UNIVERSITÉ DU HAVRE - UFR DES SCIENCES ET TECHNIQUES

Cours d'Optimisation Combinatoire et Métaheuristiques

Approches avancées de résolution exacte - Partie 3

Cédric Joncour

Bureau B112 (aile B, 1er étage)
cedric.joncour@univ-lehavre.fr
http://lmah.univ-lehavre.fr/~joncour

Master 2 - Année 2013 - Semestre 2

UNIVERSITÉ DU HAVRE - UFR DES SCIENCES ET TECHNIQUES

Cours d'Optimisation Combinatoire et Métaheuristiques

Approches avancées de résolution exacte - Partie 3

Cédric Joncour

Bureau B112 (aile B, 1er étage)
cedric.joncour@univ-lehavre.fr
http://lmah.univ-lehavre.fr/~joncour

Master 2 - Année 2013 - Semestre 2

Références bibliographiques

- *Recherche Opérationnelle - Tome 1*
  de Jacques Teghem
- *Programmation mathématique*
  de Michel Minoux
- *Optimisation combinatoire 1-5*
  de Vangelis Th. Paschos
- *Optimisation discrète*
  d'Alain Billionnet
- *Combinatorial Optimization : Theory and applications (traduction)*
  de Bernhard Korte et Jens Vygen
- *Integer Programming*
  de Laurence A. Wolsey
- *Integer and Combinatorial Programming*
  de George L. Nemhauser et Laurence A. Wolsey
- *Computers and intractibility : a guide to the theory of NP-completeness*
  de Michael R. Garey et David S. Johnson
- *Linear Programming : Foundations and extensions*
  de Robert J. Vanderbei
- Pour plus d'infos... `en.wikipedia.org` & `www.roadef.org/`
- *Livre blanc de la recherche opérationnelle* édité par la ROADEF

Le contexte de l'optimisation 1/2

Définition de l'optimisation

La science cherchant *à analyser et à résoudre* analytiquement ou numériquement les *problèmes* qui consistent à déterminer le *meilleur élément* d'un ensemble, au sens d'un *critère quantitatif* donné

Un problème : Question que l'on se pose liée à une situation ?

Analyser et résoudre : Comprendre le problème puis proposer une ou des solutions pour répondre au problème

Critère quantitatif : Toutes les solutions n'ont pas la même importance

Meilleur élément : Chercher la solution la plus intéressante

Domaines d'applications

- Civil (services publics, hopitaux, transport public, informatique)
- Industriel (automobile, aviation, énergie, télécom, production)
- Financier (gestion de portefeuille)
- Militaire (gestion des ressources, logistique)

Le contexte de l’optimisation 2/2

L’optimisation et la recherche opérationnelle

- Prendre des décisions : choix à faire/décider
  - organiser un plan, contrôler les opérations
  - allouer des ressources, ranger des tâches
  - définir des chemins ou des tournées
- En optimisant un critère (ou objectif) : but cherché du problème
- Et en respectant des contraintes : restrictions liées au problème

Quelques exemples simples

- Organiser sa journée de travail pour finir à temps chaque tâche
- Plannifier ses dépenses pour répondre à ses principaux besoins
- Trouver le plus court chemin entre son appartement et l’université

But recherché de l’optimisation

Trouver la meilleure solution réalisable pour un problème donné

Terminologie

Problème d'optimisation : problème de la forme

$$\min / \max\{f(x) : x \in \mathcal{S}\}$$

Variable de décision : élément $x \in \mathbb{R}^n$ représentant l'inconnu

Région réalisable : ensemble des solutions (affectation de valeur à x) vérifiant les contraintes du problème $\leadsto$ ensemble $\mathcal{S}$

Solution réalisable : solution vérifiant les contraintes du problème

$$\hat{x} \text{ est réalisable} \quad \Leftrightarrow \quad \hat{x} \in \mathcal{S}$$

Critère à optimiser : fonction objectif $f : \mathcal{S} \to \mathbb{R}$

Solution optimale : meilleur solution réalisable selon l'objectif f

$$x^* \in \mathcal{S} \text{ est optimale} \quad \Leftrightarrow \quad f(x^*) \leq f(\hat{x}), \quad \forall \hat{x} \in \mathcal{S}$$

But recherché de l'optimisation

Etant donné $f : \mathcal{S} \to \mathbb{R}$, trouver $x^* \in \mathcal{S}$ tel que $f(x^*) \leq f(x), \quad \forall x \in \mathcal{S}$

En pratique, la solution x^* peut s'avérer difficile à trouver

Objectif du cours

Etude de problèmes combinatoires classiques

- Problème du sac-à-dos et du placement de boîtes
- Problème du plus court chemin et du voyageur de commerce
- Problème de coloration de graphe

Méthodes avancées de résolution exacte

- Notion de qualité de formulation
- Résolution des problèmes de grand taille (variables/contraintes)
- Méthodes de décomposition des problèmes
- Approche polyèdrale

Quelques algorithmes dédiés (si on a le temps)

- Classification des problèmes
- Problèmes d'ordonnancement

Sommaire du cours

Première partie I

TECHNIQUES DE MODÉLISATION

Graphes non orientés

Graphe non orienté

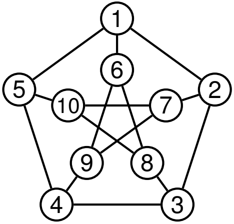
Un graphe non orienté G est un couple (V, E) tel que

$$E \subseteq \{X \subseteq V : |X| = 2\}$$

- V : ensemble des sommets
- E : ensemble des arêtes

Une arête est une paire de sommets :

$$\{u, v\} \in E \quad \text{avec} \quad u, v \in V$$



Soit le graphe non orienté $G = (V, E)$ avec
$V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

$$E = \left\{\begin{array}{l} \{1,2\}, \{2,3\}, \{3,4\}, \{4,5\}, \{5,1\} \\ \{1,6\}, \{2,7\}, \{3,8\}, \{4,9\}, \{5,10\} \\ \{6,8\}, \{6,9\}, \{7,9\}, \{7,10\}, \{8,10\} \end{array}\right\}$$

Graphes orientés

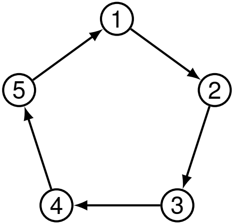
Graphe orienté

Un graphe orienté G est un couple (V, A) tel que $A \subseteq V \times V$

- V : ensemble de sommets
- A : ensemble d'arcs

Chaque arc est un couple de sommets :

$$(u, v) \in A \quad \text{avec} \quad u, v \in V$$



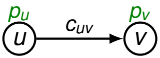
Soit le graphe orienté $G = (V, A)$ avec
$V = \{1, 2, 3, 4, 5\}$
$A = \Big\{(1, 2), (2, 3), (3, 4), (4, 5), (5, 1)\Big\}$

Graphes pondérés

Graphe pondéré

Un graphe pondéré est un graphe auquel on a associé une (ou plusieurs) fonction de valuation

Un graphe peut être pondéré/valué sur ses sommets et/ou ses arêtes



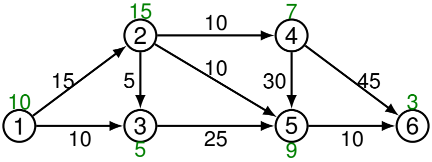


Schéma d’étude d’un problème d’optimisation

- Quelles sont les données du problème ?
  récolter les données du problème, comprendre le problème,...
- Comment modéliser le problème ?
  - Quelles décisions doit-on prendre ?
    sélectionner/placer des objets, définir un ordre ou une quantité, choisir un évènement, effectuer une opération particulière...
  - Quel est l’objectif recherché ?
    maximiser un profit, minimiser des coûts ou une quantité,...
  - Quels sont les contraintes du problème ?
    respecter des capacités ou des contraintes de précédence,...
- Quelle est la complexité de ce problème ?
- Comment résoudre le problème ?
  concevoir des algorithmes (exacts vs approchés) donnant des solutions réalisables/optimales, développer des méthodes alternatives ou hybrides,...

Qu'est ce qu'un problème d'optimisation combinatoire ?

L'optimisation combinatoire/discrète

Soit $N = \{e_1, \ldots, e_n\}$ un ensemble fini discret avec :
c_j un coût de l'élément $j \in N$
Soit $\mathcal{S}$ un ensemble de sous-ensembles réalisables de N

Trouver une solution X de $\mathcal{S}$ de coût minimum/profit maximum :

$$\min_{X \subseteq N} / \max_{X \subseteq N} \left\{ c(X) : X \in \mathcal{S} \right\} \qquad \text{avec} \quad c(X) = \sum_{j \in X} c_j \text{ fonction de coût}$$

- N : ensemble des décisions possibles
- $\mathcal{S}$: ensemble des solutions réalisables (ensemble de décisions)

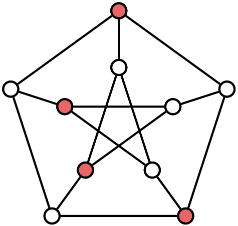
Comme l'espace des candidats solutions est discret fini, il est possible

- d'énumérer chacune des solutions réalisables
- de vérifier qu'une solution est réalisable
- de calculer le coût d'une solution réalisable
- de conserver la meilleure solution après énumération exhaustive

Le problème de la recherche d'ensemble stable

Problème d'ensemble stable de poids max (max weight independant set)

Trouver un sous ensemble de sommets S d'un graphe, de poids max, tel que chaque sommet de S n'est relié à aucun autre sommet de S



Applications

- Stable de cardinalité maximale
- Allocation de fréquences
- Gestion de conflit
- Coloration de graphe

Données : $G = (V, E)$ graphe non orienté avec
w_i poids du sommet $i \in V$

Objectif : sélectionner des sommets du graphe dont le poids total est maximale

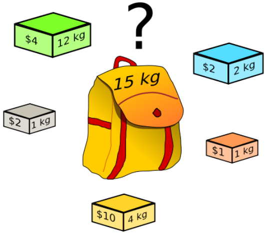
Contraintes : choisir un seul sommet entre deux sommets adjacents

Le problème de sac-à-dos

Problème du sac-à-dos (knapsack problem)

Choisir un sous-ensemble d'objets d'utilité maximale à placer dans un sac tel que la capacité du sac soit respectée

source : wikipédia - auteur : Dake



Applications

- Gestion de portefeuille
- Découpe de matériaux
- Chargement de camions
- Placement de boîtes

Données : W : capacité maximale du sac
I : ensemble de n objets avec, pour chaque objet $i \in I$,
p_i l'utilité et w_i le poids

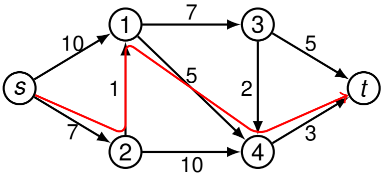
Objectif : choisir des objets afin de maximiser l'utilité totale

Contraintes : ne pas dépasser la capacité du sac

Le problème du plus court chemin

Problème du plus court chemin (shortest path problem)

Trouver un chemin de longueur minimale entre deux villes données : une ville de départ (source *s*) et une ville destination (puits *t*)



Applications

- Route la moins coûteuse
- Route la rapide dans un réseau
- Transfert de ressource/information
- Problème de flot de coût minimum

Données : $G = (V, A)$ graphe orienté avec deux sommets s et t, et c_{ij} longueur de l'arc $(i, j) \in A$

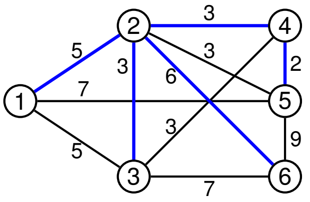
Objectif : choisir des arêtes en minimisant la longueur totale

Contraintes : former un chemin entre s et t

Le problème d'arbre couvrant de poids minimum

Problème de l'arbre couvrant de poids minimum (min spanning tree)

Choisir un sous-ensemble d'arêtes d'un graphe de coût minimum tel que ce sous-ensemble forme un arbre couvrant du graphe



Applications

- Construction d'un réseau
- Installation d'un pipeline
- Connection électrique
- Recherche d'un arbre de Steiner

Données : $G = (V, E)$ graphe non orienté, connexe avec
p_e poids de l'arête $e \in E$

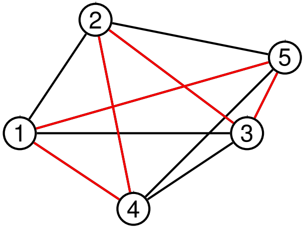
Objectif : sélectionner des arêtes afin de minimiser le poids total

Contraintes : former un arbre à l'aide des arêtes
couvrir tous les sommets à l'aide d'un arbre

Le problème du voyageur de commerce

Problème du voyageur de commerce (travelling salesman problem)

Trouver un cycle hamiltonien dans un graphe de longueur minimale : chaque sommet du graphe est visité une et une seule fois



Applications

- Tournées de véhicules
- Connection de composants dans un circuit intégré
- Ordonnancement des tâches avec de changement de configuration

Données : $G = (V, E)$ graphe non orienté avec
c_e longueur de l'arête $e \in E$

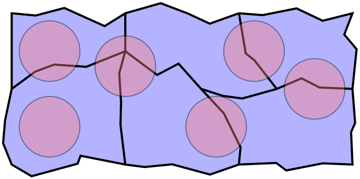
Objectif : choisir des arêtes afin de minimiser la longueur totale

Contraintes : former une tournée à l'aide des arêtes
visiter chaque ville une et une seule fois avec un tour

Le problème de recouvrement

Problème de recouvrement (covering problem)

Couvrir à coût minimum un ensemble de secteurs à l'aide d'un sous-ensemble de sites potentiellement utilisables



Applications

- Localisation d'entrepôts/antennes
- Gestion d'emploi du temps
- Plannification/ordonnancement
- Coloration de graphe
- Tournées de véhicule

Données : $N = \{1, \ldots, n\}$: ensemble de secteurs à couvrir
M : ensemble de sites potentiels avec, pour tout $i \in N$,
c_i le coût d'installation du site i
et $E_i \subseteq N$ le sous ensemble de secteurs couvert par i

Problème : choisir des éléments de M pour couvrir N à moindre coût

Difficulté des problèmes d'optimisation combinatoire

Problème du voyageur de commerce

Trouver un tour de coût minimal tel que chaque client est visité une et une seule fois

Représentation : chaque permutation des indices définit un tour :

$$\pi = (\pi_1, \pi_2, \ldots, \pi_k, \ldots, \pi_n)$$

où $\pi_k = i$ indique que la ville i est visitée en k^{e} position
Le coût associé à une solution π est :

$$\text{coût}(\pi) = c_{\pi_1\pi_2} + c_{\pi_2\pi_3} + \cdots + c_{\pi_{n-1}\pi_n} + c_{\pi_n\pi_1}$$

Nombre de solutions possibles

Si le graphe est complet, le nombre de tours possibles est $\frac{(n-1)!}{2}$

n	10	20	100	1000
$n!$	$> 10^6$	$> 10^{18}$	$> 10^{157}$	$> 10^{2567}$

On ne peut pas énumérer toutes les possibilités en pratique !

