Optimisation continue

Y. Collette

Optimisation Classique

Optimisation sous contraintes

Optimisation Sous Contraintes

Problème Général

$$\text{Minimiser} \quad f(x)$$

$$\text{avec} \quad h_i(x) = 0 \qquad i = 1, \ldots, m$$

$$h_i(x) \leq 0 \qquad i = m+1, \ldots, p$$

$$\text{et} \quad X_j^L \leq x_j \leq X_j^U \qquad j = 1, \ldots, n$$

Contraintes d'égalités

Contraintes d'inégalités

Contraintes de bornes

La Bonne Direction

La direction améliore la fobj

La fonction objectif:

$$f(x+d)-f(x)\simeq \nabla f(x)^t\cdot d$$

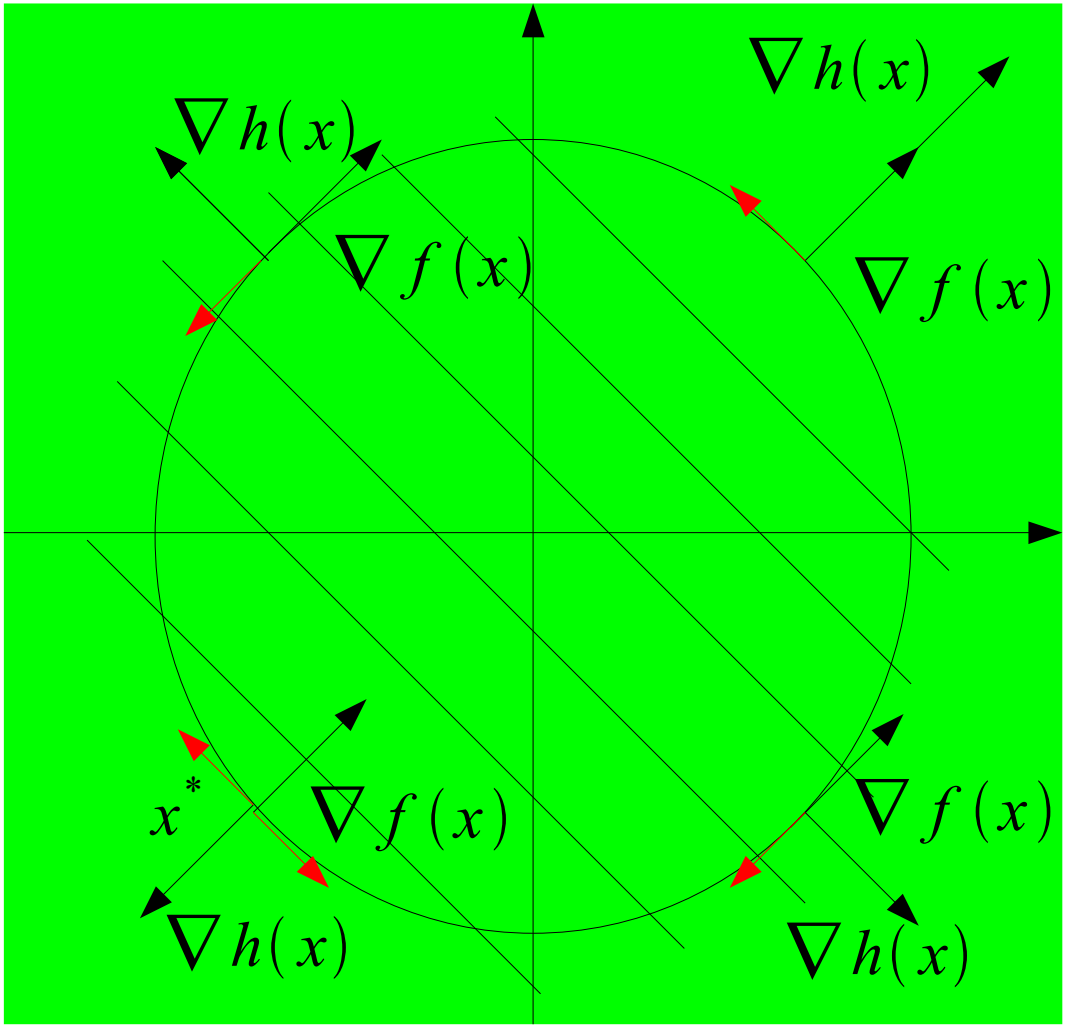
$$\nabla f(x)^t\cdot d<0$$

La contrainte:

$$0=h(x+d)\simeq h(x)+\nabla h(x)^t\cdot d$$

$$\nabla h(x)^t\cdot d=0$$

Permet de vérifier les contraintes

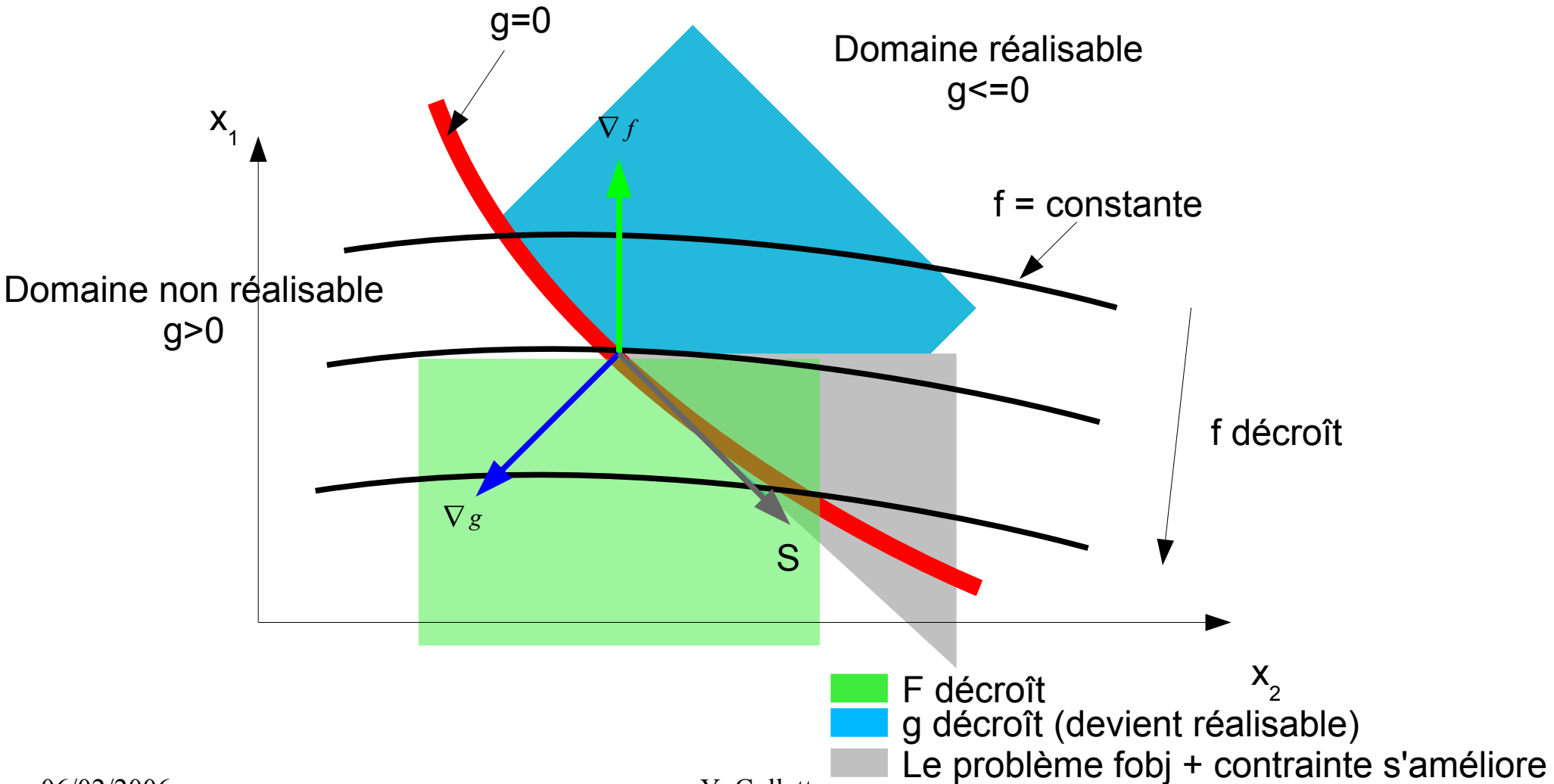


d

Minimise x_1+x_2

Avec $x_1^2+x_2^2-2=0$

La Bonne Direction



Optimisation Sous Contraintes

Conditions de Karush-Kuhn-Tucker

Si :

