Recherche Opérationnelle
Deuxième Partie

Paul Feautrier

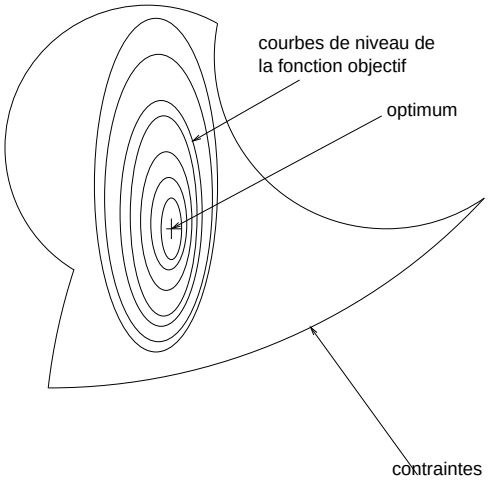
ENS Lyon

29 novembre 2005

Plan

- Introduction et principaux concepts
- Optimisation continue sans contrainte
- Programmation linéaire
- Optimisation continue sous contrainte
- Optimisation combinatoire
  - Programmation linéaire en nombres entiers.
  - Exploration
  - Métaheuristiques
  - Programmation dynamique
- Éléments de Complexité

Optimisation sous contraintes



- Résoudre :

$$\begin{array}{rcl} & \min & f(x), \\ g_i(x) & \leqslant & 0, i = 1, \ldots, n \\ x & \in & \mathbb{R}^n \end{array}$$

- L'ensemble des points faisables est $\{x \in \mathbb{R}^n \mid g_i(x) \leqslant 0, i = 1, \ldots n\}$. Les fonctions g sont les contraintes.
- La programmation linéaire est le cas particulier où f et les g_i sont linéaires. On obtient des problèmes plus ou moins difficiles suivant que l'un ou l'autre ou les deux de ces éléments sont non linéaires (resp. non convexes).

Généralisation

- Certaines contraintes peuvent être difficiles à mettre sous la forme $g_i(x) \leqslant 0$.
- Exemple: on veut que x soit entier (*i.e.* à coordonnées entières).
- On remplace la dernière contrainte par :

$$x \in S \subseteq \mathbb{R}^n.$$

Organisation de l'exposé

- Conditions nécessaires de Kuhn-Tucker
- Méthodes directes ; linéarisation.
  - Méthode des plans sécants.
- Fonction de Lagrange, point-col.
- Méthodes duales
  - Méthodes de pénalité
  - Méthodes lagrangiennes

Plan

Caractérisation de l'optimum I / III

- On suppose les fonctions f et g_i continues et à dérivées continues.
- L'optimum x^* peut être à l'intérieur de F. Dans ce cas $\nabla f(x^*) = 0$.
- L'optimum peut être sur les frontières de F. Dans ce cas $g_i(x^*) = 0$ pour un certain nombres de contraintes (les contraintes saturées) et $\nabla f(x^*)$ n'est pas nécessairement nul. On note $I \subseteq \{1, \ldots, n\}$ l'ensemble des indices des contraintes saturées.
- En particulier, si les contraintes sont toutes des contraintes d'égalité, l'intérieur de F est vide et l'on est toujours dans le dernier cas (dit de Lagrange).

Caractérisation de l'optimum II / III

- En programmation linéaire, $f(x) = c.x$, $\nabla f = c$ n'est jamais nul (ou bien le problème est trivial), donc l'optimum est sur la frontière de F.
- Une direction d est admissible en un point $x^* \in F$ si il existe $\eta > 0$ tel que $\lambda < \eta \Rightarrow x^* + \lambda d \in F$.
- x^* est un minimum si, pour toute direction admissible d,

$$\lambda < \eta \Rightarrow f(x^* + \lambda d) \geqslant f(x^*).$$

- La condition d'admissibilité peut s'écrire :

$$g_i(x^* + \lambda d) = g_i(x^*) + \lambda d.\nabla g_i(x^*) \leqslant 0, i \in I,$$

soit encore $d.\nabla g_i(x^*) \leqslant 0, i \in I$.

Caractérisation de l'optimum III / III

- Les directions admissibles en x^* appartiennent au cône

$$C = \{d \mid d.\nabla g_i(x^*) \leqslant 0, i \in I\},$$

avec $I = \{i \mid g_i(x^*) = 0\}$.

- La réciproque est fausse, sauf dans quelques cas particuliers :
  - Les fonctions g_i sont linéaires ou convexes ;
  - Les gradients sont linéairement indépendants.
- Si C est l'ensemble des directions admissibles, alors une condition nécessaire d'optimalité est :

$$d \in C \Rightarrow f(x^* + \lambda d) - f(x^*) \geqslant 0,$$

$$d \in C \Rightarrow d.\nabla f(x^*) \geqslant 0.$$

Conditions de Kuhn et Tucker

- D'après le lemme de Farkas, il existe des $\lambda_i \geqslant 0, i \in I$ tels que :

$$\begin{aligned} \forall d : -d.\sum_{i\in I} \lambda_i.\nabla g_i(x) &= d.\nabla f(x^*), \\ \nabla f(x^*) - \sum_{i\in I} \lambda_i \nabla g_i(x) &= 0. \end{aligned}$$

- Si on pose $\lambda_i = 0$ pour les contraintes non saturées, on peut étendre la sommation à toutes les valeurs de i. Une condition nécessaire pour que x^* soit un minimum est donc :

$$\begin{aligned} \exists \lambda_i \geqslant 0 \quad &, \quad i = 1, \ldots, n \\ \nabla f(x^*) \quad &- \quad \sum_{i=1}^{n} \lambda_i \nabla g_i(x^*) = 0, \\ \lambda_i.g_i(x^*) \quad &= \quad 0. \end{aligned}$$

- Les λ_i sont les multiplicateurs de Kuhn-Tucker.

