Master d'informatique - spécialité Androide - Module MAOA - Sorbonne Université

Recherche Opérationnelle
et Optimisation Combinatoire

Cahier d'exercices

---

1 Premiers exercices de modélisation

Exercice* 1 : *Problème de production (tiré d'un livre de 1ereS)

Un fabriquant de yaourt produit 2 types de yaourts à la fraise A et B à partir de fraise, de lait et de sucre. Chaque yaourt doit respecter les proportions suivantes de matières premières.

	A	B
Fraise	2	1
Lait	1	2
Sucre	0	1

Les matières premières sont en quantité limitée: 800 kilos de fraises, 700 kilos de lait et 300 kilos de sucre. La vente des yaourts A rapportent 4€ par kilo et les yaourts B 5€ . On considère que, lors du mélange à froid, aucune perte n'a lieu, ainsi le poids final du yaourt est celui de ses ingrédients.
a) Donner une formulation du problème sous la forme d'un programme linéaire.

b) Donner la représentation graphique du problème.
c) Donner la solution optimale.

Exercice* 2 : *Problème de transport

Une entreprise de construction d'automobiles possède 3 usines à Paris, Strasbourg et Lyon. Elle a besoin d'acheminer les métaux nécessaires à partir du Havre ou de Marseille. Chaque usine nécessite hebdomadairement 400 tonnes à Paris, 300 tonnes à Strasbourg et 200 tonnes à Lyon. Les ports du Havre et de Marseille peuvent fournir respectivement 550 tonnes et 350 tonnes.
Les coûts de transport entre ces villes sont donnés en kilo-€ par tonne dans le tableau suivant.

	Paris	Strasbourg	Lyon
Le Havre	5	6	3
Marseille	3	5	4

Proposer une modélisation de ce problème de transport de manière à satisfaire la demande, à partir des quantités disponibles et en minimisant les coûts de transport.

Exercice* 3 : *Combinaison optimale en vitamines (Diet Problem)

Une personne soucieuse de sa forme physique souhaite absorber chaque jour 37 unités de vitamine A, 28 unités de vitamine C et 32 unités de vitamine D. Deux marques sont succeptibles de fournir ces apports. La marque 1 coûte 3 euros et procure 2 unités de vitamine C et 8 unités de vitamine D. La marque 2 coûte 4 euros et procure 3 unités de vitamine A, 2 unités de vitamine C et 2 unités de vitamine D.
Il s'agit de trouver la combinaison respectant les exigences d'absorption quotidiennes au moindre coût.
- Définir les variables de décision de ce problème.
- Formuler la fonction objectif.
- Ecrire les contraintes d'un PLNE modélisant ce problème.

Exercice* 4 : *Centres de loisir

Une région est divisée en six zones (zones 1,...,6). La commune souhaite construire des centres de loisir dans certaines de ces zones. Et elle désire monter un nombre minimum de centres de telle manières que, pour chaque zone, il existe au moins un centre qui se trouve à au plus 15 minutes (en voiture) de cette zone. Le temps nécessaire pour aller d'une zone à l'autre est donné dans la table suivante:

de	zone 1	zone 2	zone 3	zone 4	zone 5	zone 6
zone 1	0	10	20	30	30	20
zone 2	10	0	25	35	20	10
zone 3	20	25	0	15	30	20
zone 4	30	35	15	0	15	25
zone 5	30	20	30	15	0	14
zone 6	20	10	20	25	14	0

4.1) Formuler le problème qui consiste à déterminer le nombre minimum de centres à construire ainsi que les zones où ceux-ci doivent être construits comme un programme linéaire en nombres entiers.

4.2) Modifier le programme pour qu'il corresponde à la contrainte suivante: si un centre est construit dans la zone 1, alors un centre doit être construit dans la zone 4.

4.3) Quelle inégalité permet de modéliser la contrainte suivante: une zone au moins parmi les zones 1, 2 et 3 doit avoir au moins un centre à au plus 15 minutes. Est-il nécessaire de l'ajouter au programme?

4.4) En utilisant les idées de la question précédente, peut-on ainsi réduire la formulation en ôtant des inégalités?

Exercice 5 **: *Diététique avare***

Le cuisinier de l'hôtel doit préparer le petit-déjeuner de La Castafiore. La cantatrice doit suivre un régime auquel son avarice donne raison. Le cuisinier doit donc composer un menu comportant au plus 1000 calories et qui soit le moins cher possible, à partir des ingrédients suivants:

	Poids unitaire en grammes	Calories par gramme	Prix unitaire en euros
toast	40	3,4	1,50
miel (par toast)	15	2,8	0,50
confiture (par toast)	10	5	0,40
thé (une tasse)	100	0,2	1,00
lait (un verre)	100	0,5	0,70
beurre (par toast)	10	8	0,30
oeuf (sur le plat)	60	1	0,50
bacon (la tranche)	20	8	1,00
jus d'orange	30	4	1,20
sucre (morceau)	5	4	0,10

Exercice 6 **: *Décision en production industrielle***

Une entreprise produisant des billes de plastique veut s'implanter sur une nouvelle zone géographique. Ces billes de plastique sont la matière première de nombreux objets industriels (sièges, manches d'outils, bidons,...). L'entreprise a démarché n clients et prévoit de vendre, sur un horizon de 5 années à venir, d_i tonnes de billes de plastique à chaque client $i \in \{1,...,n\}$. L'entreprise dispose de m sites potentiels $s_1,...,s_m$ pour installer ses usines. On a évalué à c_j euros le coût d'installation d'une usine sur le site s_j, $j = 1,...,m$. Les usines prévues ne sont pas toutes de même capacité de production: un site s_j aura une capacité de M_j tonnes de billes sur les 5 années à venir, $j = 1,...,m$. On suppose que le coût de production est indépendante du lieu de production. Enfin, on connaît les coûts de transports par tonne c_{ij} entre un client i et un site s_j, pour $i = 1,...,n$ et $j = 1,...,m$.

L'entreprise souhaite déterminer les sites sur lesquels établir ses usines pour pouvoir satisfaire la demande de ses clients tout en minimisant le coût total (installation, production et livraison) sur les 5 prochaines années.

Donner une modélisation de ce problème par un PLNE.

---

2 Linéarisation

Exercice* 7 : *Problème d'affectation quadratique

On considère n tâches et n processeurs. On désire affecter chacune des n tâches à chacun des n processeurs de façon à ce qu'il y ait une seule tâche par processeur et inversement. Si une tâche $i \in \{1,...,n\}$ est affectée à un processeur $k \in \{1,...n\}$, et si une tâche $j \in \{1,...,n\} \setminus \{i\}$ est affectée à un processeur $l \in \{1,...,n\} \setminus \{k\}$, cela entraîne un temps de communications entre les tâches i et j, que l'on note alors c_{ijkl}. On considère le problème d'affectation quadratique qui consiste à minimiser le coût total de communication entre les n tâches sur les n processeurs.

7.1) On considère les variables binaires x_{ik} telles que x_{ik} prenne la valeur 1 si et seulement si la tâche $i \in \{1,...,n\}$ est affectée au processeur $k \in \{1,...n\}$. Proposer une formulation à fonction objective quadratique et avec des inégalités linéaires pour ce problème en utilisant ces variables.

7.2) Etant donnés $i \in \{1,...,n\}$, $k \in \{1,...n\}$, $j \in \{1,...,n\} \setminus \{i\}$ et $l \in \{1,...,n\} \setminus \{k\}$, on considère la variables binaire s_{ijkl}. Proposer 3 inégalités linéaires liant les variables x_{ik}, x_{jl} et s_{ijkl} de façon à ce que s_{ijkl} prennent la valeur 1 si à la fois la tâche i est affectée au processeur k et si la tâche j est affectée au processeur l. En déduire une formulation PLNE pour le problème de l'affectation quadratique.

7.3) Peut-on relaxer l'intégrité de certaines variables? Comparer le nombre de variables et de contraintes des deux formulations.