Modélisation par la programmation mathématique

Problème d'optimisation combinatoire

$$\min_{X \subseteq N} \left\{ \sum_{j \in X} c_j : X \in \mathcal{S} \right\}$$

Posons $x \in \{0,1\}^{|N|}$ tel que

$$x_j = \begin{cases} 1, & \text{si l'élément } j \text{ est choisi} \\ 0, & \text{sinon} \end{cases} \qquad \forall j \in N$$

Alors le problème se formule

$$\min \left\{ \sum_{j \in N} c_j x_j : x \in P \cap \{0,1\}^{|N|} \right\}$$

où P espace formé par les contraintes exprimant les propriétés à satisfaire pour être dans l'ensemble $\mathcal{S}$:

$$x \in P \cap \{0,1\}^{|N|} \Leftrightarrow X \in \mathcal{S}$$

$P \cap \{0,1\}^{|N|}$: ensemble des solutions réalisables discret et fini

Les points de $x^s \in P \cap \{0,1\}^{|N|}$ sont les vecteurs d'incidence de la

Modélisation du problème du stable de poids maximum

Données : $G = (V, E)$ graphe non orienté avec
w_i poids du sommet $i \in V$

Problème combinatoire : $\max_{X \subseteq V} \left\{ \sum_{i \in X} w_i : X \text{ est un stable} \right\}$

Variables : $x_i = \begin{cases} 1 \text{ si le sommet } i \text{ est choisi} \\ 0 \text{ sinon} \end{cases} \qquad \forall i \in V$

Objectif : Maximiser le poids des sommets sélectionnés :

$$\max \sum_{i \in V} w_i x_i$$

Contraintes :

- Ne pas sélectionner deux sommets adjacents :

$$(x_i = 0) \vee (x_j = 0), \qquad \forall \{i, j\} \in E$$

[ou] $\qquad x_i x_j = 0, \qquad \forall \{i, j\} \in E$

- Domaine de définition des variables :

$$x_i \in \{0, 1\}, \qquad \forall i \in V$$

Est-il possible de formuler ce problème à l'aide de contraintes linéaires ?

Modélisation du problème du sac-à-dos

Données : W : capacité maximale du sac
I : ensemble de n objets avec, pour chaque objet $i \in I$,
p_i l'utilité et w_i le poids

Problème combinatoire : $\max_{X \subseteq I} \left\{ \sum_{i \in X} p_i : \sum_{i \in X} w_i \leq W \right\}$

Variables : $x_i = \begin{cases} 1 \text{ si l'objet } i \text{ est choisi} \\ 0 \text{ sinon} \end{cases} \quad \forall i \in I$

Objectif : Maximiser l'utilité des objets sélectionnés :

$$\max \sum_{i \in I} p_i x_i$$

Contraintes :

- Respect de la capacité du sac :
  $$\sum_{i \in I} w_i x_i \leq W$$
- Domaine de définition des variables :
  $$x_i \in \{0, 1\}, \qquad \forall i \in I$$

Est-il possible d'améliorer la formulation ?

Modélisation du problème du plus court chemin

Données : $G = (V, A)$ graphe orienté avec deux sommets s et t, et c_{ij} longueur de l'arc $(i, j) \in A$

Problème combinatoire : $\min\limits_{X \subseteq A} \left\{ \sum\limits_{(i,j) \in X} c_{ij} : X \text{ forme un chemin } s \to t \right\}$

Variables : $x_{ij} = \begin{cases} 1 \text{ si l'arc } (i,j) \text{ est choisi} \\ 0 \text{ sinon} \end{cases} \quad \forall (i,j) \in A$

Objectif : Minimiser la longueur des arcs choisis : $\min \sum\limits_{(i,j) \in A} c_{ij} x_{ij}$

Contraintes :

- Contraintes de respect du flot :

$$\sum_{j \in \mathcal{V}^+(i)} x_{ji} - \sum_{j \in \mathcal{V}^-(i)} x_{ij} = \begin{cases} 1 & \text{si } i = s \\ 0 & \text{si } i \neq s, t \\ -1 & \text{si } i = t \end{cases}, \quad \forall i \in V$$

- Domaine de définition des variables :

$$x_{ij} \in \{0, 1\}, \qquad \forall (i,j) \in A$$

Existe-t-il une formulation idéale du problème ?

Modélisation du problème de l'arbre couvrant de poids m

Données : $G = (V, E)$ graphe non orienté, connexe avec
p_e poids de l'arête $e \in E$

Problème combinatoire : $\min\limits_{X \subseteq E} \left\{ \sum\limits_{e \in X} p_e : X \text{ forme un arbre couvrant} \right\}$

Variables : $x_e = \begin{cases} 1 \text{ si l'arête } e \text{ est choisie} \\ 0 \text{ sinon} \end{cases} \quad \forall e \in E$

Objectif : Minimiser le poids des arêtes choisies : $\min \sum\limits_{e \in E} p_e x_e$

Contraintes :

- Choisir $|V| - 1$ arêtes pour former un arbre :
$$\sum_{e \in E} x_e = |V| - 1$$
- Eviter d'obtenir un sous-graphe non connexe :
$$\sum_{e \in E(S)} x_e \leq |S| - 1, \quad \forall S \subset V$$
- Domaine de définition des variables :
$$x_e \in \{0, 1\}, \qquad \forall e \in E$$

Comment résoudre une formulation avec un nombre élevé de contraintes ?

Modélisation du problème de voyageur de commerce

Données : $G = (V, E)$ graphe non orienté avec c_e coût de l'arête $e \in E$

Problème combinatoire : $\min\limits_{X \subseteq E} \left\{ \sum\limits_{e \in X} c_e : X \text{ est un cycle hamiltonien} \right\}$

Variables : $x_e = \begin{cases} 1 \text{ si l'arête } e \text{ est choisie} \\ 0 \text{ sinon} \end{cases} \quad \forall e \in E$

Objectif : Minimiser les coûts des arêtes choisies : $\min \sum\limits_{e \in E} c_e x_e$

Contraintes :

- Visiter chaque sommet une fois :
$$\sum_{e \in E(i)} x_e = 2, \qquad \forall i \in V$$
- Eliminer les sous-tours non connectés :
$$\sum_{e \in E(S)} x_e \leq |S| - 1, \quad \forall S \subset V : 2 \leq |S| \leq |V| - 1$$
- Domaine de définition des variables :
$$x_e \in \{0, 1\}, \qquad \forall e \in E$$

Est-il possible de formuler le problème autrement ?

Modélisation du problème de recouvrement

Données : $N = \{1, \ldots, n\}$: ensemble de base à couvrir
$\{E_i \subseteq N\}_{i \in M}$ ensemble de sous-ensembles de N avec
c_i coût associé à l'ensemble E_i, pour $i \in M$
et $a_i \in \mathbb{B}^n$ un vecteur des éléments de E_i ($a_{ij} = 1$ si $j \in E_i$)

Problème combinatoire : $\min\limits_{X \subseteq M} \left\{ \sum\limits_{i \in X} c_i : \bigcup\limits_{i \in X} E_i = N \right\}$

Variables : $x_i = \begin{cases} 1 \text{ si l'élément } E_i \text{ est choisi} \\ 0 \text{ sinon} \end{cases} \quad \forall i \in M$

Objectif : Minimiser les coûts des ensembles choisis : $\sum\limits_{i \in M} c_i x_i$

Contraintes :

- Couvrir chaque élément de base au moins une fois :
$$\sum_{i \in M} a_{ij} x_i \geq 1 \qquad \forall j \in N$$
- Domaine de définition des variables :
$$x_i \in \{0, 1\} \qquad \forall i \in M$$

Comment résoudre une formulation avec un nombre élevé de variables ?

Problèmes de recouvrement/partitionnement/remplissage

Problème de recouvrement (set covering) : $\min_{X \subseteq M} \left\{ \sum_{i \in X} c_i : \bigcup_{i \in X} E_i = N \right\}$

$$\min \left\{ \sum_{i \in M} c_i x_i : \sum_{i \in M} a_{ij} x_i \geq 1 \ \ \forall j \in N; \ \ x_i \in \{0,1\} \ \ \forall i \in M \right\}$$

Problème de partitionnement : $\min / \max_{X \subseteq M} \left\{ \sum_{i \in X} c_i : \bigcup_{i \in X} E_i = N, \bigcap_{i \in X} E_i = \emptyset \right\}$

$$\min / \max \left\{ \sum_{i \in M} c_i x_i : \sum_{i \in M} a_{ij} x_i = 1 \ \ \forall j \in N; \ \ x_i \in \{0,1\} \ \ \forall i \in M \right\}$$

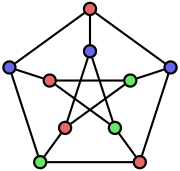
Problème de remplissage (set packing) : $\max_{X \subseteq M} \left\{ \sum_{i \in X} c_i : \bigcap_{i \in X} E_i = \emptyset \right\}$

$$\max \left\{ \sum_{i \in M} c_i x_i : \sum_{i \in M} a_{ij} x_i \leq 1 \ \ \forall j \in N; \ \ x_i \in \{0,1\} \ \ \forall i \in M \right\}$$

Cas particuliers du problème de recouvrement 1/2

Problème de coloration de graphe (graph coloring problem)

Colorier avec le moins de couleur tous les sommets d'un graphe tel que chaque paire de sommets adjacents n'ont pas la même couleur



Données : $G = (V, E)$ graphe non orienté

Objectif : chercher une coloration minimale des sommets

Contraintes : deux couleurs différentes pour les sommets adjacents

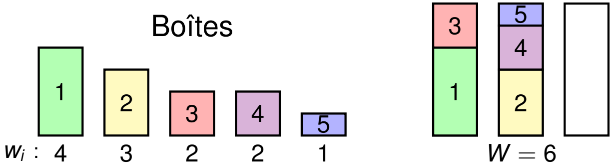
Cas particuliers du problème de recouvrement

- N : ensemble des sommets du graphe
- M : ensemble des stables du graphe avec $c_i = 1, \; \forall i \in M$

Cas particuliers du problème de recouvrement 2/2

Problème de placement de boîtes (bin packing problem)

Placer un ensemble de boîtes de longueurs différentes dans le moins de cartons possible de longueur fixée



Données : N : ensemble de n boîtes avec
w_i longueur de la boîte $i \in N$ et W longueur des cartons

Objectif : minimiser le nombre de cartons utilisés

Contraintes : ne pas dépasser la capacité d'un carton

Cas particuliers du problème de partitionnement

- N : ensemble des boîtes
- M : ensemble des remplissages possibles du carton

Forme générale d'un problème combinatoire

Un problème d'optimisation combinatoire se modélise comme suit :

$$\min \left\{ \sum_{j \in N} c_j x_j : x \in P \cap \{0,1\}^{|N|} \right\}$$

Cependant, certaines décisions seront similaires

Forme générale d'une modélisation d'un problème combinatoire

$$\min \left\{ f(x) : x \in P \cap \mathbb{Z}^n \right\}$$

Objectif de ce cours

Apprendre à :

- modéliser nos problèmes par des programmes linéaires en nombres entiers
- résoudre des programmes linéaires en nombres entiers de façon exacte

La programmation linéaire

Programme linéaire

Optimiser une fonction linéaire sous des contraintes linaires

$$(PL) \begin{cases} \max & cx \\ \text{s.c.} & Ax \leq b \\ & x \geq 0 \end{cases}$$

- $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ et $A \in \mathbb{R}^{m \times n}$: données du problème
- $x \in \mathbb{R}^n$: variables/inconnus du problème
- cx : fonction objectif du problème (fonction linéaire)
- $Ax \leq b, x \geq \mathbf{0}_n$: contraintes linéaires du problème

Polyèdre/Région réalisable : ensemble des solutions réalisables

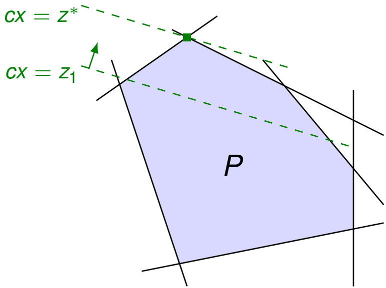
$$P = \{x \in \mathbb{R}^n : Ax \leq b, x \geq \mathbf{0}_n\}$$

Représentation graphique

Programme linéaire

$$(PL) \quad Z^{PL} = \max\{cx : x \in P\}$$

avec $P = \{x \in \mathbb{R}^n : Ax \leq b, x \geq \mathbf{0}_n\}$



Pour une solution x donné, on notera $z = cx$ la valeur de cette solution

Idée de la résolution

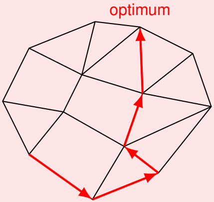
Remarques

Contraintes et objectif linéaire $\Rightarrow$ existence d'un optimum sur un point extrême de la région réalisable

Donc, il suffit d'examiner les points extrêmes

Idée de résolution

Passer de points extrêmes en points extrêmes jusqu'à l'optimum



Extension de la programmation linéaire

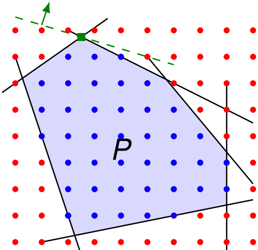
Soit le programme linéaire en nombres entiers :

$$(PLE)\begin{cases} \max \quad cx \\ \left.\begin{aligned} Ax &\leq b \\ x &\geq \mathbf{0}_n \end{aligned}\right\} x \in P \\ x \in \mathbb{Z}^n \end{cases}$$

Problèmatique

Certaines valeurs ne peuvent être fractionnaires : décision oui/non, quantité indivisible,...

Rajouter une condition supplémentaire : $x \in \mathbb{Z}^n \rightarrow$ difficile à résoudre



Modélisation par la programmation linéaire entière

Le problème du stable de poids maximum

Question : Quels sommets sélectionner afin de maximiser le poids total tel que les sommets sélectionnés soient non adjacents ?