- x^* satisfait les contraintes du problème précédent
- les fonctions $f(x), g_j(x)$ et $h_j(x)$ sont dérivables une fois

alors il existe des vecteurs de multiplicateurs de Lagrange u^*, w^* tels que (x^*, u^*, w^*) satisfasse les conditions suivantes:

(1) $h_j(x^*)=0 \qquad j=1,\dots,m$

(2) $g_j(x^*)\le 0 \qquad j=m+1,\dots,p$

(3) $u_j^* \cdot g_j(x^*)=0 \qquad j=m+1,\dots,p$

(4) $u_j^* \ge 0 \qquad j=m+1,\dots,p$

(5) $\nabla L(x^*, u^*, w^*)=0$

Conditions du premier ordre

Les contraintes d'inégalités sont soient actives soient vérifiées

$$\nabla L(x^*, u^*, w^*)=\nabla f(x^*)+\sum_{j=1}^{m} w_j^* \cdot \nabla h_j(x^*)+\sum_{j=m+1}^{p} u_j \cdot \nabla g_j(x^*)$$

Optimisation Sous Contraintes

Conditions de Karush-Kuhn-Tucker

Conditions du second ordre

Ensemble des index des contraintes d'égalité

$$A(x)=E\cup\{i\in I \text{ tels que } g_i(x)=0\}$$

Ensemble des index des contraintes d'inégalité

Condition LICQ (Linear Independant Constraint Qualification)

Etant donné x* et A(x*), on dit qu'il y a qualification d'indépendance linéaire des contraintes quand l'ensemble des gradients des contraintes $\{\nabla c_i(x^*), i\in A(x^*)\}$ est linéairement indépendant

Cône tangent à l'ensemble des valeurs réalisable en x*

$$F_1=\left\{\alpha\cdot d \text{ tels que } \alpha>0, \begin{array}{ll} d^t\cdot\nabla h_i^*=0 & \forall i\in E \\ d^t\cdot\nabla h_i^*\geq 0 & \forall i\in A(x^*)\cup I \end{array}\right\}$$

$$F_2(\lambda^*)=\left\{\omega\in F_1 \text{ tels que } \nabla h_i(x^*)^t\cdot\omega=0, \forall i\in A(x^*)\cup I \text{ avec } \lambda_i^*>0\right\}$$

Optimisation Sous Contraintes

Conditions de Karush-Kuhn-Tucker

Condition nécessaire du second ordre:
Supposons que x* soit une solution du problème contraint et que la condition LICQ soit satisfaite. Soit λ* un multiplicateur de Lagrange qui vérifie les contraintes du premier ordre. Alors:

$$\omega^t \cdot \nabla_{xx} L(x^*, \lambda^*) \cdot \omega \geq 0 \quad \forall \omega \in F_2(\lambda^*)$$

Conditions du second ordre

Condition suffisante du second ordre:
Supposon que pour x*, il existe un multiplicateur de Lagrange λ* tel que les conditions du premier ordre de KKT soient vérifiées. Supposons aussi que:

$$\omega^t \cdot \nabla_{xx} L(x^*, \lambda^*) \cdot \omega \geq 0 \quad \forall \omega \in F_2(\lambda^*), \omega \neq 0$$

Alors x* est un optimum local strict du problème contraint.

