Programmation Lineaire

Nicolas Bousquet

November 28, 2024

Jusqu'à présent dans els cours, le but a été de présenter des techniques algorithmiques (gloutons, programmation dynamique, algorithmique probabiliste). Le problème étant qu'à chaque nouveau problème on doit:

- concevoir un nouvel algorithmique (ce qui nécessite des connaissances, des compétences, mais aussi du temps)
- prouver et/ou évaluer une nouvelle fois l'efficacité de l'algorithme (certains algo e.g. gloutons sont efficaces pour certains problèmes et pas pour d'autres).

Dans cette partie du cours, on va faire le chemin dans le sens inverse: on va donner des algorithmes généraux qui fonctionnent efficacement et le but va être de les utiliser pour résoudre ou approximer des solutions de problèmes plus difficiles. Alors que la partie principale consistait à concevoir des algorithmes dans la partie précédentes, ici le gros du travail portera sur la modélisation.

Une bonne première partie du cours portera d'ailleurs sur la modélisation, sous différentes formes, de problèmes issus de domaines assez variés.

1 Optimisation et modélisation

1.1 Généralités sur les problèmes d'optimisation

Dans les prochains cours (comme dans les précédents), on va considérer des problèmes d'optimisation. Un *problème d'optimisation* est un problème de la forme

$$\text{maximize } f(x) \text{ soumis à } x \in F$$

où

- $F \subseteq \mathbb{R}^n$ est appelé *l'ensemble possible*.
- $f : F \to \mathbb{R}$ est appelée la *fonction objectif*.
- x est un vecteur de *varibles de décision*.

Les problèmes d'optimisation apparaissent partout et vous en avez déjà vu des milleurs (que ce soit en optimisation discrète ou continue). Ainsi les problèmes de plus court chemin, de compression, d'optimisation continue, de production industrielle sont tous des problèmes d'optimisation.

Comme vous l'avez déjà vu également, quand on veut résoudre des problèmes d'optimisation, on va parfois regarder quelles sont les propriétés vérifiées par notre problème. En particulier, quand on résout un problème on se rend parfois compte que sa résolution permet en fait de résoudre bien plus que le problème lui même parce qu'on sait résoudre des problèmes bien plus génériques. Cette remarque, à priori inutile sauf pour des vues théoriques à première vue (pourquoi résoudre un problème que je ne me suis pas posé), est en fait importante. En effet:

- Il arrive que les problèmes que l'on ait à résoudre soient légèrement modifiée dans le temps, la méthode que l'on a conçu fonctionne-t-elle toujours?

- Si je suis capable de concevoir une méthode générique pour résoudre des problèmes alors il y a de bonne chance que je puisse résoudre de nombreux problèmes qui pourraient intéresser d'autres industriels.

Notons que si on remplace f par $-f$ on a un problème de minimisation plutot que de maximisation. Donc maximiser $f(x)$ soumis à $x \in F$ correspond à minimiser $-f(x)$ soumis à $x \in F$. Quand F est vide, on dit que le problème n'est *pas réalisable*. Dans de nombreux cas trouver l'existence d'une solution est aussi difficile que de trouver une solution optimale alors que dans d'autres, l'existence d'une solution est triviale. Cela dépend beaucoup du problème qu'on tente de résoudre des contraintes associées.

Quand un vecteur x satisfait toutes les contraintes, on dit qu'il est *réalisable*. Un vecteure x est une *solution optimale* si $f(x)$ est maximisé sur l'ensemble des solutions réalisables. On dit que $f(x)$ est *la valeur optimale*. Quand le problème n'est pas fini (l'ensemble réalisable n'est pas fini) il peut arriver que la valeur optimale soit infinie si, pour tout C, il existe $x \in F$ tel que $f(x) > C$. On note alors $+\infty$ la valeur optimale. Dans la cas d'un problème de minimisation on peut avoir une valeur optimale de $-\infty$.

