Optimisation

Programmation linéaire

Louis-Claude Canon
louis-claude.canon@univ-fcomte.fr

Master 2 Informatique – Semestre 9

Plan

Plan

Problème politique

- Un politicien souhaite remporter une élection sur une circonscription.
- La circonscription comprend 3 zones : ville (100 000 électeurs), banlieues (200 000) et campagne (50 000).
- Pour être légitime, le politicien souhaite au moins la moitié des voix dans chaque zone.
- L'objectif est de dépenser le moins possible de frais en publicité.

Données du problème

Nombre de milliers d'électeurs gagnés pour 1000 € dépensés en publicité :

Stratégie	ville	banlieues	campagne
Apocalypse zombie	-2	5	3
Requins avec laser	8	2	-5
Autoroutes pour voitures volantes	0	0	10
Vote des dauphins	10	0	-2

Exemple de solution

On peut dépenser :

- 20 000 € pour la préparation à une apocalypse zombie,
- 0 € pour l'équipement de laser sur des requins,
- 4 000 € pour construire des autoroutes pour voitures volantes,
- 9 000 € pour permettre aux dauphins de voter.

Dans ce cas, on obtient :

- 50 000 électeurs en ville (sur 100 000),
- 100 000 électeurs en banlieues (sur 200 000),
- 82 000 électeurs en campagne (sur 50 000).

Le coût total serait de 33 000 euros.

Exemple de solution

On peut dépenser :

- 20 000 € pour la préparation à une apocalypse zombie,
- 0 € pour l'équipement de laser sur des requins,
- 4 000 € pour construire des autoroutes pour voitures volantes,
- 9 000 € pour permettre aux dauphins de voter.

Dans ce cas, on obtient :

- 50 000 électeurs en ville (sur 100 000),
- 100 000 électeurs en banlieues (sur 200 000),
- 82 000 électeurs en campagne (sur 50 000).

Le coût total serait de 33 000 euros. Est-ce optimal ?

Terminologie

- Il faut formuler mathématiquement le problème (on dit aussi le *modéliser*).
- On a 4 *variables* pour décider les sommes investies : une par stratégie. Ce sera x_1 à x_4 (e.g., x_1 étant le budget pour l'apocalypse zombie).
- On a des *contraintes* sur le nombre d'électeurs obtenus : une par zone de la circonscription. Par exemple pour la ville : $-2x_1 + 8x_2 + 0x_3 + 10x_4 \geq 50$.
- On définit l'*objectif* : minimiser la somme totale investie (ici, $x_1 + x_2 + x_3 + x_4$).

Exemple de programme linéaire

Le programme linéaire ainsi obtenu est :

$$
\begin{array}{lrcrcrcrcr}
\text{minimiser} & x_1 & + & x_2 & + & x_3 & + & x_4 & & \\
\text{sous les contraintes} & -2x_1 & + & 8x_2 & + & 0x_3 & + & 10x_4 & \geq & 50 \\
 & 5x_1 & + & 2x_2 & + & 0x_3 & + & 0x_4 & \geq & 100 \\
 & 3x_1 & - & 5x_2 & + & 10x_3 & - & 2x_4 & \geq & 25 \\
 & & & & & & & x_1, x_2, x_3, x_4 & \geq & 0
\end{array}
$$

Plan

Fonction linéaire

- Soit a_1, a_2, …, a_n des valeurs réelles.
- Soit x_1, x_2, …, x_n des variables réelles.
- Une *fonction linéaire* f est définie par :

$$f(x_1, x_2, \dots, x_n) = a_1x_1 + a_2x_2 + \dots + a_nx_n = \sum_{j=1}^{n} a_jx_j$$

- Exemple : $f(x_1, x_2) = 2x_1 - x_2$.
- Une *égalité linéaire* (avec b une valeur réelle) : $f(x_1, x_2, \dots, x_n) = b$.
- Les *inégalités linéaires* : $f(x_1, x_2, \dots, x_n) \leq b$ et $f(x_1, x_2, \dots, x_n) \geq b$.
- Une *contrainte linéaire* est soit une égalité, soit une inégalité linéaire.

