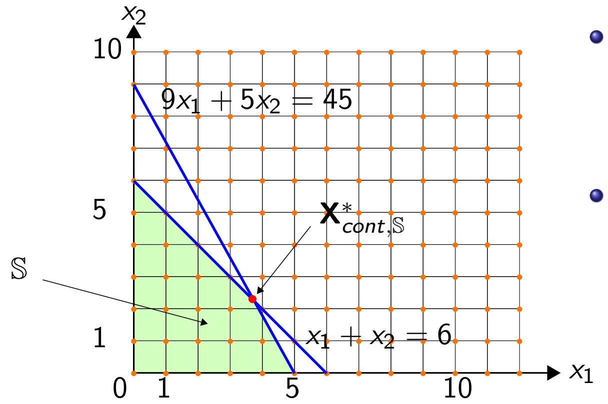


- Si la solution est entière alors on s'arrête et l'optimum entier est déterminé

Exemple (suite)

On commence par résoudre le PL en relâchant les contraintes d'intégrités (**relaxation linéaire du problème en nombres entiers**) càd : $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}$ et $\mathbf{X} \in \mathbb{R}^2$.

On a la solution $z^*_{cont,\mathbb{S}} = \frac{165}{4} = 41.25$ et $\mathbf{X}^*_{cont,\mathbb{S}} = \begin{pmatrix} 15/4 \\ 9/4 \end{pmatrix} = \begin{pmatrix} 3.75 \\ 2.25 \end{pmatrix}$.



- Si la solution est entière alors on s'arrête et l'optimum entier est déterminé
- **La solution n'est pas entière**, $z^*_{cont,\mathbb{S}}$ est une borne supérieure de la solution optimale en nombres entiers et on partitionne $\mathbb{S}$

Exemple (suite)

La solution optimale du problème relaxé n'étant pas entière, on partitionne $\mathbb{S}$:

- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière

Exemple (suite)

La solution optimale du problème relaxé n'étant pas entière, on partitionne $\mathbb{S}$:

- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière
  $\rightarrow x_1$ par exemple

Exemple (suite)

La solution optimale du problème relaxé n'étant pas entière, on partitionne $\mathbb{S}$:

- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière
  $\rightarrow x_1$ par exemple
- on applique des contraintes supplémentaires dues à la nature de la variable

Exemple (suite)

La solution optimale du problème relaxé n'étant pas entière, on partitionne $\mathbb{S}$:

- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière
  $\rightarrow x_1$ par exemple
- on applique des contraintes supplémentaires dues à la nature de la variable
  $\rightarrow x_1 \leq 3$ ou $x_1 \geq 4$ (car dans la solution optimale du problème relaxé on a $3 < x_1^* < 4$)

Exemple (suite)

La solution optimale du problème relaxé n'étant pas entière, on partitionne $\mathbb{S}$:

- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière
  $\rightarrow x_1$ par exemple
- on applique des contraintes supplémentaires dues à la nature de la variable
  $\rightarrow x_1 \leq 3$ ou $x_1 \geq 4$ (car dans la solution optimale du problème relaxé on a $3 < x_1^* < 4$)

On a séparé le problème ("branch") et on obtient ainsi deux sous-problèmes :

Exemple (suite)

La solution optimale du problème relaxé n'étant pas entière, on partitionne $\mathbb{S}$:

- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière
  $\rightarrow x_1$ par exemple
- on applique des contraintes supplémentaires dues à la nature de la variable
  $\rightarrow x_1 \leq 3$ ou $x_1 \geq 4$ (car dans la solution optimale du problème relaxé on a $3 < x_1^* < 4$)

On a séparé le problème ("branch") et on obtient ainsi deux sous-problèmes :

- $\max \mathbf{C'X}$ s.l.c $\mathbf{X} \in \mathbb{S}_1$ avec $\mathbb{S}_1$ défini par ($\mathbb{S}$ **et** $x_1 \leq 3$)

Exemple (suite)

La solution optimale du problème relaxé n'étant pas entière, on partitionne $\mathbb{S}$:

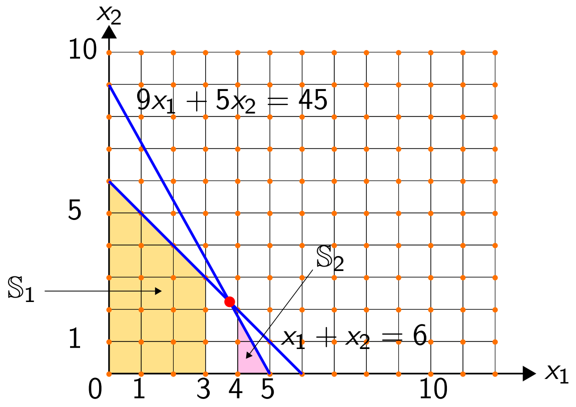
- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière
  $\rightarrow x_1$ par exemple
- on applique des contraintes supplémentaires dues à la nature de la variable
  $\rightarrow x_1 \leq 3$ ou $x_1 \geq 4$ (car dans la solution optimale du problème relaxé on a $3 < x_1^* < 4$)

On a séparé le problème ("branch") et on obtient ainsi deux sous-problèmes :

- $\max \mathbf{C'X}$ s.l.c $\mathbf{X} \in \mathbb{S}_1$ avec $\mathbb{S}_1$ défini par ($\mathbb{S}$ **et** $x_1 \leq 3$)
- $\max \mathbf{C'X}$ s.l.c $\mathbf{X} \in \mathbb{S}_2$ avec $\mathbb{S}_2$ défini par ($\mathbb{S}$ **et** $x_1 \geq 4$)

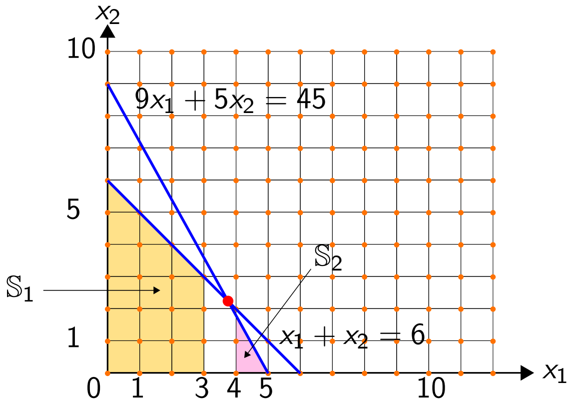
Exemple (suite)

