

Master d'Informatique - Spécialité Androide Module MAOA

Recherche Opérationnelle et Optimisation Combinatoire

Partie A - PLNE compact et solveurs

Pierre Fouilhoux

Sorbonne Université

2019-2020

1. Programmation linéaire (en nombres entiers)

2. Formulation compacte et non-compacte

3. Techniques de modélisation

4. Non-linéarité

5. Linéarisation

2/62

Recherche Opérationnelle et Optimisation Combinatoire Partie A - PLNE compact et solveurs
└ Programmation linéaire (en nombres entiers)

Recherche Opérationnelle et Optimisation Combinatoire Partie A - PLNE compact et solveurs
└ Programmation linéaire (en nombres entiers)

1. Programmation linéaire (en nombres entiers)

- 1.1 Complexité
- 1.2 Solveurs PL/PLNE
- 1.3 Difficulté de la PLNE

2. Formulation compacte et non-compacte

3. Techniques de modélisation

4. Non-linéarité

5. Linéarisation

Programme linéaire en nombres entiers (PLNE)

Ce cas étant largement le plus étudié, on l'appelle parfois même simplement *Programme entier* (ou *mixte*) (Integer Program ou Mixed Integer Program) (MIP).

$$\begin{aligned} & \text{Maximiser } c_1^T x_1 + c_2^T x_2 \\ & \text{sous les contraintes} \\ & A_1 x_1 + A_2 x_2 \leq b \\ & x_1 \in \mathbf{R}^{n_1} \\ & x_2 \in \mathbf{Z}^{n_2}. \end{aligned}$$

où c_1, c_2 sont des vecteurs et A_1 et A_2 des matrices avec x_1 partie continue de la solution et x_2 partie entière de la solution.

Les contraintes $x_2 \in \mathbf{Z}^{n_2}$ sont les *contraintes d'intégrité* (ou d'entiéreté en Belgique ou d'intégralité au Québec), appelées integrity or integrality constraint en anglais.

Complexité

Un problème linéaire continu peut être résolu en temps polynomial par la **méthode des ellipsoïdes** (Khachiyan 1979).

Il existe des algorithmes polynomiaux efficaces pour résoudre un programme linéaire : les algorithmes dits de **points intérieurs** initiés par Karmarkar (1984).

Néanmoins l'**algorithme du simplexe** (Dantzig 1947) est le plus célèbre (et le plus efficace dans le cas général) des algorithmes de résolution, bien qu'il ne soit pas polynomial !

L'algorithme du simplexe repose sur le fait qu'une solution optimale d'un programme linéaire peut être prise parmi les sommets du polyèdre de \mathbf{R}^n déterminé par $Ax \leq b$: on ramène ainsi un problème d'optimisation continue à un problème combinatoire !!!

En revanche, la PLNE est un problème NP-difficile. Il est facile de montrer que la PLNE est un problème NP-difficile car de nombreux problèmes NP-difficiles peuvent être exprimés comme des PLNE.

5/62

Solveurs PL/PLNE

Il existe de nombreux solveurs de PL : des solveurs commerciaux Cplex (IBM), Xpress, Gurobi, et même Matlab ou Excel... ; des solveurs académiques Lp de COIN-OR, Soplex de la ZIB ; et des solveurs libres comme Glpk (gnu). Les meilleurs d'entre eux peuvent résoudre des PL jusqu'à 200000 variables et 200000 contraintes en quelques secondes.

En revanche, les solveurs entiers performants sont beaucoup moins performants : ils sont en général liés aux solveurs PL : Glpk par exemple ne dépassent pas quelques 100 aine de variables et contraintes ; les solveurs commerciaux Cplex ou Gurobi sont les plus performants (Xpress est un peu en-dessous) pouvant réussir parfois quelques milliers de variables/contraintes ; un solveur "universitaire" les rattrape : SCIP de la ZIB. Un des objectifs de ce cours est de comprendre comment et dans quels cas ces solveurs atteignent de telles capacités.

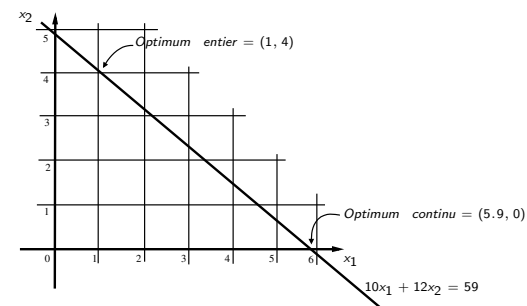
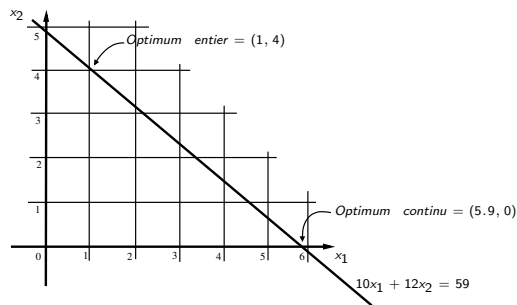
6/62

La première remarque qui peut sauter aux yeux est d'imaginer que résoudre un PLNE revient à "arrondir" la solution de sa relaxation continue. L'exemple suivant témoigne de l'insuffisance de cette remarque :

Prenons un PLNE à deux variables et une seule contrainte, ce qui constitue le cas le plus simple que l'on puisse imaginer.

$$\begin{aligned} & \text{Maximiser } 10x_1 + 11x_2 \\ & 10x_1 + 12x_2 \leq 59 \\ & x_1 \text{ et } x_2 \geq 0 \\ & x_1, x_2 \text{ entiers.} \end{aligned}$$

En dessinant le domaine de définition, on obtient la figure suivante :



On remarque alors que l'optimum de la relaxation continue a une valeur objective de 59 et celui de l'optimum entier est de 54 seulement. Mais surtout, on peut noter l'écart complet de structure et de position des deux points optimum (qui sont ici chacun solution optimale unique).

7/62

8/62

1. Programmation linéaire (en nombres entiers)

2. Formulation compacte et non-compacte

- 2.1 Problème d'OC et formulation PLNE
- 2.2 Problème du sac-à-dos 0/1
- 2.3 Recouvrement, pavage et partition
- 2.4 Problème du stable
- 2.5 Problème du voyageur de commerce
- 2.6 Problème de coloration

3. Techniques de modélisation

4. Non-linéarité

5. Linéarisation

9/62

10/62

Formulation algébrique "naïve"

Associer une variable à chaque solution :
 $t_F = 1$ si la solution $F \in \mathcal{F}$ est prise et 0 sinon

$$\begin{aligned} \text{Max } & \sum_{F \in \mathcal{F}} c(F) t_F \\ & \sum_{F \in \mathcal{F}} t_F \leq 1 \\ & t_F \in \{0, 1\} \text{ pour tout } F \in \mathcal{F}. \end{aligned}$$

Cette formulation prouve que tout problème d'Optimisation combinatoire peut s'écrire sous forme d'un PLNE!
Mais peut-on résoudre une telle formulation ayant nombre de variables égales au nombre de solutions ?

11/62

Problème d'optimisation combinatoire

Déterminer un plus grand (petit) élément dans un ensemble fini valué.

Etant donné :

- un ensemble fini d'éléments $E = \{e_1, \dots, e_n\}$
- un vecteur-poids $c = (c(e_1), \dots, c(e_n))$ associé à E
- une famille \mathcal{F} de sous-ensembles de E , les **solutions**.

Un problème d'optimisation ("linéaire") consiste à trouver un ensemble $F \in \mathcal{F}$ de poids $c(F) = \sum_{e \in F} c(e)$ maximum (ou minimum), *i.e.*

$$\text{max ou min } \{c(F) \mid F \in \mathcal{F}\}.$$

Formulation algébrique "naturelle"

Associer une variable 0-1 à chaque élément E :
 $x_e = 1$ si l'élément e est pris dans la solution et 0 sinon

$$\begin{aligned} \text{Max } & \sum_{e \in E} c(e) x_e \\ & Ax \leq B \\ & x_e \in \{0, 1\} \text{ pour tout } e \in E. \end{aligned}$$

Cette formulation PLNE demande :
- de pouvoir définir par des inégalités $Ax \leq B$ le fait que les solutions décrites par la variable doivent être dans l'ensemble \mathcal{F} .
- de pouvoir résoudre la formulation ainsi obtenue.
Ce n'est pas toujours possible...

12/62

Exemple : le problème du sac-à-dos (knapsack) 0/1

Données : n objets $i = 1, \dots, n$, de bénéfice c_i et de poids a_i ,

Objectif : Ranger les objets dans un "sac" de poids maximum b avec un bénéfice maximal.

Le **problème de sac-à-dos (knapsack)** consiste à choisir les objets à prendre parmi les n objets de manière à avoir un bénéfice maximal et respecter la contrainte du poids à ne pas dépasser.

Ce problème se rencontre bien entendu dès que l'on part en randonnée en voulant emmener le plus possible d'objets utiles (nourriture, boissons,...). Mais ce problème est plus fréquemment utilisé pour remplir les camions de transport, les avions ou bateaux de fret et même pour gérer la mémoire d'un microprocesseur.

La formulation PLNE du problème de sac-à-dos est très simple. On utilise pour chaque objet $i \in \{1, \dots, n\}$, une variable entière x_i correspondant au nombre de fois où l'objet i est choisi.

$$\begin{aligned} \text{Max } & \sum_{i=1}^n c_i x_i \\ & \sum_{i=1}^n a_i x_i \leq b, \\ & x_i \in \{0, 1\}, \text{ pour } i = 1, \dots, n. \end{aligned}$$

L'unique contrainte est dite *contrainte de sac-à-dos*.

Elle est l'unique contrainte de ce problème qui est pourtant NP-difficile!

Recouvrement, pavage et partition

Soit $E = \{1, \dots, n\}$ un ensemble fini d'éléments.

Soit E_1, \dots, E_m des sous-ensembles de E .

A chaque ensemble E_j on associe un poids c_j , $j = 1, \dots, m$.

Une famille $F \subseteq \{E_1, \dots, E_m\}$ est dite

- un **recouvrement** de E si $\cup_{E_j \in F} E_j = E$, pour tout $j \in \{1, \dots, m\}$,

- un **pavage** de E si $E_j \cap E_k = \emptyset$, pour tout $j \neq k \in \{1, \dots, m\}$,

- une **partition** de E si F est à la fois un recouvrement et un pavage.

On peut interpréter les contraintes du problème de recouvrement (resp. pavage, partition) comme le fait qu'un élément de E doit être pris au moins une fois (resp. au plus une fois, exactement une fois).

Le **problème de recouvrement** (resp. *pavage*, *partition*) consiste à déterminer un recouvrement (resp. pavage, partition) dont la somme des poids des ensembles qui le forment est de poids minimum (resp. maximum, minimum/maximum).

Recouvrement, pavage et partition

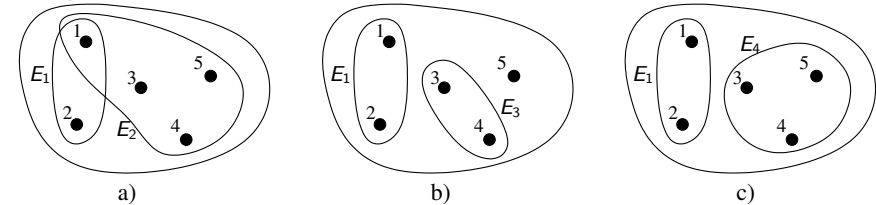
Exemple

Données :

- 5 éléments $E = \{1, 2, 3, 4, 5\}$

- et 4 sous-ensembles $E_1 = \{1, 2\}$, $E_2 = \{1, 3, 4, 5\}$, $E_3 = \{3, 4\}$ et $E_4 = \{3, 4, 5\}$.

Quelques solutions :



a) $\{E_1, E_2\}$ est un recouvrement

b) $\{E_1, E_3\}$ est un pavage

c) $\{E_1, E_4\}$ est une partition.

Recouvrement, pavage et partition

Modélisation :

variables binaires x_1, \dots, x_m associées aux sous-ensembles E_1, \dots, E_m .

Pour cela, on considère A la matrice en 0-1

dont les lignes correspondent aux éléments $1, \dots, n$

et les colonnes aux sous-ensembles E_1, \dots, E_m

et dont les coefficients sont $A_{ij} = 1$ si $i \in E_j$ et 0 sinon.

Recouvrement

Pavage

Partition

$$\text{Min } \sum_{j=1}^m c_j x_j$$

$$Ax \geq \mathbb{1}$$

$$x \in \{0, 1\}^m$$

$$\text{Max } \sum_{j=1}^m c_j x_j$$

$$Ax \leq \mathbb{1}$$

$$x \in \{0, 1\}^m$$

$$\text{Max (ou Min) } \sum_{j=1}^m c_j x_j$$

$$Ax = \mathbb{1}$$

$$x \in \{0, 1\}^m$$

où $\mathbb{1}$ est un vecteur dont chaque composante est 1.

Recouvrement, pavage et partition

Dans l'exemple avec 4 sous-ensembles

$E_1 = \{1, 2\}$, $E_2 = \{1, 3, 4, 5\}$, $E_3 = \{3, 4\}$ et $E_4 = \{3, 4, 5\}$.

La matrice A est

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

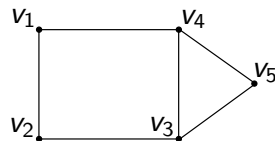
Pour le cas recouvrement, la première inégalité du PLNE est alors

$$x_1 + x_2 \geq 1$$

correspondant au fait qu'il faut choisir E_1 et/ou E_2 pour couvrir l'élément 1.

Exemple : le problème du stable

- $G = (V, E)$ un graphe non-orienté et $c(v)$ poids d'un sommet.
- Un **stable** de G est un sous-ensemble S de sommets de V tel qu'il n'existe aucune arête de E entre 2 sommets de S .
- **Problème du stable de poids maximum** : déterminer un stable S de G tel que $c(S) = \sum_{v \in S} c(v)$ soit maximum.



Problème NP-difficile !

Formulation compacte pour le problème du stable

$x(v) = 1$ si le sommet v est pris dans la solution et 0 sinon

$$\text{Max } \sum_{u \in V} c(u)x(u)$$

$$x(u) + x(v) \leq 1, \quad \text{pour tout } uv \in E,$$

$$x(u) \in \{0, 1\}, \quad \text{pour tout } u \in V.$$

L'inégalité $x(u) + x(v) \leq 1$ est appelée **inégalité aux arêtes**.

Formulation **compacte** i.e. possède un nombre polynomial de contraintes et de variables.

Formulation PLNE difficile à résoudre par Branch-and-Bound.

Aperçu de la suite du cours

Comment “renforcer” une formulation PLNE

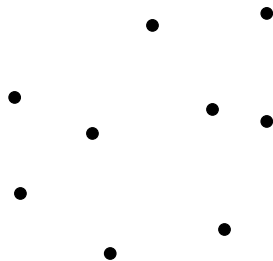
Une **clique** est un ensemble de sommets induisant un sous-graphe complet.
 Or comme il y a au plus 1 sommet dans une clique K dans un stable, l'inégalité

$$\sum_{u \in K} x(u) \leq 1 \quad \text{pour toute clique } K$$

est vérifiée pour tout stable.

Modélisation du problème du voyageur de commerce (TSP)

Données : n villes avec c_{ij} coût de transport de i à j (asymétrique).
Objectif : Déterminer un circuit passant par toutes les villes de plus petit coût.



Formulation non-compacte pour le problème du stable

On peut alors proposer une autre formulation non-compacte pour le problème du stable

$$\begin{aligned} \text{Max } & \sum_{u \in V} c(u)x(u) \\ & \sum_{u \in K} x(u) \leq 1 \quad \text{pour toute clique } K, \\ & x(u) \in \{0, 1\} \quad \text{pour tout } u \in V. \end{aligned}$$

Il y a un nombre exponentiel de cliques dans un graphe... donc cette formulation a un nombre exponentiel de contraintes...

Est-ce que cette formulation est “meilleure” ?

Voir cours “Branchement”.

Parmi toutes ces inégalités, sont-elles toutes utiles ?

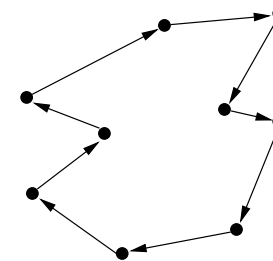
Voir cours “Approches polyédrales”

Comment résoudre une telle formulation ?

Voir cours “Algorithmes de coupes”

Modélisation du problème du voyageur de commerce (TSP)

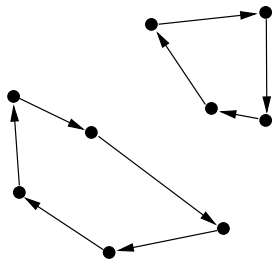
Données : n villes avec c_{ij} coût de transport de i à j (asymétrique).
Objectif : Déterminer un circuit passant par toutes les villes de plus petit coût.



Formulation en variables naturelles

$x_{ij} = 1$ si l'arc (i, j) est dans le circuit et 0 sinon.

$$\begin{aligned} \text{Min } & \sum_{i,j} c_{ij} x_{ij} \\ & \sum_{j \in V} x_{ij} = 1 \quad \forall i \in V, \\ & \sum_{i \in V} x_{ij} = 1 \quad \forall j \in V, \\ & x_{ij} \in \mathbf{N} \quad \forall i \in V, j \in V \setminus \{i\}. \end{aligned}$$



On peut ajouter également les inégalités $x_{ij} + x_{ji} \leq 1$.

25/62

“Elimination des sous-tours” par la formulation MTZ

Formulation de Miller-Tucker-Zemlin (MTZ).

Ajout des variables réelles $u_i, i = 1, \dots, n$, associées aux villes

Ajout des contraintes :

$$\begin{aligned} u_1 &= 1, \\ 2 \leq u_i &\leq n \quad \forall i \in V \setminus \{1\}, \\ u_i - u_j + 1 &\leq n(1 - x_{ij}) \quad \forall i \in V \setminus \{1\}, j \in V \setminus \{1, i\}. \end{aligned}$$

On appelle ces dernière inégalités *inégalités MTZ*.

27/62

Formulation en variables naturelles

- Relaxation linéaire de ce PLNE entière dans de nombreux cas de graphes (la matrice est totalement unimodulaire pour les graphes bipartis).
- Malheureusement, la solutions peut être faite de plusieurs cycles orientées (appelées **sous-tours**).

Il faut ajouter des inégalités pour “briser les sous-tours”.

26/62

“Elimination des sous-tours” par la formulation MTZ

Elles permettent d'éliminer les sous-tours :

- ▶ pour tout arc (i, j) où $x_{ij} = 1$, elles forcent $u_j \geq u_i + 1$, (dans le cas où $x_{ij} = 0$, elles restent valides car indiquent alors $u_i - u_j \leq n - 1$ qui est toujours vrai).
- ▶ si une solution entière de la formulation contient plus d'un sous-tour, alors l'un deux au moins ne contient pas le sommet 1 et, sur ce sous-tour, les variables u_i s'incrémentent à l'infini.

Remarque : dans le cas d'une solution réalisable pour la formulation, les variables $u_i, i = 1, \dots, n$ indiquent la position de la ville i dans le tour.

28/62

“Elimination des sous-tours” par la formulation MTZ

Cette formulation possède n variables (continues) de plus, mais elle a l’énorme avantage d’être compacte !

On peut donc directement utiliser une procédure de branchement et séparation (Branch& Bound) et donc un solveur entier.

En revanche, c’est une **Très Très mauvaise formulation**

car elle possède une mauvaise relaxation linéaire (voir chapitre Branchement).

29/62

“Elimination des sous-tours” par des flots

Formulation utilisant des flots.

Ajout des variables de flots réelles z_{ij} , pour tout $i \in V, j \in V \setminus \{1, i\}$.

Ajout des contraintes :

$$\begin{aligned} \sum_{j \in V \setminus \{1\}} z_{1j} &= |V| - 1, \\ \sum_{j \in V \setminus \{1, i\}} z_{ij} + 1 &= \sum_{j \in V \setminus \{i\}} z_{ji} \quad \forall i \in V \setminus \{1\}, \\ z_{ij} + z_{ji} &\leq (|V| - 1)(x_{ij} + x_{ji}) \quad \forall i \in V, j \in V \setminus \{1, i\}, \\ z_{ij} &\geq 0 \quad \forall i \in V, j \in V \setminus \{1, i\}. \end{aligned}$$

On peut remarquer que les variables z porte un flot de valeur $|V| - 1$ sortant du sommet 1, mais qu’il n’y pas d’inégalité de flot entrant au sommet 1.

30/62

“Elimination des sous-tours” par des flots

Ces inégalités permettent d’éliminer les sous-tours.

Preuve : En effet, pour x entier, le PLNE devient un PL de flot dans un graphe limité aux arcs sélectionnés par x . Supposons que ce graphe possède un sous-tour qui contient donc au moins 2 sommets. Le sommet 1 est dans un circuit ne passant pas par tous les sommets. Notons C le circuit passant par 1 qui contient donc au plus $|V| - 2$ sommets. En sortant de 1, le flot vaut $|V| - 1$ et à chaque sommet successif de C , le flot diminue d’une unité. Au dernier sommet i de C avant 1, le flot entrant est donc strictement supérieur à 3 et la somme des arcs sortant de i est strictement supérieure à 2 : ce qui est impossible car le seul arc sortant de 1 retenu dans le graphe est $(i, 1)$ qui n’est pas associé à une variable z : donc tous les z sortant de i sont nuls. Contradiction.

Cette formulation possède n^2 variables (continues) de plus et elle est encore compacte ! On peut l’utiliser directement dans un solveur entier.

La valeur de relaxation linéaire est bien meilleure que celle utilisant MTZ. En revanche, les contraintes liant z et x (contraintes de “big M”) ne permettent pas d’avoir une très bonne relaxation.

31/62

“Elimination des sous-tours” par des flots désagrégés

Formulation utilisant des flots “désagrégés” (dite de Maculan).

Ajout de $|V|$ ensemble de variables de flots réelles z_{ij}^k , pour tout $i \in V, j \in V \setminus \{1, i\}$

et pour tout $k \in V \setminus \{1\}$.

Ajout des contraintes :

$$\begin{aligned} \sum_{j \in V \setminus \{1\}} z_{1j}^k &= 1 \quad \forall k \in V \setminus \{1\} \\ \sum_{j \in V \setminus \{1, i\}} z_{ij}^k &= \sum_{j \in V \setminus \{i\}} z_{ji}^k \quad \forall k \in V \setminus \{1\}, \forall i \in V \setminus \{1, k\}, \\ \sum_{j \in V \setminus \{1, k\}} z_{kj}^k + 1 &= \sum_{j \in V \setminus \{k\}} z_{jk}^k \quad \forall k \in V \setminus \{1\}, \\ z_{ij}^k + z_{ji}^k &\leq x_{ij} + x_{ji} \quad \forall i \in V, j \in V \setminus \{1, i\}, \forall k \in V \setminus \{1\} \\ z_{ij}^k &\geq 0 \quad \forall i \in V, j \in V \setminus \{1, i\}, \forall k \in V \setminus \{1\}. \end{aligned}$$

On peut remarquer que chaque ensemble de variables z^k porte un flot de valeur 1 sortant du sommet 1, mais qu’il n’y pas d’inégalité de flot entrant au sommet 1.

32/62

“Elimination des sous-tours” par des flots désagrégés

Ces inégalités permettent d'éliminer les sous-tours.

Preuve : En effet, pour x entier, le PLNE devient un PL de k flots indépendants circulant sur des graphes limités aux arcs où x est non nul. S'il existe un sommet k qui n'est pas dans le circuit issu du sommet 1, le flot z^k partant de 1 reste à la valeur 1 jusqu'au dernier sommet i de C avant 1. Le flot sortant de 1 doit être porté par un z^k mais ils sont tous nuls car le seul arc de x non nul est $(i, 1)$. Contradiction.

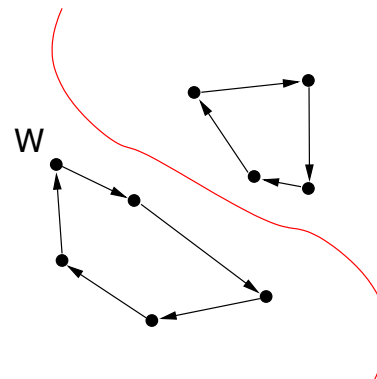
En fait la formulation par les flots précédentes peut être vues comme une agrégation des k flots z^k en un seul. On dit donc que cette formulation est “désagrégée” par rapport à la précédente. Son but est de faire disparaître les inégalités de “big M” reliant z et x .

Cette formulation a une très bonne valeur de relaxation. Toutefois elle possède n^3 variables (continues) supplémentaires ainsi qu'un nombre d'inégalité en n^2 ...

“Elimination des sous-tours” par la connexité

Théorème de Menger : un graphe est fortement connexe si et seulement si toute coupe du graphe contient au moins un arc.

$$\sum_{e \in \delta^+(W)} x(e) \geq 1, \text{ pour tout } W \subsetneq V \text{ et } W \neq \emptyset,$$

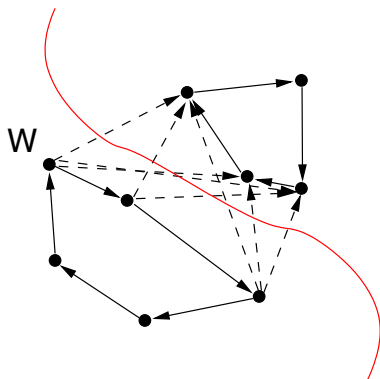


où $\delta^+(W) = \{(i, j) \mid i \in W \text{ et } j \notin W\}$ est la coupe sortante de W .

“Elimination des sous-tours” par la connexité

Théorème de Menger : un graphe est fortement connexe si et seulement si toute coupe du graphe contient au moins un arc.

$$\sum_{e \in \delta^+(W)} x(e) \geq 1, \text{ pour tout } W \subsetneq V \text{ et } W \neq \emptyset,$$



où $\delta^+(W) = \{(i, j) \mid i \in W \text{ et } j \notin W\}$ est la coupe sortante de W .

Ce n'est plus une formulation compacte : elle possède un **nombre exponentiel de contraintes** !

Formulation du TSP symétrique

$$\text{Min} \sum_{e \in E} c(e)x(e)$$

$$\sum_{e \in \delta(v)} x(e) = 2, \text{ pour tout } v \in V,$$

$$\sum_{e \in \delta(W)} x(e) \geq 2, \text{ pour tout } W \subsetneq V \text{ et } W \neq \emptyset,$$

$$x(e) \in \{0, 1\}, \text{ pour tout } e \in E.$$

Formulation “vedette” du TSP symétrique.

Renforcée par d'autres inégalités, elle est la technique record.

Coloration de graphe

Une **k-coloration** des sommets d'un graphe $G = (V, E)$ est une fonction $r : V \rightarrow \{1, \dots, k\}$ telle que $r(u) \neq r(v)$ pour tout couple de sommets adjacents u, v .

Le **problème de coloration** consiste à déterminer le plus petit k tel que G soit k -coloriable.

Première formulation

- on associe à chaque sommet u de V un vecteur binaire à K dimensions $x_u = (x_u^1, \dots, x_u^K)$, où K est une borne supérieure sur la coloration de G (au maximum $K = |V|$).
- on ajoute une variable binaire w_l par couleur $l = 1, \dots, K$ indiquant si cette couleur a été utilisée ou non.

Le problème est donc équivalent au programme

$$\begin{aligned} \text{Min } & \sum_{l=1}^K w_l \\ & \sum_{l=1}^K x_u^l = 1, \quad \text{pour tout } u \in V, & (1) \\ & x_u^l + x_v^l \leq w_l, \quad \text{pour tout } e = uv \in E \text{ et } 1 \leq l \leq K, & (2) \\ & x_u^l \in \{0, 1\}, \quad \text{pour tout } u \in V \text{ et } 1 \leq l \leq K. \end{aligned}$$

Cette formulation contient énormément de **symétrie** c'est-à-dire des solutions très "proches" de même coût.

Deuxième formulation

Remarque : une coloration est une partition des sommets en stables.

Soit \mathcal{S} l'ensemble des stables non vides de G .

On associe à chaque stable $S \in \mathcal{S}$ une variable binaire t_S .

Le problème de coloration est alors équivalent au programme en nombres entiers suivant (Mehrotra et Trick 1995).

$$\begin{aligned} \text{Min } & \sum_{S \in \mathcal{S}} t_S \\ & \sum_{S \in \mathcal{S} \mid u \in S} t_S = 1, \quad \text{pour tout } u \in V, & (3) \\ & t_S \in \{0, 1\}, \quad \text{pour tout } S \in \mathcal{S}. & (4) \end{aligned}$$

Ici, il n'y a pas de symétries artificielles dues à une numérotation des colorations.

Elle possède un nombre polynomial d'inégalités... mais un nombre exponentiel de variable. (Voir cour "Génération de colonnes")

En conclusion

Un problème d'optimisation combinatoire possède de nombreuses formulations :

- compacte
- avec un nombre exponentiel de contraintes
- avec un nombre exponentiel de variables
- avec un nombre exponentiel de variables et de contraintes !

Comment choisir une bonne formulation ?

Cela dépend du cadre algorithmique de résolution : il n'existe pas d'outils génériques "magiques" pour toutes ces formulations et tous les problèmes...

Nous verrons les principaux outils algorithmiques et les plus célèbres de ces problèmes.

1. Programmation linéaire (en nombres entiers)

2. Formulation compacte et non-compacte

3. Techniques de modélisation

4. Non-linéarité

5. Linéarisation

Cette section donne des pistes d'idées pour modéliser certains liens logiques entre des variables ou des contraintes du problème.

• *Cas simples*

Soient a, b et c des événements correspondant aux variables de décisions binaires x_a, x_b et x_c .

- ▶ Si a et b ne peuvent pas se produire tous les deux : $x_a + x_b \leq 1$.
- ▶ Si au moins des événement parmi a et b doit se produire : $x_a + x_b \geq 1$.
- ▶ Si a se produit alors b doit se produire : $x_a \leq x_b$.
- ▶ On peut noter que l'inégalité précédente modélise heureusement la contraposée de la proposition logique correspondante : si b ne se produit pas, a ne doit pas se produire.
- ▶ Si a se produit, alors b et/ou c doivent se produire : $x_a \leq x_b + x_c$.
- ▶ Si a ne se produit pas, alors une quantité y positive doit être nulle, sinon y est libre dans \mathbf{R} . On doit fixer une quantité M telle que y ne peut jamais être supérieure à M lorsqu'on atteint l'optimum du problème. Une telle constante M existe, car sinon le problème serait non borné. On peut écrire $y \leq Mx_a$ (contrainte dite de "big M").

• *Maximiser la valeur minimale d'un ensemble de fonctions linéaires*

Si on veut maximiser la plus petite valeur prise par un ensemble de fonctions linéaires $a^i x, i = 1, \dots, m$, il suffit d'ajouter une variable z et les contraintes $z \leq a^i x, i = 1, \dots, m$: la fonction objective devient alors Max z .

• Le "ou" numérique

On veut représenter une variable x devant prendre des valeurs soit 0, soit être plus grande que L où L et x sont bornées par une valeur M .

On ajoute une variable $y \in \{0, 1\}$ et on utilise les contraintes

$$x \geq Ly \quad \text{et} \quad x \leq My.$$

• *Satisfaire le plus possible d'inégalités*

On dispose d'un lot de n contraintes $a^1 x \leq b^1, a^2 x \leq b^2, \dots, a^n x \leq b^n$ qui potentiellement forment un ensemble de solutions vides. On souhaite une solution qui satisfasse le plus possible de contraintes. Pour chacune des contraintes $a^i x \leq b^i, i = 1, \dots, n$, on détermine une valeur M_i suffisamment grande pour que $a^i x \leq b^i + M_i$ soit satisfaite quelque soit x .

On pose alors y_1, \dots, y_n des variables binaires et ce cas de figure se modélise alors de la façon suivante.

$$\begin{array}{l} a^1 x \leq b^1 \\ a^2 x \leq b^2 \\ \dots \\ a^n x \leq b^n \end{array} \quad \Rightarrow \quad \begin{array}{l} a^1 x \leq b^1 + M_1 y_1 \\ a^2 x \leq b^2 + M_2 y_2 \\ \dots \\ a^n x \leq b^n + M_n y_n \\ \text{Min} \sum_{i=1}^n y_i \end{array}$$

Remarquons que si une seule contrainte doit être satisfaite parmi deux (c'est-à-dire si $k = 1$ et $n = 2$), on peut utiliser une seule variable binaire y en posant $y_1 = y$ et $y_2 = 1 - y$.

• *Implication entre contraintes*

Soit $a^1x \leq b^1$ et $a^2x \leq b^2$ deux contraintes telles que si $a^1x < b^1$, alors $a^2x \leq b^2$ doit être satisfaite, mais que, par contre, si $a^1x \geq b^1$, alors $a^2x \leq b^2$ peut ou non être satisfaite.

On prend M tel que $-a^1x \leq -b^1 + M$ et $a^2x \leq b^2 + M$ soient vérifiées pour toute valeurs de x . On peut utiliser une variable de décision binaire y et écrire alors :

$$-a^1x \leq -b^1 + M(1 - y) \tag{5}$$

$$a^2x \leq b^2 + My \tag{6}$$

En effet, si $a^1 < b$, alors la contrainte (5) implique que $y = 0$, et ainsi la contrainte (6) est équivalente à $a^2x \leq b^2$ qui doit donc être satisfaite. Pour le cas contraire (i.e. si $a^1x \geq b^1$), alors y peut prendre la valeur 0 ou 1, c'est-à-dire que $a^2x \leq b^2$ peut être satisfaite ou non.

1. Programmation linéaire (en nombres entiers)

2. Formulation compacte et non-compacte

3. Techniques de modélisation

4. Non-linéarité

4.1 Programmation mathématique

4.2 Convexification

5. Linéarisation

Programme mathématique

Un *Programme Mathématique* (mathematical program), noté PM, est un problème d'optimisation sous contrainte (\mathcal{P}) qui peut s'écrire de la façon suivante :

$$\begin{aligned} &\text{Maximiser } f(x) \\ &\text{sous les contraintes} \\ &g_i(x) \leq 0 \quad i = 1, \dots, m \\ &x \in S. \end{aligned}$$

où

- S est une partie de \mathbb{R}^n et x est un vecteur appelé *variable*, ces n composantes sont dites les *inconnues* du problème,
- la fonction $f : S \rightarrow \mathbb{R}$ est appelée *fonction objective* ou *objectif* (objective function),
- les fonctions $g_i : S \rightarrow \mathbb{R}$, $i = 1, \dots, m$, forment des inégalités qui sont appelées les *contraintes* (constraint) du problème.