Problème de programmation linéaire

- Un *problème de programmation linéaire* consiste à minimiser ou maximiser une fonction linéaire soumise à des contraintes linéaires.
- Soit n le nombre de variables et m le nombre de contraintes.
- On cherche les valeurs des variables x_1, x_2, ..., x_n pour :

$$\begin{aligned} \text{maximiser} \quad & \sum_{j=1}^{n} c_j x_j \\ \text{sous les contraintes} \quad & \sum_{j=1}^{n} a_{ij} x_j \leq b_i \text{ pour } i = 1, 2, \ldots, m \\ & x_j \geq 0 \text{ pour } j = 1, 2, \ldots, n \end{aligned}$$

- Les c_j sont les coefficients réels de la *fonction objectif*.
- Les a_{ij} et b_i sont les coefficients réels des *contraintes*.
- La dernière ligne représente les contraintes de non-négativité.

Forme standard

La *forme standard* utilise une forme plus compacte en utilisant une matrice A (qui contient les a_{ij}) et 3 vecteurs colonnes b (b_i), c (c_j) et x (x_j) :

$$\begin{array}{ll} \text{maximiser} & c^T x \\ \text{sous les contraintes} & Ax \leq b \\ & x \geq 0 \end{array}$$

Les bibliothèques de résolution s'appuient souvent sur cette forme.

Solutions

- Soit $\bar{x}$ un ensemble de valeurs réelles pour le vecteur x.
- Si les contraintes sont respectées, $\bar{x}$ est une *solution réalisable* et sa *valeur de l'objectif* est $c^T\bar{x}$.
- S'il n'existe pas de solution avec un meilleur objectif, on dit que c'est la *solution optimale* et qu'elle atteint la *valeur de l'objectif optimale*.

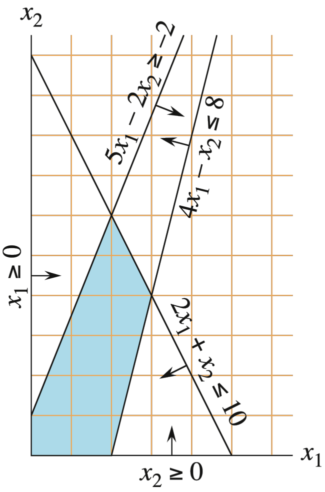
Résolution sur un exemple

On considère le problème de programmation linéaire suivant :

$$
\begin{array}{lrcrcr}
\text{maximiser} & x_1 & + & x_2 & & \\
\text{sous les contraintes} & 4x_1 & - & x_2 & \leq & 8 \\
 & 2x_1 & + & x_2 & \leq & 10 \\
 & 5x_1 & - & 2x_2 & \geq & -2 \\
 & & & x_1, x_2 & \geq & 0
\end{array}
$$

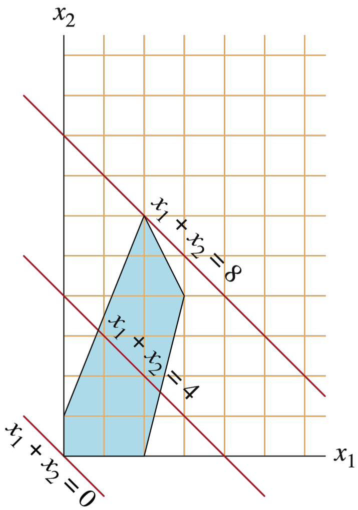
Représentation graphique (1/2)

- Les contraintes sont définies par la matrice $A = \left\{ \begin{array}{cc} 4 & -1 \\ 2 & 1 \\ -5 & 2 \end{array} \right\}$ et le vecteur $b = \{8, 10, 2\}$.
- Les 5 contraintes définissent un polygone, c'est-à-dire une région de réalisabilité (toute solution dans cette région est réalisable).