Exercice* 8 : *Linéarisation d'un fonction objective en valeur absolue

Soit x et y deux variables réelles quelconques. On considère une formulation dont la fonction objective est de minimiser la valeur absolue $|x - y|$ (et où cette valeur absolue n'apparaît que dans la fonction objective).

1) Prouver que, quelque soit une valeur réelle A, il existe toujours deux valeurs u et v réelles telles que $A = u - v$ et $|A| = u + v$.
2) En déduire une linéarisation de la formulation et prouvez-là.

Exercice* 9 *: Linéarisation de la non-égalité entre variables

Soit x et y deux variables entières positives. On désire modéliser linéairement le fait que x et y doivent prendre des valeurs différentes. On suppose qu'il existe une grande valeur M telle que x et y soit toujours plus petite strictement que M.
Donner une écriture linéaire (en nombres entiers) de cette différence entre variables.

Exercice* 10 *: Linéarisation d'un quotient

On désire linéariser le rapport de deux fonctions linéaires de variables bivalentes, c'est-à-dire donner un programme linéaire en nombres entiers équivalent à la formulation non-linéaire suivante:

$$(D)\left\{\begin{array}{ll} Max & \dfrac{a_0 + \sum_{i=1}^{n} a_i x_i}{b_0 + \sum_{i=1}^{n} b_i x_i} \\ & \quad Tx \leq \alpha \\ & \quad x_i \in \{0,1\} \qquad \forall i \in \{1,...,n\}. \end{array}\right.$$

où $Tx \leq \alpha$ est un ensemble de contraintes linéaires.

On suppose que

- $a_i \geq 0$ pour $i = 0, 1, ..., n$
- $b_i \geq 0$ pour $i = 1, ..., n$ et $b_0 > 0$

1) Montrer que la fonction objective est toujours bornée par une valeur M.
2) Ajouter à la formulation une variable réelle z égale à la valeur de la fonction objective. Réécrivez alors (D) comme une formulation contenant entre autres les termes quadratiques $(zx_i)_{i\in\{1,...,n\}}$, mais où le quotient a disparu.
3) On désire ajouter des variables réelles $(t_i)_{i\in\{1,...,n\}}$. Proposer 3 lots de n inégalités linéaires impliquant que t_i ait la valeur du produit zx_i pour $i = 1, ..., n$.
4) En déduire une formulation PLNE équivalente à (D).

---

3 Branchement

Les exercices suivants sur le branchement peuvent être réalisés “à la main”.

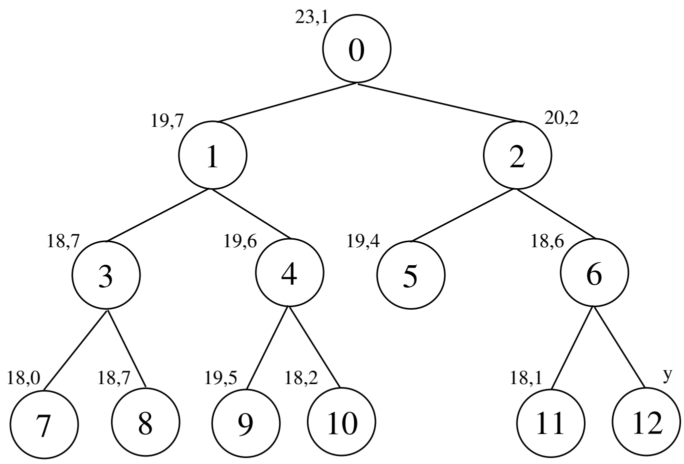
Exercice* 11 : *Représentation graphique

Résoudre le PLNE suivant par Branch-and-Bound en utilisant une représentation graphique des solutions:

$$\begin{aligned} \text{Min } & 4x_1 + 5x_2 \\ & x_1 + 4x_2 \geq 5 \\ & 3x_1 + 2x_2 \geq 7 \\ & x_1 \geq 0, x_2 \geq 0 \\ & x_1, x_2 \in \mathbb{N}. \end{aligned}$$

Exercice* 12 : *Arborescence à compléter

On considère la représentation ci-dessous de l’exploration d’un arbre de branchements d’un problème linéaire en nombres entiers. Ce PLNE consiste à maximiser une fonction objectif dont les coefficients sont tous entiers. Cette exploration a été stoppée juste après avoir exploré les sous-problèmes indiqués par des cercles. Leur numéro est là uniquement pour leur donner un nom. La valeur indiquée au-dessus d’un sous-problème est l’évaluation de ce sous-problème.



12.1) Comment est obtenu cette évaluation? Qu’indiquent les évaluations des sous-problèmes? Que peut-on déduire pour le sous-problème 5?

12.2) En regardant uniquement le niveau 0 (sous-problème 0) (resp. le niveau 1 (1 et 2), le niveau 3 (3 à 6), quelle borne supérieure de l’arborescence peut-on déduire?

12.3) Avant l’évaluation y du sous-problème 12, quelle est la valeur de la meilleure borne supérieure pour le problème?

12.4) Après évaluation y du sous-problème 12, quelle sera alors la valeur de la meilleure borne supérieure pour le problème?

12.5) Pour le sous-problème 7, l'évaluation 18 a été obtenue par une solution entière du problème, que peut-on déduire pour la suite de l'exploration?

12.6) A partir de maintenant, on souhaite poursuivre l'exploration de l'arbre de branchement est effectuée par stratégie BFS "au meilleur sous-problème d'abord". Quel est le sous-problème qui sera traité après le sous-problème 12?

12.7) Une heuristique primale vient de produire une solution entière valide pour le problème qui a pour valeur 19. Que peut-on en déduire pour la suite de l'exploration?

Exercice* 13 : *Programme quadratique binaire

On considère le programe quadratique à contraintes quadratiques binaires suivant:

$$\begin{aligned} &\text{Max } x_1^2 x_2 + 2x_2 + 2x_1x_2x_3 + 3x_1 + x_1x_3 \\ &\quad x_1x_2 + 3x_2 - x_1x_3 \le 4 \\ &\quad 3x_1 + 2x_1x_2 - x_1x_3 \le 2 \\ &\quad x_i \in \{0,1\} \quad i = 1, ..., 3. \end{aligned}$$

13.1) Expliquer pourquoi, en fixant chaque variable du problème à la valeur 1, on obtient une évaluation de la fonction objective du programme.

13.2) En utilisant le principe classique de branchement, expliquer pourquoi, au cours de l'exploration, on pourra trouver une meilleure évaluation de certains sous-problèmes.

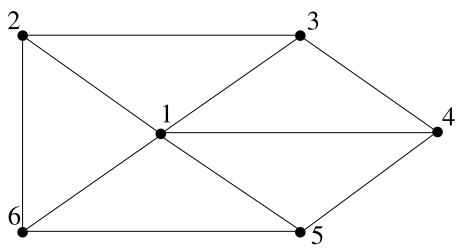
13.3) En utilisant cette évaluation et un arbre de branchement classique, résoudre le programme.

---

4 Points extrêmes et coupes

Exercice* 14 : *Contraintes de roues pour le problème du stable

On considère le problème du stable de poids maximum sur le graphe $H_1 = (V, E, c)$ (où le c est un poids 1 sur chaque sommet) suivant:



On s'intéresse à la formulation du problème du stable suivante:

$$
\begin{aligned}
\text{Max} \sum_{u \in V} c(u)x(u) & \\
x(u) + x(v) \leq 1 \qquad & \text{pour tout } uv \in E, && (1)\\
\sum_{u \in V(C)} x(u) \leq \left\lfloor \frac{|C|}{2} \right\rfloor \quad & \forall \text{ cycle impair } C, && (2)\\
0 \leq x(u) \leq 1 \qquad & \text{pour tout } u \in E, && (3)\\
x(u) \text{ entier}, \qquad & \text{pour tout } u \in V. &&
\end{aligned}
$$

On peut noter que H_1 ne contient pas de clique de tailles supérieures à 3 et que toutes les inégalités associées à des cliques de tailles 2 (arêtes) et de tailles 3 (cycles impairs de taille 3) sont déjà dans la formulation.