Conditions de Lagrange

- Un contrainte d'égalité $g_i = 0$ peut se représenter par deux contraintes d'inégalité $g_i \geqslant 0$ et $g_i \leqslant 0$.
- Il lui correspond deux multiplicateurs de Kuhn-Tucker, λ_i^+ et λ_i^- positifs.
- On peut les regrouper en un seul dont le signe est quelconque. À une contrainte d'égalité correspond un multiplicateur non contraint en signe.
- Si toutes les contraintes sont des égalités, les multiplicateurs peuvent être de signe arbitraire. Ils prennent le nom de multiplicateurs de Lagrange.
- Dans ce cas particulier, toutes les contraintes doivent être saturées.

Plan

Méthode des plans sécants I / VII

- Soit à calculer :

$$\begin{array}{rcl} & \min & f(x), \\ g_i(x) & \leqslant & 0, i = 1, \ldots, n \\ x & \in & \mathbb{R}^n \end{array}$$

où on suppose que les fonctions f et g_i sont convexes.

- On remarque que l'on peut supposer f linéaire. Sinon, on peut remplacer le problème ci-dessus par le problème équivalent :

$$\begin{array}{rcl} & \min & z, \\ z - f(x) & \geqslant & 0, \\ g_i(x) & \leqslant & 0, i = 1, \ldots, n \\ x & \in & \mathbb{R}^n \end{array}$$

Méthode des plans sécants II / VII

- On procède de façon itérative. A l'étape k, on suppose que l'on connaît un polyèdre convexe $Q^{(k)}$ tel que :

$$Q^{(k)} \subseteq F = \{x \mid g(x) \leqslant 0\}.$$

- On résout le programme linéaire :

$$\begin{array}{rcl} & \min & f(x), \\ x & \in & Q^{(k)}. \end{array}$$

  Soit $x^{(k)}$ le point obtenu.
- Si $x^{(k)} \in F$, c'est le minimum cherché. Sinon, il existe i tel que $g_i(x^{(k)}) > 0$. On forme :

$$Q^{(k+1)} = Q^{(k)} \bigcap \left\{ x \;\middle|\; g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T.(x - x^{(k)}) \leqslant 0 \right\},$$

  et on recommence.

Lemme

Si f est convexe, alors $f(x) \geqslant f(a) + \nabla f(a)^T.(x-a)$.

Démonstration.

Pour simplifier on va supposer que x est un scalaire. Soit par exemple $b > a$. Par définition, $f(x) - f(a) \leqslant (x-a)\frac{f(b)-f(a)}{b-a}$. En faisant tendre x vers a on en déduit $f'(a) \leqslant \frac{f(b)-f(a)}{b-a}$, ce qui n'est autre que le résultat cherché. La démonstration est analogue pour $b < a$. □

Lemme
$Q^{(k+1)} \subseteq F$.

Démonstration.
Soit en effet x un point de F. Puisque g_i est convexe, on a $g_i(x) \geqslant g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T.(x - x^{(k)})$. Or $g_i(x) \leqslant 0$, donc $x \in Q^{(k+1)}$. □

Lemme
$x^{(k)} \notin Q^{(k')}, k' > k$

Démonstration.
Il suffit d'observer que $x^{(k)}$ ne satisfait pas la contrainte $g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T.(x - x^{(k)}) \leqslant 0$. □

Lemme

Si $x^{(k)} \in F$, c'est le minimum cherché.

Démonstration.

En effet, d'une part $x^{(k)} \in F$, et d'autre part,

$$x \in F \Rightarrow x \in Q^{(k)} \Rightarrow f(x) \geqslant f(x^{(k)}).$$

□

Méthode des plans sécants VI / VII

Théorème

Si F est borné, tout point d'accumulation de la suite $x^{(k)}$ est un optimum.

Démonstration.

Soit y^* un point d'accumulation, et soit $y^{(k)}$ une suite extraite de x^* et convergeant vers y^*. Montrons d'abord que $y^* \in F$. On supposera pour fixer les idées que à chaque pas, la contrainte utilisée pour construire une coupe est celle qui est la moins satisfaite, c'est à dire celle pour laquelle $g_i(x^(k)) > 0$ est maximum. Supposons que y^* ne soit pas dans F, et soit g_i la contrainte la moins satisfaite. Puisque $y^{(k)}$ converge vers y il existe un k^* suffisamment grand pour que g_i soit la contrainte la moins satisfaite en $y^{(k)}$. On ajoute la contrainte :

$$g_i(y^{(k)}) + \nabla g_i(y^{(k)})^T.(x - y^{(k)}) \leqslant 0.$$

Démonstration.

Il est facile de voir que la distance $||y^{[k')} - y^{(k)}||, k' > k$ ne peut être inférieure à $\frac{|g_i(y^{[k]}|}{|\nabla g_i(y^{[k]})|}$, ce qui contredit le fait que y^* est limite des $y^{(k)}$.