Représentation graphique (2/2)

- La fonction objectif est $f(x_1, x_2) = x_1 + x_2$ avec $c_1 = c_2 = 1$.
- La solution $\bar{x} = \{0, 0\}$ est réalisable et sa valeur de l'objectif est 0.
- La solution optimale est $\bar{x} = \{2, 6\}$ et la valeur de l'objectif optimale est 8.



Algorithme du simplexe

- L'algorithme du simplexe est très efficace en pratique pour résoudre les problèmes de programmation linéaire (bien que de complexité exponentielle en temps au pire cas).
- Il part d'une solution réalisable ($\bar{x} = 0$ par exemple) et se déplace de sommet en sommet sur les bords du polyèdre (polygone en dimension n) formant la région de réalisabilité.
- Il s'arrête sur la solution optimale.
- D'autres méthodes, de complexité en temps polynomiale, se déplacent à l'intérieur de la région de réalisabilité (méthodes de points intérieurs).
- En pratique, on s'appuie sur un solveur comme CPLEX ou COIN-OR LP qui peuvent traiter des centaines de milliers de variables et de contraintes.

Gestion des valeurs réelles

- En informatique, les valeurs réelles n'existent pas.
- Souvent, on utilise simplement des nombres flottants (simple ou double précision).
- C'est une approximation souvent suffisante mais qui est sensible à des problèmes de stabilité numérique.
- Il faut donc garder en tête la vérification du résultat.

Question

Lequel de ces programmes linéaires n'admet pas de solution réalisable ?

1.
$$\begin{array}{lrcrcr} \text{minimiser} & -2x_1 & + & 3x_2 & & \\ \text{contraintes} & x_1 & + & x_2 & = & 7 \\ & x_1 & - & 2x_2 & \leq & 4 \\ & & & x_1 & \geq & 0 \end{array}$$

2.
$$\begin{array}{lrcrcr} \text{minimiser} & 2x_1 & + & x_2 & & \\ \text{contraintes} & x_1 & - & x_2 & = & 7 \\ & 3x_1 & & & \geq & 24 \\ & & & x_2 & \geq & 0 \end{array}$$

3.
$$\begin{array}{lrcrcr} \text{minimiser} & 3x_1 & - & 2x_2 & & \\ \text{contraintes} & x_1 & + & x_2 & \leq & 2 \\ & -2x_1 & - & 2x_2 & \leq & -10 \\ & & & x_1, x_2 & \geq & 0 \end{array}$$

4.
$$\begin{array}{lrcrcr} \text{minimiser} & x_1 & - & x_2 & & \\ \text{contraintes} & -2x_1 & + & x_2 & \leq & -1 \\ & -x_1 & - & 2x_2 & \leq & -2 \\ & & & x_1, x_2 & \geq & 0 \end{array}$$

Question

Lequel de ces programmes linéaires n'admet pas de solution réalisable ?

1.

$$\begin{array}{lrcrcr} \text{minimiser} & -2x_1 & + & 3x_2 & & \\ \text{contraintes} & x_1 & + & x_2 & = & 7 \\ & x_1 & - & 2x_2 & \leq & 4 \\ & & & x_1 & \geq & 0 \end{array}$$

2.

$$\begin{array}{lrcrcr} \text{minimiser} & 2x_1 & + & x_2 & & \\ \text{contraintes} & x_1 & - & x_2 & = & 7 \\ & 3x_1 & & & \geq & 24 \\ & & & x_2 & \geq & 0 \end{array}$$

3. ✓

$$\begin{array}{lrcrcr} \text{minimiser} & 3x_1 & - & 2x_2 & & \\ \text{contraintes} & x_1 & + & x_2 & \leq & 2 \\ & -2x_1 & - & 2x_2 & \leq & -10 \\ & & & x_1, x_2 & \geq & 0 \end{array}$$

4.