14.1) Lister les inégalités de la formulation pour le graphe H_1.

14.2) Montrer qu'une solution serrant les inégalités de cycles impaires correspondant aux cycles $(1,2,3),(1,3,4),(1,4,5),(1,5,6),(1,6,2)$ et $(2,3,4,5,6)$ est fractionnaire.

14.3) Donner les arguments prouvant que ce point est un point extrême fractionnaire du polyèdre définir par les inégalités triviales (3), d'arêtes (1) et de cycles impairs (1) pour ce graphe H_1.

14.4) Sommer les inégalités de triangles (cycle de tailles 3) issues de la formulation pour H_1 selon le principe de l'arrondi de Chvatal: donner une inégalité valide nouvelle.

14.5) Re-prouvez la validité de cette inégalité sans utiliser la somme de Chvatal.

14.6) Coupe-t-elle le point fractionnaire de la première question?

Exercice* 15 : *Inégalités de recouvrement (cover inequalities) pour le problème du sac-à-dos

Considérons un problème de sac à dos P en 0-1 (0-1 knapsack problem).

$$
\begin{gathered}
Max \quad c_1x_1 + c_2x_2 + ... + c_nx_n\\
a_1x_1 + a_2x_2 + ... + a_nx_x \leq b\\
x_i \in \{0,1\} \text{ pour } i = 1,...,n
\end{gathered}
$$

On suppose que $a_i > 0$ pour $i = 1,...,n$ et $b > 0$. On considère le polyèdre issue de la relaxation linéaire de cette formulation.

Un sous-ensemble R de $\{1,...,n\}$ est dit *recouvrement* (cover) de P si $\sum_{i \in R} a_i > b$.

15.1) Montrer que si R est un recouvrement de P alors la contrainte suivante est valide pour le problème de sac-à-dos

$$\sum_{i \in R} x_i \leq |R| - 1.$$

15.2) Soit le problème de sac-à-dos Q suivant

$$
\begin{aligned}
Max \quad z = 10x_1 + 19x_2 + 12x_3 + 12x_4 + x_5 + 3x_6 + x_7 \\
11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19 \\
x_i \in \{0,1\} \text{ pour } i = 1, ..., 7.
\end{aligned}
$$

Les ensembles suivants sont-ils des recouvrements pour le problème de sac-à-dos Q:
$R_1 = \{1,2,3\}$, $R_2 = \{2,3,4\}$ et $R_3 = \{3,4,5,6\}$? Donner les contraintes de recouvrement correspondantes pour ceux qui le sont.

15.3) Donner pour chacune des contraintes données dans 2) un point extrême du domaine de $\tilde{Q}$ (la relaxation linéaire de Q) qui peut être coupé par cette contrainte. Justifier votre réponse.

15.4) *Séparation des inégalités de recouvrement.* Commes les inégalités de recouvrements sont potentiellement en nombre exponentiel, on désirer proposer un algorithme de séparation. Soit $\tilde{x}$ un point fractionnaire issue de la relaxation linéaire du problème de sac-à-dos (ou d'une relaxation enrichie d'inégalités valides): le problème de séparation revient donc à déterminer s'il existe ou non un recouvrement R tel que $sum_{i\in R}\tilde{x}_i > |R| - 1$ et s'il en existe le produire.
a) On pose $\tilde{x}'_i = 1 - \tilde{x}_i$ pour tout $i \in \{1, \dots, n\}$. Montrer que, pour un R donné, l'inégalité peut s'écrire alors $\sum_{i\in R} \tilde{x}'_i \leq 1$.
b) Montrer que ce problème de séparation se réduit à un problème où $\tilde{x}'$ est le coefficient de la fonction objectif. A quelle "famille" de problème appartient ce problème?
c) En admettant que cette réduction est en faite une équivalence, quelle est la complexité de ce problème de séparation?
d) Expliquer comment utiliser en pratique ces inégalités.

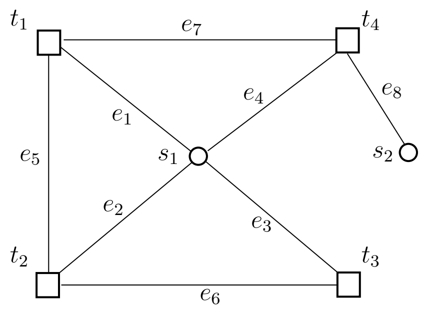
5 Problème de l'Arbre Steiner

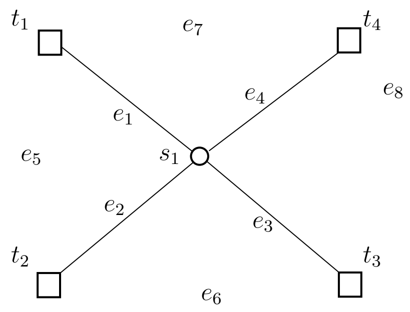
Soit un graphe $G = (V, E)$ où V est l'ensemble des sommets et E l'ensemble des arêtes. On associe à toute arête $e \in E$ un poids $w(e)$ strictement positif. L'ensemble V des sommets est partitionné entre l'ensemble T des *terminaux* et l'ensemble S des *sommets Steiner*. Le *problème de l'arbre Steiner* dans G consiste à déterminer un sous-graphe de G qui soit un arbre couvrant tous les terminaux et dont le poids total des arêtes est minimal. Remarquons qu'un arbre Steiner peut contenir ou non des sommets Steiner, qui peuvent être ainsi vus comme des sommets optionnels: un sommet Steiner ne sera pris dans la solution que s'il est nécessaire ou s'il permet d'obtenir une meilleure solution.

On rappelle qu'un un arbre est un graphe connexe sans cycle. Pour un sous-graphe H de G, on notera $w(H)$ la somme des poids des arêtes de H.

Pour illustrer les définitions, ci-dessous à gauche un graphe $G_1 = (V_1, E_1)$ associé à l'ensemble des terminaux $T_1 = \{t_1, t_2, t_3, t_4\}$ et comportant 2 sommets Steiner s_1 et s_2 et à droite un arbre Steiner qui contient les 4 terminaux et un seul sommet Steiner.

Déterminer un arbre Steiner minimum est un problème NP-difficile. Toutefois, il existe des algorithmes pseudo-polynomiaux pour le résoudre. Nous allons ici étudier une formulation PLNE permettant de le résoudre exactement et efficacement au travers d'un algorithme de Branch-and-Cut.





Exercice* 16 : *Conception de circuits électroniques

Le problème de la recherche d'un arbre Steiner dans un graphe trouve une application naturelle dans la conception de circuits électroniques. La conception de circuits électroniques consiste à placer les composants et les pistes conductrices d'un circuit sur support, par exemple une carte. Parmi toutes les étapes nécessaires, l'une d'entre elles consiste à relier tous les composants (puces, condensateurs,...) aux bornes générales d'alimentation électrique du circuit, qui sont généralement placées dans un coin de la carte. La patte correspondant à la borne "+" (respectivement "-") de chacun des composants doit donc être reliée à la borne générale "+" (respectivement "-"). On appelle cette étape *le routage d'alimentation électrique* qui a lieu sur les différentes couches du circuit (ainsi deux des couches sont réservées à l'alimentation). Tous les emplacements de pistes ne sont pas utilisables: on doit donc choisir les pistes parmi un sous-ensemble connu de pistes possibles: ces pistes potentielles se croisent en différents *points* du circuits dont certains sont les pattes des bornes des composants et d'autres sont simplement des croisements entre deux pistes potentielles. On désire évidemment réduire le poids et le coût d'un circuit, on veut donc effectuer ce routage en utilisant une longueur totale minimale de pistes conductrices.