Supposons maintenant qu'il existe dans F un autre point y' tel que $f(y') < f(y^*)$. Comme $y' \in Q^{(k)} \forall k$, il s'en suit que l'algorithme de programmation linéaire devrait toujours construire un point $x^{(k)}$ tel que $f(x^{(k)}) \leqslant f(y')$. Par suite de la continuité de f, toute limite x^* de la suite $x^{(k)}$ doit être telle que $f(x^*) \leqslant f(y')$ ce qui est contradictoire. □

Plan

Méthodes de pénalités

- Principe : au lieu d'exclure les points qui violent les contraintes, ajouter à la fonction économique une pénalité d'autant plus élevée que la contrainte est moins respectée. Si la fonction de pénalité est régulière, on peut utiliser les méthodes d'optimisation sans contrainte.
- En général, la pénalité dépend d'un paramètre qui permet de régler son importance. Quand la pénalité devient très grande devant la fonction objectif, on tend vers l'optimum sous contrainte. Mais la fonction à optimiser devient de plus en plus mal conditionnée.
- Deux variétés :
  - La fonction de pénalité est nulle dans le domaine faisable. L'optimum pénalisé n'est pas faisable. C'est une méthode extérieure.
  - La fonction de pénalité devient infinie quand on sort du domaine faisable. L'optimum est faisable, mais la méthode a besoin d'un point faisable initial.

Méthodes extérieures

- On considère la fonction $h(x)$ égale à 0 si $x \leqslant 0$ et à x^2 sinon. Il est facile de voir qu'elle est continue et à dérivé continue.
- On remplace le problème :

$$\begin{array}{rcl} P: & \min & f(x), \\ g_i(x) & \leqslant & 0, i = 1, \ldots, n \\ x & \in & \mathbb{R}^n \end{array}$$

par la suite de problèmes :

$$P_k : \min f(x) + S_k \sum_{i=1}^{n} h(g_i(x)).$$

où les S_k forment une suite croissante tendant vers l'infini.

- On note $H(x) = \sum_{i=1}^{n} h(g_i(x))$ et x_k la solution de P_k.

Convergence I / III

Théorème

Si f est continue, si l'ensemble des points faisables est fermé et si soit $f(x)$ soit $H(x)$ tend vers l'infini quand x tend vers l'infini, alors tout point d'accumulation de la suite x_k est une solution de P.

On note $\varphi_k = f(x_k) + S_k H(x_k)$, et x^* une solution de P.

Lemme

Les φ_k forment une suite décroissante.

Démonstration.

On a $f(x_{k+1}) + S_{k+1} H(x_{k+1}) > f(x_{k+1}) + S_k H(x_{k+1})$ parce que $S_{k+1} > S_k$ et $f(x_{k+1}) + S_k H(x_{k+1}) \geqslant f(x_k) + S_k H(x_k)$ puisque x_k est la solution de P_k.
De plus, $\varphi_k \leqslant f(x^*) + S_k H(x^*)$. Mais comme x^* est faisable, la pénalité est nulle. On a donc l'encadrement

$$f(x_k) \leqslant \varphi_k \leqslant f(x^*).$$

Convergence II / III

Lemme

Les $H(x_k)$ forment une suite décroissante.

Démonstration.

Comme chaque x_k est la solution d'un problème de minimum, on a :

$$\begin{aligned} f(x_k) + S_k H(x_k) &\leqslant f(x_{k+1}) + S_k H(x_{k+1}), \\ f(x_{k+1}) + S_{k+1} &\leqslant f(x_k) + S_{k+1} H(x_k). \end{aligned}$$

En additionnant et simplifiant :

$$(S_k - S_{k+1})(H(x_{k+1}) - H(x_k)) \geqslant 0.$$

Comme le premier terme est négatif, l'autre l'est aussi. □

Convergence III / III

- Les x_k appartiennent à un ensemble borné, soit parce que $f(x_k) \leqslant f(x^*)$ et que f tend vers l'infini à l'infini, soit parce que $H(x_k) \leqslant H(1)$ et que H tend vers l'infini à l'infini. On peut donc extraire de x_k une sous-suite $x_\ell, \ell \in L$ qui converge vers $\hat{x}$.
- Par continuité, $f(x_\ell)$ tend vers $f(\hat{x})$, et comme $f(x_k) \leqslant f(x^*)$, $f(\hat{x}) \leqslant f(x)$.
- Comme $\varphi_k \leqslant f(x^*)$, φ_k a une limite $phi^* \leqslant f(x^*)$.
- $\lim S_\ell H(x_\ell) = \varphi^* - f(\hat{x})$. Donc $H(x_\ell)$ tend vers 0, et par continuité $H(\hat{x}) = 0$.
- Donc $\hat{x}$ est un point faisable, donc $f(\hat{x}) \geqslant f(x^*)$, donc $f(\hat{x}) = f(x^*)$ et $\hat{x}$ est un optimum.

Méthodes intérieures

- On prend comme fonction de pénalité une fonction qui tend vers l'infini au voisinage de 0, par exemple $h(x) = -1/x$.
- On minimise $f(x) + R \sum_{i=1}^{n} h(g_i(x))$.
- Dans les mêmes conditions que ci-dessus, on montre que l'optimum du problème P_R tend vers l'optimum de P quand R tend vers 0.
- Toutes les solutions intermédiaires sont faisables, mais il faut disposer d'un point faisable pour commencer les calculs.