A l'issue de la séparation, on a les deux sous-problèmes dont les représentations graphiques sont les suivantes :



Exemple (suite)

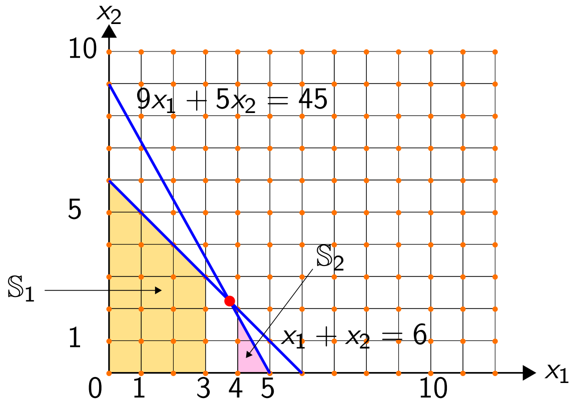
A l'issue de la séparation, on a les deux sous-problèmes dont les représentations graphiques sont les suivantes :



- On remarque qu'on a ainsi éliminé la solution $\mathbf{X}^*_{cont,\mathbb{S}}$

Exemple (suite)

A l'issue de la séparation, on a les deux sous-problèmes dont les représentations graphiques sont les suivantes :



- On remarque qu'on a ainsi éliminé la solution $\mathbf{X}^*_{cont,\mathbb{S}}$
- On peut résoudre ces deux sous-problèmes en considérant à nouveau leur relaxation linéaire

Exemple (suite)

On décide de résoudre tout d'abord la relaxation linéaire du deuxième sous-problème càd $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_2$ et $\mathbf{X} \in \mathbb{R}^2$.

Exemple (suite)

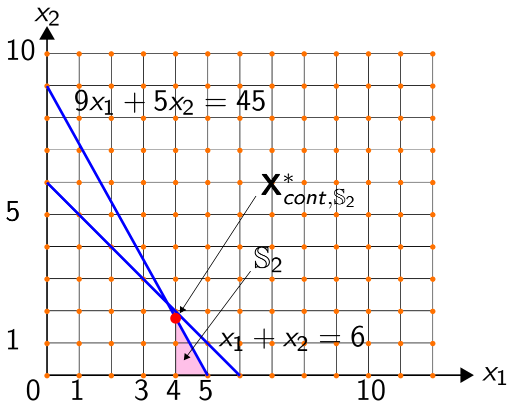
On décide de résoudre tout d'abord la relaxation linéaire du deuxième sous-problème càd $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_2$ et $\mathbf{X} \in \mathbb{R}^2$.

On obtient la solution $z^*_{cont,\mathbb{S}_2} = 41$ et $\mathbf{X}^*_{cont,\mathbb{S}_2} = \begin{pmatrix} 4 \\ 9/5 \end{pmatrix} = \begin{pmatrix} 4 \\ 1.8 \end{pmatrix}$.

Exemple (suite)

On décide de résoudre tout d'abord la relaxation linéaire du deuxième sous-problème càd $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_2$ et $\mathbf{X} \in \mathbb{R}^2$.

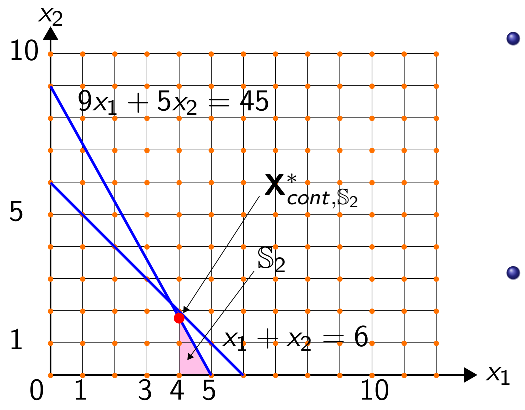
On obtient la solution $z^*_{cont,\mathbb{S}_2} = 41$ et $\mathbf{X}^*_{cont,\mathbb{S}_2} = \begin{pmatrix} 4 \\ 9/5 \end{pmatrix} = \begin{pmatrix} 4 \\ 1.8 \end{pmatrix}$.



Exemple (suite)

On décide de résoudre tout d'abord la relaxation linéaire du deuxième sous-problème càd $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_2$ et $\mathbf{X} \in \mathbb{R}^2$.

On obtient la solution $z^*_{cont,\mathbb{S}_2} = 41$ et $\mathbf{X}^*_{cont,\mathbb{S}_2} = \begin{pmatrix} 4 \\ 9/5 \end{pmatrix} = \begin{pmatrix} 4 \\ 1.8 \end{pmatrix}$.



- La solution trouvée est une borne supérieure de la solution optimale en nombres entiers sur le domaine considéré : $z^*_{cont,\mathbb{S}_2} < z^*_{cont,\mathbb{S}}$
- **La solution n'est pas entière**, on partitionne à nouveau le domaine considéré

Exemple (suite)

La solution optimale du problème relaxé sur $\mathbb{S}_2$ n'étant pas entière, on partitionne $\mathbb{S}_2$:

- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière

Exemple (suite)

La solution optimale du problème relaxé sur $\mathbb{S}_2$ n'étant pas entière, on partitionne $\mathbb{S}_2$:

- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière
  $\rightarrow x_2$ dans ce cas

Exemple (suite)

La solution optimale du problème relaxé sur $\mathbb{S}_2$ n'étant pas entière, on partitionne $\mathbb{S}_2$:

- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière
  $\rightarrow x_2$ dans ce cas
- on applique des contraintes supplémentaires dues à la nature de la variable

Exemple (suite)

La solution optimale du problème relaxé sur $\mathbb{S}_2$ n'étant pas entière, on partitionne $\mathbb{S}_2$:

- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière
  $\rightarrow x_2$ dans ce cas
- on applique des contraintes supplémentaires dues à la nature de la variable
  $\rightarrow x_2 \leq 1$ ou $x_2 \geq 2$ (car dans la solution optimale du problème relaxé sur $\mathbb{S}_2$, on a $1 < x_2^* < 2$)

Exemple (suite)

La solution optimale du problème relaxé sur $\mathbb{S}_2$ n'étant pas entière, on partitionne $\mathbb{S}_2$:

- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière
  $\rightarrow x_2$ dans ce cas
- on applique des contraintes supplémentaires dues à la nature de la variable
  $\rightarrow x_2 \leq 1$ ou $x_2 \geq 2$ (car dans la solution optimale du problème relaxé sur $\mathbb{S}_2$, on a $1 < x_2^* < 2$)

On a séparé à nouveau le problème ("branch") et on obtient ainsi deux nouveaux sous-problèmes :

Exemple (suite)

La solution optimale du problème relaxé sur $\mathbb{S}_2$ n'étant pas entière, on partitionne $\mathbb{S}_2$:

- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière
  $\rightarrow x_2$ dans ce cas
- on applique des contraintes supplémentaires dues à la nature de la variable
  $\rightarrow x_2 \leq 1$ ou $x_2 \geq 2$ (car dans la solution optimale du problème relaxé sur $\mathbb{S}_2$, on a $1 < x_2^* < 2$)

On a séparé à nouveau le problème ("branch") et on obtient ainsi deux nouveaux sous-problèmes :

- $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,1}$ avec $\mathbb{S}_{2,1}$ défini par ($\mathbb{S}_2$ **et** $x_2 \leq 1$)

Exemple (suite)

La solution optimale du problème relaxé sur $\mathbb{S}_2$ n'étant pas entière, on partitionne $\mathbb{S}_2$:

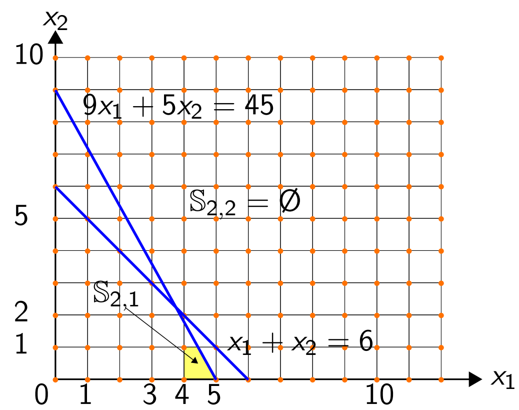
- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière
  $\rightarrow x_2$ dans ce cas
- on applique des contraintes supplémentaires dues à la nature de la variable
  $\rightarrow x_2 \leq 1$ ou $x_2 \geq 2$ (car dans la solution optimale du problème relaxé sur $\mathbb{S}_2$, on a $1 < x_2^* < 2$)

On a séparé à nouveau le problème ("branch") et on obtient ainsi deux nouveaux sous-problèmes :

- $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,1}$ avec $\mathbb{S}_{2,1}$ défini par ($\mathbb{S}_2$ **et** $x_2 \leq 1$)
- $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,2}$ avec $\mathbb{S}_{2,2}$ défini par ($\mathbb{S}_2$ **et** $x_2 \geq 2$)

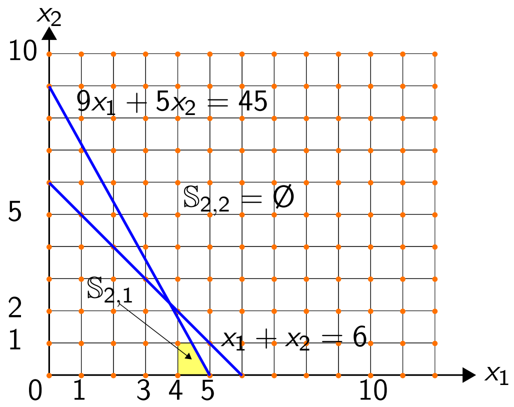
Exemple (suite)

A l'issue de la séparation de $\mathbb{S}_2$, on a deux nouveaux sous-problèmes dont les représentations graphiques sont les suivantes :



Exemple (suite)

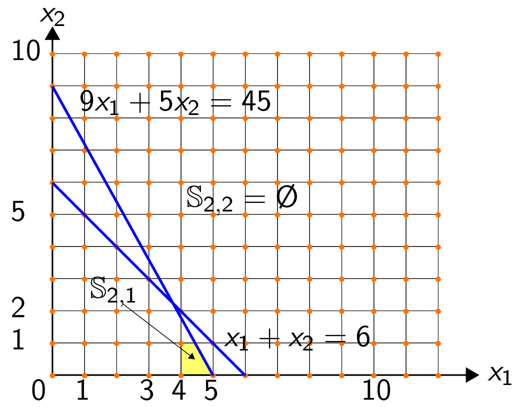
A l'issue de la séparation de $\mathbb{S}_2$, on a deux nouveaux sous-problèmes dont les représentations graphiques sont les suivantes :



- On élimine ainsi la solution $\mathbf{X}^*_{cont,\mathbb{S}_2}$. Par ailleurs $\mathbb{S}_{2,2}$ est vide

Exemple (suite)

A l'issue de la séparation de $\mathbb{S}_2$, on a deux nouveaux sous-problèmes dont les représentations graphiques sont les suivantes :



- On élimine ainsi la solution $\mathbf{X}^*_{cont,\mathbb{S}_2}$. Par ailleurs $\mathbb{S}_{2,2}$ est vide
- On peut résoudre le nouveau sous-problème en considérant à nouveau sa relaxation linéaire

Exemple (suite)

On résoud la relaxation linéaire du sous-problème max $\mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,1}$ et $\mathbf{X} \in \mathbb{R}^2$.

Exemple (suite)

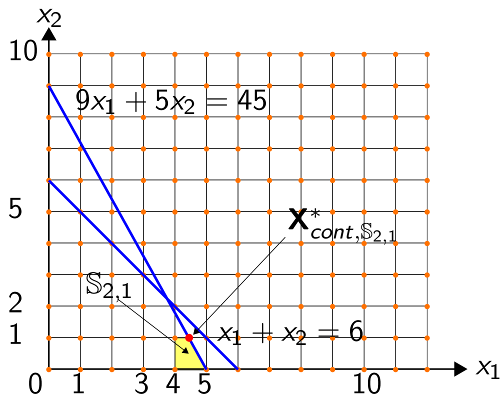
On résoud la relaxation linéaire du sous-problème max $\mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,1}$ et $\mathbf{X} \in \mathbb{R}^2$.