On peut remarquer qu'un PM peut être une maximisation ou une minimisation (il suffit de poser la fonction $f' = -f$).

On appelle *inégalités* une contrainte $g_i(x) \leq 0$ ou $g_i(x) \geq 0$: en cas de présences des deux contraintes $g_i(x) \leq 0$ et $g_i(x) \geq 0$, on parle alors d'égalité $g_i(x) = 0$.

Un vecteur \bar{x} vérifiant les contraintes d'un PM est dit *solution* ou *solution réalisable* du PM. L'ensemble des solutions d'un PM forme un *domaine de définition*. Le domaine de définition d'un PM peut être : vide (dans ce cas, le problème n'admet pas de solutions), dans le cas contraire, le PM admet des solutions. Sous certaines conditions, il peut exister des solutions x^* dites *optimales*, c'est-à-dire qui maximisent la fonction $f(x)$ sur toutes les solutions du PM.

Plusieurs cas de PM sont à mettre en évidence :

- si l'ensemble S est continu, on parle de *programme mathématique continu* (continuus) ,
- si l'ensemble S est discret (c'est-à-dire isomorphe à \mathbb{N}^n), on parle de *programme mathématique discret* (discrete) que l'on notera ici (PMD) ; on le dit également *entier* (integer program) si $S \subset \mathbb{N}^n$ ou même *binaire* si $S \subset \{0, 1\}^n$. En fait tout PMD peut se ramener au cas d'un programme entier, voir même d'un programme binaire.
- si certaines composantes du vecteur x solution prennent leurs valeurs dans un ensemble discret et les autres dans un ensemble continu, on le dit *programme mathématique mixte*.

Dans le cas des programmes entiers (donc discrets également), on peut noter alors un (PMD) de la façon suivante :

$$\begin{aligned} & \text{Maximiser } f(x) \\ & \text{sous les contraintes} \\ & g_i(x) \leq 0 \quad i = 1, \dots, m \\ & x \in \mathbb{Z}^n. \end{aligned}$$

On désigne alors $x \in \mathbb{Z}^n$ comme étant la *contrainte d'intégrité* (ou d'entiereté en Belgique ou d'intégralité au Québec) (integrity or integrality constraint). On appelle *relaxation* le fait de "relâcher", c'est-à-dire supprimer une contrainte du problème. Ainsi, un programme relaxé désignera un programme où l'on aura supprimé une ou plusieurs contraintes.

On appelle *relaxation continue* le fait de "relâcher" les contraintes d'intégrité du problème. Par abus de langage, on appelle aussi souvent *relaxation continue* le fait de résoudre le programme que où l'on a relâché les contraintes d'intégrité (l'expression désigne même parfois la solution optimale obtenue).

Programme convexe

Soit un ensemble $S \subset \mathbb{R}^n$ convexe (un ensemble de points tels que tout segment entre deux points est entièrement dans l'ensemble convexe). Une fonction $f : S \rightarrow \mathbb{R}$ est dite *fonction convexe* si elle vérifie

$$\forall x_1, x_2 \in S, \forall \lambda \in [0, 1], f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2).$$

Une fonction est dite *concave* si la fonction $-f$ est convexe. Considérons alors le (PM) suivant

$$\begin{aligned} & \text{Minimiser } f(x) \\ & x \in S. \end{aligned}$$

Si la fonction objective f d'un (PMD) est convexe (ou concave) et que l'ensemble S est un convexe fermé non-discret, on parle de *programme convexe*. Pour un programme convexe, tout optimum local est global. Les cas non-convexe (ou non-concave pour une maximisation), n'ont pas toujours d'optimum global.

Pour le cas des programmes convexes, suivant les propriétés de la fonction f : (continuité, différentiabilité,...), il existe des algorithmes plus ou moins efficace pour déterminer le minimum d'une fonction (méthode de Newton, méthode de sous-gradient,...).

Programme convexe sous contraintes

Considérons à présent le (PM) suivant

$$\begin{aligned} & \text{Minimiser } f(x) \\ & \text{sous les contraintes} \\ & g_i(x) \leq 0 \quad i = 1, \dots, m \\ & x \in S. \end{aligned}$$

Si f est convexe et S convexe non discret, on parle de *programme convexe avec contrainte*. On sait déterminer par les conditions de Karush-Kuhn-Tucker dans quel cas ce programme possède ou non un optimum. Il existe également des dérivés des méthodes citées précédemment pour résoudre le problème.

La résolution de ces programmes est souvent appelée *Programmation non-linéaire* et constitue un domaine de recherche à part entière. Il existe des solveurs continus efficaces capables de résoudre un grand nombre de cas de ces programmes. Citons par exemple le freeware SolvOpt <http://www.kfunigraz.ac.at/imawww/kuntsevich/solvopt>.

On appelle aujourd'hui *programmation non-linéaire discrète* l'étude générale des programmes convexes sous contraintes le plus souvent linéaire. Il s'agit d'un domaine de recherche récent et très riche (mais que nous n'aborderons pas ici).

Programme quadratique

Si la fonction f est quadratique et les fonctions g sont linéaires : on parle de *programme quadratique discret* (PQD). Si f est quadratique et convexe et les fonctions g linéaires, on dit que le (PMD) est un *programme quadratique convexe discret*. On peut l'écrire sous la forme :

$$\begin{aligned} & \text{Maximiser } x^T Qx + Lx = \sum_{i=1}^n \sum_{j=1}^n q_{ij} x_i x_j + \sum_{i=1}^n l_i x_i \\ & \text{sous les contraintes} \\ & g(x) \leq 0 \\ & x \in \mathbb{Z}^n. \end{aligned}$$

Si les fonctions f et g sont quadratiques, on parle parfois également de programme quadratique, on le nommera ici *programme quadratique à contraintes quadratique*.

Une matrice est dite *positive* (resp. *semi-définies positives*) si, pour tout $x \in \mathbb{R}^n$, $x^T A x = \sum_{i=1}^n \sum_{j=1}^n A_{ij} x_i x_j > 0$ (resp $x^T A x \geq 0$). Dans le cas où la matrice Q d'un PQ est définie positive, la fonction f est convexe et il existe un optimum local. Résoudre la relaxation continue d'un PQD est un problème NP-difficile en général. Il existe de très performants algorithmes pour résoudre la relaxation continue des PQD avec contraintes linéaires. Ces algorithmes sont en fait inspirés des méthodes pour la PL en utilisant des principes proches de l'algorithme du simplexe ou des points intérieurs (appelé parfois dans ce cas Barrier algorithm). Les implémentations dans Cplex permettent de résoudre également de grands programmes. Des solveurs libres existent OpenOpt par exemple.

Programme quadratique

La version discrète de ces problèmes est difficile et les tailles solvables sont assez réduites (Cplex atteint quelques centaines de contraintes/variables).

Si Q est définie positive, c'est-à-dire de la *programmation quadratique convexe* à *contrainte quadratique*, la résolution est polynomiale : on sait la résoudre efficacement. On se ramène alors au cas d'un programme semi-défini.

Il existe un cas très particulier mais utile de contrainte non-linéaire que l'on sait bien résoudre (par exemple par CPLEX et GUROBI) : les formes coniques du 2nd ordre (second order cone program)

53/62

Programme semi-défini

Des extensions des méthodes de points intérieurs permettent de résoudre efficacement la SDP : on trouvera des références à des techniques et des solveurs à <http://www-user.tu-chemnitz.de/~helmborg/semidef.html>.

On commence peu à peu à considérer des SDP discrets. Par exemple l'outil BigCrunch <http://www-lipn.univ-paris13.fr/BiqCrunch> qui s'appuie sur la programmation semi-définie pour résoudre tout problème quadratique entier.

Et surtout, les SDP jouent un grand rôle pour obtenir de bonnes relaxations de problèmes NP-difficiles : problème de coloration, problème de la coupe maximale,...

55/62

Programme semi-défini

Un autre type de programme continu, qui a priori sort un peu du cadre défini ici de la (PM) classique est celui du programme semi-défini.

On définit alors un *programme semi-défini* comme étant celui des deux problèmes ci-dessous

Maximiser $\text{Trace}(A_0^T X)$ sous les contraintes $A_i X \leq c_i, i = 1, \dots, m$ X matrice semi-définie positive.	Minimiser $c^t y$ sous les contraintes $\sum_{i=1, \dots, m} A_i y_i - A_0$ est semi-définie positive $y \in \mathbf{R}^m$
--	---

où $A_i, i = 0, \dots, m$, sont des matrices semi-définies positives et $c \in \mathbf{R}^m$.

En fait, ces deux programmes sont dit duaux l'un de l'autre et ont la même solution optimale. Le programme dual écrit ci-dessus se rapproche de l'écriture d'un programme mathématiques.

Il est facile de montrer que tout programme quadratique convexe peut se ramener à un programme semi-défini et que la programmation semi-définie est une généralisation de la programmation linéaire.

54/62

Convexification

Lorsqu'un programme mathématique n'est pas convexe, on peut parfois le ramener au cadre convexe pour le résoudre. On appelle *convexification* l'ensemble des astuces permettant de rendre convexe ou d'améliorer la convexité d'un PM ou d'un PMD.

Par exemple, si la matrice de la fonction objective d'un PMD n'est pas définie positive, il est possible de la rendre définie positive en la réécrivant différemment (ou en lui ajoutant des variables fictives bien choisies).

C'est un domaine d'étude à part entière pour lequel nous ne regardons ici qu'un exemple.

56/62

Convexification

Considérons l'exemple suivant :

La fonction

$$q(x_1, x_2) = Cx_1x_2 \quad \forall (x_1, x_2) \in \{0, 1\}^2$$

où C est une constante positive. Cette fonction n'est pas convexe car, pour tout x_1, x_2 pris dans \mathbf{R} , elle n'est pas toujours positive ou nulle. En revanche, la fonction

$$\tilde{q}(x_1, x_2) = \frac{1}{2}C(x_1 + x_2)^2 - \frac{1}{2}C(x_1 + x_2) \quad \forall (x_1, x_2) \in \{0, 1\}^2$$

est convexe. En effet, la partie quadratique de \tilde{q} est positive ou nulle pour toute valeur des variables dans \mathbf{R} .

On peut remarquer que pour tout $(x_1, x_2) \in \{0, 1\}^2$, $x_1^2 = x_1$ et $x_2^2 = x_2$ pour des variables binaires ! Ainsi sur les nombres binaires, les deux fonctions coïncident !

$$\tilde{q}(x_1, x_2) = \frac{1}{2}C(x_1^2 + x_2^2 - 2x_1x_2) - \frac{1}{2}C(x_1 + x_2) = q(x_1, x_2)$$

On peut donc utiliser \tilde{q} qui est convexe à la place de q .

Il existe ainsi plusieurs procédé de convexification automatique ou non.

57/62

58/62

Linéarisation

On appelle *linéarisation* l'ensemble des astuces permettant de transformer en un PL ou un PLNE un PMD qui n'est pas linéaire à l'origine. Généralement, ces transformations demandent l'ajout de nombreuses variables supplémentaires.

Cette section comporte quelques astuces pour effectuer une telle linéarisation à partir d'une forme quadratique.

ATTENTION ces transformations ont un coût en nombres de variables et contraintes ajoutées : les techniques citées auparavant sont parfois bien plus efficaces !

59/62

Ecriture en variables binaires

Trois résultats permettent de formuler toute variable discrète par des variables binaires.

- *Une variable à valeurs dans un espace discret*

Soit x une variable prenant ses valeurs parmi les n possibilités $v_1, \dots, v_n \in \mathbf{R}$. On peut alors poser n variables de décisions binaires y_1, \dots, y_n telle que $y_i = 1$ si $x = v_i$ et 0 sinon.

Ce cas peut alors se modéliser par les deux contraintes $x = \sum_{i=1}^n v_i y_i$ et $\sum_{i=1}^n y_i = 1$.

- *Ecriture en variables binaires*

Si l'on peut déterminer des bornes sur les variables, on peut utiliser l'idée du cas précédent pour écrire un PLNE en variables entières comme un PLNE à variables binaires. En effet, considérons une variable x à valeurs entières entre 0 et u , $u \in \mathbf{N}$. Soit n tel que $2^n \leq u < 2^{n+1}$. On remplace alors x par sa représentation binaire :

$$x = \sum_{i=1}^n 2^i y_i \quad \text{où } y_i \in \{0, 1\} \text{ pour } i = 1, \dots, n.$$

60/62

- Carré d'une variable binaire

Soit $x \in \{0, 1\}$. Alors la variable x^2 est équivalente à la variable x .

- Produit de deux variables binaires :

Soit $x \in \{0, 1\}$ et $y \in \{0, 1\}$, on veut obtenir une variable e ayant la valeur $e = xy$.

En fait, on a le résultat suivant :

$$e = xy \Leftrightarrow \begin{cases} e \leq x \\ e \leq y \\ e \geq x + y - 1 \\ e \geq 0 \\ e \in \mathbf{R} \end{cases}$$

On peut généraliser ce résultat au produit d'une variable binaire par une variable entière (bornée), au produit de plusieurs variables binaires, au carré d'une variables binaires,...

Conclusion : Au total de ces linéarisations, on peut remarquer que toute forme quadratique peut se ramener à un PLNE binaire! Mais cela se fait au prix fort, en ajoutant de nombreuses variables et contraintes.

Conclusion

Tout au long de ce cours, nous verrons à quel point l'écriture sous forme d'un PLNE est un puissant outil de modélisation. En fait, on peut le voir souvent comme l'écriture naturelle algébrique d'un problème.

Inversement, comme nous l'avons cité, il n'existe pas de méthodes génériques efficaces pour résoudre un PLNE. Le fait de ramener aussi facilement un problème à un PLNE est donc parfois dangereux : on voit souvent des chercheurs ou des ingénieurs R&D conclure un travail de modélisation en affirmant que "comme on a ramené notre problème à un PLNE que les solveurs n'arrivent pas à résoudre, notre problème est difficile et nous allons utiliser des méthodes approchées". Cet état d'esprit, fort répandu malheureusement, est doublement faux. Tout d'abord, ramener un problème à un PLNE ne prouve en rien la difficulté d'un problème : ce n'est pas une preuve de complexité, il peut ainsi exister d'autres pistes théoriques ou algorithmiques pour résoudre le problème d'origine.

Il peut exister plusieurs PLNE modélisant le problème et différentes techniques pour le résoudre : c'est justement cette étude que nous allons mener dans ce cours.

Master d'Informatique - Spécialité Androide
Module MAOA

Recherche Opérationnelle
et Optimisation Combinatoire

Partie B - Cas polynomiaux

Pierre Fouilhoux

Sorbonne Université

2019-2020

1. Polyèdre et points extrêmes
2. Totale unimodularité
3. Encadrement min-max et Totale duale intégralité
4. Caractérisation complète

2/42

Voici quelques cas particuliers où la relaxation linéaire d'un PLNE fournit une solution entière... plus exactement où l'algorithme du simplexe fournit systématiquement une solution entière.

1. Polyèdre et points extrêmes
2. Totale unimodularité
3. Encadrement min-max et Totale duale intégralité
4. Caractérisation complète

Polyèdre et points extrêmes

Un polyèdre est tout simplement une figure géométrique définie comme la partie délimitée par des "plans". Dans un espace à une dimension, les "plans" sont des points et les polyèdres sont des intervalles connexes. Dans un espace à deux dimensions, les "plans" sont des droites et les polyèdres sont des carrés, des rectangles, ... Dans un espace de dimension 3, les "plans" sont des plans et les polyèdres sont des cubes, des dodécaèdres, ...

En fait, dans \mathbf{R}^n , on appelle *hyperplan* H un sous-espace de \mathbf{R}^n défini comme l'ensemble des points vérifiant une équation linéaire, c'est-à-dire qu'il existe $a_1, \dots, a_n, b \in \mathbf{R}$ tels que $H = \{x \in \mathbf{R}^n \mid a_1x_1 + \dots + a_nx_n = b\}$.

Definition

Un *polyèdre* $P \subseteq \mathbf{R}^n$ est l'ensemble des solutions d'un système fini d'inégalités linéaires, i.e.,

$$P = \{x \in \mathbf{R}^n \mid Ax \leq b\},$$

où A est une matrice $m \times n$ (m et n entiers positifs) et $b \in \mathbf{R}^m$.

Nous dirons alors que le système $Ax \leq b$ **définit** ou **caractérise** le polyèdre P .

Dans ce cours, nous considérons des polyèdres uniquement rationnels, c'est-à-dire pour lesquels les coefficients du système $Ax \leq b$ sont tous rationnels.

Si A est une matrice $m \times n$, on désigne par A_i (resp. A^j) la $i^{\text{ème}}$ ligne (resp. $j^{\text{ème}}$ colonne) de A pour $i = 1, \dots, m$ (resp. $j = 1, \dots, n$).

Un **point** d'un polyèdre est donc défini par des coordonnées $\tilde{x} \in \mathbf{R}^n$ tel que $A\tilde{x} \leq b$.

Un **polytope** est un polyèdre borné, c'est-à-dire qu'un polyèdre $P \subseteq \mathbf{R}^n$ est un polytope s'il existe $x^1, x^2 \in \mathbf{R}^n$ tel que $x^1 \leq x \leq x^2$, pour tout $x \in P$.

Definition

Un point x d'un polyèdre P est dit **point extrême** (ou parfois *sommet*) (vertex) de P s'il n'existe pas deux solutions x^1 et x^2 de P , $x^1 \neq x^2$, telles que $x = \frac{1}{2}x^1 + \frac{1}{2}x^2$.

En d'autre terme, un point extrême de P est un point de P qui n'est pas le milieu d'un segment contenu dans P .

Theorem

Soit $P = \{Ax \leq b\}$ un polyèdre de \mathbf{R}^n . Soit $\tilde{x} \in \mathbf{R}^n$ un point. On note alors $\tilde{A}x \leq \tilde{b}$ la sous-matrice des contraintes de $Ax \leq b$ formée par les inégalités vérifiées à l'égalité par \tilde{x} .

Alors \tilde{x} est un point extrême de P si et seulement l'ensemble $\tilde{A}x \leq \tilde{b}$ est de rang n .

Rappel d'algèbre linéaire :

- un ensemble de vecteurs sont dits *linéairement indépendants* si on ne peut pas obtenir l'un d'entre eux en combinant linéairement les autres.
- le rang d'une matrice est le nombre maximum de lignes linéairement indépendantes de la matrice.

Autrement dit, un point \tilde{x} est extrême si on peut produire n inégalités de la matrice A qui sont vérifiées par \tilde{x} à l'égalité et qui sont linéairement indépendantes.

Points fractionnaires du problème du sac-à-dos

Considérons le problème de sac-à-dos suivant :

$$\begin{aligned} \text{Max } & c_1x_1 + c_2x_2 + c_3x_3 \\ & 2x_1 + 2x_2 + 5x_3 \leq 8 \\ & x_i \leq 1 \quad i = 1, \dots, 3, \\ & x_i \geq 0 \quad i = 1, \dots, 3, \\ & x_i \in \mathbf{N} \quad i = 1, \dots, 3. \end{aligned}$$

On peut montrer que ce problème admet des points extrêmes fractionnaires et donner un cas où il est optimal.

Réponse : Considérons le point $\tilde{x} = (1, 1, \frac{4}{5})$.

On peut remarquer que ce point vérifie à l'égalité 3 inégalités de la relaxation linéaire du PLNE : $x_1 = 1$, $x_2 = 1$ et l'inégalité principale.

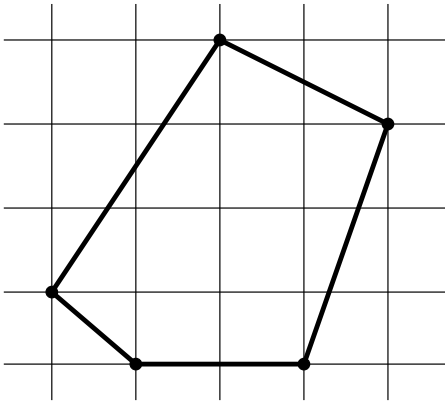
De plus, ces 3 inégalités sont clairement linéairement indépendantes.

Donc ce point est un point extrême fractionnaire du problème.

Par exemple, pour le poids $\tilde{c} = (10, 10, 1)$ ce point est clairement optimal.

Polyèdres entiers

Un point de \mathbb{R}^n est **entier** si ces coordonnées sont entières.
 Un polyèdre est dit **entier** si tous ses points extrêmes sont entiers.



Rappels : Programmation Linéaire

Rappelons deux résultats de la programmation linéaire :

- ▶ Tous les solutions optimales d'un PL se situe sur l'un des hyperplans définissant le polyèdre des solutions.
- ▶ L'ensemble des points extrêmes du polyèdre des solutions contient au moins une solution optimale. On peut donc limiter la recherche des solutions optimales aux points extrêmes du polyèdre des solutions.

Donc un PLNE dont la relaxation linéaire correspond à un polyèdre entier se résoud polynomialement !

Theorem (Equivalence Polyèdre entier/PLNE)

Un polytope rationnel P est entier si et seulement si pour tout vecteur entier c , la valeur optimale de $\text{Max}\{c^T x \mid x \in P\}$ est entière.

Preuve : Le sens direct est immédiat.
 Inversement, considérons $v = (v_1, \dots, v_n)^T$ un point extrême de P (il en existe un car il est pointu). Admettons que l'on puisse prouver qu'il existe un vecteur entier w tel que v soit l'unique solution optimale de $\text{max}\{w^T x \mid x \in P\}$ (admettons-le). Prenons $\lambda \in \mathbb{Z}$ tel que $\lambda w^T v \geq \lambda w^T u + u_1 - v_1$ pour tout u point extrême de P . On peut noter que v est encore l'unique solution optimale de $\text{max}\{\lambda w^T x \mid x \in P\}$. Ainsi, en posant le poids $\bar{w} = (\lambda w_1 + 1, \lambda w_2, \dots, \lambda w_n)^T$, on voit que v est aussi solution optimale de $\text{max}\{\bar{w}^T x \mid x \in P\}$ car $\lambda w^T v > x$ pour tout $x \in P$. Or par construction $\bar{w}^T v = \lambda w^T v + v_1$ et comme, par hypothèse, $\bar{w}^T v$ et $\lambda w^T v$ sont entiers, alors v_1 est entier. On peut reproduire cela pour toutes les composantes donc v est entier. □

(Il est possible d'étendre ce résultat aux polyèdres non bornés.)

Définitions :

- Un polyèdre est dit *pointu* s'il contient au moins un point extrême.
- Par exemple, le polyèdre $\{(x_1, x_2) \in \mathbb{R}^2 : x_1 \geq 0\}$ ne contient pas de points extrêmes.
- Un polyèdre est dit *rationnel* s'il peut être défini par un système où toutes les inégalités ont des coefficients rationnels.

On ne considère ici que des polyèdres rationnels pointus, ce qui n'est pas restrictif d'un point de vue informatique.

Un exemple simple

Considérons le problème de sac-à-dos suivant où b est entier.

$$\begin{aligned} \text{Max} \quad & \sum_{i=1}^n c_i x_i \\ & \sum_{i=1}^n x_i \leq b \\ & x_i \leq 1 \quad i = 1, \dots, n, \\ & x_i \geq 0 \quad i = 1, \dots, n, \\ & x_i \in \mathbb{N} \quad i = 1, \dots, n. \end{aligned}$$

Montrons que ce système est entier.

Réponse : Un point extrême de ce système doit vérifier à l'égalité n inégalités linéairement indépendantes de ce système. Or il y a peu d'inégalités ici qui peuvent être vérifiées à l'égalité !

Il y a deux cas :

- soit l'inégalité principale n'est pas vérifiée à l'égalité : dans ce cas, seules les inégalités triviales sont serrées et le point est entier.
- soit l'inégalité principale est vérifiée à l'égalité et alors il y a $n - 1$ inégalités triviales serrées. Donc le point est composée de $n - 1$ composantes entières 0 ou 1. Notons alors N^+ les composantes à 1. La dernière composante inconnue est donc de valeur $b - |N^+|$: notons que cette quantité est nécessairement positive, entière et vaut au plus 1. Donc cette dernière composante est elle aussi entière.

1. Polyèdre et points extrêmes

2. Totale unimodularité

Problème du flût de coût minimum
Problème de couplage

3. Encadrement min-max et Totale duale intégralité

4. Caractérisation complète

Unimodularité

Il serait très intéressant de pouvoir caractériser les matrices correspondant à des polyèdres entiers.

Une matrice **carrée** A est dite **unimodulaire** si A est entière et si son déterminant vaut $+1$ ou -1 .

Lemma

Soit A une matrice carrée de taille m entière, inversible. Alors $A^{-1}b$ est un vecteur entier pour tout vecteur entier b de taille m si et seulement si A est unimodulaire.

Preuve : Par un résultat classique d'algèbre linéaire, on sait que $A^{-1} = \frac{A^{adj}}{\det(A)}$ où A^{adj} est la matrice adjointe à A , c'est-à-dire la matrice obtenue en transposant la matrice des cofacteurs $C_{ij} = (-1)^{i+j} M_{ij}$ et où M_{ij} est le déterminant de la sous-matrice obtenue depuis A en supprimant la ligne i et la colonne j . Donc si A est entière, A^{adj} est aussi entière. Donc si A est unimodulaire, A^{-1} est entière et donc $A^{-1}b$ est un vecteur entier. Inversement, si $A^{-1}b$ est un vecteur entier pour tout vecteur entier b de taille m , alors en particulier $A^{-1}e_i$ est entier pour e_i le $i^{\text{ème}}$ vecteur unité pour tout $i = 1, \dots, m$. Donc A^{-1} est entière et donc $\det(A)$ et $\det(A^{-1})$ sont tous deux entiers. Comme $\det(A) \cdot \det(A^{-1}) = 1$, on a donc $\det(A) = 1$ ou -1 . \square

Unimodularité

Le lemme précédent mène alors à la définition suivante.

Une matrice A $m \times n$ avec $n \geq m$ de rang m est dite **unimodulaire** si A est entière et si la matrice associée à chacune de ses bases a un déterminant $+1$ ou -1 .
(Une base de A est un ensemble de m vecteurs colonnes linéairement indépendants et la matrice associée à une base est donc une sous-matrice carrée $m \times m$ inversible).

Theorem

[Veinott et Dantzig]

Soit A une matrice $m \times n$ entière de plein rang. Le polyèdre défini par $Ax = b, x \geq 0$ est entier pour tout vecteur b entier si et seulement si A est unimodulaire.

Totale unimodularité

On appelle une matrice **totale unimodulaire** (TU) une matrice telle que toutes ses sous-matrices carrées ont un déterminant valant $0, 1$ ou -1 .
Ainsi, pour une matrice unimodulaire, les coefficients sont donc uniquement $0, 1$ et -1 .

On peut remarquer qu'en fait une matrice A $m \times n$ est TU si et seulement si la matrice $[A|I]$ de taille $m \times (m+n)$ (qui est obtenue en collant une matrice identité à A) est unimodulaire.

Une conséquence du théorème précédent donne alors le théorème important suivant.

Theorem

[Hoffman-Kruskal]

Soit A une matrice $m \times n$ TU. Alors le polyèdre défini par $Ax \leq b, x \geq 0$ est entier pour tout vecteur b entier.

On peut noter que ce théorème n'est pas une caractérisation des polyèdres entiers. De plus, il n'est pas évident de détecter si une matrice est TU. Un résultat essentiel (et complexe) de Seymour (1980) prouve en fait que ces matrices peuvent être construites selon un schéma particulier.

Cas particuliers de matrices TU

Si ce résultat est assez complexe, il existe un cas particulier de matrice TU très facile à repérer.

Theorem (Poincaré [1900])

] Soit A une matrice dont les coefficients sont 0, 1 ou -1 et telle que chaque colonne contient au plus une fois le coefficient 1 et au plus une fois le coefficient -1. Alors A est TU.

Et donc, si le théorème s'applique, toute solution d'un PL utilisant A comme matrice des coefficients des inégalités correspond à un polyèdre entier.

17/42

Problème du flot de coût minimum (sans capacité)

On considère un **réseau** $G = (V, A)$ qui est un graphe orienté comportant un sommet s , appelé source, sans prédecesseur à partir duquel on peut atteindre tout sommet de G et un sommet t , appelé puits, qui est accessible depuis tout sommet de G .

Un *s-t-flot* est un vecteur positif $x \in \mathbb{R}^m$ s'il vérifie la contrainte de "conservation de flot"

$$\sum_{a \in \delta^+(u)} x(a) - \sum_{a \in \delta^-(u)} x(a) = 0 \quad \forall u \in V \setminus \{s, t\}.$$

On associe un coût $w(a) \in \mathbb{R}_+$ chaque arc $a \in A$.

Le **problème du flot de coût minimum** ou **min-cost flow** consiste à rechercher un flot tel que la valeur $\sum_{a \in A} w(a)x(a)$ soit minimale.

19/42

Cas particuliers de matrices TU

Ce théorème plus général généralise celui de Poincaré

Theorem

Soit A une matrice dont les coefficients sont 0, 1 ou -1 et telle que

- ▶ chaque colonne contient au plus deux coefficients non nuls.
- ▶ les lignes de A peuvent être partitionnées entre deux ensembles I_1 et I_2 tels que
 - ▶ si une colonne a deux coefficients de signes différents alors leurs lignes sont dans le même ensemble I_1 ou I_2
 - ▶ si une colonne a deux coefficients de même signe alors leurs lignes sont l'une dans I_1 et l'autre dans I_2 .

Alors A est TU.

Ce deuxième théorème implique le premier quand $I_2 = \emptyset$.

18/42

Problème du flot de coût minimum (sans capacité)

Considérons le PL suivant :

$$\begin{aligned} \text{Min} \quad & \sum_{a \in A} w(a)x(a) \\ & \sum_{a \in \delta^+(u)} x(a) - \sum_{a \in \delta^-(u)} x(a) = 0 \quad \forall u \in V \setminus \{s, t\} \end{aligned}$$

Le résultat de Poincaré prouve que la matrice des contraintes de ce PL, appelée **matrice de flots** est TU.

En effet, la variable $x(a)$ apparaît deux fois dans sa colonne : une fois pour chacune de ses extrémités, soit avec un coefficient -1, soit avec un coefficient +1.

Donc ce PL correspond à un polyèdre entier : il aura toujours une solution entière.

Mais cet exemple, où le flot n'est pas borné, a une solution optimale nulle... ce n'est pas un bon exemple !

20/42

Problème du flot de coût minimum (le vrai)

Soit un réseau G muni d'un poids w sur les arcs
 ... et d'une capacité minimale **entière** de flot $b(a) \in \mathbf{N}$ associée à chaque arc $a \in A$.

Ce qui se formule par

$$\begin{aligned} \text{Min} \quad & \sum_{a \in A} w(a)x(a) \\ & \sum_{a \in \delta^+(u)} x(a) - \sum_{a \in \delta^-(u)} x(a) = 0 \quad \forall u \in V \setminus \{s, t\} \\ & x(a) \geq b(a) \quad \forall a \in A. \end{aligned}$$

On peut prouver que ce PL correspond aussi à un polyèdre entier.

Soit un point extrême \bar{x} du polyèdre. Si une composante $x(a)$ vérifie inégalité de capacité à l'égalité, cette composante est entière. Prenons A^f l'ensemble des arcs de A de flot $\bar{x}(a) > b(a)$. Regardons le système issu des inégalités de conservation du flot suivante : on transforme les inégalités d'origine en fixant, à leurs bornes $b(a)$, les composantes $\bar{x}(a)$ avec $a \notin A^f$. Ce système est uniquement composée de la matrice de flots limitée aux arcs A^f donc la solution issue de cette matrice aura des composantes entières.

21/42

Problème d'affectation (couplage) biparti

Considérons le PL suivant qui formule le problème :

$$\begin{aligned} \text{Max} \quad & \sum_{e \in E} c(e)x(e) \\ & \sum_{e \in \delta(u)} x(e) \leq 1 \quad \forall u \in V_1 \\ & \sum_{e \in \delta(u)} x(e) \leq 1 \quad \forall u \in V_2 \\ & x(e) \geq 0 \quad \forall e \in E. \end{aligned}$$

La matrice de ce PL est TU par le théorème en utilisant $I_1 = V_1$ et $I_2 = V_2$.

23/42

Problème d'affectation (couplage) biparti

Soit un graphe biparti complet $G = (V_1 \cup V_2, E)$ associé à un poids $c \in \mathbf{N}^m$ associé aux arêtes de E .

On appelle **couplage** de G un ensemble d'arêtes deux à deux non incidentes.

Le **problème du couplage biparti** consiste à déterminer un couplage qui maximise la somme des poids de ses arêtes.

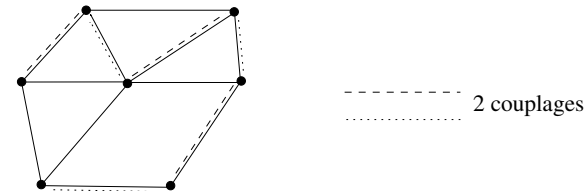
Il existe plusieurs algorithmes polynomiaux efficaces pour résoudre ce problème classique (dont le célèbre algorithme hongrois).

22/42

Problème d'affectation (couplage)

Généralisons le problème de couplage biparti.

Dans un graphe non-orienté $G = (V, E)$, on appelle **couplage** un ensemble d'arêtes deux à deux non-adjacentes (c'est-à-dire sans sommet commun). Le graphe est muni d'un poids $w(e)$ associé à chaque arête $e \in E$.



Le **problème du couplage maximum** consiste à rechercher un couplage de cardinalité maximum (ou de poids maximum).

24/42

Problème d'affectation (couplage)

Une formulation possible qui généralise la précédente est :

$$\begin{aligned} \text{Max } & \sum_{e \in E} c(e)x(e) \\ & \sum_{e \in \delta(u)} x(e) \leq 1, \quad \forall u \in V, \\ & 0 \leq x(e) \leq 1, \quad \forall e \in E, \\ & x(e) \text{ entier}, \quad \forall e \in E. \end{aligned}$$

En effet, dans un couplage, il y a au plus une arête incidente à chaque sommet.

- Si G est biparti, il peut s'écrire selon la formulation précédente : le PL a une matrice TU et le polyèdre correspondant est donc entier.
- Si G n'est pas biparti, que se passe-t-il ?

Problème d'affectation (couplage)

On ne peut pas trouver d'ensembles I_1 et I_2 satisfaisant le théorème. En effet, comme le graphe n'est pas biparti, on ne peut pas "bipartitionner" l'ensemble des lignes comme demandé dans le théorème.