Exemple

- Le problème du cas le pire de Fourier-Motzkin:

$$\begin{array}{rcl} & \max & xy + z, \\ x + y + z & = & n \end{array}$$

- On élimine z à l'aide de la dernière contrainte et on applique la méthode des pénalités :

$$\max xy + n - x - y + S(n - x - y)^2.$$

- On utilise un système de calcul algébrique pour achever les calculs.

Plan

Fonction de Lagrange, point-col

- Soit à résoudre :

$$\begin{array}{rcl} & \min & f(x), \\ g_i(x) & \leqslant & 0, i = 1, \ldots, n \\ x & \in & S \subseteq \mathbb{R}^n \end{array}$$

- La fonction de Lagrange associée est :

$$L(x, \lambda) = f(x) + \lambda.g(x), \lambda \geqslant 0.$$

- g est le vecteur dont les composantes sont les g_i.

Point-col

- (x^*, λ^*) est un point-col si et seulement si :

$$x^* \in S, \quad \lambda^* \geqslant 0.$$

$$\forall x \in S : L(x^*, \lambda^*) \leqslant L(x, \lambda^*),$$

$$\forall \lambda \geqslant 0 : L(x^*, \lambda^*) \geqslant L(x^*, \lambda).$$

- Caractérisation d'un point-col :

$$\begin{aligned} L(x^*, \lambda^*) &= \min_{x \in S} L(x, \lambda^*), \\ g(x^*) &\leqslant 0, \\ \lambda^* . g(x^*) &= 0. \end{aligned}$$

Preuve

Soit (x^*, λ^*) un point-col. La première propriété est une conséquence directe de la définition.
La deuxième propriété entraîne :

$$f(x^*) + \lambda^*.g(x^*) \geqslant f(x^*) + \lambda.g(x^*),$$

$$(\lambda - \lambda^*).g(x^*) \leqslant 0.$$

S'il existait un $g_i(x^*)$ positif, il suffirait de prendre le λ_i correspondant suffisamment grand pour violer cette inégalité. Donc $\forall i : g_i(x^*) \leqslant 0$.
Pour $\lambda = 0$ on trouve :

$$-\lambda^*.g(x^*) \leqslant 0.$$

Mais les deux termes du produit sont non négatifs, donc le produit est nul.

Preuve, réciproque

La première caractéristique entraîne directement la première propriété du point-col.
On déduit de la troisième caractéristique que $L(x^*, \lambda^*) = f(x^*)$.
Enfin :

$$L(x^*, \lambda) = f(x^*) + \lambda.g(x^*) \leqslant f(x^*) = L(x^*, \lambda^*),$$

puisque $g(x^*) \leqslant 0$.

Intérêt

Théorème

Si (x^*, λ^*) *est un point-col, alors* x^* *est un minimum global.*

Démonstration.

D'après la définition, $x^* \in S$ et $g(x^*) \leqslant 0$, donc x^* est faisable.
D'autre part :

$$\forall x \in S, g(x) \geqslant 0 : f(x^*) = L(x^*, \lambda^*) \leqslant L(x, \lambda^*) = f(x) + \lambda^* g(x) \leqslant f(x).$$

puisque $\lambda^* \geqslant 0$ et $g(x) \leqslant 0$.
x^* est donc bien minimum global. □

Mais il existe des problèmes qui n'ont pas de point-col.

Fonction de Lagrange et pénalités

- La fonction de Lagrange est une fonction de pénalité.

$$L(x, \lambda) = f(x) + \lambda.g(x),\ \lambda \geqslant 0.$$

- En effet, dans la région infaisable, $g(x) > 0$, donc le second terme augmente la valeur de L, alors qu'on recherche un minimum.
- Toutefois, ce terme est négatif dans la région faisable, ce qui diminuerai artificiellement la valeur du minimum, si l'on n'avait pas la contrainte $\forall i : \lambda_i.g_i(x^*) = 0$.
- Comme les méthodes de pénalité, l'emploi de la fonction de Lagrange permet de remplacer un problème avec contraintes par un problème sans contrainte.

Une mauvaise idée

- Maximiser $L(x, \lambda)$ par rapport à $\lambda \geqslant 0$ pour x fixé. Soit $g(x)$ le résultat. Minimiser ensuite $g(x)$ sans contraintes.
- Il est facile de trouver le maximum.
  - Si $g_i(x) > 0$, il suffit de faire tendre $\lambda_i \to \infty$ pour obtenir un maximum infini.
  - Sinon, le maximum est égal à $f(x)$.
- Ceci revient donc à étendre f en une fonction discontinue non dérivable, ce qui ne se prête pas à l'optimisation.

Une bonne idée

- Minimiser $w(\lambda) = \min L(x, \lambda)$ sans contrainte. Maximiser $w(\lambda)$ sous la contrainte $\lambda \geqslant 0$.

Lemme

$w(\lambda)$ est une fonction concave.

Démonstration.

Soit λ_1, λ_2 deux valeurs de λ, α et β deux nombres positifs tels que $\alpha + \beta = 1$, et x un point arbitraire. Par définition des minima :

$$\begin{aligned} f(x) + \lambda_1 g(x) &\geqslant w(\lambda_1), \\ f(x) + \lambda_2 g(x) &\geqslant w(\lambda_2), \\ f(x) + (\alpha\lambda_1 + \beta\lambda_2) g(x) &\geqslant \alpha w(\lambda_1) + \beta w(\lambda_2). \end{aligned}$$

et cette propriété vraie partout s'étend au minimum $w(\alpha\lambda_1 + \beta\lambda_2)$.