16.1) Montrer qu'un arbre Steiner minimum d'un graphe G est un sous-graphe connexe de coût minimum de G contenant tous les terminaux.

16.2) Expliquer brièvement pourquoi déterminer le routage d'alimentation électrique d'un circuit électronique revient à rechercher deux arbres Steiner minimaux.

Exercice* 17 : *Formulation du problème de l'Arbre Steiner

Soit χ un vecteur binaire sur les arêtes de G vérifiant toutes les inégalités suivantes

$$\sum_{e\in\delta(W)} \chi(e) \geq 1 \qquad \begin{array}{l} \forall W \subset V,\;\; \emptyset \neq W \neq V, \\ W \cap T \neq \emptyset \text{ et } (V \setminus W) \cap T \neq \emptyset, \end{array} \tag{1}$$

où $\delta(W)$ est l'ensemble des arêtes ayant une extrémité dans W et l'autre dans $V \setminus W$.
Soit $H = (V(F), F)$ le sous-graphe de G défini par l'ensemble d'arêtes $F = \{e \in E \mid \chi(e) = 1\}$ et l'ensemble $V(F)$ des sommets extrémités des arêtes de F.

17.1)
a) Prouvez que H contient tous les terminaux.
b) Soit deux terminaux t_1 et t_2 de T. Prouvez que H possède une composante connexe contenant t_1

et t_2. (Une composante connexe d'un graphe est un sous-graphe dans lequel toute paire de sommets est reliée par une chaîne.).
c) En déduire que si H est de poids minimum, alors H est connexe. Montrer également que H est alors un arbre (c'est-à-dire sans cycle).

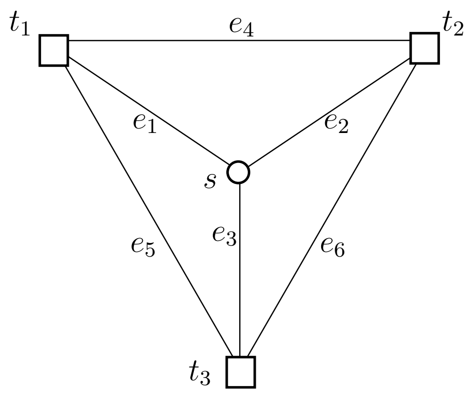
17.2) En déduire que le problème de l'arbre Steiner dans G peut se formuler comme le programme en nombres entiers suivant.

$$(\mathcal{P}) \begin{cases} Min \quad \sum_{e\in E} w(e)x(e) & \\ \sum_{e\in\delta(W)} x(e) \geq 1 \quad \begin{array}{l} \forall W \subset V, \quad \emptyset \neq W \neq V, \\ W \cap T \neq \emptyset \text{ et } (V \setminus W) \cap T \neq \emptyset, \end{array} & (1) \\ 0 \leq x(e) \leq 1 \quad \forall e \in E & (2) \\ x(e) \text{ entier} \quad \forall e \in E. & (3) \end{cases}$$

17.3) Indiquer le nombre de variables et d'inégalités que possède cette formulation? Quelle solution algorithmique peut-on envisager pour la résoudre?

Exercice* 18 : *Points extrêmes et contraintes valides pour le problème de l'Arbre Steiner

On considère le graphe $G = (V, E)$ comme suit (les carrés représentent les terminaux et le cercle un sommet Steiner).



Soit y^* la solution donnée par $y^*(e_1) = y^*(e_2) = y^*(e_4) = \frac{1}{2}$, $y^*(e_3) = 1$ et $y^*(e_5) = x(e_6) = 0$.

18.1) **a)** Montrer que y^* est une solution du programme donné par les contraintes(1) et (2).
b) Donner 6 contraintes de ce polyèdre vérifiée à l'égalité par y^* et indiquer comment on peut prouver (sans le faire) que y^* est point extrème de ce polyèdre.

18.2) On considère la partition des sommets du graphe G entre $V_1 = \{t_1\}$, $V_2 = \{t_2\}$ et $V_3 = \{s, t_3\}$. On définit le graphe G' en contractant chacun de ces ensembles en un sommet: V_1 (resp. V_2, V_3) devient u_1 (resp. u_2, u_3). (Par cette opération, G' conserve toutes les arêtes de G sauf e_3).
a) Montrer que les arêtes du graphe G', qui sont prises dans une solution du problème de l'arbre

Steiner pour G, forment un graphe connexe dans G'.
b) En déduire que la contrainte

$$x(e_1) + x(e_2) + x(e_4) + x(e_5) + x(e_6) \geq 2 \tag{1}$$

est valide pour le problème de l'arbre Steiner dans G.
c) Montrer que l'ajout dans $\mathcal{P}$ de la contrainte (1) est utile.

Exercice* 19 : *Algorithme de séparation pour les inégalités (1)
On dispose de la fonction $cutmin(y)$
- où $y \in I\!R^{|E|}$ et $0 \leq y(e) \leq 1$ pour tout $e \in E$
- qui renvoie l'ensemble de sommets W avec $\emptyset \neq W \neq V, W \cap T \neq \emptyset$ et $(V \setminus W) \cap T \neq \emptyset$, tel que la valeur $\sum_{e \in \delta(W)} y(e)$ soit minimun par rapport au vecteur y.

Donner le schéma d'un algorithme de coupes et branchements (Branch-and-Cut) pour la formulation $(\mathcal{P})$ composée des inégalités (1), (2) et (3).

6 Problème de la coupe maximale

Soit $G = (V, E)$ un graphe non orienté et une fonction poids quelconque $c(e), e \in E$. On note $ij \in E$ une arête du graphe. Pour un sous-ensemble non vide de sommets $W \subset V$, avec $\emptyset \neq W \neq V$, on note $\delta(W)$ l'ensemble des arêtes avec exactement une extrémité dans W (et donc l'autre dans $V \setminus W$. On appelle *coupe* du graphe G, un ensemble B d'arêtes tel qu'il existe $W \subset V$, avec $\emptyset \neq W \neq V$ avec $B = \delta(W)$.
Le *problème de la coupe de poids maximum* (Max-Cut) consiste à déterminer un ensemble d'arêtes $B \subset E$ tel que B soit une coupe et que $c(B) = \sum_{e \in B} c(e)$ soit maximum.

Ce problème est NP-difficile. En revanche, lorsque les poids sont tous négatifs, il est donc équivalent à déterminer une coupe de plus petit poids (en prenant l'opposé des poids): dans ce cas, le problème est polynomial.

Pour une variable ou un vecteur v indicé sur les arêtes de E, on notera $v(B) = \sum_{e \in B} v(e)$ pour tout sous-ensemble $B \subset E$.

Exercice* 20 : *Modélisation d'un verre de spin
En physique statistique, on appelle *verre de spin* un état de la matière, caractérisé à l'échelle microscopique par une aimantation (moment magnétique ou spin) de chaque atome dans une direction particulière. A l'état de verre de spin, le système a des propriétés de supra-conducteur, comme par exemple d'être un puissant éléctro-aimant. On mesure la faculté d'un matériau à s'aimanter sous l'action d'un champ magnétique, par sa *susceptibilité magnétique*. En fait, l'état de verre de spin s'obtient à une température qui minimise la susceptibilité magnétique du système.