$$\begin{array}{lrcrcr} \text{minimiser} & x_1 & - & x_2 & & \\ \text{contraintes} & -2x_1 & + & x_2 & \leq & -1 \\ & -x_1 & - & 2x_2 & \leq & -2 \\ & & & x_1, x_2 & \geq & 0 \end{array}$$

Plan

Problèmes concernés

- La *recherche opérationnelle* est une discipline qui se base sur la formulation mathématique des problèmes d'optimisation.
- Exemples : gestion d'emploi du temps avec contraintes, organisation de chaînes logistiques, affectation de ressources, etc.
- On peut notamment formuler des problèmes de graphes et de flots comme des programmes linéaires.

Notation ensembliste

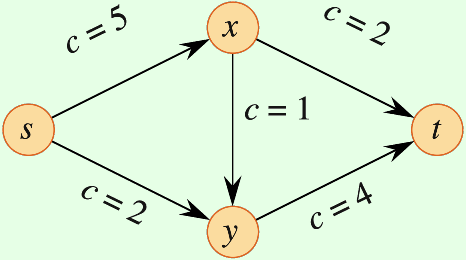
Notion mathématique

$V = \{u, v, w\}$	Ensemble V contenant 3 éléments : u, v et w.
$v \in V$	L'appartenance à un ensemble se note $\in$.
$\lvert V \rvert$	Nombre d'éléments, ou cardinalité, de l'ensemble V (ici nombre de sommets). Pour les complexités cependant, on simplifiera $O(\lvert V \rvert) = O(V)$ et $O(\lvert E \rvert) = O(E)$.
$\emptyset$	Ensemble vide.
$A \setminus B$	Opération de différence sur les ensembles (ce qui est dans A mais pas dans B).

Flot maximum : description

Notion algorithmique

- Un graphe orienté $G = (V, E)$ avec des capacités sur les arcs $c : V \times V \to \mathbb{R}$.
- Un sommet source $s \in V$ et un puits $t \in V$.
- Soit $f : V \times V \to \mathbb{R}$ le flot de chaque arc.
- On cherche à maximiser le flot total provenant de la source.
- Le flot doit respecter les contraintes de capacité et se conserver.



Flot maximum : programme linéaire

$$
\begin{aligned}
\text{maximiser} \quad & \sum_{v\in V} f_{sv} - \sum_{v\in V} f_{vs} \\
\text{sous les contraintes} \quad & f_{uv} \leq c(u,v) \quad \text{pour chaque } u,v \in V \\
& \sum_{v\in V} f_{uv} = \sum_{v\in V} f_{vu} \quad \text{pour chaque } u \in V \setminus \{s,t\} \\
& f_{uv} \geq 0 \quad \text{pour chaque } u,v \in V
\end{aligned}
$$

- La première contrainte garantit les capacités.
- La seconde contrainte concerne la conservation du flot.
- Nombre de variables : $|V|^2$.
- Nombre de contraintes : $2|V|^2 + |V| - 2$.

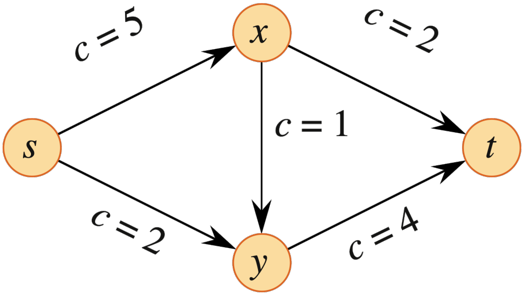
Flot maximum : programme linéaire

$$
\begin{array}{lrll}
\text{maximiser} & \displaystyle\sum_{v\in V} f_{sv} - \sum_{v\in V} f_{vs} & & \\
\text{sous les contraintes} & f_{uv} \leq c(u,v) & \text{pour chaque } u, v \in V \\
& \displaystyle\sum_{v\in V} f_{uv} = \sum_{v\in V} f_{vu} & \text{pour chaque } u \in V \setminus \{s,t\} \\
& f_{uv} \geq 0 & \text{pour chaque } u, v \in V
\end{array}
$$

- La première contrainte garantit les capacités.
- La seconde contrainte concerne la conservation du flot.
- Nombre de variables : $|V|^2$.
- Nombre de contraintes : $2|V|^2 + |V| - 2$.