Programmes linéaires, quadratiques et au-delà Supposons qu'on ait un problème d'optimisation de la forme

$$\max f(x)$$
$$F = \{x \in \mathbb{R}^n \text{ tel que } \forall j \leq n, x_j \geq 0 \text{ et } \forall i \leq m, f_i(x) \leq b_i\}$$

Si la fonction f ainsi que toutes les fonctions f_i sont linéaires alors on dit que le problème d'optimisation est un *Programme Lineaire* (ou PL). En d'autres termes, un PL est un problème d'optimisation d'une fonction linéaire sous des contraintes elles-même linéaires. Si au moins une de ces fonctions n'est pas linéaire il s'agit d'un problème d'optimisation non linéaire.

Quand une fonction f_i est linéaire, alors on peut écrire $f_i(x)$ comme $f_i(x) = \sum_{j \leq n} a_{i,j} x_i$ où les $a_{i,j}$ sont des réels et n est la tailel de x (i.e. la taille de la dimension du problème). On peut donc voir chaque f_i comme le produit scale entre un vecteur a_i de taille n et le vecteur de variables x. Notez également que les contraintes de positivité sont des contraintes linéaires (avec le vecteur a qui est nul sauf sur une coordonnée).

On denote par A la matrice où la i-ème ligne est le vecteur a. Autrement dit chaque ligne correspond au vecteur a_i. La matrice A est appelée *matrice de contraintes*. On note b le vecteur $(b_1, \ldots, b_m)$. Le vecteur b est appelé vecteur de contraintes. On peut similairement définir un vecteur c appelé *vecteur objectif* qui correspond aux coefficients de la fonction $f(x)$ que l'on optimise. Notez que m et n ne sont pas nécessairement les mêmes. n est la taille de l'espace sur lesquels on tente d'optimiser notre fonction (le nombre de variables) alors que m correspond au nombre de contraintes. Finalement un programme linéaire peut être résumé de la façon suivante:

Soit A une matrice de taille $m \times n$, $c \in \mathbb{R}^n$ et $b \in \mathbb{R}^n$. Un programme linéaire est un problème d'optimisation de la forme:

$$\begin{aligned} \text{maximiser}_{x \in \mathbb{R}^n} c^T x \\ \text{soumis} \\ Ax \leq b \text{ et} \\ x_j \geq 0 \text{ pour tout } j \leq m \end{aligned}$$

Dans la suite A sera toujours la matrice de contraintes, b le vecteur de contraintes et c le vecteur objectif.

Si les fonctions ne sont pas linéaires, on peut quand même espérer certaines propriétés: sont-elles quadratiques? Si oui on peut adopter le même genre d'approche en ayant au lieu d'un problème de la forme $Ax \leq b$, on peut espérer avoir un problème bilinéaire de la forme $y^t Ax \leq b$. Si la matrice A est définie positive on parle de programmation semi-définie. Globalement, plus les fonctions deviennent générales, plus on perd de la structure, des informations et la capacité de résoudre nos problèmes !

Optimisation Discrète vs Optimisation Continue. La *Programmation Lineaire en Nombre Entiers* est une classe importante de problème d'optimisation où les fonctions f_i and f sont linéaires mais on optimise selon un vecteur x qui peut seulement prendre des valeurs entières (et pas des valeurs réelles). Autrement dit, the domaine de notre problème devient un sous-ensemble de $\mathbb{Z}^n$ plutot qu'un sous ensemble de $\mathbb{R}^n$. Plus formellement c'est un problème de la forme:

$$\text{maximize }_{x \in \mathbb{Z}^n} c^T x \text{ subject to } x \in \mathbb{R}^n \text{ such that } Ax \leq b \text{ and } x_j \geq 0 \text{ for every } j$$

Notez que l'optimum d'un problème de maximisation est toujours plus grand on considère une optimisation en nombre réel plutot qu'en nombre entiers (une solution réelle étant entière). Et l'inverse est vrai pour les problèmes de minimisation. On peut se demander si les valeurs optimales sont toujours les mêmes. La réponse est sans surprise non. On verra par la suite que la situation peut en fait être particulièrement mauvaise quand on passe d'un problème d'optimisation réel à un problème d'optimisation entier.