□

Application à la programmation linéaire

- Il ne faut pas croire que l'optimisation de w est toujours sans contrainte. Soit par exemple le programme linéaire :

$$\begin{array}{rcl} & \min & c.x, \\ Ax - b & \geqslant & 0, \\ x & \geqslant & 0. \end{array}$$

- La fonction de Lagrange associée est :

$$L(x, \lambda) = c.x - \lambda(Ax - b) - \mu x \quad = \quad (c - \lambda A - \mu).x + \lambda b.$$

- Il est facile de voir que si $c - \lambda A - \mu$ n'est pas nul, la valeur du minimum est $-\infty$. Sinon, c'est λb. Comme $c - \lambda A - \mu = 0$ est équivalent à $c - \lambda A \geqslant 0$, on voit que l'on est amené à résoudre :

$$\begin{array}{rcl} & \max & \lambda.b, \\ c - \lambda A & \geqslant & 0, \\ \lambda & \geqslant & 0. \end{array}$$

C'est le dual du problème original!

Et s'il n'y a pas de point col ?

Théorème

Soit x^* la solution du problème avec contraintes. On a :
$\max_{\lambda \geqslant 0} w(\lambda) \leqslant f(x^*)$.

Démonstration.

Comme x^* est faisable, $g(x^*) \leqslant 0$. Donc

$$\begin{aligned} L(x^*, \lambda) &= f(x^*) + \lambda g(x^*) \leqslant f(x^*), \\ w(\lambda) &= \min_x L(x, \lambda) \leqslant L(x^*, \lambda) \leqslant f(x^*). \end{aligned}$$

Cette propriété vraie pour tout λ s'étend au maximum. □

- La solution du problème dual fournit une borne inférieure de la solution du primal.
- Il y a «saut de dualité» quand les deux solutions ne sont pas égales.
- w est dérivable à l'optimum, si et seulement si le saut de dualité est nul.

Génération de colonnes I / III

- Si l'ensemble $S = \{x_1, \ldots, x_n\}$ est fini, le problème s'écrit :

$$\max_{\lambda \geqslant 0} \min_{i=1}^{n} f(x_i) + \lambda g(x_i)$$

se ramène à un problème de programmation linéaire :

$$\begin{array}{lccl} P(n): & \max & z & \\ & z & \leqslant & f(x_i) + \lambda g(x_i),\ i = 1, \ldots, n \\ & \lambda & \geqslant & 0. \end{array}$$

- Soit x^n, λ^n la solution de $P(n)$.

Génération de colonnes II / III

- Si S est infini, on suppose que l'on a déjà construit les points $\{x_1, \ldots, x_n\}$.
- On résout le problème linéaire ci-dessus.
- On détermine le point x_{n+1} comme solution de :

$$\min_{x \in S} L(x, \lambda^n) = f(x) + \lambda^n g(x)$$

  par une méthode d'optimisation sans contraintes.
- Soit $w(\lambda^n)$ le minimum obtenu.
- On recommence avec $n+1$ points, jusqu'à convergence.

Génération de colonnes III / III

Lemme

Soit x^* la solution du problème avec contrainte: $w(\lambda^n) \leqslant f(x^*)$.

Démonstration.

Puisque $x^* \in S$, $w(\lambda^n) \leqslant f(x^*) + \lambda g(x^*) \leqslant f(x^*)$ puisque que x^* est faisable. □

Lemme

$f(x^*) \leqslant z^n$.

Démonstration.

Il est évident que $\langle f(x^*), \lambda^n \rangle$ est un point faisable pour $P(n)$. □

Lemme

$z^{n+1} \leqslant z^n$.

Démonstration.

Tout point faisable pour $P(n+1)$ est faisable pour $P(n)$. □

Convergence

- On a l'encadrement $w(\lambda^n) \leqslant f(x^*) \leqslant z^n$.
- Comme la suite z^n est décroissante, elle converge.
- Mais la convergence de $w(\lambda^n)$ vers $f(x^*)$ impliquerait que le saut de dualité est nul, ce qui n'est pas toujours le cas.

Plan

Optimisation combinatoire

- Problème d'optimisation sous contraintes où l'ensemble des points faisables est non pas continu mais discret.
- Forme du problème :

$$\begin{array}{ccc} & \min & f(x) \\ x & \in & S \end{array}$$

- Les inconnues sont les n composantes du vecteur x. S est l'ensemble discret des points faisables.
- En général, S est produit cartésien d'ensembles plus petits, et sa taille est le produit des tailles de ces petits ensembles (d'où le nom : optimisation combinatoire).
- Chaque composant de S correspond à un choix ; il faut trouver la bonne suite de choix, sachant que ceux-ci ne sont pas indépendants en général.

Exemple I / II

- On considère un ordinateur sur lequel on doit exécuter t algorithmes enchaînés $A_i, i = 1, \ldots, t$. Chaque algorithme a pour données les résultats de l'algorithme précédent.
- Pour implémenter chaque algorithme, on doit choisir une structure pour ses données. On suppose qu'il y a s structures possibles, $S_k, k = 1, \ldots, s$ et que le temps d'exécution de l'algorithme dépend de la structure choisie. Par exemple, une matrice peut être rangée par ligne ou par colonne, et, par suite des effets de cache, le temps d'exécution du produit matriciel varie suivant la structure choisie. On supposera qu'un algorithme ne modifie pas la structure des ses données. On note T_{ik} le temps d'exécution de l'algorithme i quand les données on la structure S_k.