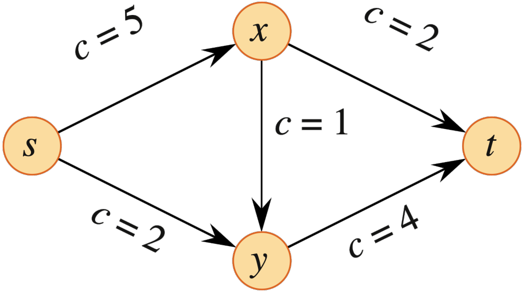
Flot maximum : exemple

$$
\begin{aligned}
\text{maximiser} \quad & f_{sx} + f_{sy} \\
\text{sous les contraintes} \quad & 0 \leq f_{sx} \leq 5 \\
& 0 \leq f_{sy} \leq 2 \\
& 0 \leq f_{xy} \leq 1 \\
& 0 \leq f_{xt} \leq 2 \\
& 0 \leq f_{yt} \leq 4 \\
& f_{sx} = f_{xy} + f_{xt} \\
& f_{sy} + f_{xy} = f_{yt}
\end{aligned}
$$



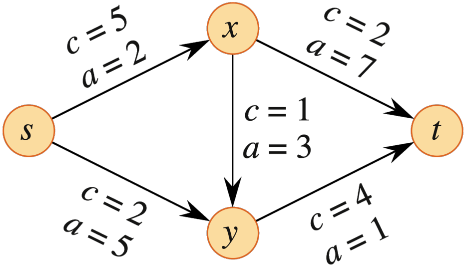
Flot maximum : solution

- $\bar{f_{sx}} = 3.$
- $\bar{f_{sy}} = 2.$
- $\bar{f_{xy}} = 1.$
- $\bar{f_{xt}} = 2.$
- $\bar{f_{yt}} = 3.$



Flot maximum à coût minimal : description

- Les deux problèmes précédents peuvent déjà être résolus par des algorithmes existants.
- L'approche par la programmation linéaire permet traiter de nouveaux problèmes.
- On modifie le problème du flot maximum en rajoutant des coût sur les arcs ($a : E \to \mathbb{R}$).
- On souhaite alors minimiser le coût ($\sum_{(u,v)\in E} a(u,v) \times f_{uv}$) en garantissant un flot minimum d.



Flot maximum à coût minimal : programme linéaire

$$
\begin{array}{lrl}
\text{minimiser} & \displaystyle\sum_{(u,v)\in E} a(u,v) \times f_{uv} & \\
\text{sous les contraintes} & f_{uv} \leq c(u,v) & \text{pour chaque } u, v \in V \\
& \displaystyle\sum_{v\in V} f_{uv} - \sum_{v\in V} f_{vu} = 0 & \text{pour chaque } u \in V \setminus \{s,t\} \\
& \displaystyle\sum_{v\in V} f_{sv} - \sum_{v\in V} f_{vs} = d & \\
& f_{uv} \geq 0 & \text{pour chaque } u, v \in V
\end{array}
$$