Si certaines variables sont réelles et d'autres sont entières, on parle de *Programme Linéaire Mixte* (MILP). Dans le cadre de ce cours on abordera simplement les programme linéaires en nombre réels et ceux en nombre entiers mais on ne parlera de techniques particulières pour résoudre les MILP.

Remark 1. Soit P un programme linéaire où on tente de minimiser une fonction:

- La valeur de la solution optimale réelle au plus égale à la valeur de la solution optimale entière.
- L'écart entre la valeur de la solution optimale réelle et entière peut être arbitrairement grand.

1.2 Modélisation comme PL(NE)

On va voir la modélisation de plusieurs problèmes comme des problèmes de Programmation Linéaire (mais pas que):

- Surveillance dans les réseaux.
- Ordonnancement
- Mix de production industrielle.

Mix de production industrielle L'entreprise 'Dovetail' produit deux types d'allumettes, les longues et les courtes. L'entreprise fait un profit de 300\$ par lot de $100,000$ boites de grandes allumettes et 200\$ pour $100,000$ boîtes de petites allumettes. L'entreprise a une machine qui lui permet d'empaqueter grandes et petite allumettes qui peut, au total, empaqueter 9 ($\times 100,000$) boites par an. Pour la production d'allumettes, il faut du bois, $3n^3$ pour $100,000$ boites de grandes allumettes et $1m^3$ pour les petites. Au total, l'entreprise a en stock 18 mètres cube de bois. De plus elle a 7 ($\times 100,000$) boites (vides) de grandes alumettes et 6 ($\times 100,000$) de petites. Le but est maximiser le profit l'année suivante.

Quand on veut représenter le problème comme un PL, on doit d'abord trouver les variables. C'est souvent "ce qu'on veut compter". Ici le nombre de petites et grandes boites produites.

- x_1 est le nombre ($\times 100,000$) de grandes boites produites.
- x_2 is the number ($\times 100,000$) de petites boites produites.

Comme le but est de maximiser le produit, on désire:

$$\max_{(x_1,x_2) \in \mathbb{R}^2} 300 \cdot x_1 + 200 \cdot x_2.$$

Maximiser cette fonction est equivalent à maximiser la même fonction où on multiplie les coefficients par le même $\lambda > 0$. Donc on veut maximiser:

$$d = \max_{(x_1,x_2) \in \mathbb{R}^2} 3 \cdot x_1 + 2 \cdot x_2.$$

Maintenant, les contraintes sont: La contrainte de bois

$$3 \cdot x_1 + x_2 \leq 18. \tag{1}$$

La contrainte machine

$$x_1 + x_2 \leq 9. \tag{2}$$

Les contraintes sur les quantités de grandes et petites boites:

$$x_1 \leq 7. \tag{3}$$

$$x_2 \leq 6. \tag{4}$$

Finalement, le nombre de boites est positif donc:

$$x_1, x_2 \geq 0$$

Remarques.

- Le programme est ici vraiment simpliste. Dans la "vraie" vie, il faut imaginer des millers de varialbes et des millions de contraintes. On sait résoudre les PL en nombre réel jusqu'à des ordres de grandeurs qui vont au million de contraintes. Pour les PLNE on s'arrete à la dizaine de milliers seulement ! (Avec beaucoup d'effort !)
- Notez qu'ici on n'a pas dit si les variables étaient entières ou réelles. En fait, on va supposer qu'elles sont réelles. C'est souvent le cas pour des problèmes de mix industriel où la quantité optimale à produire est de l'ordre de milliers voir de millions d'unités. Arrondir la solution optimale à l'entier inférieur laisse en général un probleme satisfiable et une solution "quasi-optimale". Ce n'est pas le cas pour d'autres problèmes, en particulier les problèmes de décision (ie. que l'on peut representer avec des variables dans $\{0, 1\}$) puisqu'arrondir toutes les variables à 0 si elles ne valent pas 1 donne en général une solution désastreuse !