Version soft: le Hessien est défini non-négatif

Sert plutôt à vérifier qu'une solution est optimale

Optimalité sous contrainte

Version simplifiée

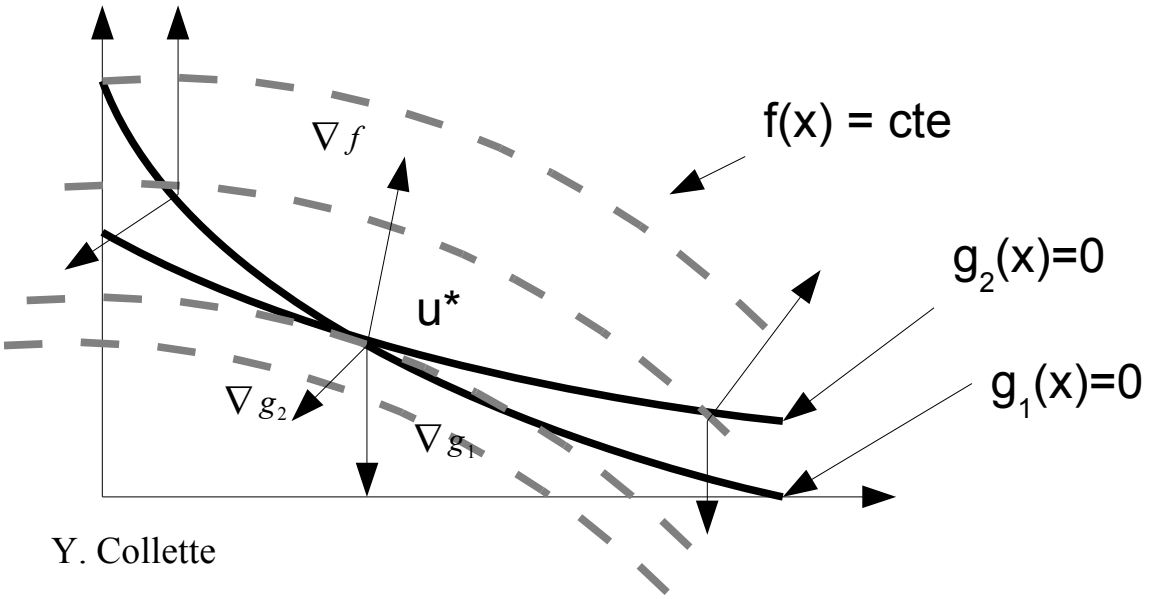
X* est un optimum si:

1 x^* vérifie les contraintes

2 $\lambda_j \cdot g_j(x^*) = 0 \quad j = 1, \ldots, m \quad \lambda_j \geq 0$

3 $\nabla f(x^*) + \sum_{j=1}^{m} \lambda_j \cdot \nabla g_j(x^*) + \sum_{k=1}^{l} \lambda_{k+m} \cdot \nabla h_k(x^*) = 0$

$\lambda_j \geq 0$ et λ_{k+m} non restreint en signe



Conditions de Karush-Kuhn-Tucker

Exemple 1

Minimiser $f(x)=x_1^2+x_2^2$

Avec $g_1(x)=-(x_1^2+x_2^2)+18\le 0$

$g_2(x)=-x_1-x_2+1\le 0$

$$\begin{bmatrix}2x_1^*\\2x_2^*\end{bmatrix}+u_1^*\begin{bmatrix}-2x_1^*\\-2x_2^*\end{bmatrix}+u_2^*\begin{bmatrix}-1\\-1\end{bmatrix}=0$$

$u_1^*(-x_1^{*2}-x_2^{*2}+18)=0$

$u_2(-x_1^*-x_2^*+1)=0$

$u_1^*\ge 0$ et $u_2^*\ge 0$

Condition 5

Condition 3

Condition 4

$x_1^*\cdot(1-2\cdot u_1^*)-u_2^*=0$

$x_2^*\cdot(1-2\cdot u_1^*)-u_2^*=0$

$u_2{}^*=x_2^*\cdot(1-2\cdot u_1^*)$

$x_1^*\cdot(1-2\cdot u_1^*)-x_2^*\cdot(1-2\cdot u_1^*)=0$

$(x_1^*-x_2^*)\cdot(1-2\cdot u_1^*)=0$

donc :

$x_1^*=x_2^*$ ou

$u_1{}^*=1/2$

On teste $u_1=u_2=0\rightarrow$ Impossible

On teste ensuite $u_2=0$ et $u_1\neq 0$

$u_1^*\neq 0, x_1^{2*}+x_2^{2*}=18\rightarrow x_2^*=\pm 3$

Si $x_1^*=x_2^*=-3$ alors une contrainte n'est pas vérifiée

Donc $x_1^*=x_2^*=+3$ et $u^{*t}=\begin{bmatrix}1/2 & 1/2\end{bmatrix}$

Conditions de Karush-Kuhn-Tucker

Exemple 2

Minimiser $f(x)=x_1^2+x_2^2$

Avec $g_1(x)=-(x_1^2+x_2^2)+18\leq 0$

$g_2(x)=-x_1-x_2+1\leq 0$

→

Minimiser $f(x)=x_1^2+x_2^2$

Avec $g_1(x)=(x_1^2+x_2^2)-18-x_3^2=0$

$g_2(x)=x_1+x_2-1-x_4^2=0$

Transformation contraintes d'inégalité en contrainte d'égalité

$$\begin{bmatrix}2x_1^*\\2x_2^*\\0\\0\end{bmatrix}+u_1^*\begin{bmatrix}2x_1^*\\2x_2^*\\-2x_3^*\\0\end{bmatrix}+u_2^*\begin{bmatrix}-1\\-1\\0\\-2x_4^*\end{bmatrix}=0$$