En fait, le PL associé ne correspond pas à un PL entier (et n'est donc pas TU). On peut produire un contre-exemple en produisant un graphe et un **point extrême fractionnaire** (c'est-à-dire non entier) :

Prenons $G = C$ limité un cycle C (impair) de 5 sommets. Le point extrême \tilde{x} associant à chaque arête la valeur $\frac{1}{2}$ vérifie chacune des 5 inégalités (1) à l'égalité. De plus, ces 5 inégalités (1) sont linéairement indépendantes : c'est bien un point extrême fractionnaire du polyèdre de la formulation.

Et pourtant le problème de couplage est polynomial même pour G non biparti !

Il n'y a pas de contradiction !

Cette formulation n'est pas "entière" mais il en existe une, donnée par Jack Edmonds en 1965 (voir section "Caractérisation de polyèdres")

25/42

26/42

1. Polyèdre et points extrêmes

2. Totale unimodularité

3. Encadrement min-max et Totale duale intégralité

Problème du flot maximum/Coupe Minimum

4. Caractérisation complète

Rappel : Dualité

Pour un programme linéaire (\tilde{P}) , appelé alors *primal*, le *dual* est le programme linéaire (\tilde{D}) suivant

$$(\tilde{P}) \left\{ \begin{array}{l} \text{Max} \quad z = c^T x \\ Ax \leq b \\ x \geq 0 \end{array} \right. \quad (\tilde{D}) \left\{ \begin{array}{l} \text{Min} \quad w = b^T y \\ A^T y \geq c \\ y \geq 0 \end{array} \right.$$

Sous forme "algébrique" :

$$(\tilde{P}) \left\{ \begin{array}{l} \text{Max} \quad z = \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n a_{ij} x_j \leq b_i \quad \forall i = 1, \dots, m \\ x_j \geq 0 \quad \forall j = 1, \dots, n. \end{array} \right. \quad (\tilde{D}) \left\{ \begin{array}{l} \text{Min} \quad w = \sum_{i=1}^m b_i y_i \\ \sum_{i=1}^m a_{ij} y_i \geq c_j \quad \forall j = 1, \dots, n \\ y_j \geq 0 \quad \forall i = 1, \dots, m. \end{array} \right.$$

Les **variables duales** de (\tilde{D}) correspondent aux inégalités de (\tilde{P}) .

La matrice de (\tilde{D}) est la transposée A^T .

Les coûts de la fonction objective et les termes de droite des inégalités échangent leurs rôles.

27/42

28/42

Encadrement Min-Max

Rappel : Dualité

- **Théorème (faible) de la dualité :**

Si (\tilde{P}) et (\tilde{D}) admettent chacun une solution \tilde{x} et \tilde{y} , alors $c^T \tilde{x} \leq b^T \tilde{y}$.

- **Théorème de la dualité :**

Si (\tilde{P}) admet une solution optimale (finie), alors (\tilde{D}) aussi et de plus elles "coïncident", i.e.

$$(\tilde{P}) \quad \max\{c^T x \mid Ax \leq b\} = \min\{b^T y \mid y^T A = c, y \geq 0\} \quad (\tilde{D}).$$

Considérons un PLNE (P) et sa relaxation linéaire (\tilde{P}) qui est un PL. Considérons un PL (\tilde{D}) dual de (\tilde{P}) et sa version PLNE (D) . Une solution entière x de (P) est solution de (\tilde{P}) . Une solution entière y de (D) est solution de (\tilde{D}) . Posons x^* une solution optimale de (\tilde{P}) alors (par la dualité faible)

$$c^T x \leq c^T x^* \leq b^T y^* \leq b^T y$$

Dans le cas TRES particulier où l'on connaît à la fois :

- un algorithme donnant une solution entière x (approchée) pour (P)
 - et un algorithme donnant une solution entière y (approchée) pour (D)
- alors on a un **encadrement Min-Max** $[x, y]$ de la solution entière de (P) (et de (D)).

29/42

30/42

Totalement dual intégralité

Une question naturelle est de savoir quand un tel encadrement est une égalité !

Considérons un PLNE (P) de relaxation linéaire $(\tilde{P}) = \{c^T x \mid Ax \leq b\}$. Et le dual de (\tilde{P}) est $(\tilde{D}) = \{b^T y \mid A^T y \geq c\}$.

Le système $Ax \leq b$ est **totalement dual intégral** (TDI)

si, pour tout vecteur entier c tel qu'il existe une solution optimale de (\tilde{P}) , alors cette solution peut être obtenue par un vecteur y entier dans (\tilde{D}) .

Theorem

Soit $Ax \leq b$ un système TDI avec $(\tilde{P}) = \{cx \mid Ax \leq b\}$ rationnel et b entier. Alors (\tilde{P}) est un polytope entier, i.e. $(P) = (\tilde{P})$.

Preuve : Par la définition de (\tilde{P}) TDI, pour toute solution optimale x du problème, il existe un vecteur y solution du dual qui soit entier et qui réalise cet optimum, c'est à dire tel que $c^T x = y^T b$. Or comme b est entier, si y est entier, $c^T x$ est donc entier. Donc par le théorème [Equivalence Polyèdre entier/PLNE], x est entier. Donc (\tilde{P}) est entier. \square

On pourrait aussi s'intéresser à la caractérisation de systèmes TDI. Or cela n'a pas vraiment de sens par rapport à la résolution d'un problème d'OC : en effet, pour tout système rationnel $Ax \leq b$, il existe un entier positif t tel que $\frac{1}{t}Ax \leq \frac{1}{t}b$ soit TDI. L'existence d'un tel système ne dit donc rien sur la structure du polyèdre P associé. En fait, on a le résultat suivant.

Theorem

[Giles and Pulleyblank]

Soit P un polyèdre rationnel. Alors il existe un système TDI $Ax \leq b$ avec A entier tel que $P = \{Ax \leq b\}$. De plus, si P est entier, alors b peut-être choisi entier.

Ce résultat nous dit donc qu'il existe toujours un système TDI pour tout polyèdre P entier et que par conséquent, il existe un système ayant toujours des solutions entières.

- Déterminer un système TDI permet de prouver l'intégrité d'un polyèdre (et de son PL).

31/42

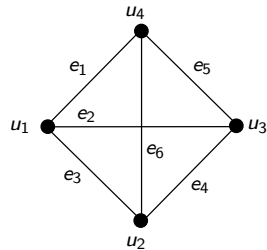
32/42

Un petit exemple non TDI

Considérons le problème (très simple) suivant.

On appelle *multi-ensemble*, un ensemble d'éléments où chaque élément peut-être représenté plusieurs fois.

Etant donné un graphe complet à 4 sommets $K_4 = (V, E)$ dont les arêtes sont munies du poids $w(e)$, $e \in E$, déterminer un multi-ensemble d'arêtes de poids maximal de K_4 tel que, pour chaque sommet du graphe, il y ait au plus 2 arêtes incidentes à ce sommet (on peut appeler ce problème le "2-couplage" maximal).



Un petit exemple non TDI

Montrons que le le système formé par les inégalités (1-1) n'est pas TDI.

Réponse : Pour cela, considérons le dual de ce système basé sur des variables y associée aux inégalités (1), c'est-à-dire aux sommets du graphe :

$$\begin{aligned} \text{Min } & \sum_{v \in V} 2y(v) \\ & y(u) + y(v) \geq w_e \quad \forall e = uv \in E, \\ & y(v) \geq 0 \quad \forall v \in V. \end{aligned}$$

On peut remarquer que, pour le poids $w(e) = 1$, $e \in E$, il est impossible qu'il existe une solution duale entière de valeur 4. Il n'existe donc pas de solutions entières du dual correspondant à une solution optimale du primal de valeur 4 : le système n'est donc pas TDI.

En effet, toute solution entière de ce dual doit avoir au moins 3 des variables y égales à au moins 1 (sinon une des inégalités ne seraient pas satisfaites pour une des arêtes) : donc toute solution entière de ce dual est au moins de valeur 6.

(Une question moins évidente serait de rechercher comment changer la formulation du problème pour le rendre TDI.)

Un petit exemple non TDI

Ce problème se formule par le PLNE (P) suivant :

$$\begin{aligned} \text{Max } & \sum_{e \in E} w(e)x(e) \\ & \sum_{e \in \delta(u)} x(e) \leq 2 \quad \forall u \in V, \\ & x(e) \geq 0 \quad \forall e \in E, \\ & x(e) \in \mathbf{N} \quad \forall e \in E. \end{aligned}$$

Remarquons que $x(e)$ est bien prise dans \mathbf{N} .

Regardons la solution optimale pour $w(e) = 1$ pour $e \in E$: la solution optimale de ce problème est clairement 4.

En effet,

- soit une arête est prise 2 fois : sans perte de généralité considérons e_1 , alors seule e_4 peut être ajoutée au plus 2 fois.
- soit chaque arête est prise au plus 1 fois : une solution optimale est alors clairement de prendre 4 arêtes formant un carré.

Problème du flot maximum

On considère un **réseau** $G = (V, A)$ qui est un graphe orienté comportant un sommet s , appelé source, sans prédecesseur à partir duquel on peut atteindre tout sommet de G et un sommet t , appelé puits, qui est accessible depuis tout sommet de G .

Un *s-t-flot* est un vecteur positif $x \in \mathbf{R}^m$ s'il vérifie la contrainte de "conservation de flot"

$$\sum_{a \in \delta^+(u)} x(a) - \sum_{a \in \delta^-(u)} x(a) = 0 \quad \forall u \in V \setminus \{s, t\}.$$

On associe une capacité maximale $b(a) \in \mathbf{N}$ associé à chaque arc $a \in A$. Un flot est dit *réalisable* si $x(a) \leq b(a)$ pour tout arc $a \in A$.

On pose $v = \sum_{a \in \delta^+(s)} x(a)$ la valeur du flot entrant en s (qui est exactement aussi la valeur du flot sortant par p : $v = \sum_{a \in \delta^-(s)} x(a)$).

Le **problème du flot de capacité maximale** ou **problème du flot max** consiste à rechercher un flot tel que sa valeur soit maximale.

On sait depuis le théorème de Ford-Fulkerson que si c est entier, alors il existe un flot optimal entier et on sait le déterminer en temps largement polynomial.

Dualité Flot Max / Coupe Min

En reprenant l'énoncé du problème du flot max, on peut définir un deuxième problème appelé **problème de la coupe minimum** ou encore **coupe min**.

On appelle **s-t-coupe** (ou plutôt ici **s-t-coupe orientée**) un ensemble d'arcs C sortant d'un ensemble de sommets W tel que $s \in W$ et $t \notin W$, i.e. $C = \delta^+(W)$. (de même $C = \delta^-(\bar{W})$ où $\bar{W} = V \setminus W$.)

On appelle **capacité d'une coupe** C la somme des capacités des arcs de la coupe :

$$b(C) = \sum_{a \in C} b(a).$$

D'après le théorème de Ford-Fulkerson concernant les flots maximum, on sait que pour tout flot maximal, on peut déterminer algorithmiquement une coupe de capacité minimum associée tel que la valeur du flot soit égale à la valeur de la coupe. On parle de la dualité Flots-Max/Coupe-Min.

Problème du flot maximum

Considérons la formulation PL suivante.

$$\begin{aligned} \text{Max} \quad & v \\ & \sum_{a \in \delta^+(u)} x(a) - \sum_{a \in \delta^+(u)} x(a) = 0 \quad \forall u \in V \setminus \{s, t\} \\ & \sum_{a \in \delta^+(s)} x(a) - v = 0, \\ & \sum_{a \in \delta^-(t)} x(a) + v = 0, \\ & x(a) \leq b(a) \quad \forall a \in A, \\ & v \geq 0, \\ & x(a) \geq 0 \quad \forall a \in A. \end{aligned}$$

37/42

38/42

Montrons que le PL du flot max est associée à un PL dual formant un système TDI. Indication : les solutions du problème de min-cut sont solutions du dual.

Réponse : Ecrivons le dual de la façon suivante : on note $\pi(u)$ les variables duales associées aux 3 premières inégalités pour $u \in V$ et on note $\gamma(u, v)$ les variables duales associées aux arcs $a = (u, v)$ des contraintes de capacités.

$$\begin{aligned} \text{Min} \quad & \sum_{a=(u,v) \in A} b(a)\gamma(u, v) \\ & \pi(u) - \pi(v) + \gamma(u, v) \geq 0 \quad \forall a = (u, v) \in A, \\ & -\pi(s) + \pi(t) \geq 1, \\ & \pi(u) \leq 0 \quad \forall u \in V, \\ & \gamma(u, v) \geq 0 \quad \forall a = (u, v) \in A. \end{aligned}$$

Etant donnée un flot maximal, c'est-à-dire une solution du primal, on sait déterminer par l'algorithme de Ford-Fulkerson, une s-t-coupe de capacité minimale. On peut montrer que cette coupe correspond à une solution entière $(\tilde{\gamma}, \tilde{\pi})$ du dual et qui est de même valeur que la solution du flot maximale : ce qui prouve que le système est TDI (enfin presque... voir page suivante).

Cette solution est donnée par $\tilde{\gamma}(u, v) = 1$ si $u \in W$ et 0 sinon et $\tilde{\pi}(u) = 1$ si $u \notin W$ et 0 sinon. En effet, on peut prouver que cette solution $(\tilde{\gamma}, \tilde{\pi})$ est solution duale en considérant les 4 cas possibles de situation d'une arête dans le graphe par rapport à la coupe (soit $u \in W, v \notin W; u \in W, v \in W, u \notin W, v \in W$ and $u \notin W, v \notin W$).

39/42

La page précédente propose une solution entière du dual pour la formulation du flot maximal...

Attention, pour pouvoir dire qu'elle est TDI, il faut effectuer cette preuve, pour toute fonction c coût entière sur les variables v et x !
 La formulation est donc celle du flot avec pour fonction objective :

$$\text{Max} \quad c_0 v + \sum_{a \in A} c_a x(a)$$

Le dual correspondant voit alors ses termes de second membre changer :

$$\begin{aligned} \pi(u) - \pi(v) + \gamma(u, v) &\geq c(a) \quad \forall a = (u, v) \in A \\ -\pi(s) + \pi(t) &\geq c_0 \end{aligned}$$

La solution $(\tilde{\gamma}, \tilde{\pi})$ solution entière du dual qui coïncide avec la valeur du primale est alors moins évidente à déterminer... (mais elle existe).

40/42

Caractérisation complète

1. Polyèdre et points extrêmes

2. Totale unimodularité

3. Encadrement min-max et Totale duale intégralité

4. Caractérisation complète

On appelle une **caractérisation complète d'un problème** la donnée d'un ensemble d'inégalités et de variables d'un programme linéaire entier dont les solutions optimales sont les solutions optimales du problème.

A moins que P soit égale à NP, il n'est pas possible de donner une caractérisation complète d'un problème NP-difficile.

Pour les problèmes polynomiaux, on peut comme on vient de le voir ici :

- trouver une formulation TU
- trouver un système TDI
- ou chercher **toutes** les inégalités nécessaires à cette description... Nous verrons cela dans le chapitre "Caractérisation".

Master d'Informatique - Spécialité Androide

Module MAOA

Recherche Opérationnelle et Optimisation Combinatoire

Partie C - Algorithmes de coupes (Branch-and-Cut)

Pierre Fouilhoux

Sorbonne Université

2019-2020

1. Inégalités valides et renforcement
2. Algorithmes de coupes
3. Algorithmes de Branch-and-Cut
4. Qu'y-a-t-il dans les solveurs entiers ?

2/49

Recherche Opérationnelle et Optimisation Combinatoire Partie C - Algorithmes de coupes (Branch-and-Cut)
└─ Inégalités valides et renforcement

Recherche Opérationnelle et Optimisation Combinatoire Partie C - Algorithmes de coupes (Branch-and-Cut)
└─ Inégalités valides et renforcement

1. Inégalités valides et renforcement
 - 1.1 Inégalités valides et renforcement
 - 1.2 Comparaison de formulations
 - 1.3 Méthodes génériques d'obtention d'inégalités valides

2. Algorithmes de coupes

3. Algorithmes de Branch-and-Cut

4. Qu'y-a-t-il dans les solveurs entiers ?

Lorsqu'un PLNE (P) n'a pas une relaxation linéaire (\tilde{P}) entière, le PL (\tilde{P}) possède donc des points extrêmes fractionnaires.

Il est utile de déterminer des inégalités que l'on appellera valides (et non redondantes voir section "Approches Polyédrales") pour un PLNE. Ces inégalités sont choisies de manière à couper ces points extrêmes fractionnaires.

On les détermine soit par intuition, soit par extrapolation de cas simples, soit en les dérivant par des procédés tels que celui de Chvátal-Gomory,...

L'ensemble de ces idées algorithmiques permettent aujourd'hui de résoudre efficacement des PLNE de bonnes tailles (quelques milliers de lignes/contraintes). Surtout dans les cas où le PLNE a une structure particulière. On nomme souvent cet aspect du domaine de recherche MIP pour Mixed Integer Programming.

Inégalités valides

Pour un PLNE donné, $(P) \text{ Max}\{c^T x \mid Ax \leq b, x \in \mathbf{N}^n\}$ avec ou non un nombre compact de contraintes et de variables, on peut ajouter des contraintes supplémentaires à sa relaxation linéaire (\tilde{P}) .

On considère le polyèdre $\mathcal{P} = \{x \in \mathbf{R}^n \mid Ax \leq b\}$ associé à (P) , appelé **polyèdre de la formulation**.

Definition

Une inégalité $ax \leq \alpha$ est dite **valide** pour (P) si elle est vérifiée par tous les points entiers de \mathcal{P} .

Ajouter une inégalité valide au PLNE (P) conserve l'espace des solutions de (P) : en revanche cela peut changer sa relaxation (\tilde{P}) .

Coupes

Une solution fractionnaire potentielle de la relaxation linéaire $(\tilde{P}) = \text{max}\{c^T x \mid Ax \leq b\}$ est donc un point extrême fractionnaire du polyèdre $\mathcal{P} = \{x \in \mathbf{R}^n \mid Ax \leq b\}$.

Considérons un tel point extrême fractionnaire \tilde{x} .

Une inégalité $ax \leq \alpha$ est dite **violée** par \tilde{x} si $a\tilde{x} > \alpha$.
Une inégalité $ax \leq \alpha$ est dite **serrée** par \tilde{x} si elle satisfait $ax \leq \alpha$ à l'égalité, c'est-à-dire $a\tilde{x} = \alpha$.

Ajouter au PLNE (P) une inégalité valide ET violée par \tilde{x} permet de couper un point extrême indésirable de l'espace des solutions optimales.
On dit alors qu'une telle inégalité est une **coupe** pour le PLNE.

De plus, ces contraintes permettent d'obtenir une relaxation plus intéressante (même si parfois c'est numériquement très faible).
On parle alors de **renforcement** de la formulation.

Exemple pour la formulation du stable

Soit un graphe $G = (V, E)$ muni d'un poids c associé aux sommets. Le problème du stable maximum se modélise selon la formulation suivante

$$\begin{aligned} \text{Max} \quad & \sum_{u \in V} c(u)x(u) \\ & x(u) + x(v) \leq 1 \quad \text{pour tout } uv \in E, & (1) \\ & 0 \leq x(u) \leq 1 \quad \text{pour tout } u \in V, & (2) \\ & x(u) \text{ entier,} \quad \text{pour tout } u \in V. \end{aligned}$$

En effet, les contraintes (2) sont trivialement valides : on les appelle d'ailleurs les contraintes triviales car elles sont vérifiées par tout vecteurs d'incidence d'un ensemble.

Les contraintes d'arêtes (1) sont valides.

En effet, prenons un stable $S \subset V$ et considérons son **vecteurs d'incidence** χ^S tel que $\chi^S[u] = 1$ si $u \in S$ et $\chi^S[u] = 0$ sinon.

On peut voir que pour un sommet u donné, un stable contient au plus une arête incidente à u , donc $\chi^S[u] + \chi^S[v] \leq 1$ pour toute arête $uv \in E$. Ce qui est équivalent à dire que χ^S satisfait toutes les inégalités d'arêtes.

Exemple pour la formulation du stable

On vient de prouver que toute solution du problème du stable vérifie la formulation.

ATTENTION Pour prouver l'équivalence entre la formulation et le problème du stable, il faut aussi prouver le "retour", c'est-à-dire que toute solution entière de la formulation correspond à un stable.

C'est bien le cas : prenons une solution entière x^* de la formation. Par les contraintes triviales et d'intégrité, on peut poser $M = \{u \in V \mid x^*(u) = 1\}$ qui est un sous-ensemble de sommets. Si deux sommets sont adjacents dans V , on voit par les contraintes d'arêtes que ces deux sommets ne peuvent appartenir tous les deux à M : donc M est bien un stable.

Et pour terminer, on remarque que les fonctions objectives du PLNE et du problème du stable maximum sont exactement la même fonction.

La formulation est bien équivalente au problème.

Exemple pour la formulation du stable

Cette formulation du stable a une relaxation linéaire non entière (sinon le problème du stable serait polynomial...).

Par exemple, si G contient une clique (un ensemble de sommets induisant un sous-graphe complet) de k sommets, le point y suivant :

$$y(u) = \frac{1}{2} \text{ pour tout } u \in K$$

$$y(u) = 0 \text{ pour tout } u \in V \setminus K$$

vérifie k inégalités d'arêtes et $n - k$ inégalités triviales à l'égalité.

De plus ces n inégalités sont linéairement indépendantes.

C'est un point extrême fractionnaire (qui est solution optimale pour la fonction poids cardinalité).

Comment le "couper" ?

Exemple pour la formulation du stable

Supposons que le graphe G contienne à présent un cycle impair C : un cycle comportant un nombre impair de sommets (qui est le même que le nombre d'arêtes). Notons $V(C)$ les sommets correspondant au cycle C (un cycle est une suite d'arêtes).

Regardons le point y tel que :

$$y(u) = \frac{1}{2} \text{ pour } u \in V(C)$$

$$y(u) = 0 \text{ pour } u \in V \setminus V(C)$$

Ce point satisfait $|C|$ inégalités d'arêtes et $n - |C|$ inégalités triviales à l'égalité.

Ces inégalités sont linéairement indépendantes car C est impair (ce n'est pas le cas si C est pair!).

Comme précédemment ce point est un point extrême fractionnaire de la formulation

Comment le couper ?

Exemple pour la formulation du stable

Les inégalités de clique

$$\sum_{u \in K} x(u) \leq 1 \quad \text{pour toute clique } K, \tag{3}$$

sont valides pour la formulation du stable.

En effet, tout vecteur d'incidence d'un stable vérifie ces inégalités car au plus un sommet peut être pris dans une clique d'un graphe.

De plus la contrainte associée à une clique K (de taille au moins 3) coupe le point extrême y .

En effet, $\sum_{u \in K} y(u) = \frac{k}{2} > 1$ dès que $k \geq 3$.

On peut remarquer que les contraintes de clique de taille 2 sont en fait les contraintes d'arêtes.

Pour simplifier l'écriture, on écrit souvent $x(K) = \sum_{u \in K} x(u)$ et les inégalités de cliques deviennent alors $x(K) \leq 1$ pour toute clique K .

Exemple pour la formulation du stable

Les inégalités de cycle impair

$$\sum_{u \in V(C)} x(u) \leq \frac{|C|-1}{2} \quad \text{pour tout cycle impair } C, \tag{4}$$

sont valides pour la formulation du stable.

En effet, prenons S un stable de V .

On peut prendre au plus un "sommet sur deux" du cycle dans S . Et on voit qu'aucun stable ne peut contenir $\frac{|C|+1}{2}$ sommets du cycle. Donc un stable contient au plus

$\frac{|C|-1}{2}$ d'un cycle impair : l'inégalité est bien valide.

Et elle coupe le point y .

En effet, $\sum_{u \in V(C)} y(u) = \frac{|C|}{2} > \frac{|C|-1}{2}$.

Remarque : On note souvent $x(V(C)) = \sum_{u \in V(C)} x(u)$

Comparaison de formulations

Ajouter une inégalité valide qui coupe une solution fractionnaire optimale de la relaxation améliore la valeur de la relaxation linéaire : on parle de **renforcement**.

Dans un cadre plus général, il n'est pas évident de prouver "à l'avance" si une inégalité valide va être systématiquement un renforcement.

On le démontre au cas par cas.

13/49

Le problème du voyageur de commerce asymétrique

Une autre formulation mène à une relaxation bien plus intéressante dites formulation par les coupes

$$\begin{aligned} \text{Min } & \sum_{i,j} c_{ij} x_{ij} \\ & \sum_{j \in V} x_{ij} = 1 \quad \forall i \in V, \\ & \sum_{i \in V} x_{ij} = 1 \quad \forall j \in V, \\ & \sum_{e \in \delta^+(W)} x(e) \geq 1 \quad \text{pour tout } W \subsetneq V \text{ et } W \neq \emptyset, \\ & x_{ij} \in \mathbf{N} \quad \forall (i,j) \in V \times V. \end{aligned}$$

Cette formulation apporte en pratique de bien meilleure valeur (expérimentale) de relaxation.

Et on peut prouver cela de manière théorique !

15/49

Le problème du voyageur de commerce asymétrique

Soient n villes avec c_{ij} coût de transport de i à j (asymétrique).

Déterminer un circuit passant par toutes les villes de plus petit coût.

Avec les variables $x_{ij} = 1$ si l'arc (i,j) est dans le circuit et 0 sinon et des variables additionnelles u réelles, la formulation MTZ que nous avons vue est une formulation du problème.

$$\begin{aligned} \text{Min } & \sum_{i,j} c_{ij} x_{ij} \\ & \sum_{j \in V} x_{ij} = 1 \quad \forall i \in V, \\ & \sum_{i \in V} x_{ij} = 1 \quad \forall j \in V, \\ & u_1 = 1, \\ & 2 \leq u_i \leq n \quad \text{pour tout } i \neq 1, \\ & u_i - u_j + 1 \leq n(1 - x_{ij}) \quad \text{pour tout } i \neq 1, j \neq 1, \\ & u_i \in \mathbf{R} \quad \forall i \in V \\ & x_{ij} \in \mathbf{N} \quad \forall (i,j) \in V \times V. \end{aligned}$$

Cette formulation est compacte !

Mais elle n'est pas très performante en terme de valeur de relaxation linéaire. Un point x avec de très petites valeurs x (non entière) va pouvoir vérifier les inégalités MTZ et être optimales : cette valeur est très basse.

14/49

Le problème du voyageur de commerce asymétrique

La difficulté ici est que les formulations ne sont pas tout à fait sur le même espace de variables : on doit faire "disparaître" les x .

Un façon de le faire est de sommer les inégalités MTZ sur un circuit C : on obtient alors

$$\sum_{(i,j) \in C} x_{ij} \leq \left(1 - \frac{1}{n-1}\right) |C|$$

En notant $V(C)$ les sommets de C et en l'appliquant à la formulation par les coupes, on obtient :

$$\sum_{i,j \in V(C)} x_{ij} \leq |C| - 1$$

On voit ici que cette deuxième inégalité est bien plus forte en terme de coupes et donc de relaxation.

Cette idée peut se généraliser et on peut prouver que toutes les inégalités provenant de la formulation MTZ sont battue par une inégalité plus forte dans la formulation par les coupes : On dit qu'elle **domine** la formulation MTZ.

16/49

Méthodes génériques

Une question importante est de se demander comment produire des inégalités automatiquement pour couper des points fractionnaires et renforcer les formulations.

C'est un domaine d'étude riche qui a permis les meilleures améliorations des solveurs entiers, citons-en quelques unes :

- ▶ La méthode d'arrondi de Chvátal-Gomory (voir plus loin) : elle est malheureusement peu efficace si elle est utilisée seule mais très utile pour renforcer un PLNE quelconque.
- ▶ L'utilisation des structures sous-jacentes de "stable" dans la matrice des contraintes : on repère tout d'abord des paires de variables ne pouvant être toutes deux à 1 : cela crée un graphe dit d'incompatibilité. On sait que toute solution sera entre autre un stable de ce graphe : donc toutes les inégalités de cliques et de cycles impaires sont valides (et très très utiles).
- ▶ Par déduction à partir de certaines contraintes : par exemple les inégalité cover provenant des inégalité de knapsack (voir plus loin).

17/49

Somme de Chvátal

Une technique célèbre pour obtenir de telles inégalités valides de manière automatique est la somme de Chvátal (avec arrondi dit de Gomory) basé sur la recette suivante :

Additionner plusieurs contraintes

Diviser les coefficients de la contrainte obtenue de manière à ramener le plus petit coefficient à 1.

Utilisant le fait que certaines sommes de termes sont entières

En déduire une nouvelle inégalité valide

19/49

Méthodes génériques

- ▶ La construction de Lovasz-Schrijver : qui permet de déduire des contraintes en les "multipliant" entre elles, en linéarisant par ajouts de variables de linéarisation (principe de Sherali-Adams), puis en projetant sur l'espace d'origine. Cela mène à la méthode de Lift-and-Project initiée par Balas.
- ▶ Les inégalités disjonctives qui génèrent aujourd'hui les coupes les plus efficaces.
- ▶ et quelques autres

18/49

Somme de Chvátal

Exemple du polyèdre du stable.

Partons des inégalités aux arêtes.

Supposons que G contienne un cycle de G .

On somme les k contraintes associées aux k arêtes du cycle :

$$\begin{array}{rcl}
 x(u_1) & + & x(u_2) & & & \leq & 1 \\
 & & x(u_2) & + & x(u_3) & & \leq & 1 \\
 & & & & \dots & & & \\
 x(u_1) & & & & & + & x(u_k) & \leq & 1
 \end{array}$$

$$2 \sum_{u \in V(C)} x(u) \leq k$$

$$\sum_{u \in V(C)} x(u) \leq \frac{k}{2}$$

20/49

Somme de Chvátal

Comme le côté gauche de l'inégalité est une somme d'entier, le côté droit peut lui aussi être ramené à l'entier immédiatement inférieur. D'où, comme k est impair, on obtient finalement une inégalité valide par construction !

$$\sum_{u \in V(C)} x(u) \leq \frac{k-1}{2}.$$

Lorsque k est impair, on obtient ainsi les contraintes dites de cycles impairs pour le problème du stable :

$$\sum_{v \in V(C)} x(v) \leq \frac{|C|-1}{2} \quad \forall C \text{ cycle impair}$$

Méthodes duale fractionnaire

On peut utiliser la somme de Chvátal-Gomory pour créer un algorithme de coupes génériques et adaptables à tout PLNE : la méthode duale fractionnaire.

Elle utilise le tableau de la méthode duale de l'algorithme du simplexe afin de produire itérativement une suite d'inégalités valides qui coupent peu à peu les points extrêmes non entiers. Elle mène donc à elle-seule à une solution entière optimale ! Cette méthode converge vers une solution optimale en temps fini... mais elle est excessivement lente !!!

En revanche, les inégalités produites par cette méthode permettent d'accélérer fortement les algorithmes de coupes et branchements génériques.

21/49

22/49

Exemple d'inégalités génériques, les cover inequalities

Après transformations basiques, toute inégalité d'un PLNE peut être vue comme une inégalité de "sac-à-dos" :

$$\sum_{i=1}^n a_i x_i \leq b \text{ avec } a_i \geq 0, b \geq 0.$$

Un ensemble de variables $R \subset \{1, \dots, n\}$ est un **cover** si $\sum_{i \in R} a_i > b$

et alors l'inégalité

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

est valide pour la formulation (voir cahier d'exercice).

1. Inégalités valides et renforcement

2. Algorithmes de coupes

2.1 Coupes et séparation

2.2 Equivalence Séparation et Optimisation

3. Algorithmes de Branch-and-Cut

4. Qu'y-a-t-il dans les solveurs entiers ?

23/49

24/49

Nous avons rencontré deux cas où on considère un grand nombre de contraintes :

- ▶ les formulations PLNE contenant au départ un grand nombre de contraintes (potentiellement exponentiel) que l'on ne peut donc pas énumérer : formulation du problème du voyageur de commerce,...
- ▶ des exemples où l'ajout de contraintes supplémentaires permet d'obtenir une meilleure valeur de relaxation : on appelle cela parfois le "renforcement" de formulation : ajout des contraintes de cliques pour la formulation du problème du stable.

Comment gérer algorithmiquement l'ajout d'un grand nombre de contraintes dans un PLNE ?

Comment résoudre une formulation non-compacte ?

Considérons ici un programme linéaire

$$(P) \quad \text{Max}\{c^T x \mid Ax \leq b\}$$

comportant n variables avec n une quantité limitée en taille (quelques milliers) mais où A contient un nombre important, par exemple exponentiel par rapport à n de contraintes : on dit que le programme est non compact sur les contraintes.

Un tel programme ne peut donc pas être introduit comme instance d'un algorithme de B&B et donc ne peut pas être utilisé directement dans les solveurs entiers. Nous allons pourtant voir qu'il existe un cadre algorithmique efficace, les algorithmes de coupes pour ces programmes.

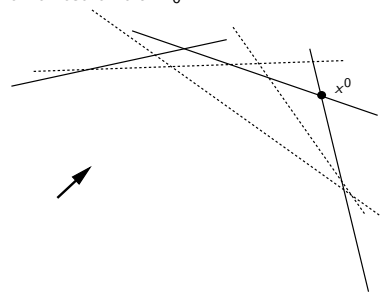
25/49

26/49

Initialisation

Choisissons $A_0 x \leq b_0$ sous-ensemble de contraintes de $Ax \leq b$ tel qu'il existe une solution finie x^0 au problème restreint à A_0 .

$$(P_0) \quad \begin{cases} \text{Max } c^T x \\ A_0 x \leq b_0 \end{cases}$$



Une solution optimale x_0 est donc un point extrême du polyèdre $A_0 x \leq b_0$.

La figure illustre en 2 dimensions, les contraintes de $A_0 x \leq b_0$ en traits pleins et les autres contraintes de $Ax \leq b$ en pointillés.

La flèche indique la "direction d'optimisation" : la direction orthogonale à toute droite d'équation $z = c^T x$.

Problème de séparation

Notons que si x_0 satisfaisait toutes les inégalités de $Ax \leq b$ restante, alors x_0 serait optimal pour tout P (et il serait point extrême de $Ax \leq b$).

Si x^0 un point extrême de $Ax \leq b$, il existe une inégalité de $Ax \leq b$ qui n'est pas satisfaite par x^0 : on dit que x^0 viole la contrainte.

Problème de séparation

Etant donné un point $\tilde{x} \in \mathbb{R}^n$, le **problème de séparation** associé à $Ax \leq b$ et \tilde{x} consiste à déterminer si \tilde{x} satisfait toutes les inégalités de $Ax \leq b$ et sinon à trouver une inégalité de $Ax \leq b$ violée par \tilde{x} .