Surveillance de réseaux Une entreprise désire surveiller les liens de son réseaux en installant des caméras aux différentes intersection. Le but est de placer des caméras "360 degrés" aux intersections de façons à ce que toutes les rues puissent être surveillées.

Le problème peut se modéliser à l'aide de graphe. Il devient alors le problème du VERTEX COVER (ou couverture d'arêtes par les sommets en français) minimum. Un *vertex cover* est un sous ensemble de sommets X du graphe tel que toute arête a au moins une de ses extrémités dans X. Le problème VERTEX COVER consiste à trouver un vertex cover de taille minimum.

On peut modéliser ce problème comme un PLNE de la façon suivante:

$$\min \sum_{v \in V} x_v$$

subject to

$$x_u + x_v \geq 1 \text{ for all } e = (u, v) \in E$$

$$x_u \in \{0, 1\}$$

Notons que le problème VERTEX COVER est NP-complet. Par conséquent l'optimisation d'un PLNE est NP-difficile.

Ordonnancement Supposons qu'on ait une machine M qui ne peut effectuer qu'une seule tâche à la fois. Et on a également un ensemble de tâches $T_1, \ldots, T_m$ telle que chaque tâche doit être effectuée sur un certain intervalle $[a_i, b_i]$ pour tout $i \leq m$. Les tâches ne sont pas fractionnables et doivent être effectuées pendant cet intervalle de temps précisement. Le but est de maximiser le nombre de tâches effectués par la machine. C'est un problème qui apparait naturellemen en ordonnancement ainsi que sa contrepartie "coloration" qui consiste à essayer de minimiser le nombre de machines nécessaires pour effectuer toutes le tâches. De nombreuses variantes de ce problèmes existent en ordonnancement et sont plus ou moins difficiles.

Modélisation par graphe de conflits. On crée un sommet par tâches. On met une arête entre deux sommets si les intervalles associés aux deux tâches s'intersectent. Autrement dit, on rajoute une arête dans le graphe de conflit si les deux tâches s'excluent mutuellement. Un ensemble de tâche qui peut être effectué est alors un ensemble de sommets deux à deux non reliés dans le graphes. Dans de nombreux exemples, on modélise le problème à l'aide d'un graphe de contraintes et on essaie alors de trouver un ensemble de sommets du graphe qui satisfait les contraintes du problème.

Modélisation par PLNE. Ce problème peut aussi être modélisé par un PLNE.

$$\max \sum_{v \in V} x_v$$

subject to

$$x_u + x_v \leq 1 \text{ for all } e = (u,v) \in E$$
$$x_u \in \{0,1\}$$

où E représente les arêtes du graphe de conflits.

2 Résolution de Programme linéaire (en nombres réels)

2.1 Réécriture des programmes linéaires

Supposons qu'on se donne un PL sous forme

$$\begin{aligned} \max c^t x \\ \text{soumis à } Ax \leq b \\ x \geq 0 \end{aligned} \tag{5}$$

Alors on peut le transformer en un PL où on remplace les inégalités par des égalités. De la façon suivante:
Contraintes d'inégalité. Supposons qu'on ait une inégalité de la forme:

$$\sum_{j=1}^{m} x_{ij} \leq s_i.$$

Alors on peut remplacer cette inégalité par une égalité en créant une variable supplémentaire y telle que $y \geq 0$ et :

$$\sum_{j=1}^{m} x_{ij} + y = s_i.$$

La variable y s'appelle alors *variable d'écart*. On remarquera, ce qui est important pour la suite que la variable d'écart est nulle si et seulement si la contrainte est *serrée* (c'est à dire que l'on a égalité) dans la formulation d'origine.