On obtient $z^*_{cont,\mathbb{S}_{2,1}} = \frac{365}{9} = 40.555$ et $\mathbf{X}^*_{cont,\mathbb{S}_{2,1}} = \begin{pmatrix} 40/9 \\ 1 \end{pmatrix} = \begin{pmatrix} 4.444 \\ 1 \end{pmatrix}$.

Exemple (suite)

On résoud la relaxation linéaire du sous-problème max $\mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,1}$ et $\mathbf{X} \in \mathbb{R}^2$.

On obtient $z^*_{cont,\mathbb{S}_{2,1}} = \frac{365}{9} = 40.555$ et $\mathbf{X}^*_{cont,\mathbb{S}_{2,1}} = \begin{pmatrix} 40/9 \\ 1 \end{pmatrix} = \begin{pmatrix} 4.444 \\ 1 \end{pmatrix}$.

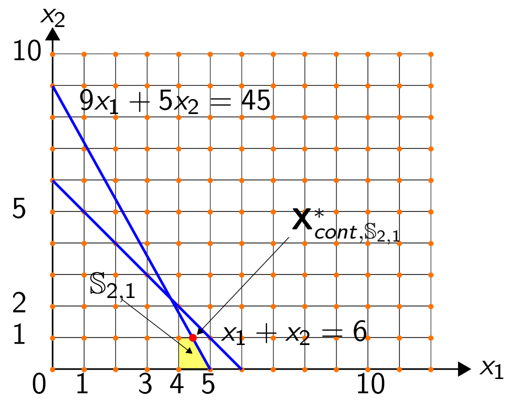


Programmation linéaire en nombres entiers | Exemple complet

Exemple (suite)

On résoud la relaxation linéaire du sous-problème max $\mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,1}$ et $\mathbf{X} \in \mathbb{R}^2$.

On obtient $z^*_{cont,\mathbb{S}_{2,1}} = \frac{365}{9} = 40.555$ et $\mathbf{X}^*_{cont,\mathbb{S}_{2,1}} = \begin{pmatrix} 40/9 \\ 1 \end{pmatrix} = \begin{pmatrix} 4.444 \\ 1 \end{pmatrix}$.



- La solution trouvée est une borne supérieure de la solution optimale en nombres entiers sur le domaine considéré : $z^*_{cont,\mathbb{S}_{2,1}} < z^*_{cont,\mathbb{S}_2}$
- **La solution n'est pas entière**, on partitionne à nouveau le domaine considéré

Exemple (suite)

La solution optimale du problème relaxé sur $\mathbb{S}_{2,1}$ n'étant pas entière, on partitionne à nouveau $\mathbb{S}_{2,1}$:

- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière

Programmation linéaire en nombres entiers | Exemple complet

Exemple (suite)

La solution optimale du problème relaxé sur $\mathbb{S}_{2,1}$ n'étant pas entière, on partitionne à nouveau $\mathbb{S}_{2,1}$:

- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière
  $\rightarrow x_1$ dans ce cas

Exemple (suite)

La solution optimale du problème relaxé sur $\mathbb{S}_{2,1}$ n'étant pas entière, on partitionne à nouveau $\mathbb{S}_{2,1}$:

- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière
  $\rightarrow x_1$ dans ce cas
- on applique des contraintes supplémentaires dues à la nature de la variable

Programmation linéaire en nombres entiers | Exemple complet

Exemple (suite)

La solution optimale du problème relaxé sur $\mathbb{S}_{2,1}$ n'étant pas entière, on partitionne à nouveau $\mathbb{S}_{2,1}$:

- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière
  $\rightarrow x_1$ dans ce cas
- on applique des contraintes supplémentaires dues à la nature de la variable
  $\rightarrow x_1 \leq 4$ ou $x_1 \geq 5$ (car dans la solution optimale du problème relaxé sur $\mathbb{S}_{2,1}$, on a $4 < x_1^* < 5$)

Programmation linéaire en nombres entiers | Exemple complet

Exemple (suite)

La solution optimale du problème relaxé sur $\mathbb{S}_{2,1}$ n'étant pas entière, on partitionne à nouveau $\mathbb{S}_{2,1}$:

- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière
  $\rightarrow x_1$ dans ce cas
- on applique des contraintes supplémentaires dues à la nature de la variable
  $\rightarrow x_1 \leq 4$ ou $x_1 \geq 5$ (car dans la solution optimale du problème relaxé sur $\mathbb{S}_{2,1}$, on a $4 < x_1^* < 5$)

On a séparé à nouveau le problème ("branch") et on obtient ainsi deux nouveaux sous-problèmes :

Programmation linéaire en nombres entiers | Exemple complet

Exemple (suite)

La solution optimale du problème relaxé sur $\mathbb{S}_{2,1}$ n'étant pas entière, on partitionne à nouveau $\mathbb{S}_{2,1}$:

- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière
  $\rightarrow x_1$ dans ce cas
- on applique des contraintes supplémentaires dues à la nature de la variable
  $\rightarrow x_1 \leq 4$ ou $x_1 \geq 5$ (car dans la solution optimale du problème relaxé sur $\mathbb{S}_{2,1}$, on a $4 < x_1^* < 5$)

On a séparé à nouveau le problème ("branch") et on obtient ainsi deux nouveaux sous-problèmes :

- $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,1,1}$ avec $\mathbb{S}_{2,1}$ défini par ($\mathbb{S}_{2,1}$ **et** $x_1 \leq 4$)

Exemple (suite)

La solution optimale du problème relaxé sur $\mathbb{S}_{2,1}$ n'étant pas entière, on partitionne à nouveau $\mathbb{S}_{2,1}$:

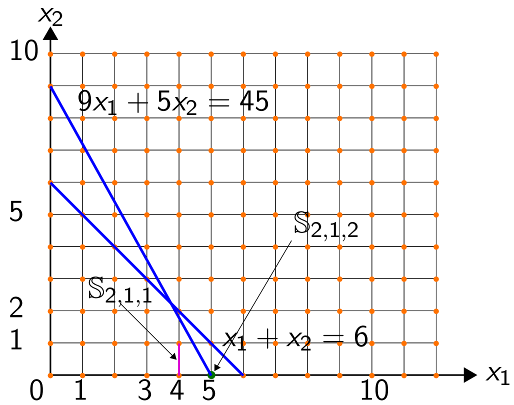
- on choisit arbitrairement une variable dont la valeur dans la solution optimale du problème relaxé est non entière
  $\rightarrow x_1$ dans ce cas
- on applique des contraintes supplémentaires dues à la nature de la variable
  $\rightarrow x_1 \leq 4$ ou $x_1 \geq 5$ (car dans la solution optimale du problème relaxé sur $\mathbb{S}_{2,1}$, on a $4 < x_1^* < 5$)