Exemple II / II

- On suppose qu'il est possible de modifier les structure de données (par exemple, de transposer une matrice). On note $\theta_{k\ell}$ le temps nécessaire pour passer de la structure S_k à la structure S_ℓ. Remarquer que $\theta_{kk} = 0$.
- La structure du programme est alors :

$$S_{k_0} R_0 S_{k_1} A_1 S_{k_1} \ldots A_t S_{k_t} R_t S_{k_{t+1}}$$

- On suppose que k_0 et k_{t+1} sont fixés par le cahier des charges.
- Trouver la séquence de restructurations donnant le temps total minimum.

Quelques méthodes de solution I / III

- Une heuristique gloutonne.
- On remarque que souvent $\theta_{kl} \ll T_{ik}$. Pour l'exemple matriciel, le temps de transposition est $O(n^2)$ alors que le temps du produit est $O(n^3)$.
- On choisit donc S_{k_j} de façon que T_{ik_j} soit minimum, et on rajoute des redistributions si nécessaire.

Quelques méthodes de solution II / III

- Programmation dynamique. On remarque que le problème a une structure analogue à celle d'un problème de plus court chemin.
- On note $P_n(k)$ le problème d'optimiser le programme analogue au programme initial, mais ou on s'arrête juste après la redistribution R_n avec une structure de données S_k. Le problème initial est $P_t(k_{t+1})$. Soit $\Omega_n(k)$ le meilleur temps d'exécution de $P_n(k)$. On a la relation de récurrence :

$$\begin{aligned} \Omega_n(k) &= \min_{\ell} \Omega_{n-1}(\ell) + T_{n\ell} + \theta_{\ell k}, \\ \Omega_0(k) &= \theta_{k_0 k}. \end{aligned}$$

- On peut calculer toutes les valeurs de Ω_{nk} à l'aide de cette récurrence, lire la valeur de $\Omega_t(k_{t+1})$ et reconstituer le chemin à suivre par retour arrière.

Quelques méthodes de solution III / III

- Codage en variables 0-1. On pose $X_{ik} = 1$ si la distribution des données à l'étape i est la distribution S_k, et 0 sinon.
- A chaque étape, il y a une et une seule distribution :

$$\sum_{k=1}^{s} X_{ik} = 1. \qquad (1)$$

- Le temps de calcul total est : $T_c = \sum_{i=1}^{t} \sum_{k=1}^{s} T_{ik} X_{ik}$.
- Le temps de redistribution de l'étape i à $i+1$ est donné par $\theta_{k\ell}$ tel que $X_{ik} = 1$ et $X_{(i+1)\ell} = 1$, ce qui peut s'écrire :

$$T_r = \sum_{i=0}^{t} \sum_{k=1}^{s} \sum_{\ell=1}^{s} X_{ik} X_{(i+1)\ell} \theta_{k\ell}.$$

- Il s'agit de minimiser la somme $T_c + T_r$ sous les contraintes (1) et $X_{ik} \in \{0, 1\}$. C'est un problème de programmation quadratique en nombres entiers.

Programmation linéaire en entiers

- Définition.

$$\begin{array}{rcl} min_{\ll} & & x, \\ Ax + b & \geqslant & 0, \\ x & \in & \mathbb{N}. \end{array}$$

- Noter que $x \in \mathbb{N}$ implique $x \geqslant 0$. Il s'agit donc de l'analogue exact du problème résolu par l'algorithme dual, à ceci près qu'il y a une contrainte d'intégrité en plus.
- On suppose en général que A et b ont des coefficients entiers.

Plan

Coque entière I / II

- Soit S un ensemble de $\mathbb{R}^n$. La coque entière de S, notée $\hat{S}$ est la coque convexe de l'ensemble des points entiers de S.

$$\begin{aligned} x \in \mathbb{Z}^n \cap S &\Rightarrow x \in \hat{S}, \\ x, y \in \hat{S}, \lambda, \mu \geqslant 0, \lambda + \mu = 1 &\Rightarrow \lambda x + \mu y \in \hat{S}. \end{aligned}$$

Propriétés de la coque entière

Lemme
Si A est convexe, $\hat{A} \subseteq A$.

Lemme
$A \subseteq B \Rightarrow \hat{A} \subseteq \hat{B}$.

Lemme
Si $\mathbb{Z}^n \cap P \subseteq S$ convexe, alors $\hat{P} \subseteq S$.

Démonstration.
Un point de $\hat{P}$ est combinaison convexe d'un certains nombres de points de $\mathbb{Z}^n \cap P$, qui appartiennent par hypothèse à S. Or S contient toutes les combinaisons convexes de ses points. □

Caractérisation de la solution entière

Théorème

La coque entière d'un polyèdre défini par $Ax + b \geqslant 0$, avec A entière, est un polyèdre.

Démonstration.

Soit $P = Q + C$ un polyèdre, où Q est borné et C un cône. On peut supposer que les vecteurs générateurs v_i de C sont entiers, et il est facile de voir que $\hat{C} = C$.

Soit $B = \{\sum_i \mu_i v_i \mid 0 \leqslant \mu_i \leqslant 1\}$. Il est clair que B est un polyèdre borné inclus dans C. Admettons que $\hat{P} = \widehat{Q + B} + C$ (ce sera montré par les deux lemmes suivants). Or $Q + B$ est borné, et pour un polyèdre borné le théorème est évident, puisque la conque entière est enveloppe convexe de ses points entiers qui sont en nombre fini. □

Coque entière II / II

Lemme

$\hat{P} \subseteq \widehat{Q+B} + C.$

Démonstration.