Données : $G = (V, E)$ graphe non orienté avec
w_i poids du sommet $i \in V$

Variables : $x_i = \begin{cases} 1 \text{ si le sommet } i \text{ est choisi} \\ 0 \text{ sinon} \end{cases} \qquad \forall i \in V$

Objectif : Maximiser le poids des sommets choisis :

$$\max \sum_{i \in V} w_i x_i$$

Contraintes :
- Ne pas sélectionner deux sommets adjacents :
$$x_i + x_j \leq 1, \qquad \forall \{i, j\} \in E$$
- Domaine de définition des variables :
$$x_i \in \{0, 1\}, \qquad \forall i \in V$$

Quelques astuces de modélisation : objectifs particuliers

Soient $e_1, \ldots, e_n$ des expressions linéaires

1. modélisation d'un objectif du type $\min \max\limits_{i=1,\ldots,n} \{e_i\}$:
   - rajouter une variable $y \in \mathbb{R}$
   - remplacer l'objectif par $\min y$
   - rajouter les contraintes $y \geq e_i, \quad i = 1, \ldots, n$

   une modélisation similaire est possible pour $\max \min\limits_{i=1,\ldots,n} \{e_i\}$
2. modélisation d'un objectif du type $\min \sum\limits_{i=1}^{n} |e_i|$:
   - rajouter n variables $y_i \in \mathbb{R}$ avec $i = 1, \ldots, n$
   - remplacer l'objectif par $\min \sum\limits_{i=1}^{n} y_i$
   - rajouter les contraintes $y_i \geq e_i, y_i \geq -e_i, \quad i = 1, \ldots, n$

Remarque

La linéarisation de $\min \min\limits_{i=1,\ldots,n} \{e_i\}$ et de $\max \sum\limits_{i=1}^{n} |e_i|$ est plus difficile

Quelques astuces de modélisation : variables particulière

Soient $v_1, \dots, v_n, l, u$ des valeurs réelles quelconques
Soit M une valeur numérique suffisamment grande

1. modélisation d'une variable discrète $x \in \{v_1, \dots, v_n\}$:
   - rajouter n variables binaires $y_i \in \{0, 1\}$ avec $i = 1, \dots, n$
   - utiliser les contraintes $x = \sum_{i=1}^{n} v_i y_i$ et $\sum_{i=1}^{n} y_i = 1$
2. modélisation d'une variable binaire $y \in \{0, 1\}$ d'activation de bornes :
   $$\text{si } y = 1 \text{ alors } l \leq x \leq u, \text{ sinon } x = 0$$
   - utiliser la contrainte $ly \leq x \leq uy$

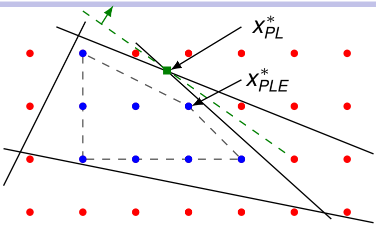
   souvent utilisé pour modéliser des coûts fixes d'utilisation d'une ressource
3. modélisation de variables incompatibles $x_i, x_j \in \mathbb{R}$:
   $$(x_i = 0) \vee (x_j = 0)$$
   - rajouter deux variables binaires $y_i, y_j \in \{0, 1\}$
   - utiliser les contraintes $x_i \leq M(1 - y_i), \quad x_j \leq M(1 - y_j), \quad y_i + y_j \leq 1$

Quelques astuces de modélisation : contraintes logiques

Soient $e, e_1, \ldots, e_n$ des expressions linéaires
Soient $l, u, v_1, \ldots, v_n$ des valeurs réelles quelconques
Soit M une valeur suffisamment grande

1. modélisation d'une contrainte disjonctive exclusif $(e \leq u) \vee (e \geq l)$:
   - rajouter une variable binaire $y \in \{0, 1\}$
   - utiliser les contraintes $e \leq u + M(1 - y), \quad e \geq l - My$
2. modélisation d'une contrainte disjonctive exclusif du type $\bigvee_{i=1}^{n} (e_i \leq v_i)$:
   - rajouter n variables binaires $y_i \in \{0, 1\}$ avec $i = 1, \ldots, n$
   - utiliser les contraintes $\sum_{i=1}^{n} y_i = 1, \quad e_i \leq v_i + M(1 - y_i), \; i = 1, \ldots, n$
3. modélisation d'un produit de variables binaires $x_i x_j$ avec $x_i, x_j \in \{0, 1\}$:
   - rajouter une variable binaire $y_{ij} \in \{0, 1\}$
   - remplacer le produit $x_i x_j$ par y_{ij} dans les contraintes
   - rajouter les contraintes $x_i \geq y_{ij}, \quad x_j \geq y_{ij}, \quad x_i + x_j \leq 1 + y_{ij}$

Vocabulaire en programmation linéaire entière



$$(PLE) \quad Z = \max\{cx : x \in P \cap \mathbb{Z}^n\}$$

où $P = \{x \in \mathbb{R}^n : Ax \leq b, x \geq \mathbf{0}_n\}$

Vocabulaire

Solution réalisable : Solution $x \in P \cap \mathbb{Z}^n$
(solutions réalisables : points bleus)

Relaxation linéaire : Problème $\max\{cx : Ax \leq b, x \geq \mathbf{0}_n, x \in \mathbb{R}^n\}$
(valeur de la relaxation linéaire : carré vert)

Polyèdre entier : Polyèdre tel que tous les points extrêmes sont des points entiers (polyèdre entier : contour pointillé gris)

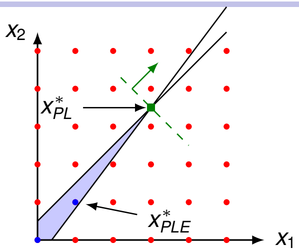
$$Z^*_{PLE} = cx^*_{PLE} \leq cx^*_{PL} = Z^*_{PL}$$

où Z^*_{PLE} et Z^*_{PLE} sont les valeurs de l'optimum et de la relaxation linéaire

Arrondir la solution de la relaxation linéaire ?

Exemple 1 :

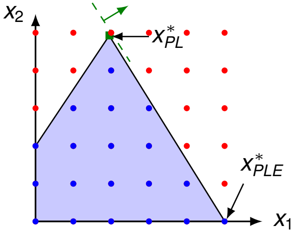
$$\begin{array}{lrcl} \max & x_1 + x_2 & & \\ \text{s. c.} & -2x_1 + 2x_2 & \leq & 1 \\ & 8x_1 - 6x_2 & \leq & 3 \\ & x_1 \ , \ x_2 & \in & \mathbb{N} \end{array}$$



Arrondir la solution donne pas forcément de solutions réalisables !

Exemple 2 :

$$\begin{array}{lrcl} \max & 25x_1 + 16x_2 & & \\ \text{s. c.} & 50x_1 + 31x_2 & \leq & 250 \\ & 3x_1 - 2x_2 & \geq & -4 \\ & x_1 \ , \ x_2 & \in & \mathbb{N} \end{array}$$



Arrondir la solution donne pas forcément une solution optimale !

Enumérer exhaustivement toutes les solutions ?

Problème du sac-à-dos

Choisir un sous-ensemble d'objets d'utilité maximale respectant une contrainte de capacité

Variables : $x_i = 1$ si l'objet $i \in N$ est rangé dans le sac ; 0 sinon

Formulation :

$$\begin{array}{ll} \max & \sum_{i \in N} p_i x_i \\ \text{s. c.} & \sum_{i \in N} w_i x_i \leq W \\ & \quad x_i \in \{0,1\} \quad \forall i \in N \end{array}$$

Explosion combinatoire : Nombre de sous ensembles possibles 2^n

complexité algorithme	taille instance/temps résolution 10	20	30	40	50	60
n	.00001 s	.00002 s	.00003 s	.00004 s	.00005 s	.00006 s
n^2	.0001 s	.0004 s	.0009 s	.0016 s	.0025 s	.0036 s
n^3	.001 s	.008 s	.027 s	.064 s	.124 s	.216 s
n^5	.1 s	3.2 s	24.3 s	1.7 m	5.2 m	13.0 m
2^n	.001 s	1 s	17.9 m	12.7 j	35.7 a	36600 a

On ne peut pas énumérer toutes les solutions en pratique !

Classes de complexité des problèmes 1/2

Problème décisionnel

Problème décisionnel : problème algorithmique posé sous la forme de question dont la réponse est soit oui, soit non :
étant donné une instance du problème, dire si elle vérifie les conditions

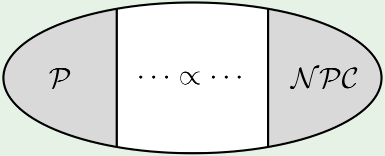
La classe $\mathcal{NP}$

Un problème décisionnel appartient à la classe $\mathcal{NP}$ ssi il existe un algorithme non déterministe résolvant le problème en temps polynomial

Pour les problèmes de la classe $\mathcal{NP}$, il existe un certificat qui détermine, en temps polynomial, la véracité d'une solution

Classification des problèmes

classe $\mathcal{NP}$



Classes de complexité des problèmes 2/2

La classe $\mathcal{P}$: problèmes faciles de la classe $\mathcal{NP}$

Un problème décisionnel (*PD*) appartient à la classe $\mathcal{P}$ ssi il existe un algorithme déterministe résolvant le problème en temps polynomial

Les problèmes de la classe $\mathcal{P}$ appartiennent à la classe $\mathcal{NP}$: $\mathcal{P} \subset \mathcal{NP}$

La classe $\mathcal{NP}$-complet : problèmes de $\mathcal{NP}$ connus comme difficiles

Un problème décisionnel (*PD*) appartient à la classe $\mathcal{NPC}$ ssi

- le problème (*PD*) appartient à la classe $\mathcal{NP}$ et
- il est aussi difficile que tous les autres problèmes de la classe $\mathcal{NP}$

$$P \propto PD, \qquad \forall P \in \mathcal{NP}$$

Les problèmes de la classe $\mathcal{NP}$-complet (ou $\mathcal{NPC}$) sont les problèmes les plus difficile de la classe $\mathcal{NP}$

Question ouverte ?

$$\mathcal{P} = \mathcal{NP}\,? \qquad \boxed{\text{ou}} \qquad \mathcal{P} \neq \mathcal{NP}\,?$$

Difficulté des problèmes d'optimisation

Soit le problème d'optimisation cominatoire (PO) :

$$(PO) \quad \max\{c(x) : x \in \mathcal{S}\}$$

Problème décisionnel (PD) d'un problème d'optimisation (PO)

Question : étant donné $\bar{z} \in \mathbb{R}$, existe-t-il $x \in \mathcal{S}$ tel que $cx \geq \bar{z}$?

La classe $\mathcal{NP}$-difficile

Un problème P est $\mathcal{NP}$-difficile ssi il est aussi difficile que tous les autres problèmes de la classe $\mathcal{NP}$:

$$P' \propto P, \qquad \forall P' \in \mathcal{NP}$$

Classes de complexité des problèmes d'optimisation

- Un problème d'optimisation (PO) appartient à la classe $\mathcal{NP}$-facile si son problème décisionnel (PD) associé appartient à la classe $\mathcal{P}$
- Un problème d'optimisation (PO) appartient à la classe $\mathcal{NP}$-difficile si son problème décisionnel (PD) associé appartient à la classe $\mathcal{NPC}$

Difficulté de la programmation linéaire

Soit le programme linéaire :

$$(PL) \quad \max\{cx : x \in P\}$$

avec $P = \{x \in \mathbb{R}^n : Ax \leq b, x \geq \mathbf{0}_n\}$

Problème décisionnel d'un programme linéaire

Question : Etant donné $\bar{z} \in \mathbb{R}$, existe-t-il $x \in P$ tel que $cx \geq \bar{z}$?

Classe de complexité d'un programme linéaire

Le problème décisionnel d'un programme linéaire appartient à la classe $\mathcal{P}$ (démonstration par l'algorithme de l'ellipsoïde)

Les programmes linéaires sont $\mathcal{NP}$-faciles !!

Remarques

- En pratique, l'algorithme du simplexe converge plus rapidement
- Un programme linéaire a un nombre infini de solutions réalisables

Difficulté de la programmation linéaire en nombres entier

Soit le programme linéaire en nombres entiers :

$$(PLE) \quad \max\{cx : x \in P \cap \mathbb{Z}^n\}$$

avec $P = \{x \in \mathbb{R}^n : Ax \leq b, x \geq \mathbf{0}_n\}$

Problème décisionnel d'un programme linéaire en nombres entiers

Question : Etant donné $\overline{z} \in \mathbb{R}$, existe-t-il $x \in P \cap \mathbb{Z}^n$ tel que $cx \geq \overline{z}$?

Classe de complexité d'un programme linéaire en nombres entiers

Le problème décisionnel d'un programme linéaire en nombres entiers appartient en général à la classe $\mathcal{NP}$-complet

Les programmes linéaires entiers sont en général $\mathcal{NP}$-difficiles ! !

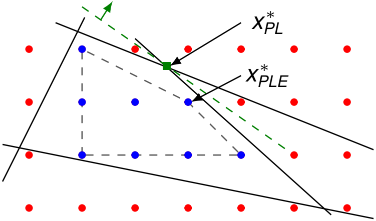
Remarques

- Certains problèmes appartiennent cependant à la classe $\mathcal{P}$
- Un programme linéaire entier a un nombre dénombrable (voire fini) de solutions réalisables

Quelques concepts de la programmation linéaire entière

$(PLE) \quad Z = \max\{cx : x \in P \cap \mathbb{Z}^n\}$

avec $P = \{x \in \mathbb{R}^n : Ax \leq b, x \geq \mathbf{0}_n\}$



Notions de bornes

BI $\leq$ Optimum $\leq$ BS

Borne primale (BP) : Borne inférieure (BI) sur le profit maximum
Exemple : valeur d'une solution réalisable

Borne duale (BD) : Borne supérieure (BS) sur le profit maximum
Exemple : valeur d'une relaxation du problème

Qualité de formulation : Qualité des bornes (écart faible des bornes)
Exemple : bonne description du polyèdre entier

Pour un problème de minimisation, les bornes sont BD=BS et BP=BI

Deuxième partie II

TECHNIQUES CLASSIQUES DE RÉSOLUTION

Résolution par énumération intelligente

Soit le problème d'optimisation combinatoire initial :

$$(PO) \quad \max\{f(x) : x \in \mathcal{S}\} \quad \text{avec } \mathcal{S} \subset \mathbb{R}^n$$

Objectif

Eviter une énumération exhaustive

Résolution exacte d'un programme linéaire entier

Procéder à des énumérations intelligentes pour trouver l'optimum en fixant des valeurs aux variables au fur et à mesure

- Algorithme de séparation et évaluation (Branch-and-Bound) : énumération basée sur le principe de bornes
- Programmation dynamique (Dynamic Programming) : énumération basée sur le principe de dominance de solutions
- Programmation par contraintes (Constraint Programming) : énumération basée sur la détection d'irréalisabilité

Algorithme de séparation et évaluation

Soit le problème d'optimisation combinatoire initial :

$$(PO^0) \quad \max\{f(x) : x \in \mathcal{S}^0\} \quad \text{avec } \mathcal{S}^0 \subset \mathbb{R}^n$$

Idée de résolution : « diviser pour mieux régner »

Résoudre p problèmes indépendants (PO^k) plus restreints :

$$(PO^k) \quad \max\{f(x) : x \in \mathcal{S}^k\} \qquad \text{avec } \mathcal{S}^0 = \bigcup_{k=1}^{p} \mathcal{S}^k$$

Algorithme de séparation et évaluation

A chaque étape, subdiviser l'ensemble $\mathcal{S}^k$ en utilisant les procédures :

Evaluation : estimer la valeur de la meilleure solution réalisable possible d'un ensemble de solutions $\mathcal{S}^k$

Séparation : séparer l'ensemble des solutions réalisables courant $\mathcal{S}^k$ en p sous-ensembles indépendants $\mathcal{S}^{n+1}, \ldots, \mathcal{S}^{n+p}$

Elaguation : éliminer les ensembles de solutions $\mathcal{S}^l$ pour lesquels on garantit qu'ils ne contiennent pas l'optimum

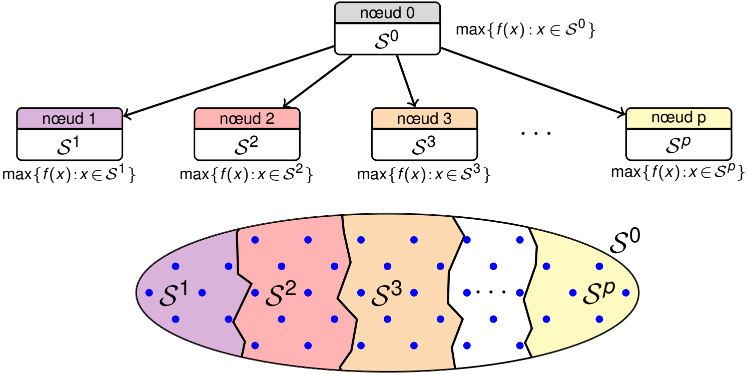
Cheminement : choisir le prochain ensemble de solutions $\mathcal{S}^l$ à étudier

Algorithme de séparation et évaluation : représentation

Représentation graphique d'un algorithme de séparation et évaluation

Utiliser un arbre enraciné (appelé arbre de séparation et évaluation) :

- un nœud représente un sous-ensemble de solutions $\mathcal{S}$ à résoudre
- un arc est créé entre deux nœuds à chaque séparation



Procédures importantes

Procédure d'évaluation : recherche d'une estimation (borne duale)

- Calculer une borne duale (résoudre une relaxation du problème)

Procédure de séparation : choix et fixation de décision (branchement)