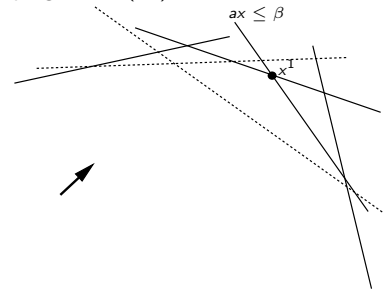
27/49

28/49

Itération

Si le problème de séparation pour $\bar{x} = x_0$ fournit une inégalité $\alpha x \leq \beta$ violée par \bar{x} , on l'ajoute au programme (P_0) pour obtenir le programme (P_1) :

$$(P_1) \begin{cases} \text{Max } c^T x \\ A_0 x \leq b_0 \\ \alpha x \leq \beta \end{cases}$$



Et on réitère !

Algorithme de séparation

Un algorithme qui répond au problème de séparation s'appelle **algorithme de séparation**.

Si on connaît un tel algorithme de séparation, on sait donc déterminer une inégalité $\alpha x \leq \beta$ qui est violée par le point extrême x^0 .

Une telle inégalité est appelée une **coupe** car elle "coupe" le point indésirable de l'espace des solutions possibles.

Dans l'illustration précédente, la contrainte $\alpha x \leq \beta$ qui était violée par x^0 a été ajoutée au système courant : on voit qu'elle a **séparé** une partie de l'espace indésirable du programme P_1

Algorithme de coupes

En réitérant l'algorithme de séparation autant tant que l'on peut ajouter des contraintes violées, on obtient alors un algorithme appelé **méthode ou algorithme de coupes** (cutting plane algorithm).

Algorithme de coupes (cutting-plane based algorithm) :

- Tant qu'il existe une inégalité de $Ax \leq B$ violée par x_i
- Ajouter l'inégalité au programme (P_i) : on obtient P_{i+1}
- Résoudre la relaxation linéaire de P_{i+1} : on obtient une solution x_{i+1}
- $i = i + 1$

Terminaison et validité :

L'algorithme de coupes se termine nécessairement : au pire il aura énuméré toutes les inégalités de $Ax \leq b$.

A la fin d'une méthode de coupes, on obtient donc nécessairement un point extrême x^* qui est point extrême du système courant et qui n'est pas violé par une contrainte de $Ax \leq b$: on obtient donc une solution optimale de (P).

Complexité de la méthode de coupes

La question importante est donc alors de connaître la complexité d'une méthode de coupes. Ceci nous est donné par le résultat extrêmement puissant suivant :

Theorem (Grötschel, Lovász, Schrijver, 1981)

Une méthode de coupes sur un système $Ax \leq b$ de contraintes est polynomiale si et seulement si l'algorithme de séparation associé à $Ax \leq b$ est polynomiale.

Ce résultat fondamental permet ainsi de manipuler des formulations exponentielles pour la PLNE ! Il indique ainsi :

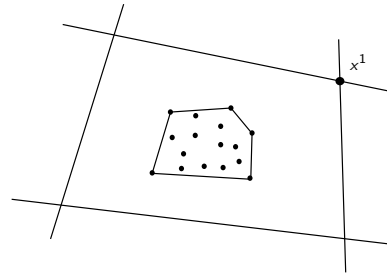
Optimiser est équivalent à Séparer.

Ce résultat est issu de l'algorithme des ellipsoïdes de Karmarkar et n'indique pas le degré du polynôme de la complexité de la méthode de coupes... En théorie il semble être élevé mais en pratique... c'est très efficace.

En pratique :

On connaît uniquement $Ax \leq B$ un premier ensemble d'inégalités tel que tout point entier les vérifiant est solution.

$$(P) \begin{cases} \text{Max } cx \\ Ax \leq B \\ x \text{ entier} \end{cases}$$



Taille du PL manipulé

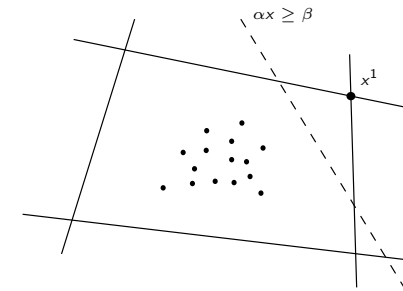
Une autre conséquence concerne la taille du système linéaire utilisé : en fait, comme on ajoute une inégalité un nombre polynomial de fois : le système courant a au plus une taille polynomiale ! Ainsi, au lieu de manipuler un système exponentiel, un système polynomial est suffisant.

Cette remarque n'est pas surprenante, en effet, pour définir un point extrême (solution) il suffit de n contraintes (linéairement indépendantes) satisfaites à l'égalité : le reste peut être donc être "écarté" du PL correspondant : la question est alors : comment trouver ces n contraintes ? En effet, si on les a, une seule utilisation de l'algorithme de séparation permettrait de tester si le point est une solution de $Ax \leq b$.

En pratique :

On connaît uniquement $Ax \leq B$ un premier ensemble d'inégalités tel que tout point entier les vérifiant est solution.

$$(P_2) \begin{cases} \text{Max } cx \\ Ax \leq B \\ \alpha x \leq \beta \\ x \text{ entier} \end{cases}$$



Le polyèdre du problème est l'inconnue... (voir section E)

Preuve de polynomialité et de NP-difficulté

Cette équivalence entre Séparation et Optimisation permet de prouver que certains problèmes sont polynomiaux : il faut pour cela avoir prouvé qu'un problème se ramène à résoudre un PL correspondant à un polyèdre entier à nombre exponentiel d'inégalités mais pour lequel vous connaissez une méthode de coupes polynomiale.

Inversement, si vous prouvez que, **pour toute fonction poids**, la méthode de coupes associées à votre PL est NP-difficile, alors le problème équivalent au PL sera à son tour NP-difficile.

Séparation pour le problème du stable maximum

Inégalités de cycles impairs

Les inégalités de cycles impairs

$$\sum_{u \text{ dans } C} x(u) \leq \left\lfloor \frac{|C|}{2} \right\rfloor \forall \text{ cycle impair } C$$

sont séparables en temps polynomial.

37/49

Inégalités de cycles impairs

Pour déterminer un cycle impair de plus petits poids par rapport à w , on crée un graphe biparti G'

- où chaque sommet i de G est dupliquée en i' et i'' : V est ainsi dupliqué en l'ensemble V' et l'ensemble V'' .
- et où chaque arête ij de G devient deux arêtes $i'j''$ et $i''j'$ portant chacune le poids w_{ij} .

On voit bien que G est biparti car il n'y a pas d'arête au sein de V' ou au sein de V'' . Dans ce cycle, tout chemin (élémentaire) de i' à i'' va correspondre à un cycle impair car il faut effectuer un nombre impair d'aller retour entre V' et V'' .

En effectuant une recherche d'un plus court chemin de i' à i'' dans G' pour tout $i \in V$, on obtient un cycle impair de plus petit poids dans G .

Comme la recherche d'un plus court chemin avec des poids w positifs est polynomial, cet algorithme de séparation revient à n fois la complexité de la recherche d'un plus court chemin et est donc polynomiale.

39/49

Inégalités de cycles impairs

Considérons un point \tilde{x} solution de la relaxation linéaire de la formulation aux arêtes du problème du stable.

Donc le point \tilde{x} vérifie toutes les inégalités d'arêtes (mais si l'on cherche à le couper, c'est qu'il est fractionnaire).

On pose alors

$$w_{uv} = 1 - \tilde{x}_u - \tilde{x}_v$$

qui est bien positif ou nul car \tilde{x} vérifie toutes les inégalités d'arêtes.

L'inégalité est alors équivalente à

$$\sum_{e \in C} w_{uv} \geq 1$$

Le problème de séparation revient donc à déterminer un plus petit cycle impair par rapport au poids w dans le graphe.

En effet :

- Si ce poids est strictement plus petit que 1, cela signifie qu'on a déterminé une inégalité violée.
- Si ce poids est plus grand que 1, cela signifie que l'inégalité avec le plus petit écart avec la violation, n'est justement pas violée : donc il n'y a pas d'inégalités violées.

38/49

Séparation pour le problème du stable maximum

Inégalités de cycles cliques

Les inégalités de cliques

$$\sum_{u \in K} x(u) \leq 1 \text{ pour toute clique } K$$

ont un problème de séparation associé NP-complet...

En effet, prouver qu'il n'y a plus d'inégalités violées est équivalent à rechercher une clique de poids maximal dans G : ce qui est un problème NP-difficile.

Donc on ne peut pas construire un algorithme de coupes polynomial pour cette formulation.

40/49

Inégalités de cycles cliques

En revanche, on peut quand même considérer la formulation suivante

$$\begin{aligned} \text{Max } & \sum_{u \in V} c(u)x(u) \\ & x(u) + x(v) \leq 1, \text{ pour tout } uv \in E, \\ & \sum_{u \in K} x(u) \leq 1, \text{ pour "quelques" cliques } K \text{ de } G, |K| \geq 3. \\ & x(u) \in \{0, 1\}, \text{ pour tout } u \in V. \end{aligned} \quad (5)$$

En effet, seules les contraintes d'arêtes sont nécessaires à la formulation : elles peuvent donc être énumérées et utilisées tout au long de l'algorithme de coupes.

Les contraintes associées à des cliques de tailles au moins 3 seront elles utilisées dans un algorithme de coupes heuristiques : c'est-à-dire qu'au lieu de rechercher s'il existe ou non une contrainte de cliques violées, on recherche heuristiquement une contrainte de cliques violée !

Elle viendront renforcer la formulation très efficacement.

41/49

Séparation pour le problème du voyageur de commerce

Inégalités de coupes

Considérons la formulation suivante pour le voyageur de commerce symétrique que nous avons vu dans le début de cette partie.

$$\begin{aligned} \text{Min } & \sum_{e \in E} c(e)x(e) \\ & \sum_{e \in \delta(v)} x(e) = 2, \text{ pour tout } v \in V, \end{aligned} \quad (6)$$

$$\begin{aligned} & \sum_{e \in \delta(W)} x(e) \geq 2, \text{ pour tout } W \subsetneq V \text{ et } W \neq \emptyset, \\ & x(e) \geq 0, \text{ pour tout } e \in E \\ & x(e) \in \{0, 1\}, \text{ pour tout } e \in E. \end{aligned} \quad (7)$$

Les contraintes (6) de degré sont au nombre de $|V|$ et sont toujours présentes dans la formulation.

En revanche les contraintes (7) sont en nombre exponentiel mais elles sont séparables en temps polynomial !

43/49

Inégalités de cycles cliques

Un algorithme très rapide et performant pour la séparation heuristique des inégalités de clique est l'algorithme glouton suivant :

Pour une solution optimale x^* de la relaxation linéaire :

- on classe les sommets par ordre de valeurs $x^*(u)$ décroissante.
- on initialise un ensemble K avec le premier sommet (de plus grande valeur)
- puis, itérativement, on essaye d'ajouter un sommet dans l'ordre des valeurs s'il forme une clique avec les sommets de K .

Cet algorithme glouton n'est évidemment pas exact mais permet d'obtenir des contraintes de cliques en temps polynomial : on peut donc le réitérer dans un processus de génération de contraintes.

On peut noter également qu'il fournit une inégalité de cliques K pour une clique K maximale au sens de l'inclusion : on verra en section F que c'est très utile car toutes les inégalités de clique non maximale sont "dominées" fortes par les inégalités de clique maximale.

42/49

Inégalités de coupes

On peut construire un algorithme de coupes en prenant comme ensemble initial les contraintes de degré et les contraintes triviales.

Le but ici est de résoudre la relaxation linéaire de cette formulation PLNE. Notons x^* une solution optimale du PL limité aux contraintes initiales.

Le problème de séparation peut alors s'écrire de la façon suivante : déterminer une contrainte (7) violée par x^* si elle existe et dire sinon qu'il n'en n'existe pas.

Ce problème se ramène en fait à déterminer une coupe min dans le graphe G en utilisant pour capacité le vecteur x^* associée aux arêtes de G .

En effet, si l'on dispose d'une coupe $\delta(W^*)$ avec $W^* \subsetneq V$ et $W^* \neq \emptyset$, on a deux cas :

- soit $\sum_{e \in \delta(W^*)} x^*(e) < 2$ dans ce cas, on a déterminé la contrainte suivante qui est violée par x^*

$$\sum_{e \in \delta(W^*)} x(e) \geq 2$$

- soit $\sum_{e \in \delta(W^*)} x^*(e) \geq 2$ dans ce cas, toutes les contraintes (7) sont non violées.

Comme on sait déterminer une coupe-min avec des capacités positives (et x^* est un vecteur positif) en temps polynomial, on sait donc résoudre en temps polynomial cette formulation exponentielle.

44/49

Algorithme de Branch-and-Cut

1. Inégalités valides et renforcement
2. Algorithmes de coupes
3. Algorithmes de Branch-and-Cut
4. Qu'y-a-t-il dans les solveurs entiers ?

A moins que $P = NP$, il semble impossible de posséder un algorithme de coupes pour résoudre un PLNE sans branchement.

Le couplage d'un algorithme de coupes et d'un branchement est appelé algorithme de coupes et branchement ou plutôt **Branch-and-Cut algorithm** :

- chaque nœud de l'arbre de branchement correspond à une relaxation résolue par une méthode de coupes : donc un nombre polynomial mais important de réitération de l'algorithme de coupes. La relaxation d'un nœud s'arrête quand il n'y a plus d'inégalités violées trouvées par l'algorithme.
- l'arbre de branchement suit le principe générique du branchement.

On peut noter que cet algorithme produit un grand nombre de points fractionnaires (qui seront coupés) : ils sont à un moment donné des points extrêmes de la formulation PL en cours de traitement. En utilisant une heuristique primale (rapide) pour chacun de ces points, on énumère un grand nombre de solutions entières... parmi elles il y a bien souvent la solution optimale!

45/49

46/49

Algorithme de Branch-and-Cut

En pratique, un algorithme de Branch-and-Cut nécessite un "framework logiciel"

- ▶ pour l'arbre de branchement
- ▶ pour la résolution des programmes linéaires (utilisant les solveurs linéaires puissants Cplex, LP,...) qui seront utilisés à chaque itération de l'algorithme de coupes, dans chaque nœud.
- ▶ pour l'assemblage du ou des algorithmes de séparation
- ▶ et d'autres astuces...

Plusieurs frameworks existent :

- au sein de Cplex et de Gurobi
- en logiciel Open Source SCIP
- et d'autres projets universitaires : Abacus, BCP de Coin-OR, Symphony,...

1. Inégalités valides et renforcement

2. Algorithmes de coupes

3. Algorithmes de Branch-and-Cut

4. Qu'y-a-t-il dans les solveurs entiers ?

47/49

48/49

Qu'y-a-t-il dans les solveurs ?

Un solveur entier comme CPLEX, GUROBI, XPRESS, GLPK, SCIP,... est en fait la mise en œuvre d'un algorithme de Branch-and-Cut automatique.

- ▶ Les algorithmes de séparations sont issus d'inégalités génériques (comme celles de covers, de cycles impairs ou de cliques) à partir de la connaissance du PLNE compact fourni en entrée.
- ▶ L'arbre de branchement est l'arbre classique du Branch-and-Bound gérés très optimalement.
- ▶ Les plus puissants possèdent en plus une phase de prétraitement par inférence logique qui permettent de réduire la taille du problème
- ▶ et il y a de nombreuses techniques connues ou secrètes : le Strong Branching, les Local Cuts, les Lifting automatiques,... qui sont pour la plupart issues de recherche publiée par des chercheurs de tout horizon.

Master d'Informatique - Spécialité Androide

Module MAOA

Recherche Opérationnelle et Optimisation Combinatoire

Partie D - Approches Polyédrales

Pierre Fouilhoux

Sorbonne Université

2019-2020

1. Introduction

2. Définitions et résultats fondamentaux

3. Approches polyédrales

2/75

1. Introduction

1.1 Présentation dans \mathbb{R}^2

1.2 Enveloppe convexe

1.3 Polytope des solutions

2. Définitions et résultats fondamentaux

3. Approches polyédrales

Rappels : polyèdre et points extrêmes

Dans \mathbb{R}^n , on appelle *hyperplan* H un sous-espace de \mathbb{R}^n défini comme l'ensemble des points vérifiant une équation linéaire, c'est-à-dire qu'il existe $a_1, \dots, a_n, b \in \mathbb{R}$ tels que $H = \{x \in \mathbb{R}^n \mid a_1x_1 + \dots + a_nx_n = b\}$.

Un *polyèdre* $P \subseteq \mathbb{R}^n$ est l'ensemble des solutions d'un système fini d'inégalités linéaires, i.e.,

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\},$$

où A est une matrice $m \times n$ (m et n entiers positifs) et $b \in \mathbb{R}^m$.

On dit alors que le système $Ax \leq b$ **définit** ou **caractérise** le polyèdre P .

Ainsi, un polyèdre est tout simplement une figure géométrique définie comme la partie délimitée par des "plans".

En dimension 1 : les hyperplans sont des points
les polyèdres sont des intervalles.

En dimension 2 : les hyperplans sont des droites
les polyèdres sont des carrés, des rectangles,...

En dimension 3 : les hyperplans sont des plans
les polyèdres sont des cubes, des dodécaèdres,...

Rappels : polyèdre et points extrêmes

Un **polytope** est un polyèdre borné, c'est-à-dire qu'un polyèdre $P \subseteq \mathbb{R}^n$ est un polytope s'il existe $x^1, x^2 \in \mathbb{R}^n$ tel que $x^1 \leq x \leq x^2$, pour tout $x \in P$.

Un **point** d'un polyèdre est donc défini par des coordonnées $\tilde{x} \in \mathbb{R}^n$ tel que $A\tilde{x} \leq b$.

Un point x d'un polyèdre P est dit **point extrême** (ou parfois *sommet*) (vertex) de P s'il n'existe pas deux solutions x^1 et x^2 de P , $x^1 \neq x^2$, telles que $x = \frac{1}{2}x^1 + \frac{1}{2}x^2$.
 En d'autre terme, un point extrême de P est un point de P qui n'est pas le milieu d'un segment contenu dans P .

THEOREME : Un point \tilde{x} d'un polyèdre P est un point extrême de P si et seulement on peut produire n inégalités de la matrice A qui sont vérifiées par \tilde{x} à l'égalité et qui sont linéairement indépendantes.

1. Introduction

1.1 Présentation dans \mathbb{R}^2

1.2 Enveloppe convexe

1.3 Polytope des solutions

2. Définitions et résultats fondamentaux

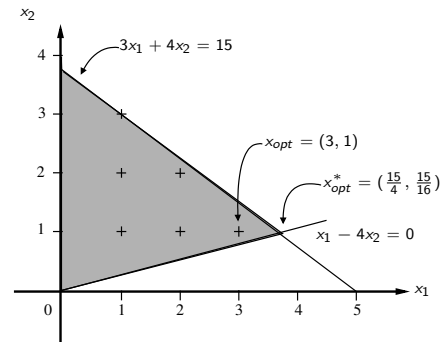
3. Approches polyédrales

5/75

6/75

Considérer le cas très simple de ce PLNE (P) à 2 variables

$$\begin{aligned} \text{Max } z &= 2x_1 + x_2 \\ x_1 - 4x_2 &\leq 0, \\ 3x_1 + 4x_2 &\leq 15, \\ x_1 &\geq 0, \\ x_2 &\geq 0, \\ x_1, x_2 &\in \mathbb{N}. \end{aligned}$$



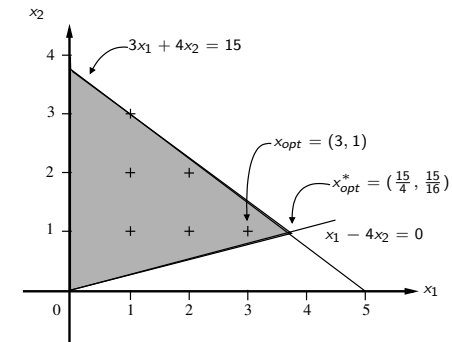
On note (P^*) sa relaxation linéaire (obtenue en enlevant la contrainte d'intégrité.)

En deux dimensions, on peut visualiser le domaine de définition de (P^*) : il est donné par les 4 contraintes du programme : c'est un polyèdre (dessiné en gris sur la figure.)

7/75

Considérer le cas très simple de ce PLNE (P) à 2 variables

$$\begin{aligned} \text{Max } z &= 2x_1 + x_2 \\ x_1 - 4x_2 &\leq 0, \\ 3x_1 + 4x_2 &\leq 15, \\ x_1 &\geq 0, \\ x_2 &\geq 0, \\ x_1, x_2 &\in \mathbb{N}. \end{aligned}$$



Notons

x_{opt} solution entière optimale de (P).

x_{opt}^* solution "fractionnaire" optimale de (P^*).

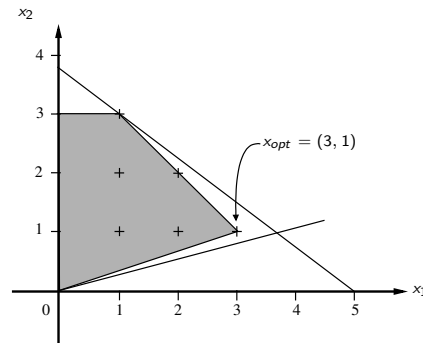
La solution optimale de (P^*) est ici un des sommets de ce polyèdre pour une valeur fractionnaire x_{opt}^* associée à la valeur $z_{opt}^* = 8 + \frac{7}{16}$.

Les points entiers contenus dans le polyèdre gris forment donc l'ensemble des solutions possibles pour (P). La solution optimale est ici unique, c'est le point x_{opt} de valeur $z_{opt} = 7$.

8/75

Regardons les points extrêmes qui sont entiers par construction).

$$\begin{aligned} \text{Max } z &= 2x_1 + x_2 \\ x_1 - 4x_2 &\leq 0, \\ 3x_1 + 4x_2 &\leq 15, \\ x_1 &\geq 0, \\ x_2 &\geq 0, \\ x_1, x_2 &\in \mathbf{N}. \end{aligned}$$



Prenons un élastique circulaire assez serré et lâchons-le “autour” des points entiers : on obtient un nouveau polyèdre (en gris sur le dessin) : ce polyèdre est l'**enveloppe convexe** des points entiers solutions.

Prenons x_{opt} point extrême optimale de l'enveloppe convexe des solutions entières de (P) : **il est par construction entier !**

Remarques importantes :

- Des solutions optimales de (P) (donc entières) peuvent être prises parmi les points extrêmes (i.e. les sommets) de ce polyèdre qu'est l'enveloppe convexe des solutions de (P) .
Connaissant l'enveloppe convexe, trouver un point extrêmes optimales (donc entier) de l'enveloppe convexe revient à résoudre un PL !
- On peut remarquer que le polyèdre convexe des points entiers est bien entendu contenu dans le polyèdre défini par les inégalités de la formulation du programme (appelé souvent domaine de définition du programme linéaire).
- Dans le cas d'une maximisation, on a naturellement $z_{opt} \leq z_{opt}^*$.
- Ces remarques sont toujours vraies quelque soit la dimension du problème.

Contourner l'énumération des solutions

Plutôt que chercher une solution optimale, on recherche alors l'enveloppe convexe !

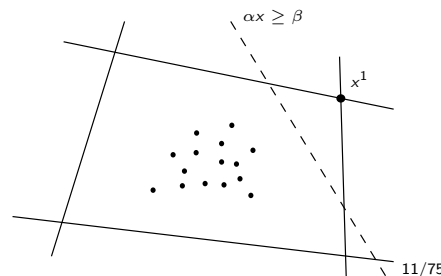
L'enveloppe convexe est alors l'inconnue ! Une fois déterminée, résoudre le programme linéaire associée à ce polyèdre donne la solution optimale de (P) .

Dans le schéma suivant, les droites schématisent les contraintes connues dans le programme (P) :

- les contraintes $Ax \leq B$ de la formulation,
- les inégalités $\alpha x \leq \beta$ valides ajoutées.

Les points noirs schématisent les solutions entières (inconnues a priori) dont on recherche l'enveloppe convexe (inconnue de même).

$$(P_2) \begin{cases} \text{Max } cx \\ Ax \leq B \\ \alpha x \leq \beta \\ x \text{ entier} \end{cases}$$



Contourner l'énumération des solutions

Toute la problématique d'une **approche polyédrale** pour un problème d'optimisation combinatoire est de savoir quelles sont les “meilleures” inégalités à ajouter pour s'approcher de l'enveloppe convexe des solutions du problème.

Si on connaît les inégalités linéaires définissant l'enveloppe convexe, résoudre un PLNE revient à résoudre un simple PL (par un algorithme polynomial !)

On contourne ainsi l'explosion combinatoire de l'énumération des solutions

Malheureusement si en petites dimensions le calcul de l'enveloppe convexe des points entiers est réalisable, on ne sait pas déterminer l'enveloppe convexe autrement que par des algorithmes exponentiels ! Cette approche ne peut être “automatiser”.

A moins que $P = NP$, on ne peut pas connaître et énumérer l'enveloppe convexe d'un problème NP-difficile.

Heureusement il y a plein de cas où l'on sait quand même déterminer toute ou partie de l'enveloppe convexe et améliorer fortement la résolution des PLNE par une approche polyédrale **dédiée** et **partielle**.

Combinaison convexe

1. Introduction

1.1 Présentation dans \mathbb{R}^2

1.2 Enveloppe convexe

1.3 Polytope des solutions

2. Définitions et résultats fondamentaux

3. Approches polyédrales

- Un ensemble $S \subset \mathbb{R}^n$ est **convexe** si, pour deux points quelconques x_1 et x_2 pris dans S , le segment $[x_1, x_2]$ est tout entier contenu dans S , i.e.

$$\forall x_1 \in S, \forall x_2 \in S, \forall \lambda \in [0, 1], \text{ alors } \lambda x_1 + (1 - \lambda)x_2 \in S.$$

- Soient p points x^1, \dots, x^p dans \mathbb{R}^n .
Un point $x \in \mathbb{R}^n$ est une **combinaison convexe** des points x^1, \dots, x^p s'il existe p scalaires $\lambda_1, \lambda_2, \dots, \lambda_p \in \mathbb{R}_+$ tels que

$$x = \sum_{i=1}^p \lambda_i x^i \quad \text{et} \quad \sum_{i=1}^p \lambda_i = 1$$

Par exemple, pour $p = 2$, tout point du segment $[x^1, x^2]$ est une combinaison convexe de x^1, x^2 .

Enveloppe convexe

Remarque :

Etant donné un ensemble de points $S = \{x^1, \dots, x^p\}$, on peut avoir besoin de tester si un point donné x^* appartient ou non à $\text{conv}(S)$. Ce problème se ramène à vérifier si x^* est combinaison convexe des points x^1, \dots, x^p , c'est-à-dire, s'il existe $\lambda_1, \dots, \lambda_p$ tels que

$$\begin{aligned} \sum_{i=1}^p \lambda_i &= 1, \\ \sum_{i=1}^p \lambda_i x^i &= x^*, \\ \lambda_i &\geq 0, \text{ pour } i = 1, \dots, p. \end{aligned}$$

C'est-à-dire vérifier que ce système d'égalités et d'inégalités contient une solution : cela peut se faire en temps polynomial (par exemple par un PL avec une fonction objective quelconque).

- Soit S un ensemble non vide de points de \mathbb{R}^n .
L'**enveloppe convexe** des points de S , notée $\text{conv}(S)$, est l'ensemble de tous les points de \mathbb{R}^n qui peuvent s'écrire comme combinaison convexe de points de S .

Remarque : l'enveloppe convexe d'un ensemble de points S peut être vue également comme le plus petit ensemble convexe contenant S .

- En fait, un ensemble $C \subset \mathbb{R}^n$ est convexe si et seulement si tout point pouvant s'écrire comme une combinaison convexe des points de C est dans C .
En d'autres termes, un ensemble C est convexe si et seulement si $C = \text{conv}(C)$.

Remarque : toute intersection d'un nombre fini d'ensemble convexe est encore convexe.

1. Introduction

- 1.1 Présentation dans \mathbb{R}^2
- 1.2 Enveloppe convexe
- 1.3 Polytope des solutions

2. Définitions et résultats fondamentaux

3. Approches polyédrales

Les résultats donnés dans ce chapitre s'appliquent pour un polyèdre quelconque. Mais l'objectif de ce document est d'étudier plus particulièrement les polyèdres définis à partir d'un problème d'optimisation combinatoire.

Soient \mathcal{P} un **problème d'optimisation combinatoire** sur n décisions (ou variables). Notons S l'ensemble des vecteurs d'incidence de toutes les solutions de \mathcal{P} . Notons c une fonction coût associée aux variables du problème.

Le problème \mathcal{P} s'écrit

$$\max \{c\chi \mid \chi \in S\}$$

Regardons l'enveloppe convexe $conv(S)$ des solutions S de \mathcal{P} .

Un exemple en dimension 3 :

Considérons le problème du sous-graphe acyclique induit, c'est-à-dire dont les solutions sont les ensembles de sommets n'induisant aucun cycle.

Considérons le cas très simple d'un graphe limité à un cycle de 3 sommets (un triangle).

Notons 1, 2 et 3 les 3 sommets de cycle.

Considérons les solutions de ce problème, c'est-à-dire des ensembles de sommets induisant des graphes sans cycle :

$$\emptyset \quad \{1\} \quad \{2\} \quad \{3\} \quad \{1,2\} \quad \{1,3\} \quad \{2,3\}$$

Quels sont leurs vecteurs d'incidence ?

$$\begin{aligned} \chi^\emptyset &= [0 \ 0 \ 0] \\ \chi^{\{1\}} &= [1 \ 0 \ 0] \\ \chi^{\{2\}} &= [0 \ 1 \ 0] \\ \chi^{\{3\}} &= [0 \ 0 \ 1] \\ \chi^{\{1,2\}} &= [1 \ 1 \ 0] \\ \chi^{\{1,3\}} &= [1 \ 0 \ 1] \\ \chi^{\{2,3\}} &= [0 \ 1 \ 1] \end{aligned}$$

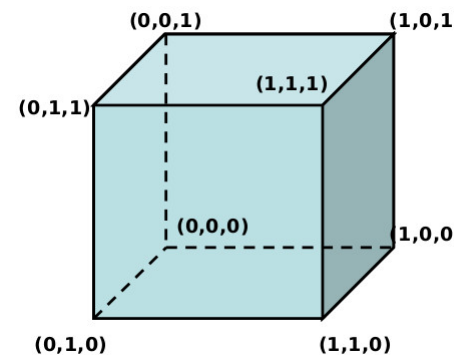
Par contre le point

$$\chi^{\{1,2,3\}} = [1 \ 1 \ 1]$$

n'est pas solution.

Un exemple en dimension 3 :

Quelle est l'enveloppe convexe de ces 7 points ?



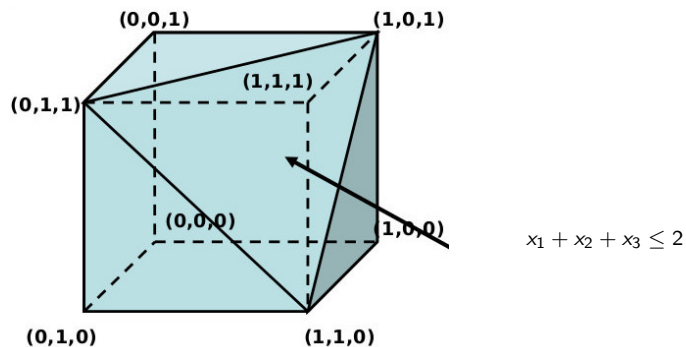
Cette enveloppe convexe est incluse dans l'hypercube de dimension 3.

Qui est simplement donnée par :

$$\begin{aligned} x_1 &\leq 1 \\ x_2 &\leq 1 \\ x_3 &\leq 1 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \\ x_3 &\geq 0 \end{aligned}$$

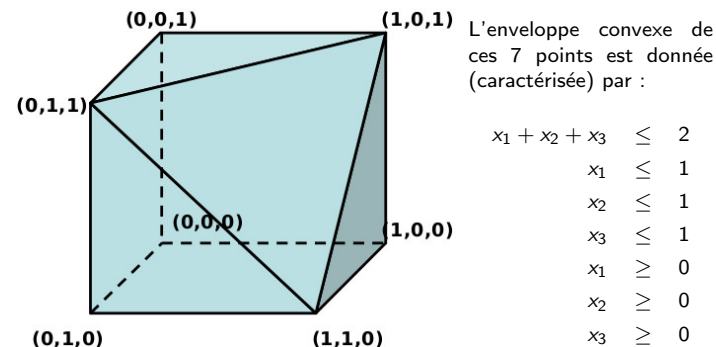
Un exemple en dimension 3 :

Quelle est l'enveloppe convexe de ces 7 points ?



Un exemple en dimension 3 :

Quelle est l'enveloppe convexe de ces 7 points ?



S versus conv(S)

Soient \mathcal{P} un problème d'optimisation combinatoire sur n décisions (ou variables).
Notons S l'ensemble des vecteurs d'incidence de toutes les solutions de \mathcal{P} .
Notons c une fonction coût associée aux variables du problème.
Le problème \mathcal{P} est

$$\max \{c\chi \mid \chi \in S\}$$

Considérons l'enveloppe convexe $conv(S)$ des solutions de \mathcal{P} .

Regardons le problème suivant

$$\max \{c\chi \mid \chi \in conv(S)\}$$

Deux remarques essentielles :

- Tous les points extrêmes de l'enveloppe convexe $conv(S)$ sont entiers par construction.
- Des points optimaux de S sont parmi les points extrêmes de l'enveloppe convexe $conv(S)$.

Mais qu'est-ce que $conv(S)$?

Une enveloppe convexe est un polyèdre

Theorem (de Minkowski)

Un ensemble (non vide) de points $P \subseteq \mathbb{R}^n$ est un polytope si et seulement s'il existe un ensemble de points S tel que $P = conv(S)$.

Conséquences essentielles :

- $conv(S)$ est un polytope (ce que nous avons déjà constaté sans le prouver).
- donc $conv(S)$ peut être décrit par un système fini d'inégalités linéaires
- et $\max \{c\chi \mid \chi \in conv(S)\}$ est un programme linéaire.

Optimiser sur $conv(S)$

La proposition suivante établit la relation entre :

- optimiser sur l'ensemble discret S
- optimiser sur l'enveloppe convexe (continue) $conv(S)$.