On a séparé à nouveau le problème ("branch") et on obtient ainsi deux nouveaux sous-problèmes :

- $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,1,1}$ avec $\mathbb{S}_{2,1}$ défini par ($\mathbb{S}_{2,1}$ **et** $x_1 \leq 4$)
- $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,1,2}$ avec $\mathbb{S}_{2,1}$ défini par ($\mathbb{S}_{2,1}$ **et** $x_1 \geq 5$)

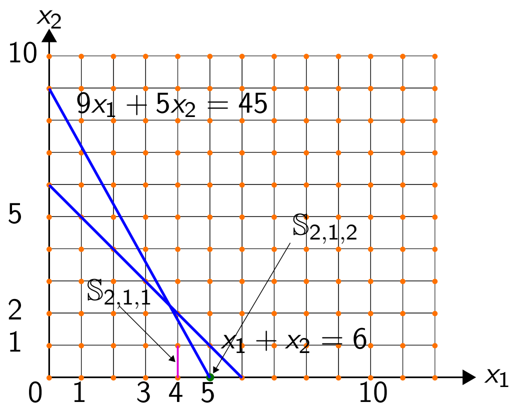
Exemple (suite)

A l'issue de la séparation de $\mathbb{S}_{2,1}$, on a deux nouveaux sous-problèmes dont les représentations graphiques sont les suivantes :



Exemple (suite)

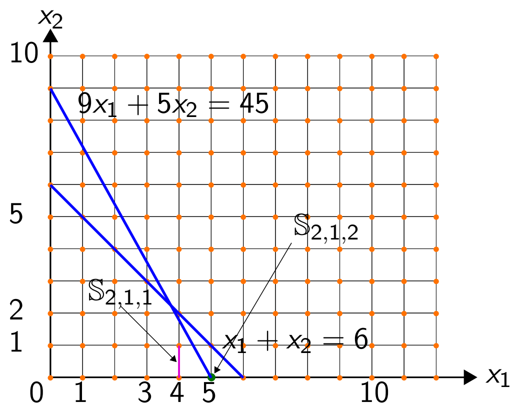
A l'issue de la séparation de $\mathbb{S}_{2,1}$, on a deux nouveaux sous-problèmes dont les représentations graphiques sont les suivantes :



- On remarque qu'on a éliminé la solution $\mathbf{X}^*_{cont,\mathbb{S}_{2,1}}$

Exemple (suite)

A l'issue de la séparation de $\mathbb{S}_{2,1}$, on a deux nouveaux sous-problèmes dont les représentations graphiques sont les suivantes :



- On remarque qu'on a éliminé la solution $\mathbf{X}^*_{cont,\mathbb{S}_{2,1}}$
- On peut résoudre les nouveaux sous-problèmes en considérant leurs relaxations linéaires

Exemple (suite)

On résoud dans un premier temps la relaxation linéaire du sous-problème $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,1,1}$ et $\mathbf{X} \in \mathbb{R}^2$.

Exemple (suite)

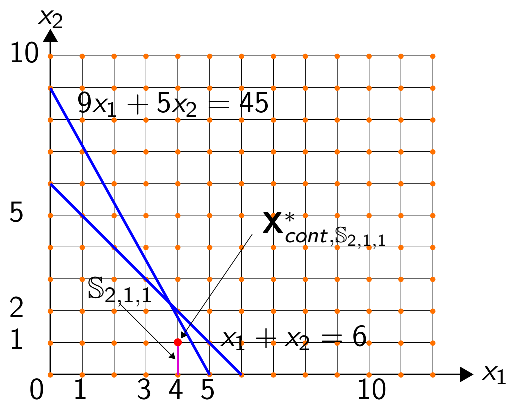
On résoud dans un premier temps la relaxation linéaire du sous-problème $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,1,1}$ et $\mathbf{X} \in \mathbb{R}^2$.

On obtient $z^*_{cont,\mathbb{S}_{2,1}} = 37$ et $\mathbf{X}^*_{cont,\mathbb{S}_{2,1,1}} = \begin{pmatrix} 4 \\ 1 \end{pmatrix} = \mathbf{X}^*_{ent,\mathbb{S}_{2,1,1}}$.

Exemple (suite)

On résoud dans un premier temps la relaxation linéaire du sous-problème $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,1,1}$ et $\mathbf{X} \in \mathbb{R}^2$.

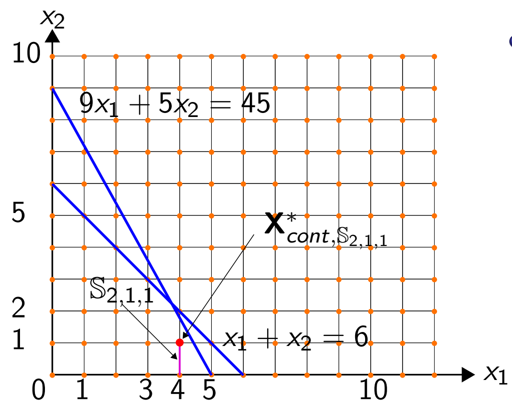
On obtient $z^*_{cont,\mathbb{S}_{2,1}} = 37$ et $\mathbf{X}^*_{cont,\mathbb{S}_{2,1,1}} = \begin{pmatrix} 4 \\ 1 \end{pmatrix} = \mathbf{X}^*_{ent,\mathbb{S}_{2,1,1}}$.



Exemple (suite)

On résoud dans un premier temps la relaxation linéaire du sous-problème $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,1,1}$ et $\mathbf{X} \in \mathbb{R}^2$.

On obtient $z^*_{cont,\mathbb{S}_{2,1}} = 37$ et $\mathbf{X}^*_{cont,\mathbb{S}_{2,1,1}} = \begin{pmatrix} 4 \\ 1 \end{pmatrix} = \mathbf{X}^*_{ent,\mathbb{S}_{2,1,1}}$.