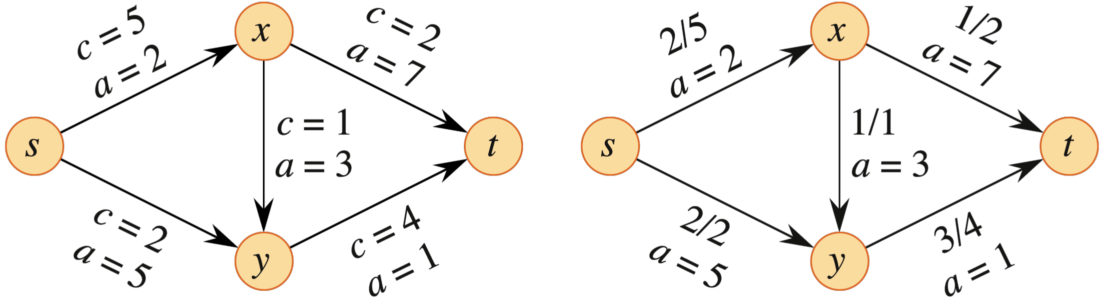
- Nombre de variables : $|V|^2$.
- Nombre de contraintes : $2|V|^2 + |V| - 1$.

Flot maximum à coût minimal : programme linéaire

$$
\begin{aligned}
\text{minimiser} \quad & \sum_{(u,v)\in E} a(u,v) \times f_{uv} \\
\text{sous les contraintes} \quad & f_{uv} \leq c(u,v) \quad \text{pour chaque } u, v \in V \\
& \sum_{v\in V} f_{uv} - \sum_{v\in V} f_{vu} = 0 \quad \text{pour chaque } u \in V \setminus \{s, t\} \\
& \sum_{v\in V} f_{sv} - \sum_{v\in V} f_{vs} = d \\
& f_{uv} \geq 0 \quad \text{pour chaque } u, v \in V
\end{aligned}
$$

- Nombre de variables : $|V|^2$.
- Nombre de contraintes : $2|V|^2 + |V| - 1$.

Flot maximum : exemple



Flot multi-produits : contexte

- L'entreprise Max & Fils produit et livre des tissus, des pots et des fleurs artificielles.
- Chaque type de produits est fabriqué dans une usine spécifique puis expédié chaque jour vers son entrepôt.
- Les différents types de produit doivent se partager le réseau de transport.

Flot multi-produits : description

- Un graphe orienté $G = (V, E)$ avec des capacités sur les arcs $c : E \to \mathbb{R}$.
- Il y a k produits différents chacun spécifié par s_i (son usine), t_i (son entrepôt) et d_i (la quantité de produit, c'est-à-dire le flot souhaité).
- Soit f_{iuv} le flot pour chaque produit sur chaque arc.
- Le flot total est la somme des flots de chaque produit $f_{uv} = \sum_{i=1}^{k} f_{iuv}$.
- On cherche à déterminer si une solution existe.

Flot multi-produits : programme linéaire

$$
\begin{array}{lrl}
\text{minimiser} & 0 & \\
\text{sous les contraintes} & \displaystyle\sum_{i=1}^{k} f_{iuv} \leq c(u,v) & \text{pour chaque } u, v \in V \\
& \displaystyle\sum_{v \in V} f_{iuv} - \sum_{v \in V} f_{ivu} = 0 & \text{pour chaque } i = 1, 2, \ldots, k \text{ et} \\
& & \text{pour chaque } u \in V \setminus \{s, t\} \\
& \displaystyle\sum_{v \in V} f_{i,s_i,v} - \sum_{v \in V} f_{i,v,s_i} = d_i & \text{pour chaque } i = 1, 2, \ldots, k \\
& f_{iuv} \geq 0 & \text{pour chaque } i = 1, 2, \ldots, k \text{ et} \\
& & \text{pour chaque } u, v \in V
\end{array}
$$

Question

Lequel de ces programmes linéaires résout le problème de flot maximum ?