Theorem

Soient $S \subseteq \mathbb{R}^n$ un ensemble de points et c un vecteur de \mathbb{R}^n . Alors

$$\max\{cx \mid x \in S\} = \max\{cx \mid x \in conv(S)\}.$$

Preuve. Soient $\bar{x} \in S$ tel que $c\bar{x} = \max\{cx \mid x \in S\}$ et $x^* \in conv(S)$ tel que $cx^* = \max\{cx \mid x \in conv(S)\}$. Comme $\bar{x} \in S \subseteq conv(S)$, alors $c\bar{x} \leq cx^*$. De plus, comme x^* est un point optimal sur un polyèdre, x^* est un point extrême de $conv(S)$. Par conséquent $x^* \in S$. Ce qui implique que $cx^* \leq c\bar{x}$, et alors $cx^* = c\bar{x}$. \square

25/75

1. Introduction

2. Définitions et résultats fondamentaux

- 2.1 Point intérieur et dimension
- 2.2 Face d'un polyèdre
- 2.3 Facettes
- 2.4 Etude faciale sur $conv(S)$
- 2.5 Description minimale

3. Approches polyédrales

27/75

Polytope des solutions

- Le théorème précédent établit le lien entre optimiser sur un ensemble de points et optimiser sur un ensemble convexe.

Optimiser \mathcal{P} correspond à optimiser sur $conv(S)$.

- Par la programmation linéaire, on sait que les points extrêmes de $conv(S)$ sont des solutions optimales de \mathcal{P} , c'est-à-dire précisément les solutions de S .

Optimiser sur $conv(S)$, c'est résoudre un PL!

Une approche polyédrale pour un problème d'optimisation combinatoire consiste à rechercher l'enveloppe convexe de ses solutions. Car si l'on sait caractériser le polyèdre $conv(S)$ par un système d'inégalités linéaires, alors on a ramené le problème \mathcal{P} à la résolution d'un programme linéaire!

Cette enveloppe convexe est appelée le polytope associé au problème ou encore **polytope des solutions du problème**.

26/75

1. Introduction

2. Définitions et résultats fondamentaux

- 2.1 Point intérieur et dimension
- 2.2 Face d'un polyèdre
- 2.3 Facettes
- 2.4 Etude faciale sur $conv(S)$
- 2.5 Description minimale

3. Approches polyédrales

28/75

Exemple : Le polytope du stable

Soit $G = (V, E)$ un graphe non orienté.

Considérons les solutions possibles pour le problème du stable de poids maximum **quelque soit le coût associé aux sommets**.

Les solutions du problème du stable maximum sont les sous-ensembles de sommets $S \subset V$ qui sont des stables (c'est-à-dire n'ayant pas d'arêtes les reliant dans G). Notons $stab(G)$ l'ensemble des stables de G , ainsi $stab(G)$ est l'ensemble des solutions du problème du stable maximum.

Regardons quelques exemples :

- $\emptyset \in stab(G)$: l'ensemble vide est un stable.
- $\{u\} \in stab(G)$ pour chaque sommet u : les singletons sont des stables.
- $\{u, v\} \in stab(G)$ pour toute paire u, v de sommets telle que $uv \notin E$: chaque non-arête est un stable.

Exemple : Le polytope du stable

On note alors $P(G)$ le **polytope des stables** de G l'enveloppe convexe des des vecteurs d'incidence des solutions du problèmes du stable, *i.e.*

$$P(G) = conv\{\chi^S \mid S \text{ est un stable de } G\}.$$

On sait que tous les points extrêmes de $P(G)$ sont des vecteurs d'incidence d'un stable de G : on ne peut pas polynomialement décrire $P(G)$ par ces points extrêmes.

Peut-on le décrire autrement ?

Quelle est déjà sa dimension : est-il dans R^3 ?... non bien sûr.

On sait que les vecteurs d'incidence d'un stable sont dans R^n avec $n = |V|$ le nombre de sommets de G : mais il peut être plus petit ? Répondons à cette question.

Exemple : Le polytope du stable

A chaque stable $S \in stab(G)$, on fait correspondre un **vecteur d'incidence** χ^S tel que

$$\chi^S[u] = \begin{cases} 1 & \text{si } u \in S \\ 0 & \text{sinon} \end{cases}$$

- Pour $\emptyset \in stab(G)$

$$\chi^\emptyset = [0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

- $\{u\} \in stab(G)$ pour chaque sommet u :

$$\chi^{\{u\}} = [0 \ 0 \ 1 \ 0 \ 0 \ 0]$$

u

- $\{u, v\} \in stab(G)$ si u, v non-arête de G .

$$\chi^{\{u,v\}} = [0 \ 0 \ 1 \ 0 \ 1 \ 0]$$

$u \qquad v$

Combinaison linéaire

Definition

Prenons un vecteur dans l'espace $x \in R^n$.

On dit que le vecteur x est une **combinaison linéaire** des k vecteurs $x^1, \dots, x^k \in R^n$ s'il existe k scalaires $\lambda_1, \lambda_2, \dots, \lambda_k \in R$ tels que $x = \sum_{i=1}^k \lambda_i x^i$.

Exemples :

Dans R^2 : un vecteur x^1 est combinaison linéaire d'un vecteur x^2 s'il existe un réel quelconque λ tel que $x^1 = \lambda x^2$: c'est-à-dire que les vecteurs représentent la même "direction".

Dans R^3 : un vecteur x^3 n'est pas combinaison linéaire de deux autres vecteurs x^1, x^2 si x^3 n'est pas dans le plan formé par x^1 et x^2 (s'ils ne sont pas coplanaires).

Dans R^n : parmi $n + 1$ vecteurs, l'un d'entre eux est nécessairement combinaison linéaire de n autres.

Indépendance linéaire

Definition

Des vecteurs $x^1, \dots, x^k \in \mathbb{R}^n$ sont dits **linéairement indépendants** si le système

$$\sum_{i=1}^k \lambda_i x^i = 0$$

admet une solution unique, $\lambda_i = 0$ pour $i = 1, \dots, k$.

Exemple dans \mathbb{R}^3

- Deux vecteurs linéairement indépendants x^1 et x^2 forment un repère à 2 dimensions, c'est-à-dire qu'ils engendrent un espace vectoriel de dimension 2 formé par tous les vecteurs qui s'écrivent $\lambda_1 x^1 + \lambda_2 x^2$ pour toutes les valeurs réelles λ_1 et λ_2 .
- Tout vecteur x^3 qui n'est pas dans cet espace vectoriel de dimension 2 engendrera alors tout l'espace \mathbb{R}^3 en formant un repère à 3 dimension.

Remarque essentielle :

Si un polyèdre contient un repère vectoriel à p dimension, ce polyèdre est au moins un objet géométrique de dimension p .

33/75

Boîte à outil de l'algèbre linéaire :

Considérons k vecteurs $x^1, \dots, x^k \in \mathbb{R}^n$.

Posons la matrice M dont les lignes (ou colonnes) sont les coordonnées de ces k vecteurs.

On appelle **rang de M** , noté $\text{rang}(M)$, le nombre maximum de lignes (ou de colonnes) de M linéairement indépendantes.

Conséquemment, si $\text{rang}(M)=k$, les vecteurs sont linéairement indépendants.

Si la matrice est carrée, $k = n$, alors les n vecteurs sont linéairement indépendants si et seulement si le **déterminant** de la matrice M est non-nul.

Quelques cas classiques de matrice à déterminant non-nul.

La matrice Identité	Matrice triangulaire à "diagonale de produit non nul"	Matrice à "diagonale faite de matrices à déterminant non nul"
$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 \\ 5 & 1 & 1 & 1 & 0 \\ 2 & 0 & 1 & 2 & 3 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

34/75

Vecteurs et points...

En prenant un ensemble S de solutions dans l'espace \mathbb{R}^n , on regarde un ensemble de **points** de l'espace : donc la connaissance des polytopes des solutions d'un problème d'optimisation combinatoire vient de points et non de vecteurs !

Le vecteur d'incidence χ^s d'une solution $s \in S$ donne les coordonnées du point s dans \mathbb{R}^n .

Pour deux points s_1 et s_2 de S , les coordonnées du vecteur $\overrightarrow{s_1 s_2}$ dans \mathbb{R}^n sont $\chi^{s_2} - \chi^{s_1}$.

Remarque :

Pour avoir d vecteurs, il faut $d + 1$ points.

Par exemple $d + 1$ points x^1, \dots, x^{d+1} forment d vecteurs $x^2 - x^1, x^3 - x^1, \dots, x^d - x^1, x^{d+1} - x^1$.

35/75

Indépendance affine

Definition

Des points $x^1, \dots, x^k \in \mathbb{R}^n$ sont dits **affinement indépendants** si $x^2 - x^1, \dots, x^k - x^1$ sont $k - 1$ vecteurs linéairement indépendants.

la combinaison linéaire concerne des vecteurs et la combinaison affine des points de l'espace.

Exemple :

Dans \mathbb{R}^2 , considérons trois points A, B, C dont les coordonnées des points sont x_A, x_B et x_C . On peut considérer les vecteurs $x_B - x_A$ et $x_C - x_A$.

Demander "Les trois points A, B, C sont-ils non-alignés ?" revient à demander "les 3 points sont-ils affinement indépendants ?"

Demander "Les 2 vecteurs $x_B - x_A$ et $x_C - x_A$ sont-ils non-parallèles ?" revient à demander "Les 2 vecteurs sont-ils linéairement indépendants ?"

Ces 4 questions sont identiques d'un point de vue "indépendance" des coordonnées !

36/75

Indépendance affine

Theorem

Des points $x^1, \dots, x^k \in \mathbb{R}^n$ sont affinement indépendants si et seulement si le système

$$\sum_{i=1}^k \lambda_i x^i = 0 \quad \text{et} \quad \sum_{i=1}^k \lambda_i = 0$$

admet une solution unique, $\lambda_i = 0$ pour $i = 1, \dots, k$.

Preuve.

(\Rightarrow) Soient $x^1, \dots, x^k \in \mathbb{R}^n$ des points affinement indépendants. Considérons le système, $\sum_{i=1}^k \lambda_i x^i = 0$ et $\sum_{i=1}^k \lambda_i = 0$. Par la 2^{de} équation, on obtient $\lambda_1 = -\sum_{i=2}^k \lambda_i$. Donc on peut en déduire que $\sum_{i=2}^k \lambda_i x^i - (\sum_{i=2}^k \lambda_i) x^1 = 0$ et ainsi $\sum_{i=2}^k \lambda_i (x^i - x^1) = 0$. Or par définition, $x^2 - x^1, \dots, x^k - x^1$ sont $k - 1$ vecteurs linéairement indépendants. Donc le système $\sum_{i=2}^k \lambda_i (x^i - x^1) = 0$ admet une solution unique, $\lambda_i = 0$ pour $i = 2, \dots, k$ et par conséquent $\lambda_1 = 0$.
 (\Leftarrow) Soient x^1, \dots, x^k des points de \mathbb{R}^n tels que $\sum_{i=1}^k \lambda_i x^i = 0$ et $\sum_{i=1}^k \lambda_i = 0$ admet une solution unique, $\lambda_i = 0$ pour $i = 1, \dots, k$. Considérons alors le système $\sum_{i=2}^k \lambda_i (x^i - x^1) = 0$. Ce système s'écrit également $\sum_{i=2}^k \lambda_i x^i - (\sum_{i=2}^k \lambda_i) x^1 = 0$ et notons que la somme des coefficients des coordonnées du système est nulle. Donc par hypothèse ce système admet pour solution unique $\lambda_i = 0$ pour $i = 2, \dots, k$ (et $-\sum_{i=2}^k \lambda_i = 0$). Ainsi les vecteurs $x^2 - x^1, \dots, x^k - x^1$ sont linéairement indépendants et les points sont donc x^1, \dots, x^k affinement indépendants. \square

Indépendance linéaire ou affine ?

Remarque algébrique : Les coordonnées d'un point x peuvent être vues comme les coordonnées d'un vecteur (par exemple en considérant le vecteur allant de l'origine au point x qui est de coordonnées x).
 Si l'on considère des éléments x^1, \dots, x^k de \mathbb{R}^n , on peut donc les voir comme des points ou des vecteurs !

Remarque utile 1 :

Par le théorème précédent, si les éléments x^1, \dots, x^k sont linéairement indépendant alors ils sont affinement indépendants.

Par contre la réciproque n'est pas vraie.

Remarque utile 2 :

Si on veut montrer que $0, x^1, \dots, x^n$ sont affinement indépendants, alors il suffit de montrer que x^1, \dots, x^n linéairement indépendants !

Dimension d'un polyèdre

Definition

Un polyèdre P de \mathbb{R}^n est de **dimension** d s'il y a au maximum $d + 1$ points affinement indépendants dans P .

On écrit alors $\dim(P) = d$.

Un polyèdre P est dit de **pleine dimension** si $\dim(P) = n$.

Quelques polyèdres :

- une droite est de dimension 1.
- une droite dans \mathbb{R}^2 n'est donc pas de pleine dimension.
- une droite dans \mathbb{R}^1 est de pleine dimension.
- une droite est de dimension 1 est de petite dimension quand elle est plongée dans un espace de dimension \mathbb{R}^n , $n \geq 1$.
- un plan (de \mathbb{R}^2) plongé dans l'espace \mathbb{R}^3 est de dimension 2 et n'est donc pas de pleine dimension.
- un cube est $[0, 1]^3$ où $[0, 1]$ désigne le segment continue de 0 à 1.
- un cube est de dimension 3 est de pleine dimension dans \mathbb{R}^3 .
- un cube n'est pas de pleine dimension dans \mathbb{R}^4 .
- un hypercube $[0, 1]^n$ est de pleine dimension dans \mathbb{R}^n .

Exemple : le polytope du stable

Soit $G = (V, E)$ un graphe simple non orienté avec $n = |V|$.

Soit $P(G)$ le polytope des stables de G , i.e.,

$$P(G) = \text{conv}\{\chi^S \mid S \text{ est un stable de } G\}.$$

On sait que $P(G) \subset \mathbb{R}^n$.

Et même que $P(G) \subset [0, 1]^n$ en notant $[0, 1]$ le segment continu de 0 à 1.

Montrons que $P(G)$ est de pleine dimension, c'est-à-dire que $\dim(P(G)) = n$.

Réponse :

Il suffit de produire $n + 1$ points de \mathbb{R}^n affinement indépendants qui sont dans $P(G)$. Pour cela on pourrait produire des points fractionnaire quelconques de cet espace "continu" qu'est un polytope... Mais on peut se limiter à énumérer des points entiers... qui sont ici des vecteurs d'incidence de stables !

Exemple : le polytope du stable

Par exemple, les vecteurs d'incidence :

- de l'ensemble vide \emptyset
- et des singletons $\{u\}$, $u \in V$.

En effet, ces ensembles sont bien tous des stables et il y en a bien $n + 1$.

De plus, les vecteurs $\chi^{\{u\}} - \chi^\emptyset$ sont bien linéairement indépendants. En effet la matrice dont les lignes sont ces vecteurs est la matrice identité qui est bien de plein rang :

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Donc les $n + 1$ points $\chi^\emptyset, (\chi^{\{u\}})_{u \in V}$ sont affinement indépendants.

Par conséquent, $P(G)$ est de pleine dimension. □

Rang et Dimension

Rappel : pour une matrice donnée M , on appelle *rang de M* , noté $\text{rang}(M)$, le nombre maximum de lignes (ou de colonnes) de M linéairement indépendantes.

Theorem

Si $P \neq \emptyset$, alors $\dim(P) = n - \text{rang}(B)$ (où B est la matrice des égalités).

Preuve. On peut remarquer que si M est de rang r et s'il existe un vecteur l tel que $\{x \in \mathbb{R}^n \mid Mx = l\} \neq \emptyset$, alors le nombre maximum de points affinement indépendants dans l'ensemble $\{x \in \mathbb{R}^n \mid Mx = l\}$ est $n - r + 1$. Posons $k = \text{rang}(B)$. Par la remarque précédente, le système $Bx = 0$ admet $n - k + 1$ solutions x^1, \dots, x^{n-k+1} affinement indépendantes. Comme $P \neq \emptyset$, par le lemme sur les points intérieurs, P contient un point intérieur \bar{x} , donc $A_i \bar{x} < b_i$ pour $i = 1, \dots, m_1$. Alors il existe $\epsilon > 0$ tel que $A_i \bar{x} + \epsilon A_i x^j \leq b_i$, pour tout $i = 1, \dots, m_1$ et $j = 1, \dots, n - k + 1$. On considère alors les points $y^j = \bar{x} + \epsilon x^j$ pour $j = 1, \dots, n - k + 1$. Nous avons $y^j \in P$ pour $j = 1, \dots, n - k + 1$. De plus, comme les points x^1, \dots, x^{n-k+1} sont affinement indépendants, les points y^1, \dots, y^{n-k+1} le sont aussi. Ce qui implique que $\dim(P) \leq n - k$. Par conséquent, $\dim(P) = n - k$. □

Corollaire : En conséquence de ce théorème, un polyèdre est de pleine dimension si et seulement s'il contient un point vérifiant toutes les contraintes du polyèdre avec inégalité stricte.

Ce résultat est néanmoins difficile à appliquer en général car les égalités du polytope sont parfois difficiles à déterminer.

Points intérieurs

Considérons maintenant un polyèdre $P \subseteq \mathbb{R}^n$ et considérons une description de P par un système de m_1 inéquations et m_2 équations, ainsi :

$$P = \left\{ x \in \mathbb{R}^n \mid \begin{array}{l} A_i x \leq b_i, \quad i = 1, \dots, m_1 \\ B_j x = d_j, \quad j = 1, \dots, m_2 \end{array} \right\}.$$

Cette écriture implique que, pour toute inéquation $A_i x \leq b_i$, $i \in \{1, \dots, m_1\}$, il existe une solution \tilde{x} de P telle que $A_i \tilde{x} < b_i$ (sinon, elle serait rangée parmi les équations). On note dans cette section A, B, b et d les matrices et vecteurs correspondant à cette écriture de P .

Un point $x^* \in P$ est appelé **point intérieur** de P si $A_i x^* < b_i$ pour $i = 1, \dots, m_1$.

Lemma

Si P est non vide, alors P contient un point intérieur.

Preuve. Pour $i = 1, \dots, m_1$, soit $x^i \in P$ tel que $A_i x^i < b_i$. Posons $\bar{x} = \frac{1}{m_1} \sum_{i=1}^{m_1} x^i$. Pour $i = 1, \dots, m_1$, $A_i \bar{x} = \frac{1}{m_1} \sum_{i=1}^{m_1} A_i x^i < \frac{1}{m_1} m_1 b_i < b_i$. De plus, pour $j = 1, \dots, m_2$, $B_j \bar{x} = d_j$. Donc \bar{x} est un point intérieur. □

Pas de théorème "fort" de Carathéodory

Soit un ensemble de points S et le polytope défini par son enveloppe convexe $\text{conv}(S)$. On peut se demander combien de points sont nécessaires pour définir par combinaison tous les points d'une enveloppe convexe...

On a le résultat suivant connu sous le nom de théorème de Carathéodory pour les ensembles convexes.

Theorem

Soient $S \subseteq \mathbb{R}^d$ un ensemble de points de \mathbb{R}^n et x un point donné de $\text{conv}(S)$. Alors x peut être écrit comme une combinaison convexe de $d' \leq n + 1$ points affinement indépendants dans $\text{conv}(S)$.

Ce théorème de Carathéodory peut se lire comme le fait que tout point $x \in \mathbb{R}^n$ dans l'enveloppe convexe de d points peut s'écrire comme la combinaison convexe d'au plus $n + 1$ points affinement indépendants de S . Or $n + 1$ est bien souvent petit devant d (dans notre cas de polytope de solutions, d est exponentiel).

Malheureusement, ceci ne signifie aucunement qu'il existe une sorte de "base convexe" de taille réduite (c'est-à-dire trouver un ensemble générateur)! En effet, cet ensemble de points générateurs est spécifique à chaque point x !

Comment décrire le polytope des solutions ?

Dans le cadre des polytope des solutions d'un problème d'optimisation combinatoire, le nombre d des solutions de S est souvent exponentiel !

Le théorème précédent n'indique en rien que l'on puisse se passer de certains de ces points solutions.

En fait, il semble naturel (et on le montre un peu plus loin) que tous les points extrêmes de $\text{conv}(S)$ sont nécessaires à décrire tout $\text{conv}(S)$... or on peut facilement construire des polytope où tous les points de S sont points extrêmes de $\text{conv}(S)$: c'est par exemple le cas lorsque S est un sous-ensemble des points extrêmes 0-1 d'un hypercube !

Alors comment décrire le polytope des solutions ?
Par ses faces !

1. Introduction

2. Définitions et résultats fondamentaux

2.1 Point intérieur et dimension

2.2 Face d'un polyèdre

2.3 Facettes

2.4 Etude faciale sur $\text{conv}(S)$

2.5 Description minimale

3. Approches polyédrales

45/75

Si on arrive à donner la dimension d'un polyèdre, on ne connaît pas forcément les inégalités qui le décrivent.

Etudions déjà les propriétés de ces inégalités.

On considère dans cette section un polyèdre $P \subseteq \mathbb{R}^n$, et supposons que nous connaissons un système qui décrit P et qui est écrit comme dans la section précédente. Nous allons étudier quelles sont les inégalités de ce système essentielles à la description de (P) .

46/75

Rappel : Inégalités valides

Une inégalité $ax \leq \alpha$ est **valide** pour un polyèdre P si tout point de P la vérifie, i.e. $ax \leq \alpha$ pour tout point $\chi \in P$: ces points peuvent être n'importe quel point (fractionnaire ou non) du polyèdre.

Si un point ne vérifie pas une inégalité, on dit que ce point **viole** l'inégalité.

Exemple du le polyèdre du stable

Sur un graphe $G = (V, E)$ non-orienté.

- Pour un sommet donné $u \in V$, les deux inégalités triviales $x(u) \geq 0$ et $x(u) \leq 1$ sont valides pour $P(G)$.

- Pour une arête donnée $uv \in E$, l'inégalité d'arête $x(u) + x(v) \leq 1$ est valide pour $P(G)$.

- Pour un cycle C donné, l'inégalité de cycle

$$\sum_{u \text{ dans } C} x(u) \leq \left\lfloor \frac{|C|}{2} \right\rfloor$$

est valide pour $P(G)$. (voir preuve dans la partie D).

47/75

48/75

Face

Definition

Soit $ax \leq \alpha$ une inégalité valide pour P .
Le sous-ensemble $F = \{x \in P \mid ax = \alpha\}$ est appelé **face** de P .
On dit aussi que F est la face définie par $ax \leq \alpha$.
Si $F \neq \emptyset$ et $F \neq P$, on dit que la face F est **propre**.

Par convention, l'ensemble vide et le polytope P lui-même sont considérés comme des faces de P .

Exemple du polyèdre du stable

Pour le polyèdre $P(G)$ associée à un graphe $G = (V, E)$ non-orienté.
Pour un sommet u donné, la face associée à l'inégalité triviale $x(u) \geq 0$ est

$$F_0(u) = \{\chi \in P(G) \mid \chi[u] = 0\}$$

Donc la face $F_0(u)$ contient tous les vecteurs d'incidence des stables ne contenant pas le sommet u .

Face

Le résultat suivant montre que, dans toute description d'une face, il existe un sous-système d'inégalités correspondant à cette face.

Theorem

Soit P un polyèdre caractérisé par les inégalités $Ax \leq b$.
Un sous-ensemble F non vide de P est une face de P si et seulement s'il existe un sous-système $A'x \leq b'$ de $Ax \leq b$ tel que $F = \{x \in P \mid A'x = b'\}$.

Preuve. (\Leftarrow) Supposons qu'il existe un sous-système $A'x \leq b'$ tel que $F = \{x \in P \mid A'x = b'\}$. Alors, par la convention d'écriture du système $Ax \leq b$, pour tout point $x \in P \setminus F$, il existe au moins une contrainte parmi les inégalités de $A'x \leq b'$ qui ne soit pas vérifiée à l'égalité par x . Considérons l'inégalité $ax \leq \alpha$ obtenue en sommant les inégalités de $A'x \leq b'$. Il est clair que $ax = \alpha$ pour tout $x \in P \setminus F$.
(\Rightarrow) Soit F une face non vide de P . On veut produire le sous-système composé des inégalités de $Ax \leq b$ telles que tout point de F vérifie ces inégalités à l'égalité. Pour cela, considérons $ax \leq \alpha$ une inégalité valide pour P telle que $F = \{x \in P \mid ax = \alpha\}$. On considère alors le programme linéaire $\max\{ax \mid x \in P\}$. Les solutions optimales de ce programme linéaire sont précisément les éléments de F . Soit (y^1, y^2) une solution optimale duale de ce programme linéaire où y_1 et y_2 sont les vecteurs duaux correspondant respectivement aux systèmes $Ax \leq b$ et $Bx = d$. Par les conditions des écarts complémentaires en programmation linéaire, le sous-système $A'x \leq b'$ de $Ax \leq b$ dont les variables duales ont une valeur positive est précisément le sous-système recherché, i.e. tel que $F = \{x \in P \mid A'x = b'\}$. \square

Donc une face est un polyèdre !

Face de dimension 0

Un point extrême d'un polyèdre peut être aussi défini comme étant une face.

Theorem

Un point $\bar{x} \in P$ est un point extrême de P si et seulement si \bar{x} est une face de dimension 0.

Preuve. Soit \bar{x} un point quelconque de P . On associe à \bar{x} le polyèdre \bar{P} obtenu à partir de P en transformant toute inéquation vérifiée à l'égalité par \bar{x} en une égalité. Notons alors $\bar{A}x \leq \bar{b}$, $\bar{B}x = \bar{d}$ le système des inéquations et des équations décrivant \bar{P} . Remarquons qu'alors $\bar{A}\bar{x} < \bar{b}$.
(\Rightarrow) Supposons par contraposée que \bar{x} n'est pas une face de dimension 0, alors $\dim(\bar{P}) > 0$ et, ainsi, par la proposition 9, $\text{rang}(\bar{B}) < n$. Par conséquent, il existe un point $y \neq 0$ tel que $\bar{B}y = 0$. Comme $\bar{A}\bar{x} < \bar{b}$, il existe un scalaire $\epsilon > 0$ tel que $\bar{A}y + \epsilon\bar{A}\bar{x} \leq \bar{b}$, et ainsi $\bar{x}^1 = \bar{x} + \epsilon y \in \bar{P} \subset P$. Aussi, ϵ peut être choisi suffisamment petit pour que $\bar{x}^2 = \bar{x} - \epsilon y \in P$. Comme ainsi $\bar{x} = \frac{1}{2}(\bar{x}^1 + \bar{x}^2)$, \bar{x} n'est pas un point extrême de P .
(\Leftarrow) Si \bar{x} est une face de dimension 0, alors \bar{P} est un polyèdre de dimension 0 et ainsi $\text{rang}(\bar{B}) = n$. Alors \bar{x} est la solution unique du système $\bar{B}x = \bar{d}$. Supposons qu'il existe \bar{x}^1 et \bar{x}^2 dans P tels que $\bar{x} = \frac{1}{2}(\bar{x}^1 + \bar{x}^2)$. Or \bar{x}^1 et \bar{x}^2 sont des solutions de $\bar{B}x = \bar{d}$. (En effet, on peut le déduire de $\bar{B}\bar{x}^1 \leq \bar{d}$, $\bar{B}\bar{x}^2 \leq \bar{d}$ et $\bar{B}\bar{x}^1 + \bar{B}\bar{x}^2 = 2\bar{d}$.) Par conséquent, $\bar{x} = \bar{x}^1 = \bar{x}^2$, et donc \bar{x} est un point extrême de P . \square

Face "minimale"

En fait, les faces peuvent être contenues les unes dans les autres. On peut tout d'abord se poser la question des faces *minimales*, c'est-à-dire celles qui ne contiennent strictement pas d'autre face.

Pour régler cette question, on a les résultats suivants.

Théorème :

Un sous-ensemble F non vide de P est une face minimale de P si et seulement s'il existe un sous-système $A'x \leq b'$ de $Ax \leq b$ tel que $F = \{x \in \mathbb{R}^n \mid A'x = b', Bx = d\}$.

Corrolaire :

Toute face minimale non vide de P est de dimension $n - \text{rang} \begin{pmatrix} A \\ B \end{pmatrix}$.

Les faces minimales non vides d'un polytope sont des points extrêmes.

La notion de face minimale se ramène donc en fait à la recherche des points extrêmes. Ce n'est pas étonnant, ce sont les faces qui sont nécessaires à une formulation "minimale" du problème, cela revient en fait à énumérer tous les points extrêmes, or cela est très coûteux et cela abandonne l'outil de la Programmation Linéaire.

Cherchons plutôt quelles sont les inégalités essentielles à la formulation.

1. Introduction

2. Définitions et résultats fondamentaux

2.1 Point intérieur et dimension

2.2 Face d'un polyèdre

2.3 Facettes

2.4 Etude faciale sur $conv(S)$

2.5 Description minimale

3. Approches polyédrales

Facette

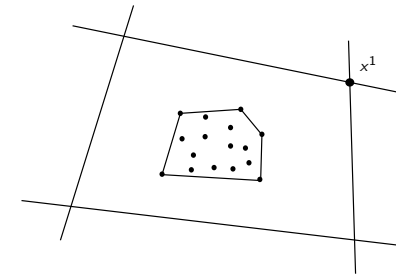
Definition

Une face F de P de dimension $dim(F) = dim(P) - 1$ est appelée *facette* de P .

Géométriquement, une facette est un des hyperplans d'appui qui définissent le polyèdre.

Dans le schéma suivant, l'inégalité qui permet de mieux couper le point x^1 (qui n'est pas solution) est d'utiliser les inégalités qui définissent des facettes du polytope des solutions.

$$(P_2) \begin{cases} \text{Max } cx \\ Ax \leq B \\ \alpha x \leq \beta \\ x \text{ entier} \end{cases}$$



Facette

Theorem

Si F est une face propre de P , alors $dim(F) \leq dim(P) - 1$.

Corollary

Si F est propre et si $dim(F) \geq dim(P) - 1$ alors F est donc une facette de P .

Theorem (Maximalité d'une facette)

Soit F une face propre qui n'est pas strictement contenue dans une autre face propre. Alors F est une facette.

On peut dire qu'une facette est maximale "au sens de l'inclusion".

1. Introduction

2. Définitions et résultats fondamentaux

2.1 Point intérieur et dimension

2.2 Face d'un polyèdre

2.3 Facettes

2.4 Etude faciale sur $conv(S)$

2.5 Description minimale

3. Approches polyédrales

Etudier $conv(S)$ uniquement avec des points de S

On peut prouver la validité d'une inégalité et son caractère définissant des facettes en **utilisant seulement des points de S** .

On a en effet les deux lemmes suivants.

Lemma

Une inégalité $ax \leq \alpha$ est valide pour S si et seulement si elle est valide pour $conv(S)$.

Lemma

Si F est une face non vide de $conv(S)$ de dimension $p - 1$, alors il existe p points affinement indépendants dans $S \cap F$.

Ces deux lemmes nous indiquent que l'on peut démontrer si une inégalité est valide et si elle définit une facette d'un polyèdre des solutions en s'intéressant uniquement aux solutions du problème.

57/75

1. Introduction

2. Définitions et résultats fondamentaux

2.1 Point intérieur et dimension

2.2 Face d'un polyèdre

2.3 Facettes

2.4 Etude faciale sur $conv(S)$

2.5 Description minimale

3. Approches polyédrales

59/75

Facettes du polytope du stable

On considère le polytope des stables $P(G)$ pour un graphe non-orienté $G = (V, E)$.
Considérons un sommet donné $v \in V$, alors l'inégalité triviale

$$x(v) \geq 0$$

est valide pour $P(G)$.

Montrons que cette inégalité triviale définit une facette de $P(G)$.

Réponse :

Posons $F_0(v) = \{\chi^S \in \mathbb{R}^n \mid S \text{ stable et } \chi^S(v) = 0\}$ la face associée à la contrainte $x(v) \geq 0$.

Pour prouver que cette face est une facette, il faut montrer qu'elle est propre et qu'elle contient n vecteurs affinement indépendants.

$F_0(v)$ est bien propre :

- car elle n'est pas vide : par exemple l'ensemble vide est un stable et son vecteur (nul) d'incidence appartient à $F_0(v)$

- et $F_0(v) \neq P(G)$: en effet, le vecteur d'incidence $\chi^{\{v\}}$ du stable singleton $\{v\}$ appartient à $P(G)$ mais pas à $F_0(v)$.

De plus, les vecteurs χ^\emptyset et $\chi^{\{u\}}$, $u \in V \setminus \{v\}$, sont bien n vecteurs d'incidence de stables affinement indépendants appartenant à $F_0(v)$. C'est bien une facette de $P(G)$.

58/75

Redondance dans une formulation linéaire

Dans la suite, nous allons voir que les inégalités qui sont nécessaires à la description de P sont celles qui définissent des faces maximales.

Définition

Une inégalité est dite **redondante** dans un système $Ax \leq b$ définissant un polyèdre P , si le sous-système, obtenu à partir de $Ax \leq b$ en supprimant cette inégalité, définit le même polyèdre P .

Inversement, une inégalité non-redondante est dite **essentielle**.

60/75

Facette et redondance

Theorem

Supposons que $P \neq \emptyset$. Alors toute contrainte valide pour P qui ne définit pas de facette de P est redondante.

Preuve : Soit $A_i x \leq b_i$ une contrainte valide essentielle à la description de P . On veut montrer que $A_j x \leq b_j$ définit une facette de P . Pour cela, remarquons que, comme cette inégalité est nécessaire dans P , alors il doit exister un point $x^* \in R^n \setminus P$ tel que

$$\begin{aligned} A_j x^* &\leq b_j, \forall j \in \{1, \dots, m_1\} \setminus \{i\}, \\ A_i x^* &> b_i, \\ Bx^* &= d. \end{aligned}$$

Puisque P est non vide, il contient un point intérieur, disons \bar{x} , et donc $A_i \bar{x} < b_i$. Soit z un point sur le segment entre x^* et \bar{x} tel que $A_i z = b_i$, i.e., il existe $0 < \lambda < 1$ avec $z = \lambda x^* + (1 - \lambda)\bar{x}$. Ainsi

$$\begin{aligned} A_j z &< b_j, \forall j \in \{1, \dots, m_1\} \setminus \{i\}, \\ A_i z &= b_i, \\ Bz &= d. \end{aligned}$$

Ceci implique que z appartient à la face de P définie par $A_i x \leq b_i$. Aussi notons que le système donné par les équations de F est $A_i x = b_i, Bx = d$ avec $A_i x = b_i$ linéairement indépendante des lignes de B (sinon, l'inégalité $A_i x \leq b_i$ ne serait pas essentielle). Par conséquent le rang de ce système est $\text{rang}(B) + 1$. Donc $\dim(F) = n - (\text{rang}(B) + 1)$ et comme $\dim(P) = n - \text{rang}(B)$, $\dim(F) = \dim(P) - 1$. $A_i x \leq b_i$ est bien une facette de P . □

61/75

Description minimale du polyèdre par des inégalités

Pour pouvoir prouver les résultats suivants, nous avons besoin des 2 lemmes de Farkas.