- **La solution est entière** : c'est une **borne inférieure de l'optimum entier**

Exemple (suite)

On résoud maintenant la relaxation linéaire du sous-problème max $\mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,1,2}$ et $\mathbf{X} \in \mathbb{R}^2$.

Exemple (suite)

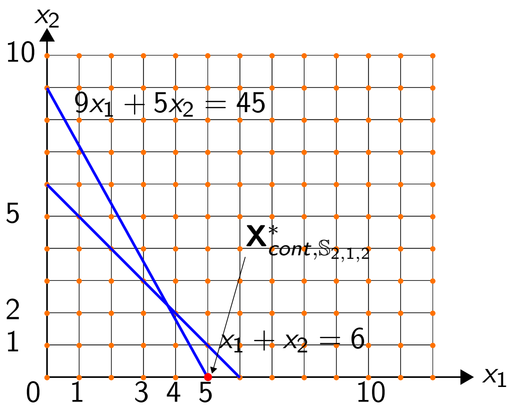
On résoud maintenant la relaxation linéaire du sous-problème max $\mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,1,2}$ et $\mathbf{X} \in \mathbb{R}^2$.

On obtient $z^*_{cont,\mathbb{S}_{2,1,2}} = 40$ et $\mathbf{X}^*_{cont,\mathbb{S}_{2,1,2}} = \begin{pmatrix} 5 \\ 0 \end{pmatrix} = \mathbf{X}^*_{ent,\mathbb{S}_{2,1,2}}$.

Exemple (suite)

On résoud maintenant la relaxation linéaire du sous-problème max $\mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,1,2}$ et $\mathbf{X} \in \mathbb{R}^2$.

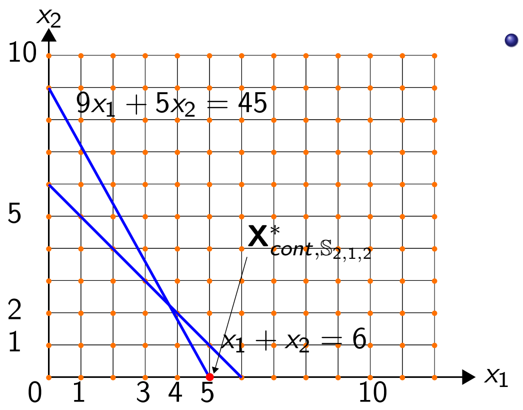
On obtient $z^*_{cont,\mathbb{S}_{2,1,2}} = 40$ et $\mathbf{X}^*_{cont,\mathbb{S}_{2,1,2}} = \begin{pmatrix} 5 \\ 0 \end{pmatrix} = \mathbf{X}^*_{ent,\mathbb{S}_{2,1,2}}$.



Exemple (suite)

On résoud maintenant la relaxation linéaire du sous-problème max $\mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,1,2}$ et $\mathbf{X} \in \mathbb{R}^2$.

On obtient $z^*_{cont,\mathbb{S}_{2,1,2}} = 40$ et $\mathbf{X}^*_{cont,\mathbb{S}_{2,1,2}} = \begin{pmatrix} 5 \\ 0 \end{pmatrix} = \mathbf{X}^*_{ent,\mathbb{S}_{2,1,2}}$.



- **La solution est entière** : c'est une **meilleure borne inférieure** de l'optimum entier : $z^*_{cont,\mathbb{S}_{2,1,2}} > z^*_{cont,\mathbb{S}_{2,1,1}}$

Exemple (suite)

- On est descendu en profondeur en considérant initialement le sous-problème $\mathbb{S}_2$. Aprés deux séparations supplémentaires, il n'y a plus intérêt à séparer

Exemple (suite)

- On est descendu en profondeur en considérant initialement le sous-problème $\mathbb{S}_2$. Aprés deux séparations supplémentaires, il n'y a plus intérêt à séparer

- A l'issue de cette descente, on a déterminé une solution réalisable en nombre entiers $\mathbf{X}^*_{ent,\mathbb{S}_{2,1,2}} = \begin{pmatrix} 5 \\ 0 \end{pmatrix}$ conduisant à une valeur de la fonction objectif égale à $z^*_{cont,\mathbb{S}_{2,1,2}} = 40$

Exemple (suite)

- On est descendu en profondeur en considérant initialement le sous-problème $\mathbb{S}_2$. Aprés deux séparations supplémentaires, il n'y a plus intérêt à séparer

- A l'issue de cette descente, on a déterminé une solution réalisable en nombre entiers $\mathbf{X}^*_{ent,\mathbb{S}_{2,1,2}} = \begin{pmatrix} 5 \\ 0 \end{pmatrix}$ conduisant à une valeur de la fonction objectif égale à $z^*_{cont,\mathbb{S}_{2,1,2}} = 40$

- Ce n'est pas fini ! Il faut descendre en profondeur en considérant désormais le sous-problème $\mathbb{S}_1$ pour déterminer s'il existe une meilleure solution que $\mathbf{X}^*_{ent,\mathbb{S}_{2,1,2}}$

Exemple (suite)

On revient donc sur la résolution du PL relaxé suivant $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_1$ et $\mathbf{X} \in \mathbb{R}^2$.

Exemple (suite)

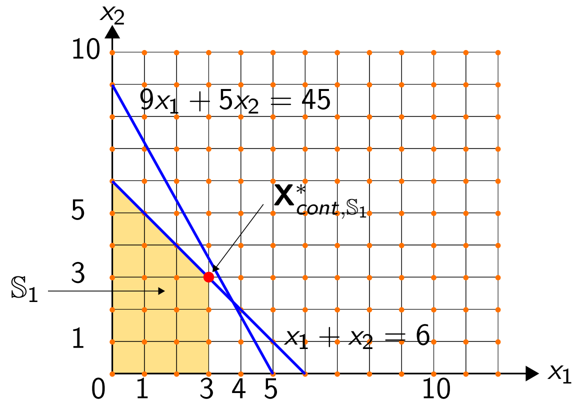
On revient donc sur la résolution du PL relaxé suivant $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_1$ et $\mathbf{X} \in \mathbb{R}^2$.

On obtient $z^*_{cont,\mathbb{S}_1} = 39$ et $\mathbf{X}^*_{cont,\mathbb{S}_1} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}$.

Exemple (suite)

On revient donc sur la résolution du PL relaxé suivant $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_1$ et $\mathbf{X} \in \mathbb{R}^2$.