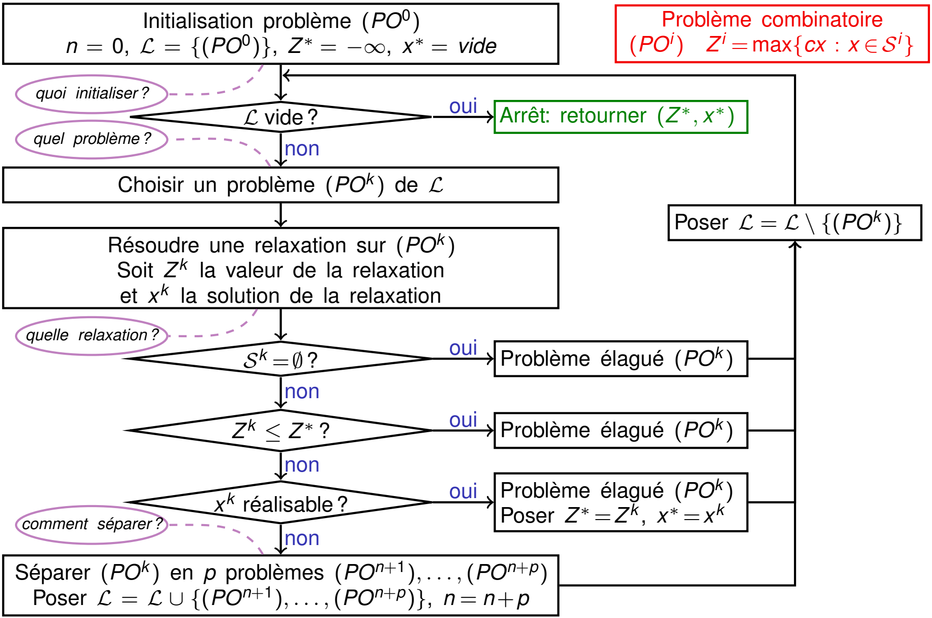
- Branchement sur des variables : choisir une variable puis fixer une valeur en traitant les différents cas possibles :
  ex : $(\mathcal{S} \cup \{x_i = \alpha_1\}) \vee (\mathcal{S} \cup \{x_i = \alpha_2\}) \vee \cdots \vee (\mathcal{S} \cup \{x_i = \alpha_p\})$
- Branchement sur des contraintes :
  ex : $(\mathcal{S} \cup \{\sum_{i=1}^{n} x_i = 0\}) \vee (\mathcal{S} \cup \{\sum_{i=1}^{n} x_i = 1\})$
- Fixer un ordre relatif de certains éléments

Procédure d'élaguation : règle de dominance de bornes

Eliminer un nœud si :

- l'évaluation est moins bonne que la meilleure solution trouvée
- la meilleure solution réalisable du nœud courant est connue
- le nœud courant ne contient pas de solutions réalisables

Diagramme d'un algorithme de séparation et évaluation



Arbre de séparation et évaluation pour un PLE

Soit le programme linéaire entier :

$$\max\{cx : x \in \mathcal{S} = P \cap \mathbb{Z}^n\} \quad \text{avec } P = \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}$$

Algorithme de séparation et évaluation utilisant une relaxation linéaire

Evaluation : utiliser une relaxation linéaire :

résoudre $\max\{cx : x \in P\}$

choisir une variable x_i tel que $x_i = \alpha \notin \mathbb{N}$ pour la relaxation

Séparation : séparer le domaine P en deux en ajoutant une contrainte disjonctive sur la variable x_i :

$$(\mathcal{S} \cap \{x_i \leq \lfloor \alpha \rfloor\}) \vee (\mathcal{S} \cap \{x_i \geq \lceil \alpha \rceil\})$$

nœud 0

$\mathcal{S}^0 = P \cap \mathbb{Z}^n$

$Z^0 = \max\left\{cx : x \in P\right\}$

$x_i \leq \lfloor \alpha \rfloor$ → **nœud 1**

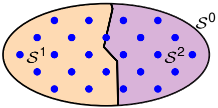
$\mathcal{S}^1 = P \cap \{x_i \leq \lfloor \alpha \rfloor\} \cap \mathbb{Z}^n$

$Z^1 = \max\left\{cx : x \in P \cap \{x_i \leq \lfloor \alpha \rfloor\}\right\}$

$x_i \geq \lceil \alpha \rceil$ → **nœud 2**

$\mathcal{S}^2 = P \cap \{x_i \geq \lceil \alpha \rceil\} \cap \mathbb{Z}^n$

$Z^2 = \max\left\{cx : x \in P \cap \{x_i \geq \lceil \alpha \rceil\}\right\}$



Exemple de résolution exacte

Résolvons le programme linéaire entier suivant :

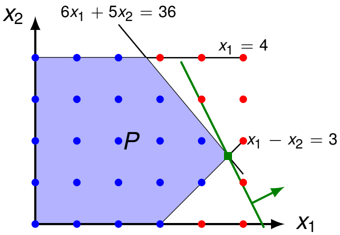
$$\max\left\{z = 2x_1 + x_2 : (x_1, x_2) \in P \cap \mathbb{Z}^2\right\}$$

$$\text{avec } P = \left\{(x_1, x_2) \in \mathbb{R}^2 : \begin{array}{ll} x_1 - x_2 \leq 3, & 6x_1 + 5x_2 \leq 36 \\ 0 \leq x_1, & 0 \leq x_2 \leq 4 \end{array}\right\}$$

Procédure d'évaluation choisie

Evaluation : relaxation linéaire

La solution de la relaxation linéaire est $\left(\frac{51}{11}, \frac{18}{11}\right)$ de valeur $\frac{120}{11}$



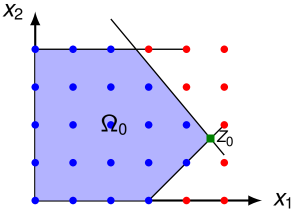
Exemple d'arbre de séparation et évaluation 1/4

$$z = 2x_1 + x_2$$

$$P = \begin{cases} x_1 - x_2 \leq 3 \\ 6x_1 + 5x_2 \leq 36 \\ x_2 \leq 4 \\ x_1 \, , \, x_2 \geq 0 \end{cases}$$

résoudre $\max_{x \in P} \{z\}$

Nœud 0 $\rightarrow \Omega_0$
Eval : $Z_0 = 10.91$
$x = (4.64, 1.64)$



Exemple d'arbre de séparation et évaluation 2/4

$$z = 2x_1 + x_2$$

$$P = \begin{cases} x_1 - x_2 \leq 3 \\ 6x_1 + 5x_2 \leq 36 \\ x_2 \leq 4 \\ x_1 \ , \ x_2 \geq 0 \end{cases}$$

résoudre $\max\limits_{x \in P} \{z\}$

Nœud 0 $\rightarrow \Omega_0$
Eval : $Z_0 = 10.91$
$x = (4.64, 1.64)$

$x_2 \leq \lfloor 1.64 \rfloor$ | $x_2 \geq \lceil 1.64 \rceil$

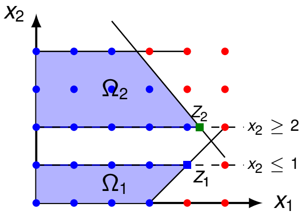
résoudre $\max\limits_{x \in P} \left\{ z \colon x_2 \leq 1 \right\}$

Nœud 1 $\rightarrow \Omega_1$
Eval : $Z_1 = 9$
$x = (4, 1)$

Solution réalisable

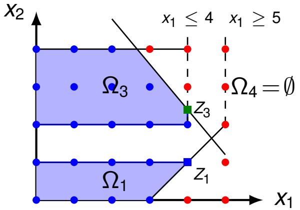
Nœud 2 $\rightarrow \Omega_2$
Eval : $Z_2 = 10.67$
$x = (4.33, 2)$

résoudre $\max\limits_{x \in P} \left\{ z \colon x_2 \geq 2 \right\}$



Exemple d'arbre de séparation et évaluation 3/4

$$z = 2x_1 + x_2$$
$$P = \begin{cases} x_1 - x_2 \leq 3 \\ 6x_1 + 5x_2 \leq 36 \\ x_2 \leq 4 \\ x_1\ ,\ x_2 \geq 0 \end{cases}$$



résoudre $\max_{x \in P}\{z\}$

Nœud 0 $\rightarrow \Omega_0$
Eval : $Z_0 = 10.91$
$x = (4.64, 1.64)$

$x_2 \leq \lfloor 1.64 \rfloor$ | $x_2 \geq \lceil 1.64 \rceil$

résoudre $\max_{x \in P}\{z : x_2 \leq 1\}$

Nœud 1 $\rightarrow \Omega_1$
Eval : $Z_1 = 9$
$x = (4, 1)$

Solution réalisable

Nœud 2 $\rightarrow \Omega_2$
Eval : $Z_2 = 10.67$
$x = (4.33, 2)$

résoudre $\max_{x \in P}\{z : x_2 \geq 2\}$

$x_1 \leq \lfloor 4.33 \rfloor$ | $x_1 \geq \lceil 4.33 \rceil$

résoudre $\max_{x \in P}\left\{z : \begin{matrix} x_1 \leq 4 \\ x_2 \geq 2 \end{matrix}\right\}$

Nœud 3 $\rightarrow \Omega_3$
Eval : $Z_3 = 10.4$
$x = (4, 2.4)$

Nœud 4 $\rightarrow \Omega_4$
Eval : $Z_4 = -\infty$
Irréalisable

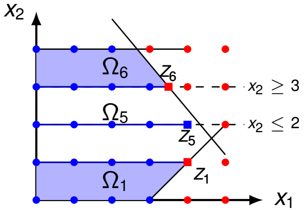
Elaguer par irréalisibilité

résoudre $\max_{x \in P}\left\{z : \begin{matrix} x_1 \geq 5 \\ x_2 \geq 2 \end{matrix}\right\}$

Exemple d'arbre de séparation et évaluation 4/4

$$z = 2x_1 + x_2$$

$$P = \begin{cases} x_1 - x_2 \leq 3 \\ 6x_1 + 5x_2 \leq 36 \\ x_2 \leq 4 \\ x_1 \;,\; x_2 \geq 0 \end{cases}$$



résoudre $\max_{x \in P}\{z\}$

Nœud 0 $\rightarrow \Omega_0$
Eval : $Z_0 = 10.91$
$x = (4.64, 1.64)$

$x_2 \leq \lfloor 1.64 \rfloor$ — $x_2 \geq \lceil 1.64 \rceil$

résoudre $\max_{x \in P}\{z : x_2 \leq 1\}$

Nœud 1 $\rightarrow \Omega_1$
Eval : $Z_1 = 9$
$x = (4, 1)$

Solution dominée

Nœud 2 $\rightarrow \Omega_2$
Eval : $Z_2 = 10.67$
$x = (4.33, 2)$

résoudre $\max_{x \in P}\{z : x_2 \geq 2\}$

$x_1 \leq \lfloor 4.33 \rfloor$ — $x_1 \geq \lceil 4.33 \rceil$

résoudre $\max_{x \in P}\left\{z : \begin{matrix} x_1 \leq 4 \\ x_2 \geq 2 \end{matrix}\right\}$

Nœud 3 $\rightarrow \Omega_3$
Eval : $Z_3 = 10.4$
$x = (4, 2.4)$

Nœud 4 $\rightarrow \Omega_4$
Eval : $Z_4 = -\infty$
Irréalisable

résoudre $\max_{x \in P}\left\{z : \begin{matrix} x_1 \geq 5 \\ x_2 \geq 2 \end{matrix}\right\}$

Elaguer par irréalisibilité

$x_2 \leq \lfloor 2.4 \rfloor$ — $x_2 \geq \lceil 2.4 \rceil$

résoudre $\max_{x \in P}\left\{z : \begin{matrix} x_1 \leq 4 \\ x_2 = 2 \end{matrix}\right\}$

Nœud 5 $\rightarrow \Omega_5$
Eval : $Z_5 = 10$
$x = (4, 2)$

Solution réalisable

Nœud 6 $\rightarrow \Omega_6$
Eval : $Z_6 = 10$
$x = (3.5, 3)$

résoudre $\max_{x \in P}\left\{z : \begin{matrix} x_1 \leq 4 \\ x_2 \geq 3 \end{matrix}\right\}$

Elaguer par dominance

Mécanismes importants

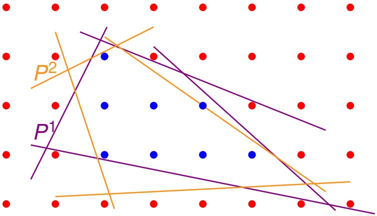
1. Initialisation :
   - calculer une solution réalisable initiale (heuristique gloutonne)
   - simplifier la formulation par des pré-traitements
2. Choix des bornes :
   - choisir une bonne relaxation (qualité relaxation vs temps calcul)
   - calculer la valeur d'un problème dual (existe pas toujours)
   - ajouter des contraintes valides (amélioration des bornes)
   - choisir une formulation de qualité (optimum proche relaxation)
3. Choix des branchements :
   - variable la plus prioritaire ou la plus fractionnaire (pour la relaxation)
   - favoriser l'émergence de solutions entières
4. Choix du cheminement des nœuds :
   - Parcours en profondeur : émergence de solutions réalisables
   - Parcours des meilleurs nœuds (valeur de la borne duale) : limiter le nombre de nœuds à étudier
5. Autres procédures :
   - utiliser des heuristiques primales : construire une solution à partir d'une relaxation
   - nettoyer la formulation après branchement

Formulation valide d'un problème d'optimisation comb.

Soit le problème d'optimisation combinatoire :

$$(PO) \quad \max\{cx : x \in \mathcal{S}\} \quad \text{avec } \mathcal{S} \subset \mathbb{R}^n$$

Notons $\mathcal{S} = \{x^s\}_{s=1}^{p}$ l'ensemble des points entiers réalisables (en bleu)



Contrairement à la programmation linéaire, un problème entier peut se formuler de différentes manières

Formulation valide

Un polyèdre $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ tel que les seuls points entiers contenus dans le polyèdre P sont les solutions réalisables au problème

$$\mathcal{S} = P \cap \mathbb{Z}^n$$

Formulation naturelle du problème de coloration de graph

variables modélisant l'affectation d'une couleur à chaque sommet

Données : $G = (V, E)$ graphe non orienté avec $V = \{1, \ldots, n\}$
$K = \{1, \ldots, k\}$ ensemble de k couleurs (ex : $k = n$)

Variables : $x_{ic} = \begin{cases} 1 \text{ si la couleur } c \text{ est affectée au sommet } i \\ 0 \text{ sinon} \end{cases} \quad \forall i \in V, c \in K$

$y_c = \begin{cases} 1 \text{ si la couleur } c \text{ est utilisée} \\ 0 \text{ sinon} \end{cases} \quad \forall c \in K$

Objectif : minimiser le nombre de couleur utilisée : $\min \sum_{c \in K} y_c$

Contraintes :

- affectation d'une couleur à chaque sommet :
  $$\sum_{c \in K} x_{ic} = 1, \qquad \forall i \in V$$
- les sommets adjacents n'ont pas la même couleur :
  $$x_{ic} + x_{jc} \leq 1, \qquad \forall \{i, j\} \in E, c \in K$$
- n'affecter des couleurs que si la couleur est utilisée :
  $$\sum_{i \in V} x_{ic} \leq n y_c, \qquad \forall c \in K$$

Formulation par lots du problème de coloration de graphe

variables modélisant des lots de sommets avec une même couleur

Données : $G = (V, E)$ un graphe non orienté avec $V = \{1, \ldots, n\}$

Variables : $w_{ij} = \begin{cases} 1 \text{ si } i \text{ est dans le lot de sommet de référence } j \\ 0 \text{ sinon} & \forall i, j \in V : \{i, j\} \notin E, i \geq j \end{cases}$

$w_{ii} = \begin{cases} 1 \text{ si le sommet } i \text{ est la référence d'un lot} \\ 0 \text{ sinon} & \forall i \in V \end{cases}$

Objectif : minimiser le nombre de sommets référence : $\sum_{i \in V} w_{ii}$

Contraintes :

- affectation d'une couleur à chaque sommet :
$$\sum_{j \in V : j \leq i} w_{ij} = 1, \qquad \forall i \in V$$
- les sommets adjacents n'ont pas la même couleur :
$$w_{ik} + w_{jk} \leq 1, \qquad \forall \{i, j\} \in E, k \in V \ (k \leq i, j)$$
- n'affecter des couleurs que si la couleur est utilisée :
$$\sum_{i \in V : j \leq i} w_{ij} \leq n w_{jj}, \qquad \forall j \in V$$

Qualité de formulation d'un problème entier

Soit le problème d'optimisation combinatoire :

$$(PO) \quad Z^* = \max\{cx : x \in \mathcal{S}\} \quad \text{avec } \mathcal{S} \subset \mathbb{R}^n$$

Dominance entre formulations

Soient P^1 et P^2 deux polyèdres valides du problème (PO)

P^1 domine P^2 ssi tout point de P^1 est dans P^2 : $P^1 \subseteq P^2$

Comparaison des relaxations linéaires

Soient P^1 et P^2 deux polyèdres valides de (PO) tel que $P^1 \subseteq P^2$

$$Z^* \leq Z^{PL}(P^1) \leq Z^{PL}(P^2)$$

« Bonne » formulation

Formulation dont la relaxation linéaire issu du polyèdre associé donne une enveloppe assez ajustée des solutions entières

Pour une bonne formulation, la relaxation linéaire

- donne une borne duale proche de l'optimum entier
- offre éventuellement une approximation (par arrondi) de l'optimum

Qualité des formulations pour le problème de coloration

La formulation par lot domine-t-elle la formulation naturelle ?