Question. Que faire avec des contraintes $\geq$ au lieu de $\leq$? Dans ce cas là, les variables s'appellent variables de surplus.
Que faire si une des variables n'est pas positive au départ (ie comment la transformer en une ou des variables positives?)

Illustration à notre exemple Dans notre exemple, le problème linéaire sous forme standard est le suivant:

$$\begin{array}{lllllll} 3 \cdot x_1 + x_2 & +x_3 & & & & =18 & \\ x_1 + x_2 & & +x_4 & & & =9 & \\ x_1 & & & +x_5 & & =7 & x_1, x_2, x_3, x_4, x_6, x_6 \geq 0 \\ x_2 & & & & +x_6 & =6 & \end{array}$$

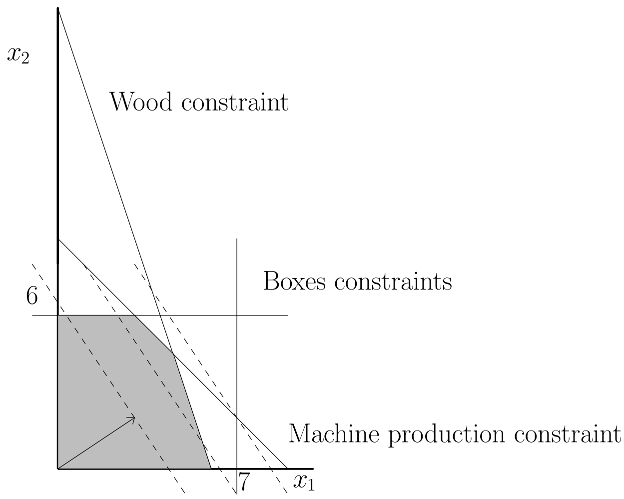

Figure 1: Représentation géométrique du programme linéaire. En pointillés, des lignes qui ont la même valeur pour la fonction objectif (les droites orthogonales au verteur objectif). En gris l'ensemble des points qui satisfont toutes les contraintes.

Autrement dit en posant le problème sous forme matricielle:

$$\begin{pmatrix} 3 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & & 00 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} 18 \\ 9 \\ 7 \\ 6 \end{pmatrix}$$

Remarques

- Notons que le PL parait plus compliqué que celui d'origine mais c'est seulement partiellement vrai dans le sens où la projection d'une solution de ce PL est une solution du PL d'origine. Similairement, quand on a une solution du PL d'origine, on peut l'étendre en une solution du nouveau PL. Les solutions sont en fait en bijection !
- Notons qu'une variable d'écart vaut 0 si et seulement si la contrainte associée est sérrée. Autrement dit, si n variables de ce nouveau PL sont égales à 0 (n étant le nombre de variable d'origine) alors ça définit un unique point à l'intersection de ces n contraintes (soit les contraintes associées aux variables d'écart, soit les contraintes $x_i \geq 0$ pour les variables x_i d'origine).

2.2 Résolution graphique

On peut résoudre géométriquement le PL en particulier quand la dimension est basse. Voilà la figure ci dessous qui modélise les contraintes.

Remark 2. Tout sommet dans un hyperplan normal au vecteur objectif a la même valeur pour la fonction objectif.

En conséquence, le but est de trouver l'hyperplan le "plus grand" qui intersecte l'ensemble réalisable du programme linéaire.

2.3 Points extremes et algorithme du Simplexe

On ne va pas donner en détail le fonctionnement de l'algorithme du Simplexe mais nous allons simplement expliquer son fonctionnement. Pour expliquer en détail sont fonctionnement il faudrait passer un ou deux cours à rappeler des notions d'algèbre linéaire et à inverser des matrices. Nous allons donc faire les choses différement et simplement donner l'intuition de cet algorithme.