On sait par exemple obtenir l'état de verre de spin pour des système magnétiques obtenus en diluant des atomes d'un matériau magnétique (Fer) dans un matériau non magnétique (Or) et en plaçant le système à la bonne température. Dans un tel système, la susceptibilité magnétique peut être définie comme la somme des énergie d'interaction magnétique entre les atomes de fers. Entre deux atomes de fer i, j, il existe une énergie d'interaction

$$H_{ij} = -J(d)S_iS_j$$

où S_i (resp. S_j) est le moment magnétique (spin) de l'atome i (resp. j) et $J(d)$ une fonction qui dépend de la distance d entre les deux atomes. Ce phénomène intrigue énormément les physiciens et ils ont proposé divers modèles pour comprendre le lien entre cette fonction de susceptibilité et l'état de verre de spin. Un des plus célèbres modèles est de considérer que les valeurs possibles pour leurs spins sont uniquement +1 ou -1 et que les interactions entre atomes n'ont lieu qu'entre les plus proches voisins. Ainsi le problème de minimisation de la fonction susceptibilité dépend uniquement de la configuration S des spins, c'est-à-dire une affection de +1 ou de -1 à chaque atome) et vaut alors

$$H(S) = \sum_{ij \in L} -J_{ij}S_iS_j$$

où L est l'ensemble des couples d'atomes ayant une interaction et J_{ij} un valeur constante (dépendant uniquement de la distance entre i et j).

Soit $G = (V, E)$ un graphe où les sommets correspondent aux atomes de fer et les arêtes aux liaisons de L. On associe à chaque arête ij le poids $w_{ij} = -J_{ij}$.

20.1) Montrer que le problème de minimisation de la fonction H pour un verre de spin revient à minimiser une fonction quadratique à valeurs bivalentes prises dans $\{-1, +1\}$.

20.2) Pour une affectation S de valeurs -1 ou +1 aux sommets de V, on définit les ensembles $S_+ = \{i \in V \mid S_i = +1\}$ et $S_- = \{i \in V \mid S_i = -1\}$. Montrer que ce problème se ramène au problème de la coupe maximum dans le graphe G. (Indication: $\sum_{ij \in E} w_{ij}$ est une quantité constante).

Exercice* 21 : *Formulation pour le problème de la coupe maximale

Soit $G = (V, E)$ un graphe non orienté et une fonction poids $c(e), e \in E$. On admet le résultat suivant:

Lemme 1: Un ensemble B d'arête de G est une coupe si et seulement si B intersecte tout cycle de G en un nombre pair d'arêtes.

21.1) **a)** Soit B une coupe de G. Montrer que pour tout cycle C de G et pour tout $F \subseteq C$ avec $|F|$ impair, $B \cap C \neq F$.

b) Soit $B \subseteq E$ un ensemble quelconque d'arêtes. On pose $\chi^B(e) = \begin{cases} 1 \text{ si } e \in B \\ 0 \text{ sinon.} \end{cases}$

Montrer que le vecteur χ^B associé à une coupe B de G vérifie l'inégalité suivante:

$$\chi^B(F) - \chi^B(C \setminus F) \leq |F| - 1 \quad \text{pour tout cycle } C, F \subseteq C, |F| \text{ impair.} \quad (1)$$

21.2) Montrer que déterminer une coupe de poids maximum dans G revient à résoudre le programme linéaire en nombres entiers suivant

$$(P)\begin{cases} \text{Maximiser } \sum_{e\in E} c(e)x(e) & & \\ \quad x(F) - x(C\setminus F) \leq |F|-1, & \text{pour tout cycle } C,\ F\subseteq C,\ |F| \text{ impair}, & (1)\\ \quad 0\leq x(e)\leq 1, & \text{pour toute arête } e\in E, & (2)\\ \quad x(e) \text{ entier}, & \text{pour toute arête } e\in E. & (3)\end{cases}$$

On appelle les contraintes de type (1) des *contraintes de cycles*, de type (2) des *contraintes triviales* et de type (3) des *contraintes d'intégrité*.

21.3) Indiquer le nombre de variables et d'inégalités que possède cette formulation? Quelle solution algorithmique peut-on envisager pour la résoudre?

Exercice 22 : *Points extrêmes et inégalité valide*

On considère dans cette question le graphe $K_5 = (V, E)$, c'est-à-dire le graphe complet à 5 sommets.

22.1) On considère un cycle à 3 sommets dans le graphe K_5 et on nomme u, v et w ces trois sommets. Donner les inégalités (1) associées à ce cycle.

22.2) Soit y le vecteur indicé sur les arêtes de K_5 tel que $y(e) = \frac{2}{3}, \forall e \in E$.
Donner 10 inégalités vérifiées à l'égalité par y.
Que faudrait-il prouver pour montrer que y est un point extrême de la formulation (P) associée à K_5?

22.3) On considére à présent un graphe G quelconque contenant une clique $K_p = (V', E')$ à p sommets (c'est-à-dire un sous-graphe complet). Montrer que l'inégalité suivante est valide pour le problème associé au graphe G

$$x(E') \leq \left\lfloor \frac{p^2}{4} \right\rfloor$$

où $\lfloor . \rfloor$ est la partie entière inférieure d'un nombre.
A partir de la question précédente, pourquoi peut-on dire que cette inégalité est utile pour résoudre le problème de la coupe maximale?

Exercice 23 : *Algorithme de séparation pour le problème de la coupe maximale*

On veut concevoir une méthode de coupes associée aux contraintes de cycles (1) et aux inégalités triviales pour un graphe $G = (V, E)$.
Soit (P^*) un programme linéaire contenant un nombre polynomial d'inégalités de cycles et toutes les inégalités triviales. On note x^* la valeur de la relaxation linéaire de $(P*)$. On rappelle alors qu'un algorithme de séparation consiste à déterminer si une solution courante x^*, obtenue par relaxation linéaire, viole une contrainte et dans ce cas, l'algorithme produit une contrainte.

23.1) Montrer qu'une inégalité de cycle peut se réécrire ainsi:

$$\sum_{e\in C\setminus F} x(e) + \sum_{e\in F}(1 - x(e)) \geq 1, \text{ pour tout cycle } C,\ F\subseteq C,\ |F| \text{ impair},$$

23.2) On suppose que l'on possède une fonction `oracle`$(G, w, \bar{w})$ où G est un graphe et w et $\bar{w}$ sont des poids positifs associés aux arêtes de G et qui détermine un cycle C et un ensemble impair $F \subset C$ tels que la somme des poids $w(C \setminus F) + \bar{w}(F)$ soit minimale. Expliquer comment résoudre le problème de séparation pour les inégalités (1) en utilisant la fonction `oracle`.

23.3) On s'intéresse à présent à un algorithme correspond à la fonction `oracle`. Pour cela, on définit un nouveau graphe $G' = (V' \cup V'', E' \cup E'' \cup E''')$ à partir de $G = (V, E)$ de la façon suivante:
- V' et V'' sont deux copies de V, *i.e.* pour chaque sommet $i \in V$, il y a un sommet $i' \in V'$ et un sommet $i'' \in V''$;
- E' et E'' sont deux copies de E, *i.e.* pour chaque arête $ij \in E$, il y a une arête $i'j' \in E'$ et une arête $i''j'' \in E''$;
- pour toute arête $ij \in E$, il y a une arête $i'j''$ et une arête $i''j'$ dans E''';
- les arêtes de E' et E'' sont valuées par w_{ij};
- les arêtes de E''' sont valuées par $\bar{w}_{ij}$.

a) Soit un sommet $i_0 \in V$. Montrer que tout chemin dans G' de i'_0 à i''_0 contient un nombre impair d'arêtes de E'''.

b) On dit qu'un chemin μ de G' est G-élémentaire si, en dehors de ses extrémités, tous ses sommets correspondent à des sommets distincts de V. Soit un chemin G-élémentaire μ de i'_0 à i''_0. Montrer que μ correspond à un cycle C et un ensemble $F \setminus C$, avec F impair.

c) On suppose, de plus, que μ est un plus court chemin G-élémentaire de i'_0 à i''_0. En déduire que μ correspond à l'inégalité de cycle (1) la plus violée correspondant à un cycle passant par i_0.

d) On peut noter que, à partir d'un plus court chemin quelconque μ', on peut déterminer en temps polynomial un sous-chemin μ qui soit G-élémentaire (il suffit de rechercher itérativement les paires de sommets (j', j'') en partant des deux extrémités de μ').

Donner un algorithme polynomial correspondant à `oracle`.