D'après un lemme ci-dessus, il suffit de considérer un point x entier de P. On peut l'écrire $x = q + c, q \in Q, c \in C$. Il en résulte $c = \sum_i \mu_i v_i$. On pose $c' = \sum_i \lfloor \mu_i \rfloor v_i$ et $b = c - c'$. Il est clair que $b \in B$, donc que $q + b \in Q + B$, que $c' \in C$ est entier, et que $q + b = x - c'$ est entier, donc que $q + b \in \widehat{Q+B}$. □

Lemme

$\widehat{Q+B} + C \subseteq \hat{P}.$

Démonstration.

Puisque $Q + B \subseteq P$,

$\widehat{Q+B} + C \subseteq \hat{P} + C = \hat{P} + \hat{C} = \widehat{P+C} = \hat{P}.$ □

Minimum entier

Lemme

Le minimum entier d'un polyèdre est le minimum rationnel de sa coque entière.

Démonstration.

Le minimum entier x^* appartient à la coque entière. Supposons qu'il existe dans la coque entière un point $x' \ll x^*$. Ce point est nécessairement à coordonnées non entières (puisque la coque entière est contenue dans le polyèdre). x' est donc combinaison convexe de points entiers qui lui sont tous supérieurs dans l'ordre lexicographique, ce qui est impossible. □

- Le problème sera résolu si on sait construire la coque entière.
- Mais la complexité de la coque entière peut être énorme.
- Heureusement, on peut se contenter de construire quelques coupes (et non toute la coque) : des contraintes affines qui excluent une partie de P mais aucun point de $\hat{P}$.

Plan

Coupe de Gomory

- Construction d'une coupe : soit

$$a/D.x + b/D \geqslant 0$$

une des contraintes du problème. D est le dénominateur commun des coefficients de x et du terme constant.

- Par construction de l'algorithme dual, la valeur de cette contrainte est l'une des variables d'écart du problème initial, qui doit être entière. Comme les x sont entiers :

$$\begin{aligned} ax + b \bmod D &= 0, \\ (a \bmod D)x &\equiv (-b \bmod D)(\bmod D), \\ (a \bmod D)x &= (-b \bmod D) + kD, \end{aligned}$$

- k est nécessairement positif, d'où :

$$(a \bmod D)x - (-b \bmod D) \geqslant 0.$$

Algorithme de Gomory

- On a mis le problème sous la forme :

$$\begin{array}{rcl} \min_{\ll} & & x \\ y = Sz + t & \geqslant & 0 \\ z & \geqslant & 0 \end{array}$$

où la matrice S et le vecteur t sont entiers, ainsi que les variables y et z. z est un extrait de y, et les n premières composantes de y sont les variables originales.
- On procède comme dans l'algorithme du Simplexe jusqu'à obtenir $t \geqslant 0$. Si l'algorithme échoue, il n'y a pas de solution entière.
- Si les composantes 1 à n de t sont entières, c'est la solution.
- Sinon, on choisit le premier t_i non entier, on construit une coupe comme ci-dessus à l'aide de la contrainte $S_i z + t_i \geqslant 0$, et on l'ajoute au tableau.
- Le nouveau tableau n'est pas faisable : le terme constant de la coupe est négatif. On reprend l'algorithme jusqu'à terminaison.

Preuve I / VI

Théorème

Si l'algorithme se termine, ou bien on a trouvé la solution entière, ou bien le problème n'a pas de solution.

Démonstration.

Soit P_n le polyèdre obtenu après la n^{e} coupe. Par construction, tous les points entiers de $P = P_0$ sont dans P_n. Si donc P_n est vide, P ne contient aucun point entier. Sinon, soit x^* le minimum lexicographique de P_n, et supposons qu'il est entier. Si P contenait un point entier plus petit, il serait dans P_n ce qui serait une contradiction. □

Preuve II / VI

Lemme

On peut toujours supposer qu'il existe une solution.

Démonstration.

On considère le problème étendu

$$\begin{array}{rcl} & \min_{\ll} & u, x \\ u + Ax + b & \geqslant & 0, \\ u & \geqslant & 0, \\ x & \geqslant & 0 \end{array}$$

Il est clair que ce problème a toujours une solution : il suffit de prendre x nul et u très grand. Si le problème initial a un minimum x^*, alors $0, x^*$ est le minimum du problème étendu. Inversement, si le problème initial n'a pas de solution, alors u ne peut être nul dans la solution du problème étendu. Les deux problèmes sont donc équivalents. □

Preuve III / VI

Lemme

Les minima successifs x_n forment une suite croissante dans l'ordre lexicographique.

Démonstration.

Évident, puisque $P_{n+1} \subseteq P_n$. □

- A une certaine étape de l'algorithme, soit $\sum_j \frac{s_j}{D} x_j + \frac{t}{D} \geqslant 0$ la ligne qui va fournir la coupe.
- A l'étape qui suit, l'algorithme du Simplexe exécute un pivot. Soit x_j la variable éliminée.
- Soit : $\sum_j \frac{s'_j}{D} x_j + \frac{t'}{D} \geqslant 0$ la ligne après l'exécution du pivot.

Preuve IV / VI

Lemme

Il existe un nombre entier Q tel que $\frac{t}{D} < Q \leqslant \frac{t'}{D}$.