Rappels d'algèbre linéaire. On rappelle quelques notions d'algèbre linéaire (sans aucune preuve). Soient $x_1, \ldots, x_m$ m vecteurs de $\mathbb{R}^n$ et $\lambda_1, \ldots, \lambda_m$ m scalaires (penser ici réels). On dit que $x = \sum_{i=1}^m \lambda_i x_i$ est une:

- *Combinaison linéaire* (de $x_1, \ldots, x_m$) si les λ_i sont quelconques.
- *Combinaison Conique* si les λ_i sont positifs ou nuls pour tout i.
- *Combinaison Convexe* si $\sum_{i=1}^m \lambda_i = 1$ et $\lambda_i \geq 0$ pour tout i.

(Dans la suite les λ représenteront toujours des scalaires). L'espace engendré par $X = \{x_1, \ldots, x_m\}$, dénoté par $Span(X)$, est l'ensemble des combinaisons linéaires des vecteurs de X. On dit que $x_1, \ldots, x_m$ est *linéairement indépendant* si $\sum_{i=1}^m \lambda_i x_i = 0$ implique $\lambda_i = 0$ pour tout i. De manière équivalent l'espace engendré par X est de *dimension* m.

Un *convexe* est un ensemble de points C tels que pour tout $x, y \in C$ et $\lambda \in [0, 1]$ alors $\lambda x + (1 - \lambda)y$ est dans C. La combinaison convexe d'un ensemble de points X est l'ensemble des points qui sont des combinaisons convexes de points de X. On peut montrer que $Conv(X)$ est le plus petit ensemble convexe qui contient X. (Exercice, montrez le !)

Un point extrême (ou sommet) d'un convexe C est un sommet qui n'est combinaison convexe d'aucun autre point de C. On utilisera la propriété suivante des convexes qui découle facilement d'anlyse de fonctions sur des ensembles convexes:

Theorem 3. *Soit C un ensemble convexe et f une fonction linéaire. Alors:*

- *Tout maximum local de f sur C est un maximum global.*
- *Le maximum (s'il n'est pas infini) est atteint sur un point extrême de C.*

3 Dualité

4 Introduction à la dualité

4.1 Definition

Etant donné un programme linéaire de la forme

$$\max c^t x$$

soumis à

$$Ax \leq b \text{ et } x \geq 0$$

Le *dual* de ce programme linéaire est le programme linéaire suivant:

$$\min b^t y$$

soumis à

$$y^t A \geq c \text{ et } y \geq 0$$

En général on appelle le problème linéaire d'origine le primal (noté (P)) et l'autre problème le dual (noté (D)).

Notez que les variables du dual, appelées *variables duales*, correspondent aux contraintes du primal. En effet au lieu de multiplier la matrice A on multiplie la matrice A^t. De manière similaire chaque contrainte du primal va correspondre à une variable du dual.

Exemple 1. Reprenons l'exemple de la fabrication d'allumettes

$$\max_{(x_1,x_2)\in\mathbb{R}^2} 3\cdot x_1 + 2\cdot x_2.$$

subject to

$$\begin{aligned} 3\cdot x_1 + x_2 \leq &\ 18 \\ x_1 + x_2 \leq &\ 9 \\ x_1 \leq &\ 7 \\ x_2 \leq &\ 6 \\ x_1, x_2 \geq &\ 0 \end{aligned}$$

The dual of this LP is:

$$\min_{(y_1,y_2,y_3,y_4)\in\mathbb{R}^4} 18\cdot y_1 + 9\cdot y_2 + 7y_3 + 6y_4.$$

subject to

$$\begin{aligned} 3y_1 + y_2 + y_3 \geq &\ 3 \\ y_1 + y_2 + y_4 \geq &\ 2 \\ y_1, y_2, y_3, y_4 \geq &\ 0 \end{aligned}$$

Interprétation économique du dual. Dans l'exemple de fabrication d'alumettes, la premiere contrainte correspond à la quantité de bois disponible pour l'entreprise. A la place d'utiliser son stock, l'entreprise pourrait décider de vendre son bois. Il y alors une question naturelle: quand est-il intéressant pour l'entreprise de vendre son bois? A quel prix est-ce intéressant d'accepter une offre pour ce bois? Notons y_1 le pris d'un mètre cube de bois. De la même façon, l'entreprise pourrait décider de vendre du temps de production à un prix y_2, des grandes boites à un prix y_3 et des petites boites à un prix y_4.