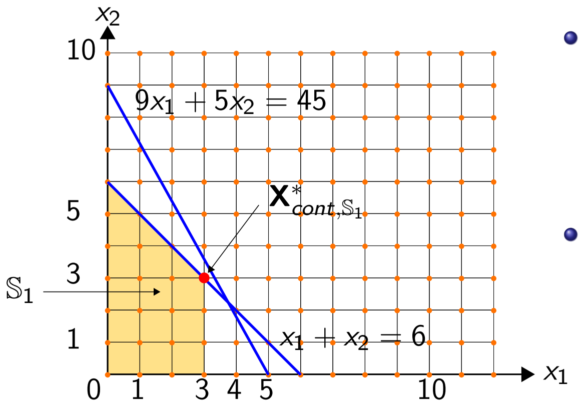
On obtient $z^*_{cont,\mathbb{S}_1} = 39$ et $\mathbf{X}^*_{cont,\mathbb{S}_1} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}$.



Exemple (suite)

On revient donc sur la résolution du PL relaxé suivant $\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_1$ et $\mathbf{X} \in \mathbb{R}^2$.

On obtient $z^*_{cont,\mathbb{S}_1} = 39$ et $\mathbf{X}^*_{cont,\mathbb{S}_1} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}$.



- La solution trouvée est une borne supérieure de la solution optimale en nombres entiers sur le domaine considéré
- Cette valeur est inférieure à la borne inférieure de la solution entière courante : **on ne descend pas plus bas ! ("Bound")**

Arboresence

$$\max \mathbf{C}'\mathbf{X}$$
$$\text{s.l.c } \mathbf{X} \in \mathbb{S}$$
$$\begin{cases} z^* = 41.25 \\ \mathbf{X}^* = \begin{pmatrix} 3.75 \\ 2.25 \end{pmatrix} \end{cases}$$

Arboresence

$$\max \mathbf{C}'\mathbf{X}$$
$$\text{s.l.c } \mathbf{X} \in \mathbb{S}$$
$$x_1 \leq 3 \qquad \begin{cases} z^* = 41.25 \\ \mathbf{X}^* = \begin{pmatrix} 3.75 \\ 2.25 \end{pmatrix} \end{cases} \qquad x_1 \geq 4$$

Arboresence

$$\max \mathbf{C}'\mathbf{X}$$
$$\text{s.l.c } \mathbf{X} \in \mathbb{S}$$
$$x_1 \leq 3 \qquad \begin{cases} z^* = 41.25 \\ \mathbf{X}^* = \begin{pmatrix} 3.75 \\ 2.25 \end{pmatrix} \end{cases} \qquad x_1 \geq 4$$

$$\max \mathbf{C}'\mathbf{X}$$
$$\text{s.l.c } \mathbf{X} \in \mathbb{S}_2$$
$$\begin{cases} z^* = 41 \\ \mathbf{X}^* = \begin{pmatrix} 4 \\ 1.8 \end{pmatrix} \end{cases}$$

Arboresence

$$\max \mathbf{C}'\mathbf{X}$$
$$\text{s.l.c } \mathbf{X} \in \mathbb{S}$$
$$x_1 \leq 3 \qquad \begin{cases} z^* = 41.25 \\ \mathbf{X}^* = \begin{pmatrix} 3.75 \\ 2.25 \end{pmatrix} \end{cases} \qquad x_1 \geq 4$$

$$\max \mathbf{C}'\mathbf{X}$$
$$\text{s.l.c } \mathbf{X} \in \mathbb{S}_2$$
$$x_2 \leq 1 \qquad \begin{cases} z^* = 41 \\ \mathbf{X}^* = \begin{pmatrix} 4 \\ 1.8 \end{pmatrix} \end{cases} \qquad x_2 \geq 2$$

Arboresence

$$\max \mathbf{C}'\mathbf{X}$$
$$\text{s.l.c } \mathbf{X} \in \mathbb{S}$$
$$x_1 \leq 3 \qquad \begin{cases} z^* = 41.25 \\ \mathbf{X}^* = \begin{pmatrix} 3.75 \\ 2.25 \end{pmatrix} \end{cases} \qquad x_1 \geq 4$$

$$\max \mathbf{C}'\mathbf{X}$$
$$\text{s.l.c } \mathbf{X} \in \mathbb{S}_2$$
$$x_2 \leq 1 \qquad \begin{cases} z^* = 41 \\ \mathbf{X}^* = \begin{pmatrix} 4 \\ 1.8 \end{pmatrix} \end{cases} \qquad x_2 \geq 2$$

$$\max \mathbf{C}'\mathbf{X}$$
$$\text{s.l.c } \mathbf{X} \in \mathbb{S}_{2,2}$$
Non réalisable

Arboresence

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S} \quad \begin{cases} z^* = 41.25 \\ \mathbf{X}^* = \begin{pmatrix} 3.75 \\ 2.25 \end{pmatrix} \end{cases}$$

$x_1 \leq 3$ ↙ ↘ $x_1 \geq 4$

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S}_2 \quad \begin{cases} z^* = 41 \\ \mathbf{X}^* = \begin{pmatrix} 4 \\ 1.8 \end{pmatrix} \end{cases}$$

$x_2 \leq 1$ ↙ ↘ $x_2 \geq 2$

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S}_{2,1} \quad \begin{cases} z^* = 40.5 \\ \mathbf{X}^* = \begin{pmatrix} 4.44 \\ 1 \end{pmatrix} \end{cases}$$

$\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,2}$ Non réalisable

Arboresence

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S} \quad \begin{cases} z^* = 41.25 \\ \mathbf{X}^* = \begin{pmatrix} 3.75 \\ 2.25 \end{pmatrix} \end{cases}$$

$x_1 \leq 3$ ↙ ↘ $x_1 \geq 4$

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S}_2 \quad \begin{cases} z^* = 41 \\ \mathbf{X}^* = \begin{pmatrix} 4 \\ 1.8 \end{pmatrix} \end{cases}$$

$x_2 \leq 1$ ↙ ↘ $x_2 \geq 2$

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S}_{2,1} \quad \begin{cases} z^* = 40.5 \\ \mathbf{X}^* = \begin{pmatrix} 4.44 \\ 1 \end{pmatrix} \end{cases}$$