1.

$$\begin{array}{lll} \text{maximiser} & \displaystyle\sum_{v\in V} f_{sv} & \\ \text{contraintes} & f_{uv} \leq c(u,v) & \forall u,v \in V \\ & \displaystyle\sum_{v\in V} f_{uv} = \sum_{v\in V} f_{vu} & \forall u \in V \setminus \{s,t\} \\ & f_{uv} \geq 0 & \forall u,v \in V \end{array}$$

2.

$$\begin{array}{lll} \text{minimiser} & \displaystyle\sum_{(u,v)\in E} a(u,v) \times f_{uv} & \\ \text{contraintes} & f_{uv} \leq c(u,v) & \forall u,v \in V \\ & \displaystyle\sum_{v\in V} f_{uv} - \sum_{v\in V} f_{vu} = 0 & \forall u \in V \setminus \{s,t\} \\ & \displaystyle\sum_{v\in V} f_{sv} - \sum_{v\in V} f_{vs} = d & \\ & f_{uv} \geq 0 & \forall u,v \in V \end{array}$$

3.

$$\begin{array}{lll} \text{maximiser} & d_t & \\ \text{contraintes} & d_v \leq d_u + w(u,v) & \forall (u,v) \in E \\ & d_s = 0 & \end{array}$$

4.

$$\begin{array}{lll} \text{minimiser} & \displaystyle\sum_{(u,v)\in E} c(u,v) \times d_{uv} & \\ \text{contraintes} & d_{uv} - z_u + z_v \geq 0 & \forall (u,v) \in E, u \neq s, v \neq t \\ & d_{sv} + z_v \geq 1 & \forall (s,v) \in E, v \neq t \\ & d_{ut} - z_u \geq 0 & \forall (u,t) \in E, u \neq s \\ & d_{st} \geq 1 & \text{si } (s,t) \in E \\ & d_{uv} \geq 0 & \forall (u,v) \in E \end{array}$$

Question

Lequel de ces programmes linéaires résout le problème de flot maximum ?

1. ✓

$$\begin{array}{lll} \text{maximiser} & \displaystyle\sum_{v\in V} f_{sv} & \\ \text{contraintes} & f_{uv} \leq c(u,v) & \forall u,v \in V \\ & \displaystyle\sum_{v\in V} f_{uv} = \sum_{v\in V} f_{vu} & \forall u \in V \setminus \{s,t\} \\ & f_{uv} \geq 0 & \forall u,v \in V \end{array}$$

2.

$$\begin{array}{lll} \text{minimiser} & \displaystyle\sum_{(u,v)\in E} a(u,v) \times f_{uv} & \\ \text{contraintes} & f_{uv} \leq c(u,v) & \forall u,v \in V \\ & \displaystyle\sum_{v\in V} f_{uv} - \sum_{v\in V} f_{vu} = 0 & \forall u \in V \setminus \{s,t\} \\ & \displaystyle\sum_{v\in V} f_{sv} - \sum_{v\in V} f_{vs} = d & \\ & f_{uv} \geq 0 & \forall u,v \in V \end{array}$$

3.

$$\begin{array}{lll} \text{maximiser} & d_t & \\ \text{contraintes} & d_v \leq d_u + w(u,v) & \forall (u,v) \in E \\ & d_s = 0 & \end{array}$$

4.

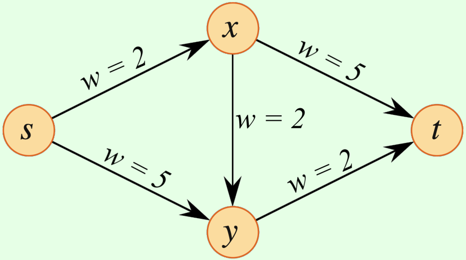
$$\begin{array}{lll} \text{minimiser} & \displaystyle\sum_{(u,v)\in E} c(u,v) \times d_{uv} & \\ \text{contraintes} & d_{uv} - z_u + z_v \geq 0 & \forall (u,v) \in E, u \neq s, v \neq t \\ & d_{sv} + z_v \geq 1 & \forall (s,v) \in E, v \neq t \\ & d_{ut} - z_u \geq 0 & \forall (u,t) \in E, u \neq s \\ & d_{st} \geq 1 & \text{si } (s,t) \in E \\ & d_{uv} \geq 0 & \forall (u,v) \in E \end{array}$$

✓(problème dual de coupe minimum)

Plus courts chemins : description

Notion algorithmique

- Un graphe orienté $G = (V, E)$ avec une pondération sur les arcs $w : E \to \mathbb{R}$.
- Un sommet source $s \in V$ et une destination $t \in V$.
- Soit d_v le poids du plus court chemin de s vers v ($d_s = 0$).
- On cherche d_t.