Soit P_{nat} (resp. P_{lot}) le polyèdre associé à la formulation naturelle (resp. par lot).

Lemme

Le polyèdre P_{lot} domine le polyèdre P_{nat}

Cependant, P_{lot} est loin de la meilleure formulation

Amélioration des formulations du problème de coloration

Amélioration de la formulation naturelle

- Remplacer $x_{ic} + x_{jc} \leq 1, \ \forall \{i,j\} \in E, c \in K$ par :
$$x_{ic} + x_{jc} \leq y_c, \quad \forall \{i,j\} \in E, c \in K$$
- Remplacer $\sum_{i \in V} x_{ic} \leq n y_c, \ \forall c \in K$ par :
$$x_{ic} \leq y_c, \quad \forall i \in V, c \in K$$

Amélioration de la formulation par lots

- Remplacer $w_{ik} + w_{jk} \leq 1, \ \forall \{i,j\} \in E, k \in V \ (k \leq i,j)$ par :
$$w_{ik} + w_{jk} \leq w_{kk}, \quad \forall \{i,j\} \in E, k \in V \ (k \leq i,j)$$
- Remplacer $\sum_{i \in V : j \leq i} w_{ij} \leq n\, w_{jj}, \ \forall j \in V$ par :
$$w_{ij} \leq w_{jj}, \quad \forall i,j \in V \ (j \leq i)$$

Remarques sur les symétries dans les formulations

Symétries des solutions

Permutation des valeurs affectées aux variables correspondant à une solution similaire

Pour éviter d'énumerer plusieurs solutions équivalentes, il est intéressant de rajouter des contraintes supplémentaires

Exemple du problème de coloration de graphes

La formulation naturelle possède des symétries de solution :

> chaque permutation de couleur sur les sommets donne des solutions réalisables équivalentes

Nouvelles contraintes pour la formulation naturelle

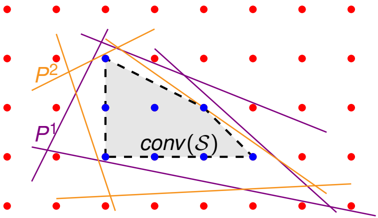
- Ne pas utiliser la c^{e} couleur si la $c-1^{\text{e}}$ n'est pas utilisée :
$$y_c \leq y_{c-1}, \quad \forall c \in K \setminus \{1\}$$
- Colorier moins de sommets avec la c^{e} couleur qu'avec la $c-1^{\text{e}}$:
$$\sum_{i \in V} x_{ic} \leq \sum_{i \in V} x_{i,c-1}, \quad \forall c \in K \setminus \{1\}$$

Formulation idéale d'un problème entier

Soit le problème combinatoire

$$(PO) \quad \max\{cx : x \in \mathcal{S}\}$$

Notons $\mathcal{S} = \{x^s\}_{s=1}^{p}$ l'ensemble des points entiers réalisables (en bleu)



$P^1 \cap P^2 \subset P^1, P^2$
$conv(\mathcal{S}) \subset P^1, P^2$

Formulation idéale

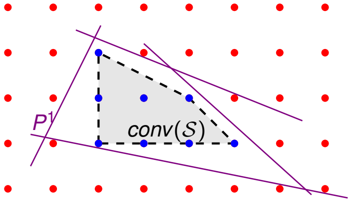
Polyèdre décrivant l'enveloppe convexe des solutions entières :

$$P = \text{conv}(\mathcal{S})$$

C'est-à-dire que les points extrêmes de P sont entiers

Obtention de la formulation idéale d'un problème entier

Soit le problème combinatoire $(PO)\ \max\{cx : x \in \mathcal{S}\}$



Comment obtenir une description complète de la formulation idéale pour un problème ?

Motivations

Si l'enveloppe convexe des points entiers est connu, il suffit de résoudre un problème linéaire (simplex retourne un point extrême)

Début de réponse

Il est aussi difficile d'obtenir une description de la formulation idéale d'un problème entier que de résoudre ce problème entier

Obtention de la formulation idéale d'un problème entier

Soit un programme entier : $\max\{cx : Ax \leq b, x \geq 0, x \in \mathbb{Z}^n\}$

Solution de la relaxation linéaire

La solution de la relaxation linéaire est de la forme $(x_{\mathcal{B}}, x_{\mathcal{N}}) = (B^{-1}b, 0)$
D'après la régle de Cramer

$$B^{-1} = \frac{com(B)}{det(B)} \quad \text{avec } com(B) \text{ comatrice de } B$$

Conséquence

Si $det(B) = \pm 1$ pour toute base B et $b \in \mathbb{Z}^m$ alors $x_{PL} \in \mathbb{Z}^n$

Matrices totalement uni-modulaire (TU)

Une matrice A est totalement uni-modulaire (TU) ssi toutes sous-matrices carrées de A a un déterminant égal à ± 1 ou 0

Propriété des matrices totalement uni-modulaire

(PL) $\max\{cx : Ax \leq b, x \geq 0\}$ a une solution optimale entière pour tout $b \in \mathbb{Z}^m$ ssi A est totalement uni-modulaire

Matrice totalement uni-modulaire

Exemples de matrice totalement uni-modulaire

- Matrice d'incidence d'un graphe

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccccc} (1,2) & (1,3) & (2,1) & (2,4) & (3,1) & (3,4) \end{array} \\ \left(\begin{array}{cccccc} +1 & +1 & -1 & 0 & -1 & 0 \\ -1 & 0 & +1 & +1 & 0 & 0 \\ 0 & -1 & 0 & 0 & +1 & +1 \\ 0 & 0 & 0 & -1 & 0 & -1 \end{array} \right) \end{array}$$

- Matrice avec la propriété de uns-consécutifs

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

Applications

- Problèmes de flot
- Problèmes de couplage

Troisième partie III

TECHNIQUES DE RÉSOLUTION PAR PLANTS COUPANTS/PAR GÉNÉRATION DE COLONNES

Aspects géométriques : polyèdres

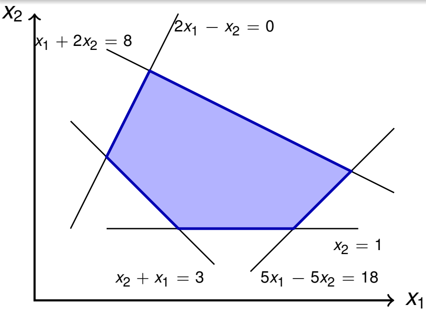
Hyperplans et polyèdres

Hyperplan : $H = \{x \in \mathbb{R}^n : \sum_{j=1}^n \alpha_j x_j = \beta,\ \alpha \neq \mathbf{0}_n\}$

Demi-espace fermé : $D = \{x \in \mathbb{R}^n : \sum_{j=1}^n \alpha_j x_j \leq \beta,\ \alpha \neq \mathbf{0}_n\}$

Polyèdre : intersection de demi-espaces fermés et d'hyperplans

$$P = \left\{x \in \mathbb{R}^n : \sum_{j=1}^n a_{ij} x_j \leq b_i \;\; i = 1, \ldots, m\right\}$$



- Un hyperplan : $2x_2 + x_1 - 8 = 0$
- Un demi-plan fermé : $2x_2 + x_1 - 8 \leq 0$
- Un polyèdre borné : $\left\{\begin{array}{l} x_1 + 2x_2 \leq 8, 5x_1 - 5x_2 \leq 18 \\ x_2 + x_1 \geq 3, 2x_1 - x_2 \geq 0, x_2 \geq 1 \end{array}\right\}$

Soit $M = \{1, \ldots, m\}$ est l'ensemble des indices des contraintes chaque inégalité $\sum_{j=1}^n a_{ij} x_j \leq b_i$ sera noté $A_i x \leq b_i$ avec $i \in M$

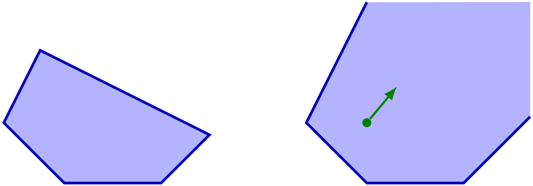
Aspects géométriques : polytopes

Soit le polyèdre $P = \{x \in \mathbb{R}^n : Ax \leq b\}$

Polyèdres bornés

Polyèdre borné (ou polytope) : P est borné si

il existe $u \geq 0$ tel que $\forall x \in P, |x_i| \leq u, \; i = 1, \ldots, n$



Polyèdre borné — Polyèdre non borné

Caractérisation des polyèdres non bornés

Le polyèdre P est non borné ssi il existe un point $\overline{x} \in P$ et une direction $d \neq \mathbf{0}_n \in \mathbb{R}^n$ tel que

$$\overline{x} + \lambda d \in P, \text{ pour tout } \lambda \geq 0$$

Aspects géométriques : points particuliers d'un polyèdre

Un polyèdre est de la forme $P = \{x \in \mathbb{R}^n : Ax \leq b\} \neq \emptyset$

Pour la suite, on considère seulement des polyèdres rationnels

C'est-à-dire que $A \in \mathbb{Q}^{m\times n}, b \in \mathbb{Q}^m$

Notations

- $(A^=, b^=)$: matrice correspondant aux lignes de (A, b) d'incides
$$M^= = \{i \in M : A_i x = b_i, \text{ pour tout point } x \in P\}$$
- $(A^\leq, b^\leq)$: matrice correspondant aux lignes de (A, b) d'indices
$$M^\leq = \{i \in M : A_i x < b_i, \text{ pour au moins un point } x \in P\}$$

Ainsi, P se réécrit $P = \left\{x \in \mathbb{R}^n : A^= x = b^=, A^\leq x \leq b^\leq\right\}$

Points particuliers d'un polyèdre

Point intérieur du polyèdre P : point $x \in P$ tel que $\forall i \in M^\leq, A_i x < b_i$

Proposition

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

Aspects géométriques : points affinements indépendants

Points affinements indépendants

Soient E un ensemble de $\mathbb{R}^n$ et $x^1, \ldots, x^\nu$ des points de E
Les points $x^1, \ldots, x^\nu$ sont affinements indépendants si l'unique solution

$$(\lambda_1, \ldots, \lambda_\nu) \in \mathbb{R}^\nu \text{ tel que } \sum_{k=1}^{\nu} \lambda_k = 0 \text{ et } \sum_{k=1}^{\nu} \lambda_k x^k = 0 \quad \text{est} \quad \lambda_k = 0, \ \forall k$$

Vecteurs linéairements indépendants

Soient E un espace vectoriel de $\mathbb{R}^n$ et $v^1, \ldots, v^\nu$ des vecteurs de E
Les vecteurs $v^1, \ldots, v^\nu$ sont linéairements indépendants si la solution

$$(\lambda_1, \ldots, \lambda_\nu) \in \mathbb{R}^\nu \text{ du système } \sum_{k=1}^{\nu} \lambda_k v^k = 0 \quad \text{est} \quad \lambda_k = 0, \ \forall k$$

Proposition

les points $x^1, \ldots, x^\nu$ sont affinements indépendants ssi
les vecteurs $x^2 - x^1, \ldots, x^\nu - x^1$ sont linéairements indépendants

Aspects géométriques : dimension d'un polyèdre

Soit le polyèdre $P = \{x \in \mathbb{R}^n : A^= x = b^=, A^\leq x \leq b^\leq\} \neq \emptyset$

Dimension d'un polyèdre

Dimension du polyèdre P **:** $\dim(P) = k$ si le nombre maximum de points de P affinement indépendants est $k+1$

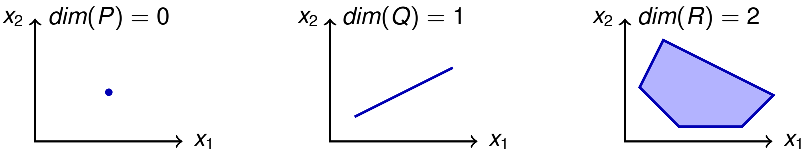
Polyèdre de pleine dimension : polyèdre P tel que $\dim(P) = n$

Proposition

La dimension de P est $\dim(P) = n - \text{rang}(A^=)$

D'où, un polyèdre P est de pleine dimension si

- P n'a pas de contraintes satisfaites à l'égalité pour tout point $x \in P$
- P admet une description minimale unique $P = \{x \in \mathbb{R}^n : A'x \leq b'\}$



Aspects géométriques : enveloppes convexe et conique

Espace convexe

Combinaison convexe des points $\mathcal{S} = \{x^s\}_{s=1}^p$ **:** $x \in \mathbb{R}^n$ tel que

$$x = \sum_{s=1}^p \lambda_s x^s \quad \text{avec } \sum_{s=1}^p \lambda_s = 1 \text{ et } \lambda_s \geq 0, s = 1, \ldots, p$$

Enveloppe convexe des points $\mathcal{S} = \{x^s\}_{s=1}^p$ **:**

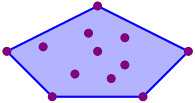
$$\text{conv}(\mathcal{S}) = \left\{ x \in \mathbb{R}^n : x = \sum_{s=1}^p \lambda_s x^s, \ \sum_{s=1}^p \lambda_s = 1, \ \lambda_s \geq 0 \ s = 1, \ldots, p \right\}$$

Combinaison positive des vecteurs $\mathcal{V} = \{r^t\}_{t=1}^q$ **:** $r \in \mathbb{R}^n$ tel que

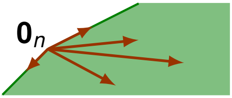
$$r = \sum_{t=1}^q \mu_t v^t \quad \text{avec } \mu_t \geq 0, t = 1, \ldots, q$$

Enveloppe conique des vecteurs $\mathcal{V} = \{r^t\}_{t=1}^q$ **:**

$$\text{cone}(\mathcal{V}) = \left\{ r \in \mathbb{R}^n : r = \sum_{t=1}^q \mu_t r^t, \ \mu_r \geq 0 \ r = 1, \ldots, q \right\}$$



Enveloppe convexe



Enveloppe conique
ou cône généré par $\mathcal{V}$

Lien entre optimisation combinatoire et prog. linéaire

Proposition

Soient $\mathcal{S} \subset \mathbb{R}^n$ un ensemble fini, dénombrable de points réalisables et $c \in \mathbb{R}^n$ un vecteur de coût, alors

$$\max \left\{ cx : x \in \mathcal{S} \right\} = \max \left\{ cx : x \in \mathrm{conv}(\mathcal{S}) \right\}$$

Remarque

Si $\mathcal{S} \subset \mathbb{R}^n$ est un ensemble fini, alors $\mathrm{conv}(\mathcal{S})$ est un polyèdre borné

Conséquence

Si $\mathrm{conv}(\mathcal{S})$ est connu, alors il suffit de résoudre un programme linéaire pour résoudre le problème $\max\{cx : x \in \mathcal{S}\}$

Il est donc utile de s'intéresser à la géométrie des polyèdres

Inégalités valides en programmation linéaire 1/2

Soit le polyèdre $P = \{x \in \mathbb{R}^n : Ax \leq b\} \neq \emptyset$

Quelles sont les inégalités nécessaires à la description de P ?