23.4) En déduire une méthode de coupes pour résoudre la relaxation linéaire de la formulation (P). Expliquer pourquoi il s'agit d'une méthode polynomiale.

7 Approches polyédrales

Exercice* 24 : *Polytope du sous-graphe 2-connexe

Soit $G = (V, E)$ un graphe non-orienté. Un arête e de E peut se noter uv où u et v sont les extrémités de e. On note une chaîne P comme une liste d'arêtes $(e_1 = u_1u_2, e_2 = u_2u_3, ..., e_l = u_{l-1}u_l)$. Pour une arête $e \in E$, on note $G \setminus e$ le graphe comprenant les sommets de V et les arêtes de $E \setminus \{e\}$. On dit que G est connexe si, pour tout couple de sommets (u, v), il existe une chaîne reliant u et v.

Deux chaînes P_1 et P_2 sont dits *arête-disjoints* si P_1 et P_2 n'ont aucune arête en commun (remarquons qu'ils peuvent avoir des sommets en commun). On dit que G est 2-connexe si, pour tout couple (u, v), il existe deux chaînes arêtes-disjoint reliant u et v. On associe à présent à chaque arête e de V, un poids $c(e)$. Le *problème du sous-graphe 2-connexe* consiste à déterminer un sous-ensemble F d'arêtes tel que le graphe $G' = (V, F)$ soit 2-connexe et tel que la somme des poids des arêtes de f soit minimum. Remarquons pour cela que G doit, au départ, être 2-connexe.

Modélisation et formulation

Le problème du sous-graphe 2-connexe a une application directe en conception de réseaux de télécommunications ou de réseaux électriques. Un tel réseau $G = (V, E)$ est composé d'un ensemble E de stations et E de liaisons entre stations. On appelle réseau *fiable* un réseau connexe qui résiste aux pannes sur les liaisons, c'est-à-dire un réseau qui reste connexe quand on une arête tombe en panne, i.e. pour toute arête $e \in E$, $G \setminus e$ reste connexe.

24.1) Montrer qu'un réseau 2-connexe est fiable.

24.2) On veut montrer qu'un réseau fiable est 2-connexe. Pour cela, considérons un graphe G fiable et un couple de sommets distincts (u, v). Comme G est connexe, il existe une chaîne P entre u et v. Montrer qu'il existe une deuxième chaîne arête-disjointe entre u et v.

On peut donc en déduire que la conception d'un réseau fiable est équivalent à la conception d'un réseau 2-connexe. On admet le théorème suivant (dit de Menger): un graphe G est 2-connexe si et seulement si toute coupe C de G contient au moins deux arêtes.
Par conséquent, le problème du sous-graphe fiable est équivalent à

$$(\mathcal{P}) \begin{cases} Min \quad \sum_{e \in E} c(e)x(e) & \\ \quad \sum_{e \in \delta(W)} x(e) \geq 2 \quad \text{pour tout ensemble } W \subset E \text{ tel que } \emptyset \neq W \neq V, & (\mathbf{1}) \\ \quad 0 \leq x(e) \leq 1 \quad \forall e \in E, & (\mathbf{2}) \\ \quad x(e) \text{ entier}, \quad \forall e \in E. & \end{cases}$$

On appelle *contraintes de coupes* les contraintes de type (2).

24.3) Quelles sont les caractéristiques de ce programme en terme de nombres de contraintes et de variables? Quels sont les possibilités pour résoudre un tel programme?

Dimension du polytope

Soit un graphe $G = (V, E)$ avec $n = |V|$ et $m = |E|$. On note $P_{2C}(G)$ les polytope des sous-graphes 2-connexes de G, i.e.

$$P_{2C}(G) = conv\{x^F \in I\!R^m \mid (V, F) \text{ est 2-connexe}\}.$$

24.4) Soit $G_1 = (V_1, E_1)$ un graphe 3-connexe, c'est-à-dire que pour toute arête e, le graphe $G_1 \setminus e$ est 2-connexe. Montrer que, dans ce cas, $P_{2C}(G_1)$ est de pleine dimension.

24.5) Soit G_2 un graphe qui est 2-connexe mais non 3-connexe.
- Montrer qu'il existe une arête e telle que l'égalité $x(e) = 1$ est valide pour $P_{2C}(G_2)$.
- On note E' l'ensemble de ces arêtes, que l'on appelle *essentielles*. En déduire une borne supérieure de la dimension de $P_{2C}(G_2)$.

- En montrant que le polyèdre $P' = P_{2C}(G_2) \cap \{x(e) = 1 \mid e \in E'\}$ est de pleine dimension, déduisez la dimension de $P_{2C}(G_2)$.

Exercice* 25 : *Facettes du polytope du sac-à-dos en 0-1
Considérons un problème de sac à dos P en 0-1 (0-1 knapsack polytope).

$$\begin{array}{rl} Max & c_1x_1 + c_2x_2 + ... + c_nx_n \\ & a_1x_1 + a_2x_2 + ... + a_nx_x \leq b \\ x_i \in \{0,1\} \text{ pour } i = 1, ..., n & \end{array}$$

On suppose que, $i = 1, ..., n$,
- $c_i > 0$ (sinon on ne prendrait jamais i dans une solution optimale).
- $a_i > 0$ (sinon, on prendrait systématiquement i dans une solution optimale).
- $a_i < b$ (sinon, on ne prendrait jamais i dans une solution).
Donc $b > 0$.
Enfin, on suppose que $sum_{i=1}^{n} a_i > b$ (sinon $\{1, \ldots, n\}$ serait trivialement la solution optimale).

On note le polytope du knapsack, comme étant l'enveloppe convexe des vecteurs entiers $\chi \in \{0,1\}^n$ qui satisfont la contrainte de sac-à-dos, *i.e.*

$$P_K(a, b) = conv\{\chi \in \{0,1\}^n \mid a\chi \leq b\}.$$

25.1) Donnez la dimension du polyèdre $P_K(a, b)$
Un sous-ensemble R de $\{1, ..., n\}$ est dit *recouvrement* (cover) de P si $\sum_{i \in R} a_i > b$. Dans un exercice précédent, on a prouvé que si R est un recouvrement de P alors la contrainte suivante (dite de recouvrement ou cover inequality) est valide pour $P_K(a, b)$

$$\sum_{i \in R} x_i \leq |R| - 1.$$

25.2) Soit un recouvrement donné R. Donnez une condition nécessaire pour qu'une contrainte de recouvrement définisse une facette de $P_K(a, b)$.

25.3) On considère un recouvrement particulier: celui contenant tous les items $\tilde{R} = \{1, \ldots, n\}$, c'est bien un recouvrement par hypothèse de l'exercice. Donnez une condition nécessaire et suffisante pour que la contrainte de recouvrement associée à $\tilde{R}$ définisse une facette de $P_K(a, b)$.

8 Caractérisation de polyèdres combinatoires

Exercice* 26 : *Etude polyédrale du problème du sous-graphe acyclique
Soit un graphe orienté $G = (V, A)$ où V est l'ensemble des sommets et A l'ensemble des arcs. On note uv un arc de A allant du sommet u au sommet v. On appelle *chemin* dans G une suite d'arcs $P = (u_0u_1, u_1u_2, ..., u_{k-1}u_k)$, on dit alors que P est de taille k. Un *circuit* C dans G est un chemin tel

que $u_0 = u_k$. On note $V(P)$ (resp. $V(C)$) l'ensemble des sommets impliqué dans un chemin (resp. un circuit).

Un graphe est dit *acyclique* s'il ne contient aucun circuit. Soit $W \subset V$ un sous-ensemble de sommets, on note $A(W)$ l'ensemble des arcs ayant leurs deux extrémités dans W. On dit alors que le graphe $(W, A(W))$ est le graphe induit par W.
Etant donnée une fonction $c : V \to I\!R$ qui associe à tout sommet $v \in V$ un poids $c(v)$, le *problème du sous-graphe acyclique induit* (PSAI) consiste à déterminer un sous-graphe acyclique induit $(W, A(W))$ de G tel que $c(W) = \sum_{v \in W} c(v)$ soit maximum.