Plus courts chemins : programme linéaire

$$
\begin{aligned}
\text{maximiser} \quad & d_t \\
\text{sous les contraintes} \quad & d_v \leq d_u + w(u, v) \text{ pour tout arc } (u, v) \in E \\
& d_s = 0
\end{aligned}
$$

- Si c'était une minimisation, $d_t = -\infty$ serait un solution optimale.
- Astuce : c'est l'inégalité triangulaire qui garantit une valeur maximale aux variables.
- Nombre de variables : $|V|$.
- Nombre de contraintes : $|E| + 1$.

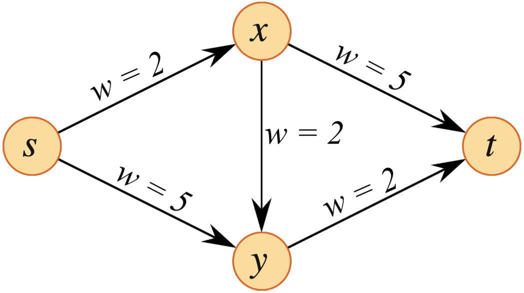
Plus courts chemins : programme linéaire

$$\begin{array}{ll} \text{maximiser} & d_t \\ \text{sous les contraintes} & d_v \leq d_u + w(u,v) \text{ pour tout arc } (u,v) \in E \\ & d_s = 0 \end{array}$$

- Si c'était une minimisation, $d_t = -\infty$ serait un solution optimale.
- Astuce : c'est l'inégalité triangulaire qui garantit une valeur maximale aux variables.
- Nombre de variables : $|V|$.
- Nombre de contraintes : $|E| + 1$.

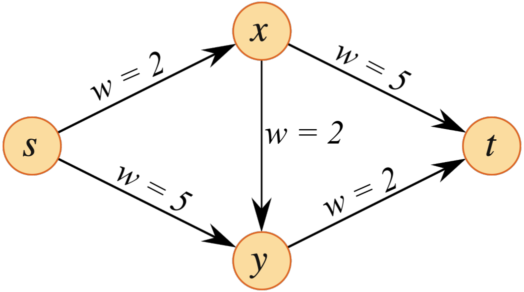
Plus courts chemins : exemple

$$
\begin{array}{ll}
\text{maximiser} & d_t \\
\text{sous les contraintes} & d_x \leq d_s + 2 \\
& d_y \leq d_s + 5 \\
& d_y \leq d_x + 2 \\
& d_t \leq d_x + 5 \\
& d_t \leq d_y + 2 \\
& d_s = 0
\end{array}
$$



Plus courts chemins : solution

- $\bar{d}_s = 0$.
- $\bar{d}_x = 2$.
- $\bar{d}_y = 4$.
- $\bar{d}_t = 6$.



Plan

Résumé

- La formulation d'un problème comme programme linéaire consiste à le modéliser avec des contraintes et une fonction objectif qui sont des sommes pondérées des variables que l'on souhaite déterminer.
- La programmation linéaire permet de traiter de nombreux problèmes d'optimisation combinatoire.
- Des outils efficaces permettent de résoudre des programmes avec de nombreuses variables et contraintes.
- Un problème de programmation linéaire peut s'adapter facilement en modifiant sa formulation.

Prochaines échéances

- Début du projet-tournoi le 23/9.
- Premier rendu intermédiaire du projet-tournoi le 6/10.
- Second rendu intermédiaire du projet-tournoi le 20/10.
- Épreuve sur table le 23/10.
- Rendu final pour le projet-tournoi le 4/11.
- Épreuve de TP le 5/11.
- Restitution du projet-tournoi le 6/11.