Lemma

(Lemme de Farkas pour les inéquations). Etant donné une matrice $m \times n$ A et un vecteur $b \in R^m$, le système $Ax \leq b$ admet une solution si et seulement s'il n'existe pas un vecteur $y \geq 0$ de R^m tel que $yA = 0$ et $yb < 0$.

Lemma

(Lemme de Farkas) Le système $Ax = b$ admet une solution (resp. solution positive) si et seulement s'il n'existe pas un vecteur y tel que $yA = 0$ et $yb < 0$ (resp. $yA \geq 0$ et $yb < 0$).

62/75

Description minimale du polyèdre par des inégalités

Le résultat suivant montre qu'une facette d'un polyèdre P est au moins définie par une contrainte valide pour P .

Theorem

Pour toute facette F de P , une des inéquations définissant F est nécessaire dans la description de P .

Preuve : Soit $I \subset \{1, \dots, m_1\}$ l'ensemble des indices des inégalités $A_i x \leq b_i$ de $Ax \leq b$ qui définissent F . Soit \bar{P} le polyèdre définie par ces inégalités. Nous allons montrer que $P \setminus \bar{P}$ n'est pas vide, ce qui montre qu'au moins une des inégalités parmi $A_i x \leq b_i, i \in I$, doit apparaître dans la description de P . Par la suite, nous considérons une contrainte $A_i x \leq b_i$ pour un $i \in I$.

Comme $F \neq \emptyset$, prenons x^0 un point intérieur de F . De plus, comme $F \neq P$, A_i est indépendant des lignes de B , c'est-à-dire que le système $yB = A_i$ n'a pas de solution. Par le lemme de Farkas, le système $\{Bx = 0, A_i x > 0\}$ admet une solution, disons x^1 . Comme $A_k x^0 < b_k$ pour tout $k \in \{1, \dots, m_1\} \setminus I$, il existe $\epsilon > 0$ tel que

$$A_k x^0 + \epsilon A_k x^1 \leq b_k \text{ pour } k \in \{1, \dots, m_1\} \setminus I.$$

Notons $x^2 = x^0 + \epsilon x^1$. D'abord, remarquons que $Bx^2 = 0$. Alors x^2 est un point du segment reliant x^0 à x^1 qui est dans \bar{P} . En effet, $A_k x^2 \leq b_k$, pour $k \in \{1, \dots, m_1\}$. De plus, comme $A_i x^1 > 0$ et $A_i x^0 = b_i$, on a $A_i x^2 = A_i x^0 + \epsilon A_i x^1 > b_i$. Par conséquent, $x^2 \notin \bar{P}$ donc $x^2 \in P \setminus \bar{P}$. □

63/75

Inégalités équivalentes

En fait une seule inégalité est nécessaire par facette dans une description de P .

Definition

Deux inégalités valides $a_1 x \leq \alpha_1$ et $a_2 x \leq \alpha_2$ sont **équivalentes** s'il existe $\lambda > 0$ et un vecteur μ de R_2^m tel que $a_2 = \lambda a_1 + \mu B$ et $\alpha_2 = \lambda \alpha_1 + \mu d$. (Dans les notations de l'écriture de P).

On a alors le résultat suivant.

Theorem

Supposons que $P \neq \emptyset$. Si deux inégalités valides de P définissent la même facette, alors elles sont équivalentes.

D'après cette proposition, pour toute facette F de P , une et une seule inégalité définissant F est nécessaire dans un système décrivant P .

64/75

Inégalités équivalentes

Comme conséquence des propositions précédentes, on a le théorème suivant.

Theorem

Le système définissant P est minimal (en nombre d'inéquations et d'équations) si et seulement si les lignes de B sont linéairement indépendantes et si toute inégalité $A_i x \leq b_i, i = 1, \dots, m_1$ définit une facette distincte de P .

La description d'un polyèdre de pleine dimension est même unique à une multiplication d'un scalaire près.

Corollary

Si P est un polyèdre de pleine dimension, alors il existe un système linéaire minimal unique (à des multiplications par des scalaires près) qui décrit P . De plus, toute contrainte de ce système définit une facette distincte de P .

65/75

Facettes du polytope du stable

On recherche à présent n points affinement indépendants dans F_K .
Considérons les $|K|$ singletons $\{u\}, u \in K$ qui sont $|K|$ stables dont les vecteurs d'incidence sont contenus dans F_K .

Il manque donc $n - |K|$ vecteurs à exhiber.

Or comme K est un sous-graphe complet maximal dans G , cela signifie que pour tout sommet $w \in V \setminus K$, il existe un sommet u_w dans K tel que w et u_w ne sont pas adjacents : en effet, dans le cas contraire, K ne serait pas maximale. Donc la paire $\{w, u_w\}$ est un stable et elle est dans F_K car u_w est dans K . On a ainsi $n - |K|$ stables.

On a au total n stables $\{u\}, u \in K$ et les $\{w, u_w\}, w \in V \setminus K$.

67/75

Facettes du polytope du stable

Soit K une clique de G (sous-graphe complet de G). Montrer que la contrainte

$$\sum_{v \in K} x(v) \leq 1$$

est bien valide pour $P(G)$ car un stable ne peut pas contenir plus d'un sommet d'un graphe complet.

Montrons qu'elle définit une facette de $P(G)$ lorsque K est une clique maximale au sens de l'inclusion : c'est-à-dire que K n'est pas contenue dans une clique plus grande.

Réponse :

La face $F_K = \{\chi^S \in \mathbb{R}^n \mid S \text{ stable et } \chi^S(K) = 1\}$ c'est-à-dire qu'elle contient tous les stables possédant exactement un sommet de K .

F_K est propre :

- car elle n'est pas vide : elle contient $\chi^{\{u\}}$ pour tout $u \in K$
- et elle est différente de $P(G)$: elle ne contient pas χ^{\emptyset} .

66/75

Facettes du polytope du stable

On peut prouver que leurs vecteurs d'incidence sont n vecteurs affinement indépendants dans F_K : en effet ces n vecteurs sont déjà linéairement indépendants !
Pour cela regardons la matrice dont les lignes sont les vecteurs et dont les colonnes sont rangées en mettant d'abord les sommets de K puis ceux de $V \setminus K$: la matrice est de la forme :

$$\begin{pmatrix} I_{|K|} & O \\ M & I_{n-|K|} \end{pmatrix}$$

Où I_p est la matrice identité de côté p , O la matrice nulle et M est une matrice quelconque (en fait avec un seul 1 par ligne). Donc cette matrice est triangulaire de diagonale remplie de 1 : elle est de plein rang.

L'inégalité de clique définit bien une facette de $P(G)$ si K est maximal au sens de l'inclusion. □

68/75

Facettes du polytope du stable

Soit uv une arête de E .

Regardons à présent l'inégalité d'arêtes

$$x(u) + x(v) \leq 1$$

qui est bien valide pour $P(G)$.

En fait, une arête est une clique de taille 2!

Donc cette inégalité est une inégalité de clique!

Par conséquent, une inégalité d'arête définit une facette si l'arête est une clique maximale dans G : c'est-à-dire s'il n'existe pas de sommet w relié à la fois à u et à v .

Facettes du polytope du stable

Inversement, est-ce que cette inégalité d'arête définit une facette s'il existe un tel sommet w relié à u et à v ?

Réponse :

Alors l'inégalité de clique

$$x(u) + x(v) + x(w) \leq 1$$

est valide pour $P(G)$.

Si on l'additionne avec l'inégalité valide $-x(w) \leq 0$, on obtient alors l'inégalité d'arête $x(u) + x(v) \leq 1$.

Cela veut dire que l'inégalité d'arête n'est pas essentielle dans la formulation : **elle est redondante... donc elle ne définit pas une facette.**

En conclusion si une arête est contenue dans une clique plus grande, alors sa face associée n'est pas une facette.

De plus, **elle peut être supprimée de la formulation** (si la contrainte de clique qui la contient est maintenue!).

Notez pourtant que les contraintes d'arêtes sont dans la formulation entière du stable d'origine...

Formulation entière

On désire résoudre un problème d'optimisation combinatoire

$$\mathcal{P} \quad \max \{cx \mid x \in \mathcal{S}\}$$

par l'approche polyédrale.

En pratique, nous avons vu dans la première section qu'un problème d'optimisation combinatoire \mathcal{P} peut souvent s'écrire comme un programme linéaire en nombres entiers de la forme

$$(P_1) \quad \max \{cx \mid Ax \leq b, 0 \leq x \leq 1, x \text{ entier}\}$$

où $Ax \leq b$ est un ensemble d'inégalités linéaires qui peut être de taille exponentielle par rapport au nombre n de variables.

Cette formulation PLNE (P_1) définit donc un ensemble de solutions entières S qui sont très exactement les solutions du problème \mathcal{P} .

1. Introduction

2. Définitions et résultats fondamentaux

3. Approches polyédrales

Renforcement et points extrêmes fractionnaires

Si on relaxe la contrainte d'intégrité dans (P_1) , on obtient la *relaxation linéaire* (P_1^*) de la formulation entière.

Elle peut parfois fournir une solution entière. C'est le cas par exemple des matrices de flot si les capacités sont entières ou des matrices correspondants au problème d'affectation. Plus généralement, c'est le dans certains types de matrices : matrices TDI (Totally Dual Integral), matrices TU (Totally Unimodular),...

Mais ce n'est pas le cas en général. Cela signifie que le polyèdre défini par (P_1^*) possède des **points extrêmes fractionnaires**. Bien entendu, il peut exister des instances dont la solution optimale correspond par chance à un point entier parmi un ensemble de points qui seraient majoritairement fractionnaires, mais ce n'est pas le cas général. On peut donc dire que cette relaxation linéaire n'est pas une description suffisante pour toute instanciation.

L'approche polyédrale consiste à **renforcer** cette formulation (P_1) pour que la relaxation linéaire propose un point extrême entier. Plus exactement, on cherche à ajouter suffisamment d'inégalités de l'enveloppe convexe dans le but d'avoir produit celles dont l'intersection des faces va donner un point extrême entier optimal, c'est-à-dire une solution optimale du problème.

73/75

Approche polyédrale

L'**approche polyédrale**, introduite par Edmonds en 1965, dans le cadre du problème du couplage, consiste à étudier le polytope $conv(S)$ afin de pouvoir résoudre \mathcal{P} comme un programme linéaire.

La caractérisation complète du polytope $conv(S)$ est généralement difficile à obtenir.

Par ailleurs, une description complète du polyèdre peut comporter un nombre exponentiel d'inégalités. Cependant, un nombre réduit de ces inégalités peut être suffisant pour résoudre le problème à l'aide d'une *méthode de coupes* ou de *coupes et branchement* (Branch&Cut method) (voir section correspondante).

75/75

Polyèdre combinatoire

Considérons l'enveloppe convexe $conv(S)$ des solutions S de \mathcal{P} : on l'appelle **polytope des solutions** ou encore **polyèdre combinatoire**.

Le problème \mathcal{P} est alors équivalent au programme linéaire

$$\max \{cx \mid x \in conv(S)\}.$$

Par conséquent, si nous caractérisons le polyèdre $conv(S)$ par un système d'inégalités linéaires, alors nous ramenons le problème \mathcal{P} à la résolution d'un programme linéaire ! Si l'on ne connaît pas toutes les inégalités définissant des facettes de $conv(S)$, une connaissance partielle de ces inégalités permet de renforcer la formulation.

74/75

Master d'Informatique - Spécialité Androide

Module MAOA

Recherche Opérationnelle et Optimisation Combinatoire

Partie E - Caractérisation

Pierre Fouilhoux

Sorbonne Université

2019-2020

1. Définitions

2. Outils de caractérisation

3. Le polytope du couplage

2/41

Définitions

1. Définitions

2. Outils de caractérisation

3. Le polytope du couplage

On appelle **caractérisation** (ou caractérisation complète) d'un polyèdre le fait de décrire l'ensemble des inégalités qui composent ce polyèdre.
Ainsi le PL composé par ces inégalités est entier.

Considérons un Problème d'Optimisation Combinatoire (\mathcal{P}).

L'expression *Formulation entière* pour (\mathcal{P}) est "dangereuse" :

- soit elle désigne un Programme linéaire en Nombres Entiers (PLNE) modélisant (\mathcal{P}) : ce qui est a priori NP-difficile à résoudre.
- soit elle désigne un Programme Linéaire dont tous les points extrêmes sont entiers, c'est-à-dire dont une caractérisation du polytope des solutions de (\mathcal{P}) : ce qui est a potentiellement polynomial.

On privilégiera ici l'expression "**formulation caractérisant le problème**" pour éviter cette ambiguïté.

Caractérisation et Algorithme

Considérons un Problème d'Optimisation Combinatoire (\mathcal{P}) pour lequel on connaît sa caractérisation par un ensemble d'inégalités $Ax \leq b$.

Résoudre (\mathcal{P}) revient à résoudre le PL composé par les inégalités $Ax \leq b$. Mais il peut y avoir **un nombre exponentiel** d'inégalités dans $Ax \leq b$!

En fait (\mathcal{P}) est polynomial :

- si on connaît sa caractérisation $Ax \leq b$

ET

- soit on sait énumérer polynomialement tout $Ax \leq b$
- soit on sait séparer en temps polynomial toutes les inégalités de $Ax \leq b$

5/41

Caractérisation partielle

Remarque : Il a été montré que si un problème d'Optimisation Combinatoire est NP-difficile, à moins que $P=NP$, on ne peut pas "décrire" la caractérisation de (\mathcal{P}). (Cette preuve définit ce mot "décrire" de manière complexe).

Notez que le nombre d'inégalités du polytope du voyageur de commerce pour n villes a été estimé à 2^{2^n} inégalités par Groeschel.

On appelle "abusivement" **caractérisation partielle** le fait de décrire une partie des facettes d'un polyèdre.

Une caractérisation partielle, couplées à un algorithme de branchement, est la base de l'efficacité des algorithmes de **Branch-and-cut** permettant de résoudre les formulations PLNE.

On devrait uniquement appeler **caractérisation partielle** le fait de caractériser complètement le **polytope associé à un sous-problème** du problème d'optimisation combinatoire considéré.

Par exemple, on connaît la caractérisation du polytope du stable pour les graphes bipartis, les graphes planaires, les graphes d'intervalles,... mais certainement pas pour un graphe quelconque.

7/41

Caractérisation minimale

Etant donné un polyèdre P .

On a vu que :

- pour toute facette F de P , une des inéquations définissant F est nécessaire dans une caractérisation de P .
- Il existe une écriture unique d'un système linéaire à des multiplications par des scalaires près qui décrit les facettes de P (en fait c'est le cas si P est de pleine dimension).

On appelle **caractérisation minimale** un ensemble d'inégalités $Ax \leq b$ en nombre minimal caractérisant (P).

Il s'agit alors d'un système où chaque inégalité correspond à exactement une facette de (P).

On peut tout à fait obtenir des caractérisations non minimales : c'est-à-dire des caractérisations où il y a "trop" d'inégalités : c'est souvent le cas des systèmes qu'on a prouvé être entier par aspect TDI par exemple.

6/41

1. Définitions

2. Outils de caractérisation

3. Le polytope du couplage

8/41

Plusieurs outils de caractérisation

Considérons un problème d'optimisation combinatoire (\mathcal{P}).
 Considérons un ensemble d'inégalités $Ax \leq b$ qui sont valides pour (\mathcal{P}) et telles que tout point entier les vérifiant est une solution de (\mathcal{P})., c'est-à-dire que $(Ax \leq b, x \text{ entier})$ forme un PLNE modélisant (\mathcal{P}).

Il existe plusieurs techniques pour prouver que $Ax \leq b$ caractérise le polyèdre des solutions de (\mathcal{P}) :

- montrer que A est TU (voir chapitre correspondant)
- montrer que le système basé sur $Ax \leq b$ est TDI (voir chapitre correspondant)
- montrer qu'il n'y a pas de points extrêmes dans le polyèdre $Ax \leq b$ qui soit fractionnaire
- énumérer toutes les facettes du polyèdre et vérifier qu'il y a une inégalité pour chaque facette dans $Ax \leq b$
- et quelques autres : par lifting d'inégalités valides, décomposition des instances et recombinaison polyédrales, utilisation de formulations étendues (avec ou sans projection dans l'espace initial, étude des points extrêmes critiques,...

9/41

Caractériser "par absence de points fractionnaires"

Exemple du problème du couplage biparti

Soit un graphe biparti complet $G = (V_1 \cup V_2, E)$ associé à un poids $c \in \mathbb{N}^m$ associé aux arêtes de E .

On appelle **couplage** de G un ensemble d'arêtes deux à deux non incidentes. Le **problème du couplage biparti** consiste à déterminer un couplage qui maximise la somme des poids de ses arêtes.

Considérons le PL suivant

$$\begin{aligned} \text{Max} \quad & \sum_{e \in E} c(e)x(e) \\ & \sum_{e \in \delta(u)} x(e) \leq 1 \quad \forall u \in V_1 \\ & \sum_{e \in \delta(u)} x(e) \leq 1 \quad \forall u \in V_2 \\ & x(e) \geq 0 \quad \forall e \in E. \end{aligned}$$

On a déjà montré que ce PL est TU.
 Cela signifie que les inégalités de ce PL sont la caractérisation (complète) du problème du couplage dans les graphes bipartis.

11/41

Cas particulier minimization : le dominant

Remarque : Dans le cas d'un problème (\mathcal{P}) de direction Minimisation, on peut noter que l'on peut seulement regarder le polytope des solutions dans la direction "minimisation" : seules les points extrêmes de ce "côté" du polytope peuvent être solution.

Si l'on nomme P le polytope des solutions de (\mathcal{P}), on appelle **dominant** de P le polyèdre $P^+ = P \cup \mathbb{R}^+$: à chaque point du polytope P , on "ajoute" toutes les valeurs d'un cadran de repère.

Dans ce polyèdre, tous les points extrêmes de P potentiellement optimaux dans la direction minimisation sont conservés !
 Minimiser sur P ou sur P^+ donnera les mêmes solutions.

Exemple :

Le polytope des coupes est inconnu : c'est normal le problème Max-Cut est NP-difficile.

En revanche, on sait résoudre Min-Cut en temps polynomial...

et on connaît le polyèdre défini par le dominant du polytope des coupes !

10/41

Caractériser "par absence de points fractionnaires"

Exemple du problème du couplage biparti

Montrons à nouveau cette caractérisation mais en utilisant une preuve du type "par absence de points fractionnaires".

Soit P le polytope associée à la formulation précédente, c'est-à-dire le polytope définie par les inégalités de cette formulation.

Soit x^* un point extrême de P

Posons $E_f = \{e \in E \mid 0 < x^*(e) < 1\}$ et supposons que E_f est non vide (on veut obtenir une contradiction).

Cas 1 : E_f contient un cycle.

On va montrer que x^* ne peut pas être un point extrême, c'est-à-dire que nous allons construire deux points y et z de P tels que $x^* = \frac{1}{2}(y + z)$.

Considérons un cycle C dans E_f .

En fait un graphe est biparti si et seulement si tous ses cycles sont de longueurs paires.

Donc comme G est biparti, C est de longueur paire.

12/41

Caractériser “par absence de points fractionnaires”

Exemple du problème du couplage biparti

Notons alors e_1, e_2, \dots, e_{2k} les arêtes de C . On pose alors

$$y(e) = \begin{cases} x^* + \epsilon & \text{si } e \in \{e_1, e_3, e_5, \dots, e_{2k-1}\} \\ x^* - \epsilon & \text{si } e \in \{e_2, e_4, e_6, \dots, e_{2k}\} \\ x^* & \text{otherwise} \end{cases} \quad z(e) = \begin{cases} x^* - \epsilon & \text{si } e \in \{e_1, e_3, e_5, \dots, e_{2k-1}\} \\ x^* + \epsilon & \text{si } e \in \{e_2, e_4, e_6, \dots, e_{2k}\} \\ x^* & \text{otherwise} \end{cases}$$

En prenant $\epsilon = \min\{x^*(e) \mid e \in E_f\}$, on voit que y et z sont positifs. Donc y et z satisfont les inégalités triviales. De plus, $\epsilon > 0$ et y et z sont bien distincts et distincts de x .

De plus, on montre que y et z satisfont bien les autres inégalités :
 Pour $u \in V_1$, prouvons que y satisfait $\sum_{e \in \delta(u)} x(e) \leq 1$.
 C'est trivialement vrai si u n'est pas dans le cycle C .
 Si u est dans C , il y a deux arête e_i et e_{i+1} de C incidentes à u : on peut voir qu'alors on a bien $\sum_{e \in \delta(u)} y(e) = \sum_{e \in \delta(u)} y(e) + \epsilon - \epsilon = \sum_{e \in \delta(u)} x^*(e) \leq 1$.
 Par symétrie, on voit que y et z satisfont bien toutes les inégalités de la formulation.

Enfin, par construction, $x^* = \frac{1}{2}(y + z)$ donc x n'est pas un point extrême, une contradiction.

13/41

Caractériser “par absence de points fractionnaires”

Exemple du problème du couplage biparti

Cas 2 : E_f ne contient pas de cycle.

Dans ce cas, considérons le plus long chemin (en nombre d'arêtes) $\mu = (e_1, e_2, \dots, e_k)$ de E_f .

On pose à présent $\epsilon = \min\{x^*(e), 1 - x^*(e) \mid e \in E_f\}$. On a bien $\epsilon > 0$ et pose alors les points y et z définis comme dans le Cas 1 qui sont distincts et distincts de x . Montrons que y et z vérifient bien les inégalités de la formulation. De la même façon que dans le Cas 1, les inégalités triviales sont vérifiées ainsi que les inégalités de degré pour les sommets en dehors μ et pour les sommets intérieurs de μ .

On prouve également que les inégalités de degré sont vérifiées par y et z pour les extrémités de μ . En effet, notons u le premier sommet du chemin μ , incident à e_1 . z vérifie clairement l'inégalité de degré pour u . Remarquons à présent que u ne peut pas être incident à une autre arête de E_f car sinon μ ne serait pas un plus long chemin. De plus, comme $x^*(e_1) > 0$, u ne peut donc être incident à une arête f telle que $x^*(f) = 1$ donc u n'est incident qu'à des arête f telles que $x^*(f) = 0$. Donc y vérifie aussi l'inégalité de degré pour u (cas symétrique pour l'autre extrémité).

Similairement au Cas1, on peut conclure que cette formulation ne contient aucun point fractionnaire : elle est donc bien une caractérisation du problème du couplage dans les graphes bipartis. □

14/41

Caractériser par preuve algébrique

Une autre technique de preuve, celle-ci très générique, est de prouver que toute inégalité définissant une facette est du type de l'une des contraintes de la formulation. Cette technique de preuve est à l'origine des approches polyédrales, mise au point pour le polytope du couplage (dans un graphe quelconque) par J. Edmonds [1965].

Soit (\mathcal{P}) un problème combinatoire consistant à rechercher une solution optimale dans un ensemble $S \subset 2^E$ de solutions où E est un ensemble fini de taille n et valué par c , i.e. $(\mathcal{P}) : \max\{c(S) \mid S \in \mathcal{S}\}$.

On pose alors χ^S le vecteur d'incidence de S tel que $\chi^S(e) = 1$ si $e \in S$ et $\chi^S(e) = 0$ si $e \in E \setminus S$.

Le polytope $P(\mathcal{S})$ associé à ce problème se définit comme l'enveloppe convexe des vecteurs d'incidence des éléments de \mathcal{S} , i .

$$P(\mathcal{S}) = \text{conv}\{\chi^S \in \{0, 1\}^n \mid S \in \mathcal{S}\}.$$

15/41

Caractériser par preuve algébrique

Une conséquence du fait qu'une facette est une face maximale au sens de l'inclusion est le Lemme suivant.

Lemme

Soit $ax \leq \alpha$ une contrainte définissant une facette F de $P(\mathcal{S})$.
 Soit $a'x \leq \alpha'$ une contrainte valide de $P(\mathcal{S})$ et soit F' la face associée à $a'x \leq \alpha'$.
 Alors si $F \subset F'$ alors $F = F'$.

Ce lemme est une des clés de certaines preuves du fait qu'une face d'une contrainte est une facette.

16/41

Caractériser par preuve algébrique

Le lemme suivant, en revanche, est spécifique aux polytopes de pleine dimension. On utilise ce lemme sous les deux formes proposées i) et ii) qui ne sont en fait que la contraposée l'une de l'autre.

Lemma

Supposons que $P(S)$ soit de pleine dimension.
 Soit $ax \leq \alpha$ une contrainte définissant une facette de $P(S)$.
 Posons S_a l'ensemble des solutions de S dont le vecteur d'incidence associé vérifie l'inégalité $ax \leq \alpha$ à l'égalité, i.e.

$$S_a = \{S \subset S \mid \text{et } a\chi^S = \alpha\}.$$

Soit $a'x \leq \alpha'$ une contrainte valide de $P(S)$.
 i) Si pour toute solution $S \in S_a$, $a'x^S = \alpha'$, alors $ax \leq \alpha$ et $a'x \leq \alpha'$ sont identiques à un coefficient multiplicateur positif prêt.
 ii) Si $ax \leq \alpha$ et $a'x \leq \alpha'$ ne sont pas identiques à un coefficient multiplicateur positif prêt, alors il existe $S \in S_a$ tel que $a'x^S < \alpha'$.

Le petit ii) de ce théorème est fréquemment la clef des preuves de caractérisation.

1. Définitions

2. Outils de caractérisation

3. Le polytope du couplage

3.1 Etude faciale

3.2 Caractérisation de $P_M(G)$

17/41

18/41

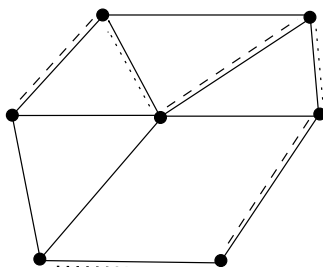
Le problème du couplage (dans un graphe quelconque)

Soit $G = (V, E)$ un graphe non orienté quelconque.

Un **couplage** (*matching*, en anglais) est un sous-ensemble d'arêtes deux à deux non-adjacentes.

Si chaque arête e de G est munie d'un certain poids $c(e)$, le **problème du couplage maximum** dans G consiste à déterminer un couplage dont le poids total des arêtes est maximum, i.e. $\sum_{e \in E} c(e)$ maximum.

Dans le graphe suivant, les arêtes en pointillé propose un couplage de 3 arêtes et les arêtes en tiret un autre couplage de 3 arêtes.



----- 2 couplages

Le problème du couplage (dans un graphe quelconque)

Donnons tout d'abord une formulation du problème comme un programme linéaire en variables binaires.

Soit $\chi(e)$, $e \in E$, des variables binaires associées aux arêtes de G telle que

$$\chi(e) = \begin{cases} 1 & \text{si } e \text{ prise dans le couplage} \\ 0 & \text{sinon} \end{cases}$$

Comme dans un couplage, il y a au plus une arête incidente à chaque sommet, le problème du couplage maximum est équivalent au PLNE \mathcal{P} suivant.

$$\begin{aligned} \text{Max } & \sum_{e \in E} c(e)x(e) \\ & \sum_{e \in \delta(u)} x(e) \leq 1, \quad \forall u \in V, & (1) \\ & 0 \leq x(e) \leq 1, \quad \forall e \in E, & (2) \\ & x(e) \text{ entier}, \quad \forall e \in E. \end{aligned}$$

Notons \mathcal{P}^* le système d'inégalités (1)-(2) formées par les inégalités de degré et triviales ci-dessus.

19/41

20/41

Le problème du couplage (dans un graphe quelconque)

Considérons le polyèdre associé au problème du couplage maximum. On note $P_M(G)$ l'enveloppe convexe des vecteurs d'incidence des couplages de G , i.e.

$$P_M(G) = \text{conv}\{\chi^M \in \mathbb{R}^n \mid M \text{ couplage de } G\}.$$

On a vu (par deux techniques de preuves) que les inégalités (1)-(2) de la formulation précédente sont suffisantes pour caractériser le polytope $P(G)$ lorsque G est biparti, c'est-à-dire lorsqu'il ne contient pas de cycle impair.

Qu'en est-il pour un graphe quelconque ?

Le problème du couplage (dans un graphe quelconque)

On veut donc "couper" ce point extrême fractionnaire pour "tendre" vers une caractérisation de $P_M(G)$.

Pour cela, on peut remarquer qu'un couplage dans ce cycle contient au plus deux arêtes. Par conséquent, l'inégalité $\sum_{i=1}^5 x(e_i) \leq 2$ est valide pour le polyèdre $P_M(G)$ du petit exemple. De plus, son ajout dans la description coupe ce point fractionnaire indésirable.

Tentons de généraliser cette contrainte.

En fait si on considère un sous-ensemble de sommets $S \subseteq V$ avec $|S|$ impair. Alors il est clair qu'un couplage de G ne peut pas contenir plus de $\frac{|S|-1}{2}$ arêtes de $E(S)$. Par conséquent, considérons l'inégalité

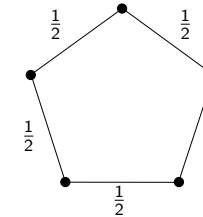
$$\sum_{e \in E(S)} x(e) \leq \frac{|S|-1}{2}, \text{ pour tout } S \subseteq V \text{ avec } |S| \text{ impair} \quad (3)$$

qui est valide pour $P_M(G)$.

On peut remarquer que cette contrainte généralise bien la contrainte basée sur le cycle du petit exemple.

Le problème du couplage (dans un graphe quelconque)

Malheureusement, les inégalités \mathcal{P}^* (1)-(2) ne caractérisent pas le polytope $P_M(G)$. En effet, \mathcal{P}^* contient des points extrêmes fractionnaires dans le cas général. Par exemple, considérons l'exemple suivant.



Soient e_1, \dots, e_5 les 5 arêtes de ce graphe. Les valeurs fractionnaires représentent le vecteur $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ où toutes les composantes correspondent à une arête. Il est facile de voir que ce vecteur vérifie toutes les contraintes de \mathcal{P}^* . De plus, il vérifie à l'égalité les 5 contraintes de type (1) associée à chacun des 5 sommets. Comme de plus, ces contraintes sont linéairement indépendantes, ce point est bien un point extrême du domaine de définition de \mathcal{P}^* . Comme ce point est fractionnaire, il ne peut appartenir à $P_M(G)$. (Remarquons que si le poids était $c(e_i) = 1, i = 1, \dots, 5$, ce point serait même la solution maximale de \mathcal{P}^*).

Le problème du couplage (dans un graphe quelconque)

On obtient alors l'ensemble d'inégalités (1)-(3)

$$\begin{aligned} \sum_{e \in \delta(u)} x(e) &\leq 1 && \forall u \in V, \\ \sum_{e \in E(S)} x(e) &\leq \frac{|S|-1}{2} && \text{ pour tout } S \subseteq V \text{ avec } |S| \text{ impair} \\ 0 &\leq x(e) \leq 1 && \forall e \in E, \end{aligned}$$

Mais comment savoir si ces contraintes citées sont suffisantes pour décrire tout $P_M(G)$?

Etude faciale de $P_M(G)$

1. Définitions

2. Outils de caractérisation

3. Le polytope du couplage

3.1 Etude faciale

3.2 Caractérisation de $P_M(G)$

Nous allons approfondir l'étude faciale de $P_M(G)$.

Tout d'abord, étudions la dimension du polyèdre $P_M(G)$.

Theorem

$P_M(G)$ est de pleine dimension, i.e. $\dim(P_M(G)) = m$.

Preuve : On veut déterminer $m + 1$ vecteurs d'incidence de solutions affinement indépendants dans $P_M(G)$.

L'ensemble vide \emptyset est un couplage de G . Donc le vecteur d'incidence de \emptyset $\chi^\emptyset = (0, \dots, 0) \in P_M(G)$.

Les singletons arêtes $\{e\}$ de G . Alors $\{e\}$, $e \in E$, est un couplage de G . De plus, la matrice formé par les vecteurs-lignes $\chi^{\{e\}}$, $e \in E$, est la matrice identité. Donc les vecteurs $\chi^{\{e\}} - \chi^\emptyset$, $e \in E$, sont bien linéairement indépendants et les $m + 1$ points cités sont affinement indépendants.

Par conséquent, $P_M(G)$ est de pleine dimension. □

25/41

26/41

Etude faciale de $P_M(G)$

Nous connaissons trois familles, (2), (1) et (3), de contraintes valides pour $P_M(G)$, mais définissent-elles des facettes de $P_M(G)$?

Theorem

Les contraintes triviales $x(e) \geq 0$, $e \in E$, définissent des facettes de $P_M(G)$.

Preuve : Tout d'abord, il faut prouver que ces contraintes sont bien valides. Ce qui est évident dans ce cas présent.

Soit $e \in E$. Considérons la face $F = \{x \in P_M(G) \mid x(e) = 0\}$.

Nous montrons en premier que F est propre. En effet, $F \neq \emptyset$ car $\chi^\emptyset \in P_M(G)$ par exemple. De plus, $F \neq P_M(G)$ car $\chi^{\{e\}}(e) = 1$ et donc $\chi^{\{e\}} \in P_M(G) \setminus F$.

Enfin, nous exhibons m vecteurs affinement indépendants dans F . Ce sont par exemple, les vecteurs χ^\emptyset et $\chi^{\{e'\}}$, $e' \in E \setminus \{e\}$.

Ainsi, la dimension de F est $\dim(F) = m - 1$ et F est bien une facette. □

27/41

Etude faciale de $P_M(G)$

Theorem

Les contraintes triviales $x(e) \leq 1$, $e \in E$ ne définissent pas des facettes.

Preuve : Soit $e \in E$.

L'inégalité $x(e) \leq 1$ peut-être obtenue à partir de l'inégalité de type (1) associée à l'un des sommets u incidents à e en lui additionnant les inégalités triviales $-x(e') \leq 0$ pour tout $e' \in \delta(u) \setminus \{e\}$.

Par conséquent, les contraintes triviales $x(e) \leq 1$, $e \in E$ ne définissent pas de facettes. □

On dit que les contraintes de type (1) **dominent** les contraintes triviales.