On peut trouver des inégalités qui nous disent à quel prix il n'ext pas intéressant de vendre sa production. En effet, avec 3 mètres cubes de bois, une unité de machine et 100.000 grandes boites, l'entreprise gagne 3. Autrement dit $3y_1 + y_2 + y_3 \geq 3$ autrement l'entreprise à plus intérêt à garder son stock et produire plutot que vendre le stock correspondant. De la même façon, on a $y_1 + y_2 + y_4 \geq 2$. On remarque que ces contraintes correspondent exactement aux contraintes du dual !

Supposons finalement que l'on se place du point de vue de l'acheteur des ressources. Le but est de minimiser le coût total d'achat tout en étant sûr que l'entreprise veuille bien le vendre. Autrmeent dit on veut minimiser $18y_1 + 9y_2 + 7y_3 + 6y_4$. Et pour être sûr que l'entreprise vende, on rajoute les contraintes $3y_1 + y_2 + y_3 \geq 3$ et $y_1 + y_2 + y_4 \geq 2$.

Exemple 2 - Vertex Cover et Couplage On a vu que le problème du vertex cover minimum dans un graphe pouvait s'écrire:

$$\min \sum_{v\in V} x_v$$

subject to

$$x_u + x_v \geq 1 \text{ for all } e = (u,v) \in E$$

$$x_u \in \{0,1\}$$

Remark 4. Montrer que la matrice de contrainte est matrice d'incidende du graphe avec E lignes et V colonnes et où sur chaque ligne on a exactement deux valeurs 1s aux deux extremités de l'arête.

On peut montrer que le problème dual du problème de vertex cover est le problème de couplage maximum. Un couplage est un ensemble d'arêtes dans un graphes qui sont deux à deux extrémités disjointes. Autrmeent dit:

$$\max \sum_{v\in V} y_e$$

subject to

$$\sum_{e|v \text{extremité de } e} y_e \leq 1 \text{ for all } v \in V$$

$$y_e \in \{0,1\}$$

4.2 Forme générale du dual

On a défini le programme linéaire dual quand le Programme Lineaire est sous forme canonique. Dans de nombreux cas, le problème n'est pas de cette forme. Nous avons vu qu'il etait toujours possible de mettre le PL sous forme canonique. On peut donc faire ceci et ensuite calculer le dual. Il existe une table plus simple pour calculer le dual si on veut. Il est résumé dans le tableau suivant:

$Primal$	$Dual$
max	min
Vector of constraint	Objective function
Objective function	Vector of constraint
Variables	Constraints
Constraints	Variables
Constraint $\leq$	Variable ≥ 0
Constraint $\geq$	Variable ≤ 0
Constraint $=$	Variable unconstrained
Variable ≥ 0	Constraint $\geq$
Variable ≤ 0	Constraint $\leq$
Variable unconstrained	Constraint $=$

On ne va pas prouver toutes ces équivalences pendant le cours mais une analyse détaillée permet de le prouver assez facilement.

Exercice 5.
- Montrer que le dual d'une variable positive ou nulle est une contrainte $\geq$.
- Montrer que le dual d'une variable négative ou nulle est une contrainte $\leq$.