Inégalités linéaires valides

Inégalité valide : $\alpha x \leq \beta$ avec $\alpha \in \mathbb{R}^n$ et $\beta \in \mathbb{R}$ est valide pour P si

$$P \subseteq \{x \in \mathbb{R}^n : \alpha x \leq \beta\}$$

C'est-à-dire que l'inégalité est satisfaite par tous les points de P

Proposition

$\alpha x \leq \beta$ est une inégalité valide pour P ssi

il existe $u \in \mathbb{R}^m_+$ tel que $\alpha \leq uA$ et $ub \leq \beta$

Conséquence

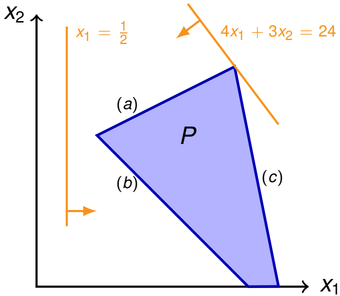
Toute combinaison d'inégalités de P est une inégalité valide :

pour tout vecteur $u \in \mathbb{R}^m_+$, $(\alpha, \beta) = (uA, ub)$ est valable

Inégalités valides en programmation linéaire 2/2

Soit $Z^{PL} = \max\{4x_1 + 3x_2 : (x_1, x_2) \in P\}$
avec

$$P = \left\{ (x_1, x_2) \in \mathbb{R}^2 : \begin{array}{llll} -x_1 + 2x_2 \leq 4 & (a), & 5x_1 + x_2 \leq 20 & (c) \\ 2x_1 + 2x_2 \geq 7 & (b), & x_1 \;,\; x_2 \geq 0 & (d)-(e) \end{array} \right\}$$



Exemples d'inégalités valides pour P

- $-2x_1 \leq -1$ valide car :
  en posant $u = (\frac{2}{3}, \frac{2}{3}, 0, 0, 0)$ on a
  $uA = (-2, 0)$ et $ub = -2$
  et ainsi $\alpha = uA$ et $\beta \geq ub$
- $\frac{2}{3}(a) + \frac{2}{3}(b)$ donne $x_1 \geq 1$
  ici $u = (\frac{2}{3}, \frac{2}{3}, 0, 0, 0)$
- $(a) + (c)$ donne $4x_1 + 3x_2 \leq 24$
  ici $u = (1, 0, 1, 0, 0)$
  remarque : $Z^{PL} \leq 24$ et u représente une solution duale

Aspects géométriques : faces d'un polyèdre

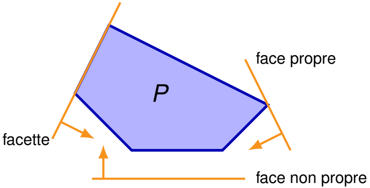
Soit le polyèdre $P = \{x \in \mathbb{R}^n : A^{\leq}x \leq b^{\leq}, A^{=}x = b^{=}\} \neq \emptyset$

Faces

Face : F est une face de P s'il existe une inégalité valide $\alpha x \leq \beta$ (induisant la face) tel que :

$$F = \{x \in P : \alpha x = \beta\}$$

Face propre : F est une face propre de P si $F \neq \emptyset$ et $F \neq P$



Remarques

- Toute face propre définit un polyèdre tel que $\dim(F) \leq \dim(P) - 1$
- Une face $F = \{x \in P : \alpha x = \beta\}$ est non vide ssi $\max\{\alpha x : x \in P\} = \beta$

Aspects géométriques : facettes d'un polyèdre

Soit le polyèdre $P = \{x \in \mathbb{R}^n : A^{\leq}x \leq b^{\leq}, A^{=}x = b^{=}\} \neq \emptyset$

Facettes

Facette : F est une facette de P si F est une face propre de P telle que :

$$\dim(F) = \dim(P) - 1$$

Remarque

Les facettes sont les faces propres maximales (au sens de l'inclusion)

Proposition (condition suffisante à la description de P)

Supposons que $P \neq \emptyset$. Alors toute contrainte de $A^{\leq}x \leq b^{\leq}$ qui n'induit pas une facette est redondante

Description du polyèdres par les facettes

Proposition

Un sous-ensemble F non vide de P est une face ssi il existe un sous-système $A'x \leq b'$ de $A^{\leq}x \leq b^{\leq}$ tel que $F = \{x \in P : A'x = b'\} \neq \emptyset$

Proposition (condition nécessaire à la description de P)

Pour toute facette F de P, une des inégalités définissant F est nécessaire à la description de P

Théorème

Le polyèdre P définissant P est minimal si et seulement si

- les lignes de $A^{=}$ sont linéairement indépendantes
- toute inégalité de $A^{\leq}x \leq b^{\leq}$ définit une facette distincte de P

Corollaire

Si P est un polyèdre de pleine dimension, alors il existe un système linéaire minimal unique qui décrit P

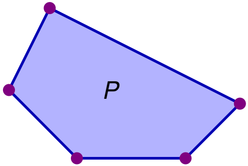
Aspects géométriques : points extrêmes

Soit le polyèdre $P = \{x \in \mathbb{R}^n : Ax \leq b\} \neq \emptyset$

Quels sont les points nécessaires à la description de P ?

Points extrêmes d'un polyèdre

Point extrême du polyèdre P : $\widetilde{x} \in P$ point extrême de P si il n'existe pas $x^1, x^2 \in P$ $(x^1 \neq x^2)$ tel que $\widetilde{x} = \frac{1}{2}x^1 + \frac{1}{2}x^2$



L'ensemble convexe des points extrêmes de P est noté :

$$\boxed{\text{conv}(P) = \text{conv}(\{x^s\}_{s=1}^p)}$$

Proposition

$\widetilde{x} \in P$ est un point extrême ssi $\{\widetilde{x}\}$ est une face de dimension 0

Remarque

Les points extrêmes sont les faces propres minimales

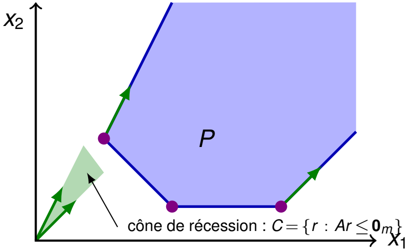
Aspects géométriques : rayons extrêmes

Soit le polyèdre $P=\{x \in \mathbb{R}^n : Ax \leq b\}$ et le cône $C=\{r \in \mathbb{R}^n : Ar \leq \mathbf{0}_m\}$

Rayons d'un polyèdre

Rayon du polyèdre P : $\tilde{r}$ rayon de P si $\tilde{r} \in C \setminus \{\mathbf{0}_n\}$ ($\tilde{r} \neq \mathbf{0}_n$)

Rayon extrême du polyèdre P : $\tilde{r} \in C$ rayon extrême de P si il n'existe pas $r^1, r^2 (r^1 \neq \lambda r^2)$ rayons de P tel que $\tilde{r} = \frac{1}{2} r^1 + \frac{1}{2} r^2$



L'ensemble conique des rayons extrêmes de P est noté :

$$\boxed{\text{cone}(P) = \text{cone}(\{r^t\}_{t=1}^q)}$$

En particulier, le cône de récession de P vérifie $\text{cone}(P) = C$

Proposition

$\tilde{r} \in C$ est un rayon extrême ssi $\{\lambda \tilde{r} : \lambda \geq 0\}$ est une face de C de dim 1

Propriétés diverses sur les polyèdres

Soit $P = \{x \in \mathbb{R}^n : Ax \leq b\} \neq \emptyset$

Propriétés des polyèdres bornés

- aucun rayons
- $\max\{cx : x \in P\}$ admet une solution optimale en un point extrême
- pour tout point extrême x^*, il existe un vecteur coût $c \in \mathbb{R}^n$ tel que :

$$x^* \text{ est optimal pour } \max\{cx : x \in P\}$$

Propriétés des polyèdres non bornés

- $r \in \text{cone}(P)$ est un rayon de P ssi

$$\{y \in \mathbb{R}^n : y = x + \lambda r, \lambda \geq 0\} \subseteq P \text{ pour tout } x \in P$$

  C'est-à-dire que tout déplacement suivant un rayon est réalisable
- si l'objectif croît le long d'un rayon $r \in \text{cone}(P)$ de P alors le problème est non borné : s'il existe $r \in \text{cone}(P)$ tel que $cr > 0$ alors

$$\forall x \in P,\ (x + \lambda r) \xrightarrow[\lambda \to \infty]{} \infty$$

Description du polyèdre par les points et rayons

Soit un polyèdre $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ tel que rang$(A) = n$

Théorème

Soient $\mathcal{X} = \{x^s\}_{s=1}^p$ et $\mathcal{R} = \{r^t\}_{t=1}^q$ les points et rayons extrêmes de P
Alors, pour tout vecteur $c \in \mathbb{R}^n$, le problème $\max\{cx : x \in P\}$

- est irréalisable ssi $p = 0$
- est non borné ssi $p > 0$ et il existe t tel que $cr^t > 0$
- a une solution optimale z^* ssi $p > 0$ et $cr^t \leq 0$ pour tout t

Théorème de Minkowski

Si $P \neq 0$, alors

$$P = \left\{ x = \sum_{s=1}^{p} \lambda_s x^s + \sum_{t=1}^{q} \mu_t r^t : \sum_{s=1}^{p} \lambda_s = 1,\ \lambda_s \geq 0 \ \ \forall s,\ \mu_t \geq 0 \ \ \forall t \right\}$$

Avec $\{x^s\}_{s=1}^p$ et $\{r^t\}_{t=1}^q$ l'ensemble des points et rayons extrêmes de P

$$P = \text{conv}(P) + \text{cone}(P)$$

Résumé de la situation

1. Un polyèdre peut être décrit de deux façons différentes :
   - soit à l'aide de facettes
   - soit à l'aide de points/rayons extrêmes
2. Application à l'optimisation combinatoire :
   - essayer de déterminer des facettes de conv$(\mathcal{S})$
   - essayer de connaître les points extrêmes de conv$(\mathcal{S})$

Chacune de ces visions donne des méthodes différentes :

1. Génération de coupes
2. Génération de colonnes

Soit

$$(PO) \quad \max\{cx : x \in \mathrm{conv}(\mathcal{S}\})$$

Comment trouver des inégalités valides pour (PO) ?

Comment montrer que ces inégalités sont des facettes ?

Optimisation et Séparation

Soit le problème d'optimisation :

$$(PO) \ \min\{cx : x \in P\}$$

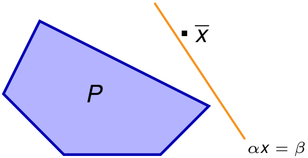
avec $P = \{x \in \mathbb{R}^n : Ax \leq b\}$

Problème de séparation (PS)

Etant donné $\overline{x} \in \mathbb{R}^n$, déterminer si $\overline{x} \in P$

Et si $\overline{x} \notin P$, déterminer un hyperplan séparant $\overline{x}$ de P

C'est-à-dire chercher une inégalité valide $\alpha x \leq \beta$ de P tel que $\alpha\overline{x} > \beta$



Thèorème optimisation/séparation

Le problème d'optimisation (PO) est résolu en temps polynomial ssi le problème de séparation (PS) est résolu en temps polynomial

Application à l'optimisation combinatoire

Soit un problème d'optimisation combinatoire :

$$\max_{S \subseteq N} \left\{ \sum_{j \in S} c_j : S \in \mathcal{S} \right\}$$

où

- $N = \{e_1, \ldots, e_n\}$ est un ensemble de base fini
- $\mathcal{S}$ est une famille de sous-ensemble de N formant l'ensemble des solutions réalisables

vecteur d'incidence

A chaque élément S de l'ensembe $\mathcal{S}$ est associé un vecteur d'incidence $x^S \in \{0, 1\}^{|N|}$:

$$x_j^S = \begin{cases} 1 & \text{si } j \in S \\ 0 & \text{sinon} \end{cases}$$

Ainsi, l'ensemble des solutions réalisables $\mathcal{S}$ est décrit par :

$$\mathcal{S} = \{x^s\}_{s=1}^{p}$$

où x^s est un vecteur d'incidence d'une solution réalisable S

Description du polytope des solutions

Soit un problème d'optimisation combinatoire :

$$\max\{cx : x \in \mathcal{S}\} \qquad \text{avec } \mathcal{S} = \{x^s\}_{s=1}^{p}$$

Polytope des solutions

$$\text{conv}(\mathcal{S}) = \left\{ \sum_{s=1}^{p} \lambda_s x^s : \sum_{s=1}^{p} \lambda_s = 1,\ \lambda_s \geq 0\ s = 1, \ldots, p \right\}$$

Proposition

Si $\tilde{x} \notin \text{conv}(\mathcal{S})$, alors il existe une inégalité qui sépare $\tilde{x}$ et $\text{conv}(\mathcal{S})$

Théorème

Un ensemble non vide de points $P \subseteq \mathbb{R}^n$ est un polytope ssi il existe un ensemble de points $\mathcal{S}$ tel que $P = \text{conv}(\mathcal{S})$

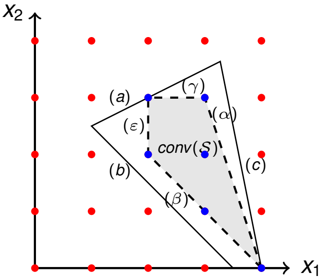
Conséquences

- Si $\mathcal{S}$ est fini, $\text{conv}(S)$ est donc un polytope
- On a équivalence entre $\max\{cx : x \in \mathcal{S}\}$ et $\max\{cx : x \in \text{conv}(\mathcal{S})\}$

Motivations d'une étude polyèdrale 1/2

Soit $\mathcal{S} = P \cap \mathbb{Z}^2$ avec

$$P = \left\{ (x_1, x_2) \in \mathbb{R}^2 : \begin{array}{llll} -x_1 + 2x_2 \leq 4 & (a), & 5x_1 + x_2 \leq 20 & (c) \\ 2x_1 + 2x_2 \geq 7 & (b), & x_1 \;,\; x_2 \geq 0 & (d)-(e) \end{array} \right\}$$



$\mathcal{S} = \{x^t\}_{t=1}^T$: points réalisables entiers

Ici, $\mathcal{S} = \left\{ \begin{array}{l} (2,2),(2,3),(3,1) \\ (3,2),(3,3),(4,0) \end{array} \right\}$

$$\mathrm{conv}(\mathcal{S}) = \left\{ (x_1, x_2) \in \mathbb{R}^2 : \begin{array}{llll} x_1 + x_2 \geq 4 & (\alpha), & x_1 \geq 2 & (\gamma) \\ 3x_1 + x_2 \leq 12 & (\beta), & x_2 \leq 3 & (\varepsilon) \end{array} \right\}$$

Est-il possible de trouver rapidement l'enveloppe convexe des solutions réalisables ?