Par conséquent, en présence des inégalités de type (1), les inégalités triviales $x(e) \leq 1$, $e \in E$ peuvent être omises de toute formulation.

28/41

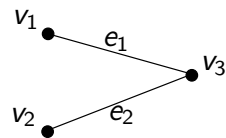
Etude faciale de $P_M(G)$

Pour les contraintes de type (1), il existe des cas où ces contraintes sont redondantes et ne définissent donc pas de facettes.

Theorem

Les contraintes de type (1) ne définissent pas en général de facettes.

Preuve : Il suffit de produire un contre-exemple. Considérons donc un graphe limité à deux arêtes adjacentes par une extrémité comme indiqué sur la figure suivante :



Si l'on prend les contraintes de type (1) associée aux sommets v_1, v_2 et v_3 . On obtient le système

$$\begin{array}{rcl} x(e_1) + x(e_2) & \leq & 1 \\ x(e_1) & \leq & 1 \\ x(e_2) & \leq & 1 \end{array}$$

On voit donc que la première de ces trois contraintes domine les deux autres. Ces deux dernières ne définissent donc pas des facettes. □

29/41

30/41

Caractérisation de $P_M(G)$

En fait, le système (1)-(3) d'inégalités caractérisent bien le polytope du couplage. Ce résultat donné par Edmonds en 1965 et redémontré de la façon suivante par Lovász en 1979.

Theorem

Pour tout graphe $G = (V, E)$, le polyèdre des couplages $P_M(G)$ est donné par les inégalités (2), (3) et (1).

31/41

Caractérisation complète de $P_M(G)$

Pour prouver ce résultat, nous utilisons le lemme suivant, qui est l'adaptation au cas du couplage d'un lemme de la section précédent.

Pour $ax \leq \alpha$ une contrainte définissant une facette de $P_M(G)$, on pose C_a l'ensemble des couplages dont le vecteur d'incidence associé vérifie l'inégalité $ax \leq \alpha$ à l'égalité, i.e.

$$C_a = \{C \in E \mid C \text{ couplage de } G \text{ et } a\chi^C = \alpha\}.$$

Lemma (Lemme de caractérisation)

Soit $ax \leq \alpha$ une contrainte définissant une facette de $P_M(G)$. Soit $a'x \leq \alpha'$ une contrainte valide de $P_M(G)$.

i) Si pour tout couplage $C \in C_a$, $a'\chi^C = \alpha'$, alors $ax \leq \alpha$ et $a'x \leq \alpha'$ sont identiques à un coefficient multiplicateur positif prêt.

ii) Si $ax \leq \alpha$ et $a'x \leq \alpha'$ ne sont pas identiques à un coefficient multiplicateur positif prêt, alors il existe un couplage $C \in C_a$ tel que $a'\chi^C = \alpha'$.

32/41

Caractérisation complète de $P_M(G)$

[Technique de preuve très intéressante]

Avant de nous lancer dans la preuve du théorème, donnons quatre lemmes techniques, spécifiques au problème du couplage qui nous seront nécessaires pour la preuve.

Lemma

Supposons $ax \leq \alpha$ une contrainte définissant une facette de $P_M(G)$ qui soit différente des contraintes triviales (2). Alors $a(e) \geq 0$ pour tout $e \in E$.

Notons que ce résultat indique que les seules inégalités utiles pour le polyèdre du couplage ont des coefficients positifs (en dehors de l'inégalité triviale $-x(e) \leq 0$).

Preuve : Soit $e_0 \in E$ et une inégalité $ax \leq \alpha$ une contrainte définissant une facette de $P_M(G)$ qui soit différente de la contrainte triviale $x(e_0) \geq 0$. Par le théorème précédent il existe donc un couplage C un couplage de C_a tel que $e_0 \notin C$. On pose alors $C' = C \setminus \{e_0\}$ qui est encore un couplage. Comme $ax \leq \alpha$ est une inégalité valide : $a\chi^{C'} \leq \alpha$. Or $a\chi^{C'} = a\chi^C - a(e_0)$ et donc $a\chi^C - a(e_0) \leq \alpha$. Comme $a\chi^C = \alpha$, on en déduit que $-a(e_0) \leq 0$, c'est-à-dire $a(e_0) \geq 0$. \square

33/41

Grphe support

On appelle **graphe support** G_a associé à l'inégalité $ax \leq \alpha$ le graphe composé des arêtes $e \in E$ telles que $a(e) > 0$.

Ce graphe peut-être vu comme la structure combinatoire de l'inégalité.

Lemma

G_a est connexe.

Preuve : Supposons que G_a soit l'union de deux graphes G_1 et G_2 disjoints. On pose alors a_1 (respectivement a_2) le vecteur obtenu depuis a en mettant à zéro les coefficients associés aux arêtes de G_1 (respectivement G_2). On note alors $\alpha_i = \max\{a_i\chi^C \mid C \text{ couplage de } G\}$, pour $i = 1, 2$. Alors $\alpha = \alpha_1 + \alpha_2$ et $a = a_1 + a_2$ donc $ax \leq \alpha$ s'obtient en sommant deux inégalités valides pour $P(G)$. Ce qui est impossible car elle définit une facette de $P(G)$. \square

35/41

Caractérisation complète de $P_M(G)$

Lemma

Supposons $ax \leq \alpha$ une contrainte définissant une facette de $P_M(G)$ qui soit différente des contraintes de couplage (1).

Alors, pour tout sommet v , il existe un couplage C de C_a tel que C ne contienne pas d'arête incidente à v .

Preuve : Pour cela, nous allons montrer que, dans ce cas, $ax \leq \alpha$ est alors de type (1). Supposons qu'il existe un sommet $v_0 \in V$ tel que tout couplage de $c \in C_a$, C contient une arête incidente à v_0 . C'est-à-dire, $\chi^C(\delta(v)) = 1$. Ce qui revient à dire par le lemme de caractérisation que l'inégalité $ax \leq \alpha$ est de type (1). \square

34/41

Caractérisation complète de $P_M(G)$

Lemma

Soit $e = uv$ une arête de G telle que u et v soit dans G_a et telle qu'il existe un couplage $C \in C_a$ non incident à u et v . Alors l'arête e n'est pas dans G_a .

Preuve : Comme C n'est ni incident à u , ni à v , alors $C' = C \cup \{e\}$ est encore un couplage de G , donc, par validité, $a\chi^{C'} \leq \alpha$. Or, comme $C \in C_a$, $a\chi^C = \alpha$ et $a\chi^{C'} = a\chi^C - a(e)$, on a $a(e) \leq 0$. Or, par définition du graphe support, $a(e) > 0$, une contradiction. \square

36/41

Caractérisation complète de $P_M(G)$

Revenons enfin au théorème d'Edmonds!

Theorem

Pour tout graphe $G = (V, E)$, le polyèdre des couplages $P_M(G)$ est donné par les inégalités (2), (3) et (1).

Preuve du théorème :

Tout d'abord, notons que toutes les contraintes sont bien valides.

Soit $ax \leq \alpha$ une contrainte définissant une facette de $P_M(G)$.

Le schéma de la preuve est le suivant :

- on suppose que $ax \leq \alpha$ définit une facette n'est pas de type (2) et (1), on va alors prouver que dans ce cas $ax \leq \alpha$ est de type (3)

- Pour prouver cela, on va exhiber un sous-ensemble $S_0 \subset V$ tel que

$\chi^C(\delta(S_0)) = \frac{|S_0|-1}{2}$ pour tout couplage de $C \in C_a$: ce qui prouvera alors que $ax \leq \alpha$ est de type (3).

L'ensemble S_0 est en fait l'ensemble des sommets de V incidents aux arête e de G_a (i.e. telle que $a(e) > 0$). Considérons alors la contrainte (3) associée à S_0 , c'est-à-dire

$$x(E(S_0)) \leq \frac{|S_0|-1}{2} \quad (4)$$

37/41

Caractérisation complète de $P_M(G)$

[La preuve est ici donnée à titre de culture générale]

On peut alors montrer que C_2 est incident u et v . En effet, supposons que, par exemple, C_2 ne couvre pas u , dans ce cas, u et z sont deux sommets de G_a non couverts par C_2 . De plus, la chaîne qui les relie dans G_a est de taille strictement inférieure à celle reliant u et v (car $a(e) > 0$ pour tout e dans G_a), d'où contradiction. On obtient la même contradiction pour v .

Considérons alors l'ensemble d'arêtes $M = C_1 \cup C_2$. On peut déduire des deux paragraphes précédents que les sommets u, v et z sont de degré 1 dans le graphe G^* formé par les arêtes de M .

Sans perte de généralité, on peut supposer que la composante connexe de G^* contenant u consiste en une chaîne Q ne passant pas par z . Considérons alors les deux couplages

$$\bar{C}_1 = (C_1 \setminus Q) \cup (C_2 \cap Q)$$

et

$$\bar{C}_2 = (C_2 \setminus Q) \cup (C_1 \cap Q),$$

obtenus à partir de C_1 et de C_2 en permutant le statut "dedans" ou "dehors" des arêtes de Q pour chacun des couplages. Puisque $\bar{C}_1 \cup \bar{C}_2 = C_1 \cup C_2$, alors \bar{C}_1 et \bar{C}_2 sont dans C_a . Comme, \bar{C}_2 ne couvre ni u , ni z , cela contredit le choix de M_1 selon une chaîne minimale. □

39/41

Caractérisation complète de $P_M(G)$

[La preuve est ici donnée à titre de culture générale]

Supposons donc que notre contrainte $ax \leq \alpha$ soit aussi différente de cette inégalité (4) associée à S_0 .

Donc il existe un couplage $C_1 \in C_a$ qui ne vérifie pas à l'égalité (4), i.e. tel que $\chi^{C_1} x(E(S_0)) < \frac{|S_0|-1}{2}$.

Par conséquent, comme $\frac{|S_0|-1}{2}$ désigne le nombre de paires distinctes dans S_0 , S_0 contient au moins une paire u et v de sommets qui ne sont incidents à aucune arête de C_1 .

De plus, par le lemme de connexité du graphe support, G_a est connexe. On va supposer alors qu'on a choisi C_1 de manière à ce que la distance entre u et v dans G_a selon le poids a est la plus courte possible.

Par le lemme sur les arêtes de G_a , on sait aussi que u et v ne sont pas adjacents dans G_a .

Alors il existe un sommet z différent de u et v dans G_a sur la plus courte chaîne reliant u et v . Comme $ax \leq \alpha$ est différente des contraintes (1), par le lemme de caractérisation, il existe un couplage C_2 de C_a tel que C_2 ne contienne pas d'arête incidente à z .

38/41

Caractérisation complète de $P_M(G)$

En plus de ce résultat de caractérisation, il existe un algorithme de séparation polynomial pour les inégalités (3).

Ainsi ce résultat plus cet algorithme prouve que le problème de couplage est polynomial!

Bon... on connaissait ce résultat de complexité depuis Koëning...

Mais ce résultat a ouvert la porte à de nombreux autres résultats sur des problèmes que l'on ne savait pas résoudre auparavant!

On considère ce résultat comme le père des travaux sur le voyageur de commerce qui a permis de résoudre exactement jusqu'à l'instance contenant toutes les villes de la planète (et aussi tous les pubs irlandais)... et de toute une théorie de la programmation mathématique bien vivante dans le monde de l'industrie.

40/41

Pour aller plus loin

Il y aurait d'autres titres de chapitres possibles sur ce thème :

- Les opérations de liftings dans les polyèdres très utilisées pour les problèmes "non-graphiques"
- Les techniques de décompositions de polyèdres qui permettent de prouver des cas polynomiaux nouveaux de problèmes classiques
- Les formulations étendues qui donnent des formulations compactes caractérisant certains problèmes classiques
- Certains polyèdres célèbres comme celui des permutations (permutrons), des orbites de permutations (orbitopes)... qui sont la clef du traitement des symétries dans les PLNE
- Les inégalités dites de disjonction qui ont accéléré Cplex de manière importante
- ...

Master d'Informatique - Spécialité Androide
Module MAOA

Recherche Opérationnelle
et Optimisation Combinatoire

Partie F - Génération de colonnes
(Branch-and-Price)

Pierre Fouilhoux

Sorbonne Université

2019-2020

1. Principe de la génération de colonnes sur un exemple
2. Définitions et notations
3. Décomposition de Dantzig-Wolfe
4. Application au problème de coloration

2/69

Recherche Opérationnelle et Optimisation Combinatoire Partie F - Génération de colonnes (Branch-and-Price)
└─ Principe de la génération de colonnes sur un exemple

Recherche Opérationnelle et Optimisation Combinatoire Partie F - Génération de colonnes (Branch-and-Price)
└─ Principe de la génération de colonnes sur un exemple
└─ Le problème de multiflot-max

1. Principe de la génération de colonnes sur un exemple
 - 1.1 Le problème de multiflot-max
 - 1.2 Génération de colonnes pour le multiflot-max fractionnaire
 - 1.3 Le problème de multiflot de coût minimum
 - 1.4 Initialisation d'une génération de colonnes
 - 1.5 Résoudre le pricing
 - 1.6 Branchement pour le multiflot entier
2. Définitions et notations
3. Décomposition de Dantzig-Wolfe
4. Application au problème de coloration

1. Principe de la génération de colonnes sur un exemple
 - 1.1 Le problème de multiflot-max
 - 1.2 Génération de colonnes pour le multiflot-max fractionnaire
 - 1.3 Le problème de multiflot de coût minimum
 - 1.4 Initialisation d'une génération de colonnes
 - 1.5 Résoudre le pricing
 - 1.6 Branchement pour le multiflot entier
2. Définitions et notations
3. Décomposition de Dantzig-Wolfe
4. Application au problème de coloration

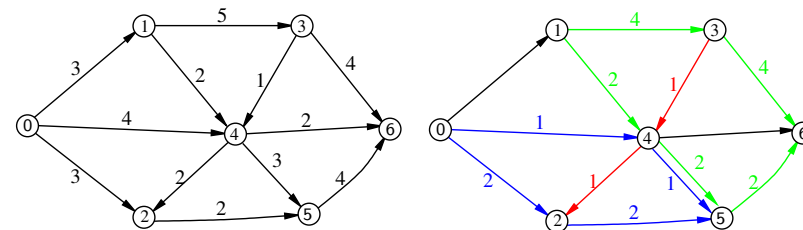
Un exemple : Le problème de multiflot-max

Soit $G = (V, A)$ un graphe orienté et k **commodités**, c'est-à-dire des paires de sommets $(s_1, t_1), \dots, (s_k, t_k)$ entre lesquelles doit circuler un flot.

Le réseau G est muni d'une capacité $b(a) \in \mathbb{N}$ associé à chaque arc $a \in A$.
 Pour chaque paire $i \in \{1, \dots, k\}$, on désire trouver des (s_i, t_i) -flots entiers tels que la somme des capacités utilisées par les flots sur un même arc respecte la capacité de cet arc.
 L'objectif du **problème de multiflot-max** est de maximiser la quantité total de flot circulant sur le graphe.

Si flots continus, le problème de **multiflot-max fractionnaire** est polynomial.
 Si flots entiers, le problème de **multiflot-max entier** est NP-difficile dès que $k \geq 2$.

Un exemple : Le problème de multiflot-max



Commodités : $(0,5), (1,6), (3,2)$

On peut noter que les trois flots reliant chacune des trois commodités sont en concurrence sur le graphe.

L'arc $(4, 5)$ porte à la fois du flot des commodités $(0, 5)$ et $(1, 6)$.

Notons que pour chaque nœud, il y a une conservation des flots individuelle (il n'y a pas de mélange des flots...).

Une première formulation naturelle

Considérons des variables $f^i(a)$ indiquant la quantité de flot passant par l'arc a pour desservir la commodité (s_i, t_i) .

$$\begin{aligned} \text{Max} \sum_{i=1}^k \sum_{u \in \delta^+(s_i)} f^i(s_i, u) \\ \sum_{v \in \delta^-(u)} f^i(v, u) &= \sum_{v \in \delta^+(u)} f^i(u, v) \quad \forall i \in \{1, \dots, k\}, \forall u \in V, \\ \sum_{i=1}^k f^i(a) &\leq b(a) \quad \forall a \in A, \\ f^i(a) &\geq 0 \quad \forall i \in \{1, \dots, k\}, \forall a \in A. \end{aligned}$$

Ce problème est donc polynomial en utilisant un PL avec km variables et $kn + m$ inégalités.

Pour un graphe de 1000 sommets, 10000 arêtes et des commodités entre toute paire de sommets du graphe, cela représente un taille de PL trop importante pour être traitée par un solveur linéaire... mais il existe une reformulation bien plus efficace !

Reformulation

Une reformulation peut être obtenue en considérant d'autres variables.

Pour chaque commodité (s_i, t_i) , $i \in \{1, \dots, k\}$, considérons $\mathcal{P}_{s_i t_i}$ l'ensemble des chemins (élémentaires) allant du sommet s_i au sommet t_i .

Posons $\mathcal{P} = \bigcup_{i=1}^k \mathcal{P}_{s_i t_i}$.

Il est bien connu qu'un (s, t) -flot est un ensemble de chemins allant de s à t et portant une part du flot allant de s à t . L'idée est de chercher, pour chaque commodité, un tel ensemble de chemins.

On associe une variable f_μ associée à chaque chemin $\mu \in \mathcal{P}$.

Reformulation

La formulation (\tilde{F}) est un programme linéaire, dite **formulation arc-chemin**, qui modélise le problème de multiflot-max fractionnaire.

$$(\tilde{F}) \quad \begin{aligned} & \text{Max} \sum_{\mu \in \mathcal{P}} f_{\mu} \\ & \sum_{\substack{\mu \in \mathcal{P} \\ a \in \mu}} f_{\mu} \leq b(a) \quad \forall a \in A \\ & f_{\mu} \geq 0 \quad \forall \mu \in \mathcal{P}. \end{aligned}$$

Si on ajoute en plus la contrainte d'intégrité, on obtient alors une formulation (F) qui est un PLNE modélisant le problème de multiflot-max entier.

$$(F) \quad \begin{aligned} & \text{Max} \sum_{\mu \in \mathcal{P}} f_{\mu} \\ & \sum_{\substack{\mu \in \mathcal{P} \\ a \in \mu}} f_{\mu} \leq b(a) \quad \forall a \in A \\ & f_{\mu} \geq 0 \quad \forall \mu \in \mathcal{P}. \\ & f_{\mu} \in \mathbf{N} \quad \forall \mu \in \mathcal{P}. \end{aligned}$$

Notez donc que (\tilde{F}) est la relaxation linéaire de (F) .

Principe de la génération de colonnes sur un exemple

La formulation PL arc-chemin

$$(\tilde{F}) \quad \begin{aligned} & \text{Max} \sum_{\mu \in \mathcal{P}} f_{\mu} \\ & \sum_{\substack{\mu \in \mathcal{P} \\ a \in \mu}} f_{\mu} \leq b(a) \quad \forall a \in A \\ & f_{\mu} \geq 0 \quad \forall \mu \in \mathcal{P}. \end{aligned}$$

possède seulement m inégalités... mais un nombre exponentiel de variables!

Ce qui veut dire que même sa relaxation linéaire ne peut pas être directement obtenue par un solveur linéaire (algo du simplexe,...). On résoud la relaxation linéaire d'une telle formulation par l'**algorithme de génération de colonnes** (column generation, en anglais)

1. Principe de la génération de colonnes sur un exemple

1.1 Le problème de multiflot-max

1.2 Génération de colonnes pour le multiflot-max fractionnaire

1.3 Le problème de multiflot de coût minimum

1.4 Initialisation d'une génération de colonnes

1.5 Résoudre le pricing

1.6 Branchement pour le multiflot entier

2. Définitions et notations

3. Décomposition de Dantzig-Wolfe

4. Application au problème de coloration

Rappel : Dualité

Pour un programme linéaire (\tilde{P}) , appelé alors *primal*, le *dual* est le programme linéaire (\tilde{D}) suivant

$$(\tilde{P}) \quad \begin{cases} \text{Max} & z = c^T x \\ & Ax \leq b \\ & x \geq 0 \end{cases} \quad (\tilde{D}) \quad \begin{cases} \text{Min} & w = b^T y \\ & A^T y \geq c \\ & y \geq 0 \end{cases}$$

Sous forme "algébrique" :

$$(\tilde{P}) \quad \begin{cases} \text{Max} & z = \sum_{j=1}^n c_j x_j \\ & \sum_{j=1}^n a_{ij} x_j \leq b_i \quad \forall i = 1, \dots, m \\ & x_i \geq 0 \quad \forall j = 1, \dots, n \end{cases} \quad (\tilde{D}) \quad \begin{cases} \text{Min} & w = \sum_{i=1}^m b_i y_i \\ & \sum_{i=1}^m a_{ij} y_i \geq c_j \quad \forall j = 1, \dots, n \\ & y_i \geq 0 \quad \forall i = 1, \dots, m \end{cases}$$

Les **variables duales** de (\tilde{D}) correspondent aux inégalités de (\tilde{P}) .

La matrice de (\tilde{D}) est la transposée A^T .

Les coûts de la fonction objective et les termes de droite des inégalités échangent leurs rôles.

Rappel : Dualité

• *Théorème (faible) de la dualité :*

Si (\tilde{P}) et (\tilde{D}) admettent chacun une solution \tilde{x} et \tilde{y} , alors $c^T \tilde{x} \leq b^T \tilde{y}$.

• *Théorème de la dualité :*

Si (\tilde{P}) admet une solution optimale (finie), alors (\tilde{D}) aussi et de plus elles "coïncident", i.e.

$$(\tilde{P}) \max\{c^T x \mid Ax \leq b\} = \min\{b^T y \mid y^T A = c, y \geq 0\} \quad (\tilde{D}).$$

• *Corollaire de la dualité :*

Soit x une solution de (\tilde{P}) et y une solution de (\tilde{D}) . Alors x et y sont solutions optimales de (\tilde{P}) et (\tilde{D}) si et seulement si $c^T x = b^T y$.

Principe de la génération de colonnes sur un exemple

Comme (\tilde{F}) est un programme linéaire, considérons son dual (\tilde{D}) . On note λ_a la variable duale associée à l'inégalité associée à un arc $a \in A$ de (\tilde{F}) .

$$\begin{aligned} \text{Max} \sum_{\mu \in \mathcal{P}} f_{\mu} & & \text{Min} \sum_{a \in A} b(a)\lambda_a \\ \sum_{\mu \in \mathcal{P} \mid a \in \mu} f_{\mu} & \leq b(a) \quad \forall a \in A & \sum_{a \in \mu} \lambda_a & \geq 1 \quad \forall \mu \in \mathcal{P}, \\ f_{\mu} & \geq 0 \quad \forall \mu \in \mathcal{P}. & \lambda_a & \geq 0 \quad \forall a \in A. \end{aligned}$$

En tant que transposé, le dual (\tilde{D}) possède alors un nombre exponentiel de lignes mais par contre un nombre fini de variables !

La condition d'optimalité donné par le corollaire de la dualité est la suivante :

Si l'on sait produire à la fois :
- une solution f^* du primal (\tilde{F})
- et une solution λ^* du dual (\tilde{D})
- qui ont même valeur objective,
alors f^* est optimale pour (\tilde{F}) .

Principe de la génération de colonnes sur un exemple

Initialisation :

Dans ce cas simple de multiflot-max fractionnaire, il est facile d'avoir une solution réalisable.

En effet, pour chaque commodité (s_i, t_i) , prenons un chemin quelconque μ_i^0 reliant s_i à t_i .

Considérons alors l'ensemble \mathcal{P}_0 de ces chemins.

Et les k variables f_{μ} associée aux chemins de \mathcal{P}_0 .

Alors la formulation arc-chemin restreinte à ces variables admet une solution et cette solution est bien une solution du problème de multiflot-max fractionnaire (au pire les valeurs des flots véhiculés sont toutes nulles).

Considérons :

- le PL initial (\tilde{F}_0) qui est la restriction de (\tilde{F}) aux k variables $f_{\mu}, \mu \in \mathcal{P}_0$.
- son dual (\tilde{D}_0) .

Notons que (\tilde{D}_0) est aussi la restriction de (\tilde{D}) aux k inégalités associées aux chemins $\mu \in \mathcal{P}_0$.

Principe de la génération de colonnes sur un exemple

Programme linéaire initial (\tilde{F}_0) et son dual (\tilde{D}_0) .

$$\begin{aligned} \text{Max} \sum_{\mu \in \mathcal{P}_0} f_{\mu} & & \text{Min} \sum_{a \in A} b(a)\lambda_a \\ \sum_{\mu \in \mathcal{P}_0 \mid a \in \mu} f_{\mu} & \leq b(a) \quad \forall a \in A & \sum_{a \in \mu} \lambda_a & \geq 1 \quad \forall \mu \in \mathcal{P}_0, \\ f_{\mu} & \geq 0 \quad \forall \mu \in \mathcal{P}_0. & \lambda_a & \geq 0 \quad \forall a \in A. \end{aligned}$$

Résoudre le PL compact (\tilde{F}_0) (et en même temps son dual (\tilde{D}_0)) est polynomial.

Notons f^0 la solution optimale de (\tilde{F}_0) et λ^0 la solution optimale de (\tilde{D}_0) .

Remarques essentielles :

- λ^0 est un vecteur candidat à être solution de (\tilde{D}) tout entier.
 - f^0 possède seulement k composantes parmi le nombre exponentiel possible dans (\tilde{F}) .
- Posons alors f^* le vecteur de (\tilde{F}) obtenu en fixant à 0 toutes les colonnes n'apparaissant pas dans \mathcal{P}_0 : **alors f^* est une solution de (\tilde{F}) !**

Principe de la génération de colonnes sur un exemple

On peut déjà tester l'optimalité de f^* :

En effet, par dualité forte, on sait que les fonctions objective de f^0 et λ^0 sont identiques pour (\tilde{F}_0) et pour (\tilde{D}_0) .

De plus, par construction (ajout des valeurs nulles), la fonction objective pour f^0 dans (\tilde{F}_0) est la même que celle de $*$ dans (\tilde{F}) .

Donc le corollaire de la dualité nous dit que :

Si λ^0 est solution de (\tilde{D}) , alors f^* est optimale pour (\tilde{F}) .

Or tester si λ^0 est solution de (\tilde{D}) revient à tester si le vecteur λ^0 vérifie les contraintes du dual, c'est-à-dire si

$$\sum_{a \in \mu} \lambda_a \geq 1 \quad \forall \mu \in \mathcal{P}$$

Cette condition est donc vérifiée si le vecteur λ^0 vérifie ce lot d'inégalités qui est en nombre exponentiel : autant qu'il y a de chemins !

On appelle ce test : le problème de "pricing" (ou en français le problème de génération d'une colonne).

17/69

Principe de la génération de colonnes sur un exemple

Validité de l'algorithme :

L'algorithme, dans le pire des cas, va ajouter toutes les variables possibles : il est donc trivialement valide.

Complexité de l'algorithme :

Il s'agit de la même question que la complexité de la méthode de coupes : en effet, le problème de pricing est le problème de séparation pour le dual de la formulation !

Donc par le résultat de Groëtschel-Lovasz-Schrijver :

si le problème de pricing se résoud en temps polynomial (i.e. si λ^0 vérifie les inégalités), alors la réitération du problème de pricing dans une méthode de génération de colonnes est polynomiale !

19/69

Principe de la génération de colonnes sur un exemple

Algorithme de génération de colonnes :

- Si le problème de pricing renvoie "vraie" : alors vecteur dual λ^0 vérifie les inégalités et la solution primale f^* est optimale.

- Sinon (le problème de pricing renvoie "faux") : alors il existe une inégalité violée par λ^0 : cette inégalité correspond à un chemin $\mu \in \mathcal{P}$. On ajoute alors la variable f_μ au PL maître

$$(\tilde{F}_{i+1}) \leftarrow (\tilde{F}_i) \text{ plus la variable } f_\mu$$

Et on recommence pour $(\tilde{F}_{i+1})!$

C'est le principe de génération de colonnes (ou de variables).

18/69

Principe de la génération de colonnes sur un exemple

Le problème de pricing de cette génération de colonnes pour le problème de multiflot est de tester les inégalités du dual pour chaque chemin de \mathcal{P} , c'est-à-dire si

$$\sum_{a \in \mu} \lambda'_a \geq 1 \quad \forall \mu \in \mathcal{P}.$$

C'est-à-dire

Tester s'il existe un chemin μ correspondant à une contrainte violée et répondre VRAI en produisant ce chemin
 Et sinon répondre FAUX.

On recherche donc un algorithme capable de résoudre le problème de pricing : un tel algorithme est appelé en anglais **pricer**.

20/69

Principe de la génération de colonnes sur un exemple

Ici ce problème de pricing se ramène à déterminer, pour chaque paire de commodités (s_i, t_i) , un plus court chemin de s_i à t_i selon les poids λ^0 associés aux arcs de A : Si le plus court de ces chemins est de valeur ≥ 1 , alors toutes les inégalités du dual sont vérifiées : le problème de pricing renvoie FAUX. Sinon, le plus court de ces chemins viole l'inégalité et ce chemin μ^* est retourné par le problème de pricing. Dans ce dernier cas, on crée la variable f_{μ^*} et on l'ajoute au problème maître.

Comme les coûts duaux λ^0 sont positifs, la recherche d'un plus court chemin est polynomial (par exemple avec l'algorithme de Dijkstra $O(m \log(n))$), le problème de pricing peut être résolu en $O(k m \log(n))$: un tel algorithme résout bien le problème de pricing (c'est un pricer). Comme ce pricer est polynomial, sa réitération dans une génération de colonnes est alors polynomiale !

La formulation arc-chemin pour le problème de multiflot-max fractionnaire peut donc être résolue en temps polynomial !

21/69

1. Principe de la génération de colonnes sur un exemple

- 1.1 Le problème de multiflot-max
- 1.2 Génération de colonnes pour le multiflot-max fractionnaire
- 1.3 Le problème de multiflot de coût minimum
- 1.4 Initialisation d'une génération de colonnes
- 1.5 Résoudre le pricing
- 1.6 Branchement pour le multiflot entier

2. Définitions et notations

3. Décomposition de Dantzig-Wolfe

4. Application au problème de coloration

23/69

Principe de la génération de colonnes sur un exemple

On peut noter que cette méthode de génération de colonnes suit le principe de l'algorithme du simplexe : entrée d'une variable en base en fonction des coûts réduits.

La méthode de génération de colonnes suit ainsi les bonnes propriétés observées (et par encore tout à fait bien comprises) de l'algorithme du simplexe comme le fait de rarement nécessiter beaucoup d'itérations (et donc de variables).

Malheureusement, la méthode de génération de colonnes suit aussi les mauvaises propriétés de l'algorithme du simplexe : bien souvent il s'agit d'itérations fortement dégénérées où plusieurs bases donnent la même solution ! Et il est plus difficile que dans le cas du simplexe de faire converger la méthode : des outils d'optimisation continue sont souvent nécessaires.

Dans le cas du problème de multiflot fractionnaire, la convergence est rapide et cette formulation est très efficace, bien plus efficace que la formulation naturelle avec un nombre polynomial mais très élevée de variables.

22/69

Le problème de multiflot de coût minimum

Le problème de multiflot-max considéré jusqu'ici est une version simplifiée du problème classique de multiflot.

Soit $G = (V, A)$ un graphe orienté et k **commodités**, c'est-à-dire des triplets $(s_1, t_1, d_1), \dots, (s_k, t_k, d_k)$ où (s_i, t_i) est une paire de sommets entre lesquelles doit circuler un flot de valeur au moins d_i .

Le réseau G est muni :

- d'une capacité $b(a) \in \mathbb{N}$ associé à chaque arc $a \in A$
- et d'un coût unitaire (positif) d'utilisation des arcs $c(a)$.

L'objectif du **problème de multiflot** (de coût minimum) est de déterminer k flots reliant les commodités avec un coût total minimal.

Ce problème de multiflot est un modèle assez réaliste pour le routage des données entre les paires de clients voulant communiquer dans un réseau de télécommunications.

Si flots continus, le problème de **multiflot fractionnaire** est polynomial. Si flots entiers, le problème de **multiflot entier** est NP-difficile dès que $k \geq 2$.

24/69

Une première formulation naturelle

Comme pour le multiflot-max, on peut considérer des variables $f^i(a)$ indiquant la quantité de flot passant par l'arc a pour desservir la commodité (s_i, t_i) .

$$\begin{aligned} \text{Min } & \sum_{i=1}^k \sum_{a \in A} c(a) f^i(a) \\ & \sum_{v \in \delta^-(u)} f^i(vu) = \sum_{v \in \delta^+(u)} f^i(uv) \quad \forall i \in \{1, \dots, k\}, \forall u \in V \setminus \{s_i, t_i\}, \\ & \sum_{v \in \delta^-(s_i)} f^i(vs_i) + d_i = \sum_{v \in \delta^+(s_i)} f^i(s_i v) \quad \forall i \in \{1, \dots, k\}, \\ & \sum_{v \in \delta^-(t_i)} f^i(vt_i) = \sum_{v \in \delta^+(t_i)} f^i(t_i v) + d_i \quad \forall i \in \{1, \dots, k\}, \\ & \sum_{i=1}^k f^i(a) \leq b(a) \quad \forall a \in A, \\ & f^i(a) \geq 0 \quad \forall i \in \{1, \dots, k\}, \forall a \in A. \end{aligned}$$

Ce problème est donc polynomial par ce PL mais sa taille est bien trop importante pour être traitée par un solveur entier.

Reformulation

On utilise la même reformulation (\tilde{F}) qui est un programme linéaire, dit **formulation arc-chemin**, qui modélise le problème de multiflot fractionnaire

$$\begin{aligned} \text{Min } & \sum_{\mu \in \mathcal{P}} c(\mu) f_\mu \\ (\tilde{F}) \quad & \sum_{\mu \in \mathcal{P} \mid a \in \mu} f_\mu \leq b(a) \quad \forall a \in A \\ & \sum_{\mu \in \mathcal{P}_{s_i t_i}} f_\mu = d_i \quad \forall i \in \{1, \dots, k\} \\ & f_\mu \geq 0 \quad \forall \mu \in \mathcal{P}. \end{aligned}$$

où la notation $c(\mu)$ désigne le coût total des arcs du chemin μ : $c(\mu) = \sum_{a \in \mu} c(a)$.

Si on ajoute en plus la contrainte d'intégrité, on obtient alors une formulation (F) qui est un PLNE modélisant le problème de multiflot entier.