5 Théorèmes de dualité

5.1 Théorèmes de dualité faibles et forts

Soit (P) un programme linéaire et (D) son dual:

$$(P) \quad \begin{aligned} & z_p = \max c^T x \\ & \text{such that} \\ & Ax \leq b \\ & x \geq 0 \end{aligned} \qquad\qquad (D) \quad \begin{aligned} z_d = \min b^T y \\ \text{such that} \\ A^T y \geq c \\ y \geq 0 \end{aligned}$$

Theorem 6 (Théorème faible de dualité). *Supposons qu'à la fois pour (P) et (D) le PL soit réalisable (il existe au moins une solution). Alors la valeur de n'importe quelle solution de (D) est plus grande ou égale à la valeur de n'importe quel solution de (P). En particulier on a*

$$z_p \leq z_d.$$

Proof. Soit x un point qui satisfait les contraintes de (P) et y un point qui satisfait les contraintes de (D). On a:

$$Ax \leq b \Rightarrow b^T y \geq (x^T A^T) y$$

comme $y \geq 0$. De plus on a

$$A^T y \geq c \Rightarrow (y^T A) x \geq c^T x$$

comme $x \geq 0$. Comme $(x^T A^T)y = (y^T A)x$ on obtient:

$$b^T y \geq c^T x$$

ce qui conclut la preuve. $\square$

En général la dualité faible se "voit" bien quand on considère deux problèmes qui sont duaux l'un de l'autre (comme pour le vertex cover et le couplage). En effet, on se rend compte que si on veut un vertex cover il faudra sélectionner au moins un sommet par arête.

Le théorère de dualité faible peut être étendu en un théorème de dualité fort:

Theorem 7 (Théorème fort de dualité)**.** *Supposons qu'à la fois pour (P) et (D) le PL soit réalisable (il existe au moins une solution). Si on note z_P la valeur optimale de (P) et z_d la valeur optimale de (D), on a:*

$$z_p \geq z_d.$$

Theorem 8. *Les théorèmes suivants sont des corollaires du théorème de dualité forte:*

- *Le flot maximum d'un graphe est égale à la coupe minimum.*
- *Le couplage maximum est égal à un vertex cover minimum dans les graphes bipartis.*

5.2 Application - Certificats d'optimalité

Considérons à nouveau le problème de production d'allumettes:

$$\max_{(x_1,x_2)\in\mathbb{R}^2} 3 \cdot x_1 + 2 \cdot x_2.$$

subject to

$$\begin{aligned} 3 \cdot x_1 + x_2 \leq&\ 18 \\ x_1 + x_2 \leq&\ 9 \\ x_1 \leq&\ 7 \\ x_2 \leq&\ 6 \\ x_1, x_2 \geq&\ 0 \end{aligned}$$

Its dual is the following LP.

$$\min_{(y_1,y_2,y_3,y_4)\in\mathbb{R}^4} 18 \cdot y_1 + 9 \cdot y_2 + 7y_3 + 6y_4.$$

subject to

$$\begin{aligned} 3y_1 + y_2 + y_3 \geq&\ 3 \\ y_1 + y_2 + y_4 \geq&\ 2 \\ y_1, y_2, y_3, y_4 \geq&\ 0 \end{aligned}$$

Quelqu'un nous assure que la solution $(3, 6)$ est optimale. Nous allons montrer que l'on peut vérifier qu'il dit vrai efficacement sans recalculer la solution optimale. Tout d'abord, on peut facilement vérifier que deux contraintes sont serrées pour cette valeur, à savoir $x_2 \leq 6$ et $x_1 + x_2 \leq 9$ et que toutes les contraintes sont satisfaites. C'est donc un candidat de solution optimale.

Utilisons le théorème des écarts complémentaires pour vérifier l'optimalité de la solution. Comme les premières et troisièmes contraintes ne sont pas serrées, les variables duales associées à ces variables devraient être égales à 0. Donc $y_1^* = y_3^* = 0$ dans une solution optimale du dual. De plus, comme x_t et x_c sont positives, les contraintes duales associées doivent être serrées. Donc on doit avoir:

$$\begin{aligned} y_2^* =&\ 3 \\ y_2^* + y_4^* =&\ 2 \end{aligned}$$

Donc une solution optimale du dual devrait être $(0, 3, 0, -1)$, qui ne satisfait pas les contraintes de positivité, ce qui donne une contradiction. Donc $(3, 6)$ n'est pas une solution optimale du primal !