Le problème du sous-graphe acyclique induit est équivalent au programme en nombres entiers (P) suivant

$$(\mathcal{P}) \left\{ \begin{array}{lll} Max & \displaystyle\sum_{u \in V} c(u)x(u) & \\ & \displaystyle\sum_{u \in W} x(u) \leq |W| - 1, \quad \forall W \subset V \text{ tel que } (W, A(W)) \text{ est un circuit} & (1) \\ & 0 \leq x(u) \leq 1, \qquad \forall u \in V & (2) \\ & x(u) \text{ entier}, \qquad \forall u \in V. & (3) \end{array} \right.$$

Etude polyèdrale générale

Soit un graphe orienté $G = (V, A)$ avec $n = |V|$. On note $P(G)$ le polytope des sous-graphes acycliques induits de G, i.e.

$$P(G) = conv\{\chi^W \in I\!R^n \mid (W, A(W)) \text{ est acyclique.}\}$$

26.1) Montrer que $P(G)$ est de pleine dimension.

26.2) Montrer que les contraines triviales $x(u) \geq 0$, $u \in V$, définissent des facettes de $P(G)$.

26.3) Soit un graphe $G_0 = (V, A)$ composé d'un unique circuit C_0 . Montrer que la contrainte de circuit associée à C_0 définit une facette de $P(G_0)$.

26.4) Soit un graphe $G_1 = (V, A)$ composé d'un unique circuit $C_1 = (v_1v_2, v_2v_3, ..., v_nv_1)$ tel qu'il existe deux sommets v_i et v_j non-consécutifs, $i \neq j$ soient reliés par un arc. Montrer que la contrainte de circuit correspondant à C_1 ne définit pas de facette de $P(G_1)$. En déduire une formulation réduite du PLNE (P).

Etude sur la diclique

Soit $K_n = (W, A_W)$ un graphe orienté di-complet (c'est-à-dire tel qu'il existe un arc reliant tout sommet de W à tout sommet de W). Lorsque K_n est un sous-graphe d'un graphe G, on appelle K_n une diclique de G. Le graphe K_4 est donné sur la figure 1.

26.5) Considérons le graphe K_4 et la solution y^* donnée par $y^*(u_i) = \frac{1}{2}$, $i = 1, ..., 4$. Montrez que y^* vérifie toutes les contraintes du programme (P) pour K_4 quand on relaxe la contrainte d'intégrité (3). Indiquez quelles conditions permettent de prouver que y^* est un sommet du polyèdre défini par les contraintes de ce programme pour K_4.

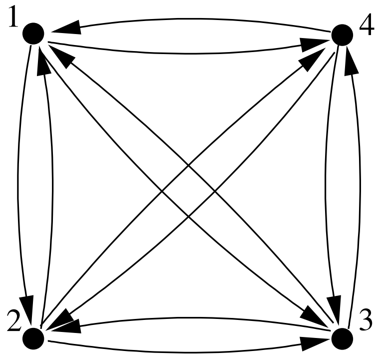


Figure 1: Diclique K_4

26.6) On s'intéresse maintenant à un graphe K_n quelconque. Prouver qu'un seul sommet de K_n peut appartenir à une solution de PSAI. En déduire une inégalité valide pour $P(G)$ lorsque G contient une diclique.

26.7) Soit $G = (V, A)$ un graphe orienté contenant une diclique $K = (W, A_W)$. On suppose que pour tout sommet u de $V \setminus W$, il existe un sommet v de W tel que l'arc uv ou l'arc vu n'existe pas dans A (cela revient à dire que K est maximale au sens de l'inclusion). Prouver que l'inégalité de la question précédente décrit alors une facette de $P(G)$. En déduire une nouvelle formulation pour P.

Caractérisation d'un cas particulier d'un unique circuit

Considérons à nouveau le graphe $\mathcal{C} = (V, E)$ composé d'un unique circuit avec $|V| \geq 2$. On veut montrer que le polytope $P(\mathcal{C})$ est donné par les contraintes triviales et l'unique contrainte de circuit.

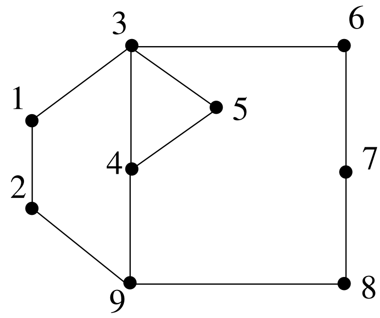
26.8) Quelle serait la conséquence de ce résultat par rapport au programme (P) correspondant au graphe $\mathcal{C}$?

26.9) On suppose qu'il existe une inégalité $ax \leq \alpha$ définissant une facette de $P(\mathcal{C})$ qui soit différente des inégalités (2) et (1).

- **i)** En utilisant le fait que $ax \leq \alpha$ n'est pas une contrainte triviale, montrer que $a(u) \geq 0$ pour tout $u \in V$.
- **ii)** En utilisant le fait que $ax \leq \alpha$ n'est pas la contrainte de circuit, montrer qu'il existe un sommet u_0 tel que $a(u_0) = 0$.
- **iii)** En utilisant le fait que $ax \leq \alpha$ n'est pas une contrainte triviale, montrer que, pour tout couple de sommets u et v distincts du circuit, $a(u) \leq a(v)$.
- **iv)** En déduire que $ax \leq \alpha$ ne définit pas une facette de $P(\mathcal{C})$.

Exercice* 27 : *Caractérisation partielle du polytope du stable

Considérons le graphe $H_{SP} = (V, E)$ suivant (le nom de ce graphe provient du fait que ce graphe est Série-Parallèle).



Le but de cet exercice est de montrer que le polytope du stable pour ce graphe est donné par les contraintes (3), (1) et (2).

$$
\begin{aligned}
&\text{Max} \sum_{u \in V} c(u)x(u) && \\
&x(u) + x(v) \leq 1 && \text{pour tout } uv \in E, && (1)\\
&\sum_{u \in V(C)} x(u) \leq \frac{|C|-1}{2} && \text{pour tout cycle impair } C, && (2)\\
&0 \leq x(u) \leq 1 && \text{pour tout } u \in V, && (3)\\
&x(u) \text{ entier,} && \text{pour tout } u \in V. &&
\end{aligned}
$$

Soit $ax \leq \alpha$ une contrainte, différente des contraintes (1), (2) et (3), qui définit une facette de $P(H_{SP})$.

27.1) Montrer que $a_v \geq 0$ pour tout v de V.

27.2) Montrer que $a_1 = a_2$ et $a_6 = a_7 = a_8$.

27.3) Montrer que $a_5 = 0$ (Indication: supposer que les coefficients du cycle (3,4,5) sont tous strictement positifs et utiliser le fait que la contrainte $ax \leq \alpha$ est différente de la contrainte (2) associée à ce cycle). Montrer de même qu'au moins une des coefficients du cycle (1,2,9,8,7,6,3) est nul.

Pour la fin de cette preuve, on considère le *graphe support* de l'inégalité $ax \leq \alpha$, c'est-à-dire le sous-graphe induit par les sommets dont les coefficients sont strictement positifs dans a.

27.4) Montrer que H_a ne contient pas d'arête pendante, c'est-à-dire une arête uv où u est de degré 1 dans H_a. (Indication: on utilisera le fait que l'inégalité $ax \leq \alpha$ est différente de l'inégalité d'arête pour uv).

27.5) Montrer que H_a est connexe.

27.6) En déduire que le polytope du stable pour H_{SP} est donné par les contraintes (1), (2) et (3). (Indication: partir de la propriété sur la nullité d'un des coefficients du cycle $(1, 2, 9, 8, 7, 6, 3)$ et regarder la structure du graphe H_a résultant).