Motivations d'une étude polyèdrale 2/2

Soit un problème d'optimisation combinatoire

$$(POC) \quad \max\{cx : x \in \mathcal{S}\}$$

avec $\mathcal{S} = \{x^s\}_{s=1}^{p}$ un ensemble fini de points réalisables

Condition de résolution de (POC) par la programmation linéaire

Connaître les inégalités linéaires décrivant conv$(\mathcal{S})$

Objectif : décrire le polyèdre conv$(\mathcal{S})$

$$\text{Trouver } \widetilde{A}, \widetilde{b} \text{ tel que } \mathrm{conv}(\mathcal{S}) = \{x \in \mathbb{R}^n : \widetilde{A}x \leq \widetilde{b}\}$$

Conjecture

Pour les problèmes linéaires entiers « difficile », il n'existe pas de description compact de conv$(\mathcal{S})$ (nombre polynomial de contraintes)

Cependant on peut tenter d'obtenir une meilleure approximation de conv$(\mathcal{S})$ (surtout autour de l'objectif) par l'ajout d'inégalités valides

Inégalités valides en programmation linéaire entière 1/2

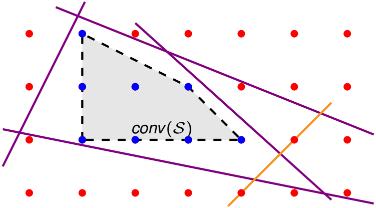
Soit un problème d'optimisation combinatoire

$$(POC) \quad \max\{cx : x \in \mathcal{S}\}$$

avec $\mathcal{S} = \{x^s\}_{s=1}^{p}$ un ensemble fini de points réalisables

Inégalités valides pour un problème d'optimisation combinatoire

Inégalité valide : $\alpha x \leq \beta$ avec $\alpha \in \mathbb{R}^n$ et $\beta \in \mathbb{R}$ est valide pour $\mathcal{S}$ si $\alpha x \leq \beta$ est satisfaite pour tout point $x \in \mathcal{S}$



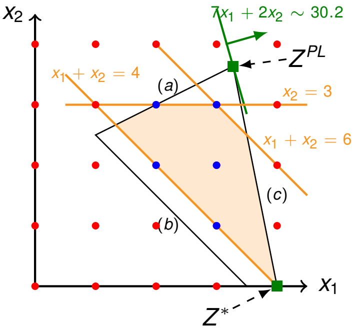
Lien avec le polytope des solutions

une inégalité valide est valide pour $\mathcal{S}$ ssi elle est valide pour le polyèdre conv($\mathcal{S}$) : conv($\mathcal{S}$) $\subseteq \{x \in \mathbb{R}^n : \alpha x \leq \beta\}$

Inégalités valides en programmation linéaire entière 2/2

Soit $Z^* = \{7x_1 + 2x_2 : (x_1, x_2) \in \mathcal{S}\}$ avec $\mathcal{S} = P \cap \mathbb{Z}^2$ et

$$P = \left\{ (x_1, x_2) \in \mathbb{Z}^2 : \begin{array}{rlrl} -x_1 + 2x_2 \leq 4 & (a), & 5x_1 + x_2 \leq 20 & (c) \\ 2x_1 + 2x_2 \geq 7 & (b), & x_1 \;,\; x_2 \geq 0 & (d)-(e) \end{array} \right\}$$



Exemples d'inégalités valides pour $\mathcal{S}$

Comme $x \in \mathbb{Z}^2$,

- (b) : $x_1 + x_2 \geq \frac{7}{2}$
  $\Rightarrow x_1 + x_2 \geq \left\lceil \frac{40}{11} \right\rceil = 4$ valide
- $5(a) + (c)$: $11x_2 \leq 40 \Leftrightarrow x_2 \leq \frac{40}{11}$
  $\Rightarrow x_2 \leq \left\lfloor \frac{40}{11} \right\rfloor = 3$ valide
- $4(a) + 3(c)$: $x_1 + x_2 \leq \frac{76}{11}$
  $\Rightarrow x_1 + x_2 \leq \left\lfloor \frac{76}{11} \right\rfloor = 6$ valide

Ajout de contraintes simples : contraintes logiques

Arrondissement des bornes

Soit $\mathcal{S} = \{x \in \mathbb{Z} : l_i \leq x_i \leq u_i\}$, alors $\lceil l_i \rceil \leq x_i \leq \lfloor u_i \rfloor$ est valide

Exemple : soit $\mathcal{S} = \{x \in \mathbb{Z} : 0.2 \leq x \leq 1.3\} \quad \Rightarrow \quad x = 1$

Ajout de contraintes logiques

Soit un problème entier avec des variables binaires : $x_i \in \{0, 1\}, \ \forall i$
Déduire des contraintes logiques simples à partir de contraintes impliquant plusieurs variables x

Exemple : considérons

$$\mathcal{S} = \{(x_1, x_2, x_3, x_4) \in \{0, 1\}^4 : 7x_1 + 3x_2 - 4x_3 - 2x_4 \leq 1\}$$

- si $x_1 = 1$, on doit avoir $x_3 = x_4 = 1$, donc
$$x_1 \leq x_3 \quad \text{et} \quad x_1 \leq x_4$$
- si $x_2 = 1$, on doit avoir $x_3 = 1$ ou $x_4 = 1$, donc
$$x_2 \leq x_3 + x_4$$
- si $x_1 = x_2 = 1$ alors la contrainte est violée, donc
$$x_1 + x_2 \leq 1$$

Approche polyèdrale

Le problème de séparation

Etant donné $\overline{x} \in \mathbb{R}^n$, déterminer si $\overline{x} \in \text{conv}(\mathcal{S})$
Et si $\overline{x} \notin \text{conv}(\mathcal{S})$, déterminer un hyperplan séparant $\overline{x}$ de $\text{conv}(\mathcal{S})$
C'est-à-dire chercher une inégalité valide $\alpha x \leq \beta$ de P tel que $\alpha \overline{x} > \beta$

Cependant, $\text{conv}(\mathcal{S})$ n'est pas connu a priori. Que faire ?

Approche polyèdrale

Etude consistant à déterminer des inégalités valides pour approximer le polyèdre $\text{conv}(\mathcal{S})$

- Générer des inégalités valides à l'aide de combinaisons de contraintes et d'arrondis
- Trouver des inégalités valides en étudiant les propriétés combinatoires du problème

Inégalités génériques : coupes de Chvátal-Gomory

Proposition

Soit $\mathcal{S} = \{y \in \mathbb{Z} : y \leq b\}$, l'inégalité suivante est valide pour $\mathcal{S}$:

$$\boxed{y \leq \lfloor b \rfloor}$$

Exemple

Soit $\mathcal{S} = P \cap \mathbb{Z}^2$ la région entière avec $P = \{x \in \mathbb{R}^2 : 11x_1 + 11x_2 \leq 76\}$
En posant $y = x_1 + x_2 \in \mathbb{Z}$, on en déduit l'inégalité $x_1 + x_2 \leq \left\lfloor \frac{76}{11} \right\rfloor = 6$

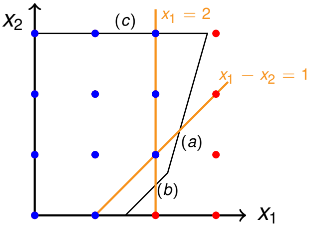
Proposition

Soit A est une matrice $m \times n$ avec $(A^1, \ldots, A^n)$ les colonnes de A
Soit $\mathcal{S} = P \cap \mathbb{Z}^n$ avec $P = \{x \in \mathbb{R}^n : Ax \leq b, x \geq \mathbf{0}_n\}$
Alors, pour tout $u \in \mathbb{R}^m_+$, l'inégalité suivante est valide pour $\mathcal{S}$

$$\boxed{\sum_{i=1}^n \lfloor uA^i \rfloor x_i \leq \lfloor ub \rfloor}$$

Exemple de coupes de Chvátal-Gomory

1. Soit $\mathcal{S} = P \cap \mathbb{Z}^2$ avec
$P = \{x \in \mathbb{R}^2_+ : 7x_1 - 2x_2 \leq 14\ (a),\ 2x_1 - 2x_2 \leq 3\ (b),\ x_2 \leq 3\ (c)\}$



En prenant $u = (\frac{1}{7}, 0, \frac{2}{7})$, on obtient l'inégalité $x_1 \leq \left\lfloor \frac{20}{7} \right\rfloor = 2$

En prenant $u = (0, \frac{1}{2}, 0)$, on obtient l'inégalité $x_1 - x_2 \leq \left\lfloor \frac{3}{2} \right\rfloor = 1$

2. Soit $\mathcal{S} = P \cap \mathbb{Z}^3$ avec
$P = \{x \in \mathbb{R}^3_+ : 13x_1 + 20x_2 + 11x_3 \geq 30 \quad (a), \quad x_1 + x_2 \geq 1 \quad (b)\}$
En prenant $u = (\frac{1}{11}, \frac{2}{11})$, on obtient $\frac{15}{11}x_1 + 2x_2 + x_3 \geq \frac{32}{11}$
Et donc,

$$x_1 + 2x_2 + x_3 \geq \left\lceil \frac{32}{11} \right\rceil = 3$$

Utilisation des coupes de Chvátal-Gomory

Les inégalités de Chvátal-Gomory sont-elles utilisables ?

Théorème

Chaque inégalité valide pour $\mathcal{S}$ peut être obtenue en appliquant la procédure de Chvátal-Gomory un nombre fini de fois

Applicabilité des inégalités de Chvátal-Gomory

- En théorie, toutes les inégalités valides qui décrivent conv($\mathcal{S}$) peuvent être obtenues à l'aide de la procédure
  Cependant, déterminer les bons cœfficients u restent difficile
- En pratique, cette procédure est utilisée pour prouver que des (familles d')inégalités sont valables

L'objectif est de trouver des familles d'inégalités valides

Qualité des inégalités valides

Soit $\mathcal{S} = P \cap \mathbb{Z}^n$ avec $P = \{x \in \mathbb{R}^n : Ax \leq b, x \geq \mathbf{0}_n\}$

Dominance d'inégalites valides

L'inégalité valide $\alpha x \leq \beta$ de $\mathcal{S}$ est dominée par l'ensemble des inégalités valides $\alpha^i x \leq \beta^i$ ($i = 1, \ldots, m$) s'il existe $\lambda \in \mathbb{R}^m_+$ tel que :

$$\big(\alpha, \beta\big) \neq \Big(\sum_{i=1}^{m} \lambda_i \alpha^i, \sum_{i=1}^{m} \lambda_i \beta^i\Big), \quad \alpha \leq \sum_{i=1}^{m} \lambda_i \alpha^i \quad \text{et} \quad \beta \geq \sum_{i=1}^{m} \lambda_i \beta^i$$

Le rang de Chvátal par rapport à P de l'inégalité valide $\alpha x \leq \beta$ est :

- 0 si elle est équivalente (ou dominée) aux inégalités $Ax \leq b$
- k si elle est équivalente (ou dominée) à une combinaison d'inégalités de rang inférieur à $k-1$ et obtenues par la procédure

Plus le rang de l'inégalité est élevé, meilleure sera l'inégalité

Description des faces du polyèdre conv($\mathcal{S}$)

Si F est une face non vide de conv($\mathcal{S}$) de dimension $p-1$, alors il existe p points $x^1, \ldots, x^p \in \mathcal{S}$ affinements indépendants tel que $x^i \in F$, $\forall i$

Techniques de preuves de facettes

Soit $\alpha x \leq \beta$ une inégalité valide de $\text{conv}(\mathcal{S}) = \{x \in \mathbb{R}^n : \widetilde{A}x \leq \widetilde{b}\}$
Supposons que $\text{conv}(\mathcal{S})$ est de pleine dimension

1. **Preuve de nécessité :** montrer que $\alpha x \leq \beta$ est essentielle à la description de $\text{conv}(\mathcal{S})$: montrer qu'il existe un point $\overline{x} \in \mathbb{R}^n$ tel que $\alpha\overline{x} > \beta$ et
   $\overline{x}$ satisfaisait toutes les autres inégalités de $\widetilde{A}x \leq \widetilde{b}$
2. **Preuve directe :** exhiber n points affinements indépendantes $x^1, \ldots, x^n$ de $\mathcal{S}$ tels que
   $$\alpha x^i \leq \beta, \qquad i = 1, \ldots, n$$
3. **Preuve par maximalité :** montrer que la face F induit par $\alpha x \leq \beta$ n'est contenue dans aucune facette :
   Supposons qu'il existe une facette de $\text{conv}(\mathcal{S})$ induite par
   $$\alpha' x \leq \beta' \text{ tel que } F \subseteq \{x \in \text{conv}(\mathcal{S}) : \alpha' x = \beta'\}$$
   montrons que, dans ce cas, $\alpha' = \lambda\alpha$ avec $\lambda > 0$

Application au polytope des stables 1/3

Etude du problème de la recherche du stable de cardinalité maximal

Données : $G = (V, E)$ un graphe non orienté

Objectif : trouver un ensemble stable de cardinalité maximale

Polytope des stables : soit $x^S \in \mathbb{R}^{|V|}$ vecteur d'incidence du stable S :

$$x_u^S = \begin{cases} 1 \text{ si } u \in S \\ 0 \text{ sinon} \end{cases} \qquad \forall u \in V$$

Le polytope des stables est noté

$$P(G) = \text{conv}\left\{x^S \in \mathbb{R}^{|V|} : S \subseteq V \text{ est un stable}\right\}$$

Modélisation : $\max\{\sum_{u \in V} x_u : x \in P(G)\}$

Proposition

Le polytope $P(G)$ est de pleine dimension

Proposition

Soit $\sum\limits_{u \in V} \alpha_u x_u \leq \beta$ une inégalité valide de $P(G)$ distincte de $x_u \geq 0, \forall u \in V$

Si l'inégalité induit une facette de $P(G)$, alors $\alpha_u \geq 0, \forall u \in V$

Application au polytope des stables 2/3

$$\max \left\{ \sum_{u \in V} x_u : x_u + x_v \leq 1,\ \forall \{u,v\} \in E;\ \ x_u \in \{0,1\},\ \forall u \in V \right\}$$

Proposition

Les inégalités $x_u \geq 0, \forall u \in S$ induisent des facettes de $P(G)$

Clique

Clique : ensemble $C \subseteq V$ tel que $\{u,v\} \in E$ pour tout $u, v \in C\,(u \neq v)$

Contraintes valides sur les cliques

Soit $C \subseteq V$ une clique de cardinalité maximale, alors
$\sum_{u \in C} x_u \leq 1$ est une inégalité valide induisant une facette de $P(G)$

Contraintes valides sur les cycles impairs

Soit C un cycle de longueur impair, alors

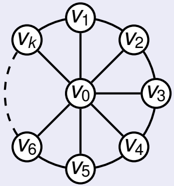
$$\sum_{u \in C} x_u \leq \left\lfloor \frac{|C|}{2} \right\rfloor = \frac{|C|-1}{2} \text{ est valide}$$

Application au polytope des stables 3/3

$$\max\left\{\sum_{u\in V} x_u : x_u + x_v \leq 1,\ \forall\{u,v\}\in E;\ \ x_u\in\{0,1\},\ \forall u\in V\right\}$$

Roue d'un graphe

Une roue de longueur k est le graphe suivant :



Cette roue sera noté $R = \{v_0, C\}$ où $C = [v_1, \ldots, v_k, v_1]$ est un cycle

Contraintes valides sur les roues impaires

Soit $R = \{v_0, C\}$ une roue de longueur impaire, alors

$$\sum_{u\in C} x_u + \frac{|C|-1}{2} x_0 \leq \frac{|C|-1}{2} \text{ est une inégalité valide}$$

induisant une facette si v_0 n'est adjacent à aucun sommet de $V \setminus R$

Motivations d'un ajout itérative d'inégalités valides

Soit le problème linéaire entier :

$$(PLE) \quad Z^* = \max\{cx : x \in \mathcal{S}\}$$

avec $\mathcal{S} = P \cap \mathbb{Z}^n$ et $P = \{x \in \mathbb{R}^n : Ax \leq b\}$
Soit $\mathcal{F} = \{(\alpha, \beta) : \alpha x \leq \beta \text{ IV pour } \mathcal{S}\}$ une famille d'inégalités valides

Ajouter toutes inégalités valides de $\mathcal{F}$ au départ ?