Pricing

Comme (\tilde{F}) est un programme linéaire, considérons son dual (\tilde{D}).
Posons λ_a la variable duale associée à l'inégalité associée à un arc $a \in A$ de (\tilde{F}),
et ω_i la variable duale associée à l'inégalité associée à une commodité $i \in \{1, \dots, k\}$.

$$\begin{aligned} \text{Min } & \sum_{\mu \in \mathcal{P}} c(\mu) f_\mu \\ (\tilde{F}) \quad & \sum_{\mu \in \mathcal{P} \mid a \in \mu} f_\mu \leq b(a) \quad \forall a \in A \\ & \sum_{\mu \in \mathcal{P}_{s_i t_i}} f_\mu = d_i \quad \forall i \in \{1, \dots, k\} \\ & f_\mu \geq 0 \quad \forall \mu \in \mathcal{P} \\ \text{Max (fonction objective inutile)} \\ (\tilde{D}) \quad & \sum_{a \in \mu} \lambda_a + \omega_i \leq c(\mu) \quad \forall i \in \{1, \dots, k\}, \forall \mu \in \mathcal{P}_i \\ & \lambda_a \leq 0 \quad \forall a \in A \\ & \omega_i \in \mathbf{R} \quad \forall i \in \{1, \dots, k\} \end{aligned}$$

1. Principe de la génération de colonnes sur un exemple

- 1.1 Le problème de multiflot-max
- 1.2 Génération de colonnes pour le multiflot-max fractionnaire
- 1.3 Le problème de multiflot de coût minimum
- 1.4 Initialisation d'une génération de colonnes
- 1.5 Résoudre le pricing
- 1.6 Branchement pour le multiflot entier

2. Définitions et notations

3. Décomposition de Dantzig-Wolfe

4. Application au problème de coloration

Initialisation

Pour appliquer le principe précédent, il faut obtenir un ensemble de chemins \mathcal{P}' tel que la **formulation restreinte** $\tilde{F}_{\mathcal{P}'}$ possède une solution...
C'est l'**étape d'initialisation de la génération de colonnes**.

Or déterminer une solution réalisable pour ce problème de multiflot n'est pas évident (bien que polynomial). Il y a plusieurs façons de procéder :

- soit déterminer heuristiquement une solution réalisable (mais cela n'est pas garanti de fonctionner)
- soit déterminer heuristiquement beaucoup de chemins (jusqu'à ce que cela soit réalisable)
- soit démarrer "à vide" c'est-à-dire partir d'une formulation avec seulement un chemin par commodité (donc pas forcément réalisable) : dans ce cas, un primal non réalisable correspond à des coûts réduits infinis... On peut par contre utiliser le pricer classique avec des coûts particuliers, dit coût de Farkas, (ils proviennent du Lemme de Farkas (voir cours sur l'approche polyédrale) : ces coûts guident vers une solution réalisable.

29/69

Pricing et Pricer

Supposons à présent que l'on a trouvé un ensemble \mathcal{P}' de chemins tels que la formulation $\tilde{F}_{\mathcal{P}'}$ ait une solution.

Le problème de pricing est de tester s'il existe un chemin $\mu \in \mathcal{P}'$ tel que

$$\sum_{a \in \mu} \lambda_a + \omega_i \leq c(\mu) \quad \forall i \in \{1, \dots, k\}, \forall \mu \in \mathcal{P}'$$

Une façon efficace de répondre à ce problème est :

- d'effectuer l'étape suivant pour chaque $i \in \{1, \dots, k\}$
- de se rappeler que $c(\mu) = \sum_{a \in \mu} c(a)$

Alors l'inégalité s'écrit

$$\sum_{a \in \mu} (c(a) - \lambda_a) \geq \omega_i$$

où ω_i est une donnée fixe pour chaque itération i .

31/69

1. Principe de la génération de colonnes sur un exemple

1.1 Le problème de multiflot-max

1.2 Génération de colonnes pour le multiflot-max fractionnaire

1.3 Le problème de multiflot de coût minimum

1.4 Initialisation d'une génération de colonnes

1.5 Résoudre le pricing

1.6 Branchement pour le multiflot entier

2. Définitions et notations

3. Décomposition de Dantzig-Wolfe

4. Application au problème de coloration

30/69

Pricing et Pricer

Posons alors une longueur auxiliaire associée aux arcs du graphe

$$l_a = c(a) - \lambda_a \quad \forall a \in A$$

Comme $\lambda_a \leq 0$, alors cette longueur est positive !!

Le pricer revient à nouveau ici, pour chaque commodité i , à rechercher un plus court chemin entre s_i et t_i selon le poids l_a .

Si cette plus courte longueur est supérieure à d_i , alors il n'existe aucune variable améliorant le problème maître dans \mathcal{P}' . Sinon on a déterminé une variable à ajouter (on peut en ajouter plusieurs à la fois).

Ce pricer est donc polynomial et la résolution de la formulation arc-chemin pour le problème du multiflot fractionnaire est donc aussi polynomiale (et très efficace).

32/69

1. Principe de la génération de colonnes sur un exemple

1.1 Le problème de multiflot-max

1.2 Génération de colonnes pour le multiflot-max fractionnaire

1.3 Le problème de multiflot de coût minimum

1.4 Initialisation d'une génération de colonnes

1.5 Résoudre le pricing

1.6 Branchement pour le multiflot entier

2. Définitions et notations

3. Décomposition de Dantzig-Wolfe

4. Application au problème de coloration

Problème de multiflot entier

Le problème de multiflot entier est en revanche NP-difficile.

Sa formulation arc-chemin est un PLNE :

$$(F) \quad \begin{aligned} \text{Min} \quad & \sum_{\mu \in \mathcal{P}} c(\mu) f_{\mu} \\ & \sum_{\mu \in \mathcal{P} \mid a \in \mu} f_{\mu} \leq b(a) \quad \forall a \in A \\ & \sum_{\mu \in \mathcal{P}_{s_i t_i}} f_{\mu} = d_i \quad \forall i \in \{1, \dots, k\} \\ & f_{\mu} \geq 0 \quad \forall \mu \in \mathcal{P} \\ & f_{\mu} \in \mathbf{N} \quad \forall \mu \in \mathcal{P} \end{aligned}$$

dont la relaxation linéaire est (\tilde{F}) est celle du multiflot fractionnaire.

Donc la relaxation linéaire de ce PLNE peut être résolue en temps polynomial par un algorithme de génération de colonnes.

Il est donc possible d'utiliser cet algorithme de génération de colonnes pour résoudre chacun des nœuds d'un arbre de branchement sur les variables f_{μ} : un tel algorithme est appelé **algorithme de génération de colonnes et branchement** ou **Branch-and-Price algorithm**, en anglais.

Problème de multiflot entier

La règle classique de branchement sur les variables entières :

- sélectionner une variable fractionnaire f_{μ} parmi celles non-entière de la relaxation \tilde{F} .

- prendre la partie entière inférieure v de f_{μ}

- créer deux sous-problèmes fils, l'un où $f_{\mu} \leq v$ et l'autre avec $f_{\mu} \geq v + 1$

Cette règle n'est pas très efficace pour cette formulation (F) : en effet, il existe de nombreux chemins possibles, certains très proches les uns des autres à une arête près.

De nombreuses règles de branchement plus efficaces ont été tentées... passant souvent par réécrire tout ou partie la formulation pour avoir un branchement efficace.

Une des difficultés est que l'ajout d'inégalités dans le problème maître conduit à créer de nouvelles variables duales à prendre en compte dans le coût dual de la colonne à calculer...

Nous verrons plus loin le cas d'un branchement complet pour le problème de coloration.

1. Principe de la génération de colonnes sur un exemple

2. Définitions et notations

3. Décomposition de Dantzig-Wolfe

4. Application au problème de coloration

Notations

Considérons une formulation PLNE décrite de la façon suivante :

$$(P) \quad \begin{aligned} \text{Min} \quad & \sum_{s \in \mathcal{S}} c(s) t_s \\ & \sum_{s \in \mathcal{S}} \chi_k^s t_s \geq b_k \quad \forall k \in \mathcal{K} \\ & t_s \in \mathbf{N} \quad \forall s \in \mathcal{S} \end{aligned}$$

où

- \mathcal{S} est l'ensemble des variables entières de la formulation (potentiellement en nombre exponentiel).
- $c(s)$ est le coût d'une variables.
- \mathcal{K} est un ensemble d'inégalités (a priori en nombre polynomial).
- pour une contrainte k donnée, χ_s^k est le coefficient de la varuble t_s pour cette inégalité : notez que le vecteur colonne χ_s correspond ainsi à un vecteur décrivant la variable t_s .

37/69

Algorithme de Branch-and-Price

Pour résoudre la formulation PLNE (P) il faut utiliser un arbre de branchement où la relaxation linéaire de chacun des PL associés aux nœuds est résolu par l'algorithme de génération de colonnes : c'est l'**algorithme Branch-and-Price**.

Or une génération de colonnes converge parfois lentement.

Néanmoins les algorithmes de Branch-and-Price ont fait leurs preuves pour :

- obtenir des bornes inférieures intéressantes pour un problème (problème d'ordonnancement, de coloration,...)
- obtenir des bonnes solutions avec une garantie expérimentale par différentes méthodes d'approximation, arrondis,...
- et, pour certains problèmes, la génération de colonnes est la meilleure méthode pour obtenir des solutions entières optimales : problème de tournées de véhicules (voir exo TD), de coloration (voir fin de ce cours), problème de découpes,...

39/69

Algorithme de génération de colonnes

L'algorithme de génération de colonnes permet de résoudre la relaxation linéaire (\tilde{P}) de (P) peut-être décrit comme suit

- 0- Partir d'un ensemble \mathcal{S}_0 de variables tel que la formulation PL (\tilde{P}_0) restreinte à \mathcal{S}_0 ait une solution.
 $\mathcal{S}^* \leftarrow \mathcal{S}_0$
- 1- Résoudre le PL compact (\tilde{P}^*) restreint aux variables de \mathcal{S}^* : soit t^* la solution optimale de (\tilde{P}^*) et λ_k^* le coût dual associé à l'inégalité $k \in \mathcal{K}$
- 2- Tester s'il existe une variable t_s dans $\mathcal{S} \setminus \mathcal{S}^*$ telle que son **coût réduit** soit strictement négatif (c'est le problème de pricing) :

$$\bar{c}(s) = c(s) - \sum_{k \in \mathcal{K}} \chi_k^s \lambda_k^*$$

- 3- S'il n'en existe pas alors STOP : t^* est solution fractionnaire optimale pour (\tilde{P})
- 4- Sinon $\mathcal{S}^* \leftarrow \mathcal{S}^* \cup \{t_s\}$ et GOTO 1

38/69

Algorithme de Branch-and-Price

Heuristique primale :

Les formulations menant à des génération de colonnes produisent en général de bonnes solutions entières :

- par arrondi des valeurs fractionnaires
- par "stop-and-solve" : on peut par exemple stopper le déroulement de la génération de colonnes à une itération quelconque et résoudre par Branch-and-Bound la formulation restreinte compacte obtenues en cours de route
- par analyse "statistique" des colonnes générées les plus utilisées dans une solution optimales fractionnaires : dans l'exemple du multiflot c'est repérer les arêtes les plus utilisées par un flot pour construire de manière gloutonne une solution
- ...

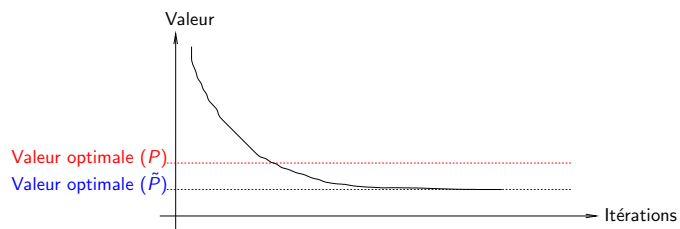
40/69

Algorithme de Branch-and-Price

Une difficulté : obtenir une borne inférieure rapidement

Une difficulté qui n'apparaît pas en génération d'inégalités (Branch-and-Cut) est que tant que la convergence de la génération de colonnes n'a pas été obtenue, la méthode ne produit pas de borne inférieure :

En effet, regardons ce schéma où la courbe suit l'amélioration de la valeur de la fonction objective pendant une génération de colonnes : on voit que la courbe tend peu à peu vers la solution optimale fractionnaire de la relaxation linéaire (\tilde{P}).



Comme la valeur optimale de (P) est au-dessus de la valeur optimale de \tilde{P} , on ne peut savoir qu'on a obtenu une borne inférieure qu'à la convergence.

41/69

Outils et mise en œuvre

La mise en œuvre des méthodes de Branch-and-Price est assez complexe :

- gestion des variables qui, même en nombre polynomial, deviennent trop importantes pour la résolution des relaxation (suppression en fonction d'une évaluation de leur activité)
- convergence de la génération de colonnes
- utilisation d'un ou plusieurs pricer pour ne pas systématiquement utiliser un pricer trop lent (utilisation de pricer heuristique)
- réaliser des compromis entre résoudre la relaxation linéaire et accélérer le branchement
- règle et stratégie de branchement efficaces...

Si l'on désire résoudre un PLNE avec un nombre exponentiel de variables, il existe des framework

- commerciaux : Concert technology de Cplex-IBM ou Gurobi.
- universitaires : SCIP de la ZIB, BCP de COIN-OR, Symphony de COIN-OR
- un produit universitaire BapCode de l'Université de Bordeaux tente même de mettre en œuvre automatiquement une méthode de décomposition (dite de Dantzig-Wolfe).
- on peut avoir besoin de gérer des formulations avec un nombre exponentiel de variables et de contraintes ! Il faut donc un algorithme de **Branch-and-cut-and-Price (BCP)** : il n'existe là pratiquement que les outils universitaires SCIP de la ZIB et BCP de COIN-OR.

43/69

Algorithme de Branch-and-Price

Une difficulté : obtenir une borne inférieure rapidement

Or cette convergence est parfois très lente.

Il existe par contre une façon d'obtenir une borne inférieure pour la valeur objective de (P).

Notons :

- z_P la valeur optimale d'une solution entière de (P)
- $z_{(P)}$ la valeur optimale de la relaxation (\tilde{P})
- z_{iter} la valeur obtenue pour une itération quelconque
- λ_{min}^* la plus petite valeur parmi les k coûts duaux λ_k^* , $k \in \mathcal{K}$
- et B une borne supérieure telle que $B \geq \sum_{s \in \mathcal{S}} t_s$

(Si $t_s \in \{0, 1\}$, la borne B est le nombre maximum de variables dans une solution).

Alors on a

$$z_{iter} + B\lambda_{min}^* \leq z_{\tilde{P}} \leq z_P$$

En effet, par le théorème faible de la dualité, la solution z_{iter} plus que B fois la valeur du plus petit coût dual (qui est négatif) est une borne inférieure de la valeur optimale.

42/69

1. Principe de la génération de colonnes sur un exemple

2. Définitions et notations

3. Décomposition de Dantzig-Wolfe

3.1 Principe général

3.2 Qualité de la relaxation de Dantzig-Wolfe

4. Application au problème de coloration

44/69

1. Principe de la génération de colonnes sur un exemple

2. Définitions et notations

3. Décomposition de Dantzig-Wolfe

3.1 Principe général

3.2 Qualité de la relaxation de Dantzig-Wolfe

4. Application au problème de coloration

Principe général

Soit $c \in \mathbb{R}^n$ un vecteur-colonne, $A \in \mathbb{R}^{n \times m}$, $a \in \mathbb{R}^m$, $B \in \mathbb{R}^{n \times m'}$ et $b \in \mathbb{R}^{m'}$. On considère le PLNE (P) suivant

$$(P) \begin{cases} \text{Min } c^T x \\ Ax \geq a \\ Bx \geq b \\ x \in \mathbb{N}^n \end{cases}$$

où :

- les inégalité $Ax \geq a$ sont "couplantes", c'est-à-dire qu'elles concernent une bonne partie des variables
- les $Bx \geq b$ sont "décomposables".

On dit que l'ensemble d'inégalités $Bx \geq b$ est **décomposable** si le problème

$$(SP) \begin{cases} \text{Min } d^T x \\ Bx \geq b \\ x \in \mathbb{N}^n \end{cases}$$

(avec d ici un poids quelconque) est la juxtaposition de plusieurs problèmes sur des espaces de variables indépendantes.

Principe général

Le problème (SP) peut être reformulé différemment en utilisant une description dite "par les solutions".

On considère pour cela l'ensemble Q des solutions de (SP) : c'est-à-dire tous les points entiers du polyèdre $Bx \geq b$ (cet ensemble est en général exponentiel sur la taille du problème!)

$$Q = \{x \in \mathbb{N}^n \mid Bx \geq b\}$$

Supposons ici que Q soit fini, c'est-à-dire que $\text{conv}(Q)$ soit un polytope : on peut alors noter $Q = \{\chi^1, \dots, \chi^{|Q|}\}$ ces solutions qui sont donc des vecteurs entiers.

Remarquons qu'alors les solutions de Q sont données par

$$\sum_{q=1}^{|Q|} \chi^q t_q \quad \text{avec} \quad \sum_{q=1}^{|Q|} t_q = 1$$

où t_q sont des variables binaires.

Principe général

Déterminer une solution de (SP) revient donc à choisir un vecteur χ dans Q , en d'autre terme cela revient à réécrire (SP) de la façon suivante :

$$(SP) \begin{cases} \text{Min } d^T x \\ x = \sum_{q=1}^{|Q|} \chi^q t_q \\ \sum_{q=1}^{|Q|} t_q = 1 \\ t_q \in \{0, 1\} \quad \forall q \in \{1, \dots, |Q|\}. \end{cases}$$

Cette écriture "naïve" revient à énumérer toutes les solutions et donner à x le vecteur la meilleure d'entre elles : on peut noter que les variables t sont ici des valeurs binaires permettant de définir quels vecteurs choisir dans Q .

Principe général

Un peu loin, nous verrons que l'idée de la relaxation de Dantzig-Wolfe est de voir que l'on peut prendre des variables t réelles pour définir (SP) .

$$(SP) \left\{ \begin{array}{l} \text{Min } d^T x \\ x = \sum_{q=1}^{|Q|} \chi^q t_q \\ \sum_{q=1}^{|Q|} t_q = 1 \\ t_q \geq 0 \quad \forall q \in \{1, \dots, |Q|\}. \end{array} \right.$$

En effet, le polyèdre associé est naturellement entier car chacun de ses points extrêmes correspond à exactement à un élément de Q (c'est-à-dire une solution de (SP)).

49/69

Principe général

Utilisons cette reformulation de (SP) dans (P) :

$$(P) \left\{ \begin{array}{l} \text{Min } c^T x \\ Ax \geq a \\ x = \sum_{q=1}^{|Q|} \chi^q t_q \\ \sum_{q=1}^{|Q|} t_q = 1 \\ t_q \in \{0, 1\} \quad \forall q \in \{1, \dots, |Q|\}. \end{array} \right.$$

Et faisons alors disparaître x !

$$(P) \left\{ \begin{array}{l} \text{Min } \sum_{q=1}^{|Q|} (c^T \chi^q) t_q \\ \sum_{q=1}^{|Q|} (A \chi^q) t_q \geq a \\ \sum_{q=1}^{|Q|} t_q = 1 \\ t_q \in \{0, 1\} \quad \forall q \in \{1, \dots, |Q|\}. \end{array} \right.$$

50/69

Principe général

On définit la **Relaxation de Dantzig-Wolfe** comme la relaxation fractionnaire (MP) de (P) ainsi définie :

$$(MP) \left\{ \begin{array}{l} \text{Min } \sum_{q=1}^{|Q|} (c^T \chi^q) t_q \\ \sum_{q=1}^{|Q|} (A \chi^q) t_q \geq a \\ \sum_{q=1}^{|Q|} t_q = 1 \\ t_q \geq 0 \quad \forall q \in \{1, \dots, |Q|\}. \end{array} \right.$$

Cette relaxation n'est pas la relaxation linéaire classique de (P) .

Il est important de remarquer que (MP) , appelé *problème maître*, est un PL mais où le fait que les solutions de (SP) soit entière a été en partie pris en compte.

51/69

Principe général

La formulation (MP) contient par construction un nombre exponentiel de variables autant que de solutions dans Q !

Afin de résoudre (MP) , on peut alors mettre en place un algorithme de génération de colonnes.

En posant π le vecteur dual des inégalités provenant de $Ax \geq a$ et ρ la valeur duale provenant de la contrainte d'égalité, le problème de pricing revient à rechercher s'il existe une solution de $\chi^q \in Q$ telle que son coût réduit est négatif, i.e.,

$$(c - \pi^T A) \chi^q - \rho < 0$$

Ainsi le pricer revient à résoudre le problème (SP) avec le coût $d = (c - \pi^T A)$: si ce poids est négatif strictement, on ajoute une colonne χ^q à Q . Et on réitère cela jusqu'à ce que (SP) renvoie une plus petite solution de coût réduit positif : ce qui signifie que la solution du master restreint est une solution optimale de (MP) .

52/69

Principe général

Le pricer peut donc se résoudre par PLNE en utilisant :

$$(SP) \begin{cases} \text{Min } (c - \pi^T A)^T x \\ Bx \geq b \\ x \in \mathbf{N}^n \end{cases}$$

On peut alors voir (SP) comme le sous-problème de la méthode de Dantzig-Wolfe.

Bien entendu, il est tout à fait possible de résoudre le problème de pricing par d'autres moyens algorithmiques, surtout s'ils sont plus efficaces.

Relaxation de Dantzig-Wolfe

Comme l'on cherche à résoudre un PLNE (P) , cette génération de colonnes ne mène qu'à une relaxation fractionnaire (MP) de ce PLNE, dite **relaxation de Dantzig-Wolfe**.

En général, cette relaxation (MP) est meilleure que la relaxation linéaire du PLNE (P) !

De nombreuses bornes ont été fournies par cette méthode pour des problèmes d'optimisation combinatoire.

De plus un algorithme de Branch-and-Price basée sur cette décomposition de Dantzig-Wolfe est souvent efficace.

1. Principe de la génération de colonnes sur un exemple

2. Définitions et notations

3. Décomposition de Dantzig-Wolfe

3.1 Principe général

3.2 Qualité de la relaxation de Dantzig-Wolfe

4. Application au problème de coloration

Amélioration de la relaxation linéaire ?

En fait, si la relaxation (MP) de Dantzig-Wolfe est plus intéressante en général que la relaxation linéaire de (P) , il existe de nombreux cas où les deux relaxations sont identiques !

C'est le cas dans ce cadre très fréquent :

Theorem

Si (SP) est entièrement décrit par $\text{conv}(x \in \mathbf{R}^+ \mid Bx \geq b)$, c'est-à-dire si cette enveloppe convexe est entière, alors les deux relaxations linéaires et de Dantzig-Wolfe sont identiques.

Ce résultat se déduit de l'écriture de (P) où l'on peut remplacer (SP) par un polyèdre dont chacun des sommets est une de ses solutions.

On peut interpréter ce résultat en disant que, dans ce cas, la décomposition de Dantzig-Wolfe sur (SP) n'apporte pas rien sur l'intégrité de (SP) .

Amélioration de la relaxation linéaire ?

A partir du résultat précédent, on peut noter que, dans tous les cas où le problème de pricing d'une génération de colonnes est polynomial, alors la relaxation linéaire de la génération de colonnes est la même qu'une version non décomposée.

Dans le cas de la relaxation du multiflot entier, ce sera bien la même valeurs !

Par contre, lorsque le pricer est NP-difficile, alors on constate le plus souvent une forte amélioration !

Dantzig-Wolfe versus décompo lagrangienne

On définit le dual lagrangien de la manière suivante

$$(LDP) \left\{ \begin{array}{l} \max_{\gamma \in R_+^{|a|}} \min_{x \in Q} c^T x - \gamma^T (Ax - a) \end{array} \right.$$

On peut linéariser le min en utilisant Q l'ensemble des solutions de (SP).

$$(LDP) \left\{ \begin{array}{l} \max \eta \\ \eta \leq c^T x^q - \gamma^T (Ax^q - a) \quad \forall q \in \{1, \dots, |Q|\} \\ \gamma \in R_+^{|a|} \\ \eta \in R \end{array} \right.$$

En notant λ_q les variables duales associées aux contraintes du dual lagrangien, on peut écrire son dual de la façon suivante (en ajoutant sans perte de généralité la normalisation des valeurs λ .)

$$(DLDP) \left\{ \begin{array}{l} \min \sum_{q=1}^{|Q|} (c^T x^q) \lambda_q \\ \sum_{q=1}^{|Q|} (Ax^q) \lambda_q \geq a \\ \sum_{q=1}^{|Q|} \lambda_q = 1 \\ \lambda_q \geq 0 \quad \forall q \in \{1, \dots, |Q|\} \end{array} \right.$$

On obtient la relaxation (MP) dans l'espace de variable correspondant à la décomposition de Dantzig-Wolfe. Donc le dual de la formulation par décomposition de Dantzig-Wolfe correspond au dual lagrangien. \square

Dantzig-Wolfe versus décompo lagrangienne

On peut se demander quelle relaxation est la plus intéressante entre la relaxation de Dantzig-Wolfe et la relaxation lagrangienne.

Theorem

Les relaxations de Dantzig-Wolfe et la relaxation lagrangienne ont même valeurs.

Preuve : En effet, la relaxation lagrangienne de (P) est

$$(LRP) \left\{ \begin{array}{l} \min c^T x - \gamma^T (Ax - a) \\ x = \sum_{q=1}^{|Q|} \chi^q t_q \\ \sum_{q=1}^{|Q|} t_q = 1 \\ t_q \in \{0, 1\} \quad \forall q \in \{1, \dots, |Q|\} \end{array} \right.$$

avec γ le vecteur colonne optimal des multiplicateurs de Lagrange.

1. Principe de la génération de colonnes sur un exemple

2. Définitions et notations

3. Décomposition de Dantzig-Wolfe

4. Application au problème de coloration

Problème de coloration

Soit $S \subseteq N$. Une *coloration* des sommets d'un graphe $G = (V, E)$ est une fonction $r : V \rightarrow S$ telle que $r(u) \neq r(v)$ pour tout couple de sommets adjacents u, v . Les éléments de l'ensemble S sont appelés les *couleurs* disponibles. Une *k-coloration* est une coloration $c : V \rightarrow \{1, \dots, k\}$. Un graphe est dit *k-coloriable* s'il possède une *k-coloration*.

Tester si un graphe est *k-coloriable* est un problème NP-complet si $k \geq 3$, et il est polynomial si $k = 2$. En effet, si $k = 2$, il suffit de tester si le graphe est biparti, c'est-à-dire s'il ne contient pas des cycle impair. Ce qui peut se faire par un simple parcours de graphe.

Soit $G = (V, E)$ un graphe non-orienté. Le *problème de coloration* consiste à déterminer le plus petit k tel que G soit *k-coloriable*.

Première formulation

On associe à chaque sommet u de V un vecteur binaire à K dimensions $x_u = (x_u^1, \dots, x_u^K)$, où K est une borne supérieure sur la coloration de G (au maximum $K = |V|$).

Comme l'on désire minimiser le nombre de couleur, on ajoute une variable binaire w_l par couleur $l = 1, \dots, K$ indiquant si cette couleur a été utilisée ou non.

Le problème est donc équivalent au programme

$$(P_C) \begin{cases} \text{Min} \sum_{l=1}^K w_l \\ \sum_{l=1}^K x_u^l = 1 & \forall u \in V, \\ x_u^l + x_v^l \leq w_l & \forall e = uv \in E \text{ et } 1 \leq l \leq K, \\ x_u^l \in \{0, 1\} & \forall u \in V \text{ et } 1 \leq l \leq K, \\ w_l \in \{0, 1\} & \forall 1 \leq l \leq K. \end{cases}$$

Les premières contraintes sont celles couplante ($Ax \leq a$) et les suivantes celles décomposables par "couleur l " ($Bx \geq b$).

61/69

62/69

Décomposition

Ainsi le sous-problème est

$$(SP_C) \begin{cases} \text{Min} \sum_{l=1}^K (d_{w_l} w_l + d_{x^l} x^l) \\ x_u^l + x_v^l \leq w_l & \forall e = uv \in E \text{ et } 1 \leq l \leq K \\ x_u^l \in \{0, 1\} & \forall u \in V \text{ et } 1 \leq l \leq K \\ w_l \in \{0, 1\} & \forall 1 \leq l \leq K \end{cases}$$

qui est en fait la somme des PLNE pour chaque couleur $l \in \{1, \dots, K\}$

$$(SP_C^l) \begin{cases} \text{Min} d_{w_l} w_l + d_{x^l} x^l \\ x_u^l + x_v^l \leq w_l & \forall e = uv \in E \\ x_u^l \in \{0, 1\} & \forall u \in V \\ w_l \in \{0, 1\}. \end{cases}$$

Le problème est donc décomposé en K sous-problèmes (SP_C)

Décomposition

On peut voir que le problème (SP_C) consiste donc à déterminer K stables !

Un vecteur solution appartenant à l'ensemble Q des solutions est donc un vecteur représentant K stables et pour chaque stable l un entier w_l indiquant si ce stable l est ou non l'ensemble vide.

Posons alors \mathcal{S} l'ensemble des stables G et $\chi^{\mathcal{S}}$ l'ensemble des vecteurs d'incidence de ces stables. Alors Q peut être défini comme un vecteur composé de toutes les combinaisons de K vecteurs $\chi^S \in \chi^{\mathcal{S}}$ chapeauté chacun de 0 si $S = \emptyset$ et 1 sinon.

Comme (SP_C) est la somme de K (SP_C^l) , on peut écrire plus simplement

$$(SP_C) \begin{cases} \text{Min} d_{w_l} w_l + d_{x^l} x^l \\ x_u^l + x_v^l \leq w_l & \forall e = uv \in E \\ x^l = \sum_{s \in \mathcal{S}} \chi^s t_s^l \\ \sum_{s \in \mathcal{S}} t_s^l = 1 \\ t_s^l \in \{0, 1\} & \forall s \in \mathcal{S}. \end{cases}$$

63/69

64/69

Décomposition

On peut remarquer que si $d_{w_l} > 0$ et pour une solution (entière) de (SP_C^l) , on a en fait $w_l = \sum_{s \in \mathcal{S}} t_s^l$. En effet, la contrainte sur les arêtes fait que seul un des terme de la somme est non nul. Ainsi, on peut prouver que les deux valeurs sont identiques. On peut alors écrire

$$(SP_C^l) \left\{ \begin{array}{l} \text{Min } d_{w_l} \sum_{s \in \mathcal{S}} t_s^l + d_{x^l} x^l \\ x_u^l + x_v^l \leq w_l \\ x^l = \sum_{s \in \mathcal{S}} \chi^s t_s^l \\ \sum_{s \in \mathcal{S}} t_s^l = 1 \\ t_s^l \in \{0, 1\} \end{array} \right. \quad \begin{array}{l} \forall e = uv \in E \\ \\ \\ \\ \forall s \in \mathcal{S}. \end{array}$$

65/69

Décomposition

En utilisant cette écriture de (SP) , on peut réécrire (P_C) de la manière suivante

$$(P_C) \left\{ \begin{array}{l} \text{Min } \sum_{l=1}^K \sum_{s \in \mathcal{S}} t_s^l \\ \sum_{l=1}^K x_u^l = 1 \quad \forall u \in V, \\ x^l = \sum_{s \in \mathcal{S}} \chi^s t_s^l \\ \sum_{s \in \mathcal{S}} t_s^l = 1 \\ t_s^l \in \{0, 1\} \quad \forall s \in \mathcal{S} \text{ et } 1 \leq l \leq K \end{array} \right.$$

66/69

Décomposition

On peut noter ici que toutes les variables et vecteurs ne dépendent pas de l ... On peut donc réécrire de la façon suivante

$$(P_C) \left\{ \begin{array}{l} \text{Min } \sum_{s \in \mathcal{S}} t_s \\ \sum_{s \in \mathcal{S}} \chi_u^s t_s = 1 \quad \forall u \in V, \\ t_s \in \{0, 1\} \quad \forall s \in \mathcal{S} \end{array} \right.$$

Cette écriture est une formulation, bien classique et bien compréhensible pour le problème de coloration !

$$(P_C) \left\{ \begin{array}{l} \text{Min } \sum_{s \in \mathcal{S}} t_s \\ \sum_{s \in \mathcal{S} \mid u \in s} \chi_u^s t_s = 1 \quad \forall u \in V, \\ t_s \in \{0, 1\} \quad \forall s \in \mathcal{S} \end{array} \right.$$

67/69

Génération de colonnes pour la coloration

On peut alors considérer la relaxation linéaire (MP_C) de ce programme

$$(MP_C) \left\{ \begin{array}{l} \text{Min } \sum_{s \in \mathcal{S}} t_s \\ \sum_{s \in \mathcal{S}} \chi_u^s t_s = 1 \quad \forall u \in V, \\ t_s \geq 0 \quad \forall s \in \mathcal{S} \end{array} \right.$$

Considérons un sous-ensemble \mathcal{S}^* de stables tels qu'il existe une coloration avec ces stables (on peut aisément obtenir une telle initialisation).

Notons (MP_C^*) la relaxation linéaire de (P_C) restreinte à \mathcal{S}^* . Notons t^* la solution optimale t^* de ce programme restreint (MP_C^*) et π_v les coûts duaux associés à ses inégalités.

Alors t^* est la solution optimale de (MP_C) si et seulement si

$$(1 - \sum_{v \in V} \pi_v \chi_v^s) \geq 0 \quad \forall s \in \mathcal{S}$$

c'est-à-dire s'il n'existe pas de stable s tel que $\sum_{v \in s} \pi_v < 1$.

68/69

Génération de colonnes pour la coloration

Le problème de pricing de génération de colonnes revient ainsi à rechercher un stable de poids maximum selon les coûts λ_v , $v \in V$. Si ce stable maximal est plus grand stable est de poids < 1 alors on a trouvé un colonne à ajouter à (MP) sinon on a atteint la relaxation linéaire de (MP) .

La borne obtenue est une des meilleures bornes inférieures connues pour le problème de coloration.

De plus, en utilisant cette génération de colonnes dans un algorithme de Branch-and-Price, on a pu obtenir des solutions optimales (entières) pour de nombreuses instances difficile.

En effet, le principe de branchement pour la formulation (MP) est assez simple à mettre en œuvre : on peut utiliser la règle de Ryan-And-Foster qui propose de créer deux fils à partir d'un couple de sommet (u, v) non adjacent du graphe : un fils où u et v ont même couleur et un fils où u et v sont de couleurs différentes. L'astuce de mise en œuvre repose sur le fait que pour un fils, on identifie les deux sommets dans le graphe et pour l'autre, on ajoute une arête : ainsi à chaque profondeur de l'arbre, le graphe étudié est simplifié.