9 Génération de colonnes

Exercice* 28 : *Tournées de véhicules

Modélisation

On considère le problème pratique suivant appelé *problème de tournées de véhicules* (Capacitated Vehicle Routing Problem (CVRP), en anglais).

Une entreprise laitière récolte du lait dans les fermes autour de son usine d'emballage. Elle dispose pour cela d'un grand nombre de véhicules identiques possédant chacun une citerne réfrigérée. Ces véhicules sont tous parqués le matin dans l'enceinte de l'usine. La contenance de chacune de ces citernes est de Q litres. L'entreprise collecte le lait dans n fermes, numérotées de 1 à n, dont chacune possède une production quotidienne de lait d_i, $i = 1, ..., n$.
Pour accéder de l'usine aux fermes et des fermes entre elles, les véhicules peuvent emprunter toutes les routes du réseau routier de la région. On considère ici que le temps de trajet entre deux points de ce réseau routier est le même dans un sens ou dans l'autre. L'entreprise désire minimiser le temps total de l'acheminement du lait jusqu'à son usine pour assurer la plus grande fraîcheur à son lait.

On considère à présent le problème de recherche de cycles dans un graphe complet qui peut se décrire de la façon suivante.
Soit un graphe non orienté complet K possédant $n+1$ sommets notés $V = \{0, 1, ..., n\}$. A chaque arête (i, j) de K, on associe un *coût* w_{ij} strictement positif véfifiant les inégalités triangulaires $w_{ij} \leq w_{ik} + w_{kj} \quad \forall i, j, k \in V$ (par exemple une distance euclidienne).
A chaque sommet $i \in V$, on associe une valeur réelle positive d_i appelée *demande* du sommet i. On définit également une valeur positive Q qui est inférieure à $\sum_{i=1}^{n} d_i$.
On appelle *Q-cycle* un cycle non vide de K passant par le sommet 0 et dont la somme des demandes des sommets est inférieure à Q. Comme le graphe K est complet, on peut écrire un Q-cycle C comme une séquence $(u_0, u_1, ..., u_p)$ de sommets où $u_0 = 0$ et $\displaystyle\sum_{l=1}^{p} d_{u_l} \leq Q$.
On note alors le coût d'un Q-cycle comme la somme du coût de ses arêtes, c'est-à-dire

$$w(C) = \sum_{l=0}^{p-1} w_{u_l u_{l+1}} + w_{u_p u_0}.$$

Le *problème de couverture par des Q-cycles* (PCQC) du graphe K consiste à déterminer des Q-cycles $(C_1, ..., C_k)$ tels que
- les Q-cycles n'ont aucun sommet en commun en dehors du sommet 0
- la somme des coûts des Q-cycles est minimum.

28.1) Montrer que le problème de tournée de véhicules se ramène au PCQC.

28.2) Lorsque $Q = \sum_{i=1}^{n} d_i$, à quel problème célèbre est équivalent le PCQC? Que peut-on en déduire quant à la complexité du problème PCQC?

Formulation à nombre exponentiel de contraintes

On considère des variables binaires x_{ij} associées aux arêtes (i,j) de K. Soit le PLNE suivant

$$(P)\left\{\begin{array}{lll} Min & \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}x_{ij} & \\ & \displaystyle\sum_{j=1}^{n} x_{ij} = 2 & \forall i \in \{1,...,n\}, \\ & \displaystyle\sum_{i\in W}\ \sum_{j\in\{0,1,...,n\}\setminus W} x_{ij} \geq 2\left\lceil \frac{\sum_{i\in W} d_i}{Q}\right\rceil & \forall W \subset \{1,...,n\}, W \neq \emptyset, \\ & x_{ij} \in \{0,1\} & \forall i \neq j \in \{0,1,...,n\}. \end{array}\right.$$

Les premières contraintes sont appelées *contraintes de degré* et les deuxièmes *contraintes de coupes.* On veut montrer que cette formulation est équivalente au problème PCQC.

28.3) **a)** Soit $T = (C_1,...,C_k)$, $k \geq 1$, un ensemble de Q-cycles couvrant les sommets de K. On pose le vecteur d'incidence χ^T indicé sur les arêtes de K de la manière suivante:

$$\chi^T_{ij} = \left\{\begin{array}{l} 1 \text{ si l'arête } (i,j) \text{ est dans la solution } T \\ 0 \text{ sinon.} \end{array}\right.$$

Montrer que χ^T vérifie toutes les contraintes de la formulation (P).

28.4) Soit x^* une solution quelconque de la formulation (P). On pose alors le graphe $H^* = (\{0,1,...,n\}, E^*)$ où $E^* = \{(i,j) \mid x^*_{ij} = 1\}$. On veut montrer que E^* est un ensemble de Q-cycles couvrant les sommets de K.

- **i)** Soit i_0 un sommet quelconque parmi $\{1,...,n\}$. Montrer qu'il existe dans E^* un chemin d'au moins 2 arêtes passant par i_0.
- **ii)** Pour un chemin dans K, on appelle *sommets intérieurs* les sommets de μ qui ne sont pas une extrémité de μ. Soit μ le chemin passant par i_0 et possédant le plus de sommets intérieurs distincts. Montrer que μ est en fait un cycle.
- **iii)** Montrer que le cycle μ passe nécessairement par le sommet 0.
- **iv)** Montrer que μ est un Q-cycle.

28.5) En conclure que la formulation (P) est bien équivalente au problème PCQC.

Formulation à nombre exponentiel de variables

On désire proposer une formulation PLNE pour ce problème. On considère pour cela l'ensemble $\mathcal{C}$ de tous les Q-cycles du graphe K.

28.6) Pour chaque Q-cycles $C \in \mathcal{C}$, on associe une variable binaire t_C. Montrer que la formulation PLNE (F) suivant est équivalent au problème PCQC.

$$(F) \left\{ \begin{array}{lll} Min & \sum_{C \in \mathcal{C}} w(C) t_C & \\ & \displaystyle\sum_{C \in \mathcal{C} \ | \ i \in C} t_C = 1 & \forall i \in \{1, ..., n\} \quad (1) \\ & t_C \geq 0, & \forall C \in \mathcal{C}, \\ & t_C \in I\!N & \forall C \in \mathcal{C}. \end{array} \right.$$

28.7) Indiquer le nombre de variables et d'inégalités que possède cette formulation? Quelle solution algorithmique peut-on envisager pour la résoudre?

28.8) On considère $(\tilde{F})$ la relaxation linéaire de (F). On suppose que l'on dispose d'une première solution réalisable pour le problème, c'est-à-dire un ensemble T' de Q-cycles couvrant tout le graphe K. On appelle alors $(\tilde{F}')$ le programme linéaire $(\tilde{F})$ restreint aux variables t_C correspondant aux Q-cycles de T' et on note $t' = (t'_C)_{C \in T'}$ la solution optimale de $\tilde{F}'$. Enfin on note $(\lambda_i)_{i \in \{1,...,n\}}$ les variables duales associées aux inégalités (1) de la formulation $(\tilde{F})$.
Définissez le problème de pricing pour cette formulation.

28.9) On dispose pour cette question d'une fonction `oracle(`ω`)` où ω est un poids quelconque associé aux arêtes de K (c'est-à-dire des poids pouvant être positifs ou négatifs): cette fonction retourne un Q-cycle C de K passant par le sommet 0 et tel que la somme des poids des arêtes de C est minimale. Cette fonction résoud donc le problème de recherche d'un plus petit Q-cycle passant par 0 avec des coûts d'arêtes quelconque (il s'agit d'un problème NP-difficile mais pseudo-polynomial que l'on sait bien résoudre en pratique).
Décrire, en le justifiant, un algorithme de génération de colonnes permettant de résoudre la relaxation linéaire $\tilde{F}$.
(Indication: on posera un poids ω_{ij} à partir de w_{ij}, λ_i et λ_j).