Démonstration.

La coupe est : $\sum_j \frac{s_j \bmod D}{D} x_j - \frac{-t \bmod D}{D} \geqslant 0$

Les formules de changement de base donnent:

$$\frac{t'}{D} = \frac{t}{D} + \frac{s_j}{D} \frac{-t \bmod D}{D} \frac{D}{s_j \bmod D} \leqslant \frac{t}{D} + \frac{-t \bmod D}{D}.$$

Soit q le quotient entier par défaut de t par D : $t = qD + t \bmod D$.

$$\frac{t'}{D} \geqslant q + \frac{t \bmod D + (-t \bmod D)}{D}.$$

Le deuxième terme est égal à 1, il suffit donc de prendre $Q = q + 1$.

D'autre part $t/D = (q+1) - (D - t \bmod D)/D < q + 1$.

L'inégalité est stricte puisque t n'est pas entier. □

Preuve V / VI

- A un instant donné du déroulement de l'algorithme, on dit qu'une ligne est active si lors d'une opération ultérieure, la valeur de son second membre changera.

Lemme

La première ligne ne peut être active qu'un nombre fini de fois.

Démonstration.

Soit $\tau_1 = \lceil t_1 \rceil$. On a l'encadrement $\tau_1 - 1 < t_1 \leqslant \tau_1$. Considérons l'évolution de τ_1 après une coupe.

Si la source de la coupe est la ligne 1, après le changement de base qui suit, τ_1 augmente au moins d'une unité. Si la source est une autre ligne, c'est que t_1 est entier. Une autre ligne est source de la coupe, et si la première ligne est active, c'est que S_{1j} est non nul. La valeur de t_1 augmente. Qu'elle prenne une valeur entière ou fractionnaire, la valeur de τ_1 augmente au moins d'une unité.

Or t_1 donc τ_1 sont bornés par la solution optimale x_1^*. □

Preuve VI / VI

Théorème

L'algorithme des coupes de Gomory converge.

Démonstration.

Nous avons vu que la première ligne ne peut être active qu'un nombre fini de fois. Après la dernière modification, tout se passe comme si le problème à résoudre avait une contrainte et une inconnue de moins (méthode de déflation). On peut donc prouver que la deuxième ligne n'est active qu'un nombre fini de fois. De proche en proche, on voit que l'algorithme se termine. □

Complexité

- Comme il faut pouvoir distinguer entre nombres entiers et nombres fractionnaires, les calculs doivent être menés en arithmétique exacte.
- Les nombres à manipuler sont des déterminants de sous-matrices $n \times n$ de la matrice des contraintes. Leur taille (nombre de bits) est donc bornée par n fois le nombre de bits du plus grand coefficient.
- Le nombre maximum de coupes est borné par le nombre de coupes nécessaires pour caractériser la coque entière de l'ensemble des solutions. Résultat de Chvatal.

Plan

Variables booléennes

- On peut coder un choix entre n possibilités à l'aide de n variables 0-1, $X_1, \ldots, X_n$.
- Chaque variable entière ne peut prendre que les valeurs 0 ou 1 : $0 \leqslant x \leqslant 1$.
- Les choix sont exclusifs : $\sum_{i=1}^{n} X_i = 1$.
- Si à chaque choix est associé une quantité a_i, la quantité associée à un jeu de variables est $\sum_{i=1}^{n} X_i t_i$.
- On peut considérer les X_i comme des booléens et coder les opérateurs :

$$\begin{aligned} Z &\equiv X \vee Y :: Z \geqslant X, Z \geqslant Y, Z \leqslant X + Y, \\ Z &\equiv X \wedge Y :: Z \leqslant X, Z \leqslant Y, Z \geqslant X + Y - 1, \\ Z &\equiv \neg X :: Z = 1 - X. \end{aligned}$$

Problèmes de graphe

- On peut représenter un graphe de nombreuses façons : matrice d'incidence, matrice de connexion, $Z_{ij} = 1$ si et seulement si il existe un arc $i \to j$.
- Chemin : $X_{ij} = 1$ si et seulement si le chemin emprunte l'arc $i \to j$. Contrainte : $X_{ij} \leqslant Z_{ij}$.
- Loi de Kirchoff : $\sum_i X_{ij} = \sum_k X_{jk}$, pour tout j excepté le début et la fin du chemin.
- Chemin simple : ne passe qu'une fois par chaque sommet. $\sum_i X_{ij} \leqslant 1$.
- Pour le début, $\sum_j X_{ij} = 1$, et pour la fin $\sum_i X_{ij} = 1$.
- Minimiser la somme $\sum_{ij} X_{ij}$ assure que le chemin n'a pas de boucles isolées.

Techniques de grands nombres

- Soit $P = \{x \mid Ax + b \geqslant 0\}$ et $Q = \{x \mid Cx + d \geqslant 0\}$ deux polyèdres dont on doit explorer les points entiers. Soit z une nouvelle variable 0-1.
- Si P et Q sont bornés, alors pour M suffisamment grand, le polyèdre :

$$P \oplus Q = \{z, x \mid Ax + b + Mz \geqslant 0, Cx + d + M(1 - z) \geqslant 0\}$$

  est l'union disjointe de P et Q.
- On peut mener l'exploration sur $P \oplus Q$ en une seule fois.
- M doit être choisi de telle façon que $x \in P$ entraîne $Cx + d + M \geqslant 0$ et réciproquement.