Notons $P^{\mathcal{F}} = P \cap \{x \in \mathbb{R}^n : \alpha x \leq \beta,\ \forall (\alpha, \beta) \in \mathcal{F}\}$

Etudier le polyèdre $P^{\mathcal{F}}$ plutôt que le polyèdre P ?

Avantages :

- obtention de meilleure bornes
- diminution de la taille d'un arbre de séparation et évaluation

Inconvénients :

- taille du problème plus important (simplex résolu avec plus de contraintes)
- nombre souvent très important d'inégalités valides à ajouter → résolution trop longue

Générer plutôt des coupes (inégalités valides) au fur et à mesure

Procédure de séparation

Soit le programme linéaire entier :

$$(PLE) \quad Z^* = \max\{cx : x \in \mathcal{S}\}$$

avec $\mathcal{S} = P \cap \mathbb{Z}^n$ et $P = \{x \in \mathbb{R}^n : Ax \leq b\}$
Soit $\mathcal{F} = \{(\alpha, \beta) : \alpha x \leq \beta$ IV pour $\mathcal{S}\}$ une famille d'inégalités valides

Procédure de séparation

Etant donné

- une famille d'inégalités valables (notée $\mathcal{F}$) et
- une solution de la relaxation linéaire de (PLE) (notée $\widehat{x}$)

algorithme qui cherche et renvoie une inégalité de $\mathcal{F}$ violée par $\widehat{x}$

Algorithme de séparation pour une famille d'inégalité $\mathcal{F}$

Entrée : la solution de la relaxation linéaire $\widehat{x} = \text{argmax}\{cx : x \in P\}$
Sortie : retourner une coupe $\alpha x \leq \beta \in \mathcal{F}$ tel que $\alpha\widehat{x} > \beta$

La procédure de séparation dépend de la famille d'inégalités considérée

Algorithme des plans coupants 1/2

Soit le problème linéaire entier :

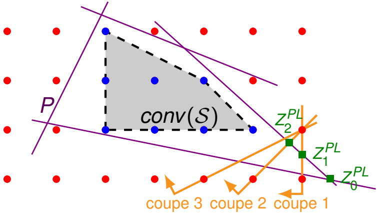
$$(PLE) \quad Z^* = \max\{cx : x \in \mathcal{S}\}$$

avec $\mathcal{S} = P \cap \mathbb{Z}^n$ et $P = \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}$

Principe de l'algorithme des plans coupants

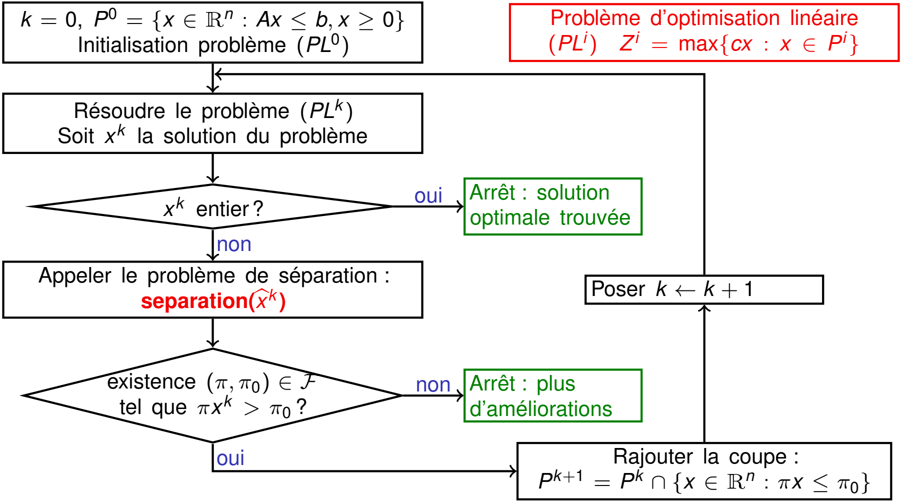
Etant donné une famille d'inégalités valides $\mathcal{F}$,

algorithme qui consiste à séparer les solutions de la relaxation linéaire par l'ajout d'inégalités valides de $\mathcal{F}$



Génération itérative d'inégalités valides de $\mathcal{F}$ (si la procédure de séparation en trouvent) coupant les solutions de la relaxation linéaire

Algorithme des plans coupants 2/2



Amélioration de la relaxation linéaire à chaque ajout d'inégalités valides

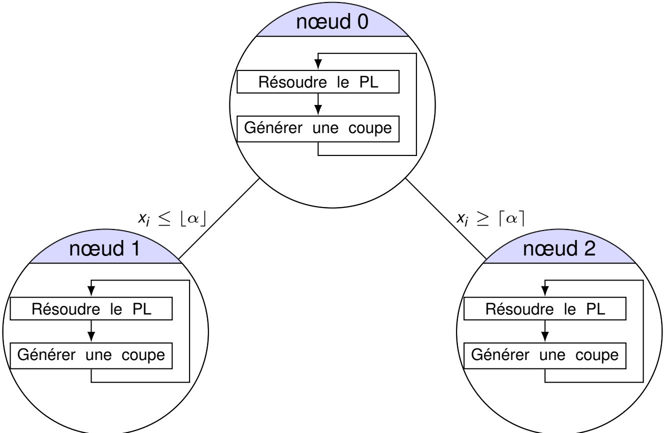
$$\text{conv}(\mathcal{S}) \subseteq P^k \subseteq \cdots \subseteq P^1 \subseteq P^0 \quad \Rightarrow \quad Z^* \leq Z_k^{PL} \leq \cdots \leq Z_1^{PL} \leq Z_0^{PL}$$

Algorithme de séparation et coupure

Algorithme de « Branch and Cut »

combinaison de deux mméthodes :

- un algorithme de séparation et évaluation
- une procédure de plans coupants à chaque nœud



Inégalités combinatoires pour le problème de sac-à-dos 1

Données : W : capacité maximale du sac
I : ensemble de n objets avec pour chaque objet $i \in I$:
p_i l'utilité et w_i le poids

Formulation : $\max \left\{ \sum_{i \in I} p_i x_i : \sum_{i \in I} w_i x_i \leq W, \quad x \in \{0,1\}^{|I|} \right\}$

Couverture du sac

L'ensemble $C \subseteq I$ couvre le sac si $\sum_{i \in C} w_i > W$

Inégalité de couverture (cover inequality)

Si l'ensemble $C \subseteq I$ couvre le sac, alors

$$\sum_{i \in C} x_i \leq |C| - 1 \text{ est valide}$$

Remarque

L'inégalité se généralise sur un problème à plusieurs contraintes de sac-à-dos ; en particulier, pour le polyèdre $P = \{x \in \mathbb{R}^n_+ : Ax \leq b\}$

Inégalités combinatoires pour le problème de sac-à-dos 2

Soit $X^{PL} = \{x \in [0,1]^{|I|} : \sum_{i \in I} w_i x_i \leq W\}$ les solutions réelles
Soit $\mathcal{F} = \{C \subseteq I : \sum_{i \in C} w_i > W\}$ la famille des couvertures

Procédure de séparation

Etant donné $\widehat{x} \in X^{PL}$,

Chercher une couverture $C \in \mathcal{F}$ **tel que** $\sum_{i \in C} \widehat{x}_i > |C| - 1$

S'il existe une telle couverture, ajouter l'inégalité $\sum_{i \in C} \widehat{x}_i \leq |C| - 1$

Recherche d'une couverture violée

Soit C la couverture cherchée, posons $z_i = \begin{cases} 1 \text{ si } i \in C \\ 0 \text{ si } i \notin C \end{cases}, \quad \forall i \in I$

Existe-t-il $z \in \{0,1\}^{|I|}$ t.q. $\sum_{i \in I} w_i z_i > W$ et $\sum_{i \in I} \widehat{x}_i z_i > \sum_{i \in I} z_i - 1$?

$\Leftrightarrow$ Résoudre $\zeta = \min\{\sum_{i \in I}(1 - \widehat{x}_i) z_i : \sum_{i \in I} w_i z_i > W, z \in \{0,1\}^{|I|}\} < 1$?

si $\zeta < 1$ on ajoute une inégalité violée ; sinon, il en existe pas

Trouver ζ revient à résoudre un problème de sac-à-dos !!

Inégalités combinatoires pour le problème de sac-à-dos 3

Soient un sac de capacité 6 ($W = 6$) et cinq objets ($n = 5$) :

i	1	2	3	4	5
p_i	30	28	27	34	7
w_i	2	2	3	4	1
p_i/w_i	15	14	9	8.5	7

Rappel : calcul de la relaxation linéaire pour le problème de sac-à-dos

1. Trier les objets dans l'ordre décroissant du rapport profit/poids ($\frac{p_i}{w_i}$)
2. Poser, pour chaque objet $k \in I$:

$$x_k = \begin{cases} 1 & \text{si } \sum_{i=1}^{k} w_i \leq W \\ \frac{W - \sum_{i=1}^{k-1} w_i}{w_k} & \text{si } \sum_{i=1}^{k} w_i > W \quad \text{et} \quad \sum_{i=1}^{k-1} w_i \leq W \\ 0 & \text{sinon} \end{cases}$$

La solution de la relaxation linéaire est donc :

$$x_{PL}^1 = (1, 1, \tfrac{2}{3}, 0, 0) \quad \text{et} \quad Z_1^{LP} = 76$$

Inégalités combinatoires pour le problème de sac-à-dos 4

i	1	2	3	4	5
p_i	30	28	27	34	7
w_i	2	2	3	4	1
p_i/w_i	15	14	9	8.5	7

et $W = 6$

1. La solution réelle est $x^1_{PL} = (1, 1, \frac{2}{3}, 0, 0)$ et $Z^{PL}_1 = 76$
   Or $w_1 + w_2 + w_3 = 7 > W$ et $x_1 + x_2 + x_3 = \frac{8}{3} > 2$
   On rajoute donc l'inégalité :
   $$x_1 + x_2 + x_3 \leq 2 \quad (a)$$
2. La nouvelle solution réelle est $x^2_{PL} = (1, 1, 0, \frac{1}{2}, 0)$ et $Z^{PL}_2 = 75$
   Or $w_1 + w_2 + w_4 = 8 > W$ et $x_1 + x_2 + x_4 = \frac{5}{2} > 2$
   On rajoute l'inégalité :
   $$x_1 + x_2 + x_4 \leq 2 \quad (b)$$
3. La nouvelle solution réelle est $x^3_{PL} = (1, \frac{4}{5}, \frac{1}{5}, \frac{1}{5}, 1)$ et $Z^{PL}_3 = 71.6$ Or $w_1 + w_4 + w_5 = 7 > W$ et $x_1 + x_4 + x_5 = \frac{11}{5} > 2$
   On rajoute l'inégalité :
   $$x_1 + x_4 + x_5 \leq 2 \quad (c)$$

Inégalités combinatoires pour le problème de sac-à-dos 5

i	1	2	3	4	5
p_i	30	28	27	34	7
w_i	2	2	3	4	1
p_i/w_i	15	14	9	8.5	7

et $W = 6$

4. La nouvelle solution réelle est $x_{PL}^4 = (1, \frac{3}{4}, \frac{1}{4}, \frac{1}{4}, \frac{3}{4})$ et $Z_4^{PL} = 71.2$
   Aucune inégalité de couverture est violée
5. Mais, en combinant les inégalités (a), (b) et (d) : $x_3 + x_4 \leq 1$
   On obtient l'inégalité :
   $$\tfrac{1}{2}(a) + \tfrac{1}{2}(b) + \tfrac{1}{2}(d) \rightarrow \; x_1 + x_2 + x_3 + x_4 \leq \left\lfloor \tfrac{5}{2} \right\rfloor = 2 \quad (e)$$
6. La nouvelle solution réelle est $x_{PL}^5 = (\frac{1}{2}, 1, 0, \frac{1}{2}, 1)$ et $Z_5^{PL} = 67$
   Or $w_2 + w_4 + w_5 = 7 > W$ et $x_1 + x_2 + x_4 = \frac{5}{2} > 2$
   On rajoute l'inégalité :
   $$x_2 + x_4 + x_5 \leq 2 \quad (f)$$

Inégalités combinatoires pour le problème de sac-à-dos 6

i	1	2	3	4	5
p_i	30	28	27	34	7
w_i	2	2	3	4	1
p_i/w_i	15	14	9	8.5	7

et $W = 6$

7. La nouvelle solution réelle est $x_{PL}^6 = (\frac{2}{3}, \frac{2}{3}, \frac{1}{3}, \frac{1}{3}, 1)$ et $Z_6^{PL} = 66$
   Aucune inégalité de couverture est violée
8. Mais, en combinant les inégalités (c), (d), (e) et (f)
   On obtient l'inégalité :

$$\frac{1}{2}(c) + \frac{1}{2}(d) + \frac{1}{2}(e) + \frac{1}{2}(f) \rightarrow \; x_1 + x_2 + x_3 + 2x_4 + x_5 \leq \left\lfloor \frac{7}{2} \right\rfloor = 3 \quad (g)$$

9. La nouvelle solution réelle est $x_{PL}^7 = (1, 1, 0, 0, 1)$ et $Z_7^{PL} = 65$
   Cette solution est entière donc elle est optimale

Conclusions et perspectives

Conclusions

L'algorithme des plans coupants :

- est complémentaire à une méthode de séparation et évaluation (inégalités classiques intégrées aux logiciels commerciaux)
- accélere la résolution exacte d'un programme linéaire entier (ajouter manuellement des inégalités)
- permet de résoudre des formulations avec un nombre élevé de contraintes (exemple : problème du voyageur de commerce)

Aller plus loin

- Les problèmes de la classe $\mathcal{NP}$-faciles possèdent :
  - une description de l'enveloppe convexe de taille polynomial ou
  - un algorithme de séparation polynomial
- Les méthodes de décompositions :
  - si $\mathcal{S} = \mathcal{S}^1 \cap \mathcal{S}^2$, étudier plutôt conv($\mathcal{S}^1$) et conv($\mathcal{S}^2$)
  - décomposition de Benders (formulation avec variables liantes)

Quatrième partie IV

Techniques de résolution par décomposition