$x_1 \leq 4$ ↙ ↘ $x_1 \geq 5$

$\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,2}$ Non réalisable

Arboresence

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S} \quad \begin{cases} z^* = 41.25 \\ \mathbf{X}^* = \begin{pmatrix} 3.75 \\ 2.25 \end{pmatrix} \end{cases}$$

$x_1 \leq 3$ ↙ ↘ $x_1 \geq 4$

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S}_2 \quad \begin{cases} z^* = 41 \\ \mathbf{X}^* = \begin{pmatrix} 4 \\ 1.8 \end{pmatrix} \end{cases}$$

$x_2 \leq 1$ ↙ ↘ $x_2 \geq 2$

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S}_{2,1} \quad \begin{cases} z^* = 40.5 \\ \mathbf{X}^* = \begin{pmatrix} 4.44 \\ 1 \end{pmatrix} \end{cases}$$

$x_1 \leq 4$ ↙ ↘ $x_1 \geq 5$

$\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,2}$ Non réalisable

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S}_{2,1,1} \quad \begin{cases} z^* = 37 \\ \mathbf{X}^* = \begin{pmatrix} 4 \\ 1 \end{pmatrix} \end{cases}$$

Arboresence

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S} \quad \begin{cases} z^* = 41.25 \\ \mathbf{X}^* = \begin{pmatrix} 3.75 \\ 2.25 \end{pmatrix} \end{cases}$$

$x_1 \leq 3$ ↙ ↘ $x_1 \geq 4$

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S}_2 \quad \begin{cases} z^* = 41 \\ \mathbf{X}^* = \begin{pmatrix} 4 \\ 1.8 \end{pmatrix} \end{cases}$$

$x_2 \leq 1$ ↙ ↘ $x_2 \geq 2$

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S}_{2,1} \quad \begin{cases} z^* = 40.5 \\ \mathbf{X}^* = \begin{pmatrix} 4.44 \\ 1 \end{pmatrix} \end{cases}$$

$x_1 \leq 4$ ↙ ↘ $x_1 \geq 5$

$\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,2}$ Non réalisable

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S}_{2,1,1} \quad \begin{cases} z^* = 37 \\ \mathbf{X}^* = \begin{pmatrix} 4 \\ 1 \end{pmatrix} \end{cases}$$

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S}_{2,1,2} \quad \begin{cases} z^* = 40 \\ \mathbf{X}^* = \begin{pmatrix} 5 \\ 0 \end{pmatrix} \end{cases}$$

Arboresence

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S} \quad \begin{cases} z^* = 41.25 \\ \mathbf{X}^* = \begin{pmatrix} 3.75 \\ 2.25 \end{pmatrix} \end{cases}$$

$x_1 \leq 3$:

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S}_1 \quad \begin{cases} z^* = 39 \\ \mathbf{X}^* = \begin{pmatrix} 3 \\ 3 \end{pmatrix} \end{cases}$$

$x_1 \geq 4$:

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S}_2 \quad \begin{cases} z^* = 41 \\ \mathbf{X}^* = \begin{pmatrix} 4 \\ 1.8 \end{pmatrix} \end{cases}$$

$x_2 \leq 1$:

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S}_{2,1} \quad \begin{cases} z^* = 40.5 \\ \mathbf{X}^* = \begin{pmatrix} 4.44 \\ 1 \end{pmatrix} \end{cases}$$

$x_2 \geq 2$:

$\max \mathbf{C}'\mathbf{X}$ s.l.c $\mathbf{X} \in \mathbb{S}_{2,2}$ Non réalisable

$x_1 \leq 4$:

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S}_{2,1,1} \quad \begin{cases} z^* = 37 \\ \mathbf{X}^* = \begin{pmatrix} 4 \\ 1 \end{pmatrix} \end{cases}$$

$x_1 \geq 5$:

$$\max \mathbf{C}'\mathbf{X} \quad \text{s.l.c } \mathbf{X} \in \mathbb{S}_{2,1,2} \quad \begin{cases} z^* = 40 \\ \mathbf{X}^* = \begin{pmatrix} 5 \\ 0 \end{pmatrix} \end{cases}$$

Rappel du Sommaire

Label des points

- L’algorithme de séparation-évaluation labellise tous les points de la région réalisable, certains de manière explicite (les noeuds de l’arbre), et d’autres implicitement

Label des points

- L’algorithme de séparation-évaluation labellise tous les points de la région réalisable, certains de manière explicite (les noeuds de l’arbre), et d’autres implicitement

- Par exemple, le point $\mathbf{X} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$ est contenu dans $\mathbb{S}_1$ dont l’optimum est en $\mathbf{X}^*_{\mathbb{S}_1} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}$, il est donc inutile de l’évaluer

Label des points

- L'algorithme de séparation-évaluation labellise tous les points de la région réalisable, certains de manière explicite (les noeuds de l'arbre), et d'autres implicitement
- Par exemple, le point $\mathbf{X} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$ est contenu dans $\mathbb{S}_1$ dont l'optimum est en $\mathbf{X}^*_{\mathbb{S}_1} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}$, il est donc inutile de l'évaluer
- L'algorithme divise pour régner, et converge nécessairement du au fait qu'il élimine toujours des points à chaque étape de séparation, alors que ceux-ci sont en nombre fini

Parcours de l'arbre

- On peut résoudre les problèmes de séparation-évaluation soit en largeur d'abord (on évalue tous les sous-problèmes à un niveau avant de séparer au niveau inférieur)

Programmation linéaire en nombres entiers Remarques

Parcours de l'arbre

- On peut résoudre les problèmes de séparation-évaluation soit en largeur d'abord (on évalue tous les sous-problèmes à un niveau avant de séparer au niveau inférieur)
- Ou alors en profondeur d'abord comme dans l'exemple

J. Ah-Pine (Univ-Lyon 2) Théorie des graphes et Optimisation M1 Informatique 2010-2011 / 173 - 173

Programmation linéaire en nombres entiers Remarques

Parcours de l'arbre

- On peut résoudre les problèmes de séparation-évaluation soit en largeur d'abord (on évalue tous les sous-problèmes à un niveau avant de séparer au niveau inférieur)
- Ou alors en profondeur d'abord comme dans l'exemple
- La pratique montre que l'exploration en profondeur d'abord fonctionne mieux, en particulier avec une bonne fonction d'estimation comme la relaxation du PL : on divise pour régner plutôt que d'explorer systématiquement