u_1^* et u_2^* non restreint en signe

u_1^* et u_2^* différents de 0

Les méthodes à pénalités

Introduction

- Principe: Créer une nouvelle fonction objectif qui pénalise la violation d'une contrainte

$$\tilde{f}(x) = f(x) + p(x)$$

- Utiliser une méthode d'optimisation sans contraintes pour trouver une solution
- Méthodes courantes:
  - Fonction de pénalité intérieure
  - Fonction de pénalité extérieure
  - Lagrangien augmenté

Les méthodes à pénalités

Pénalité Extérieure

- Expression de la pénalité pour les contraintes violées

$$p(x) = R \sum_{j=1}^{m} max\left[0, g_j(x)^2\right] + R \sum_{k=m+1}^{p} \left[h_k(x)\right]^2$$

- Algorithme

1. Commencer avec une petite valeur de R
2. Minimiser $\tilde{f}(x) = f(x) + p(x)$
3. Augmenter R (d'un facteur 10 par exemple)
4. Tant que la convergence n'est pas assurée, repartir à l'étape 2

- Intérêt
  - Facile à programmer
  - Approche de l'optimum par la région non réalisable

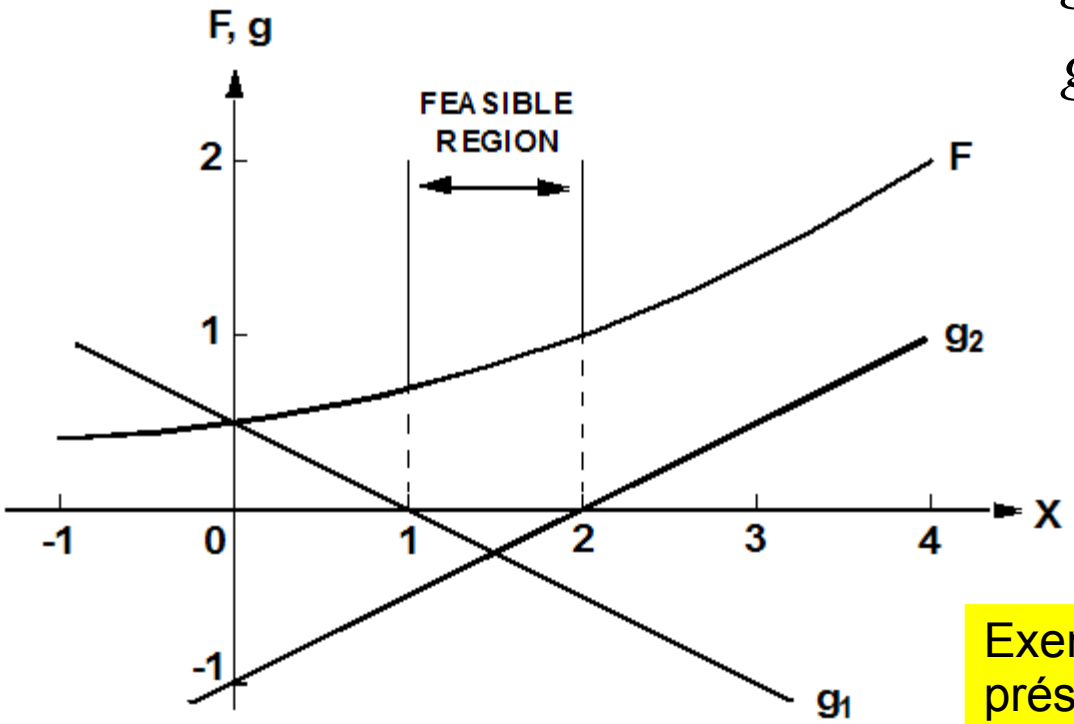
Les méthodes à pénalités

Pénalité Extérieure - Exemple

Minimiser $f(x)=\left(x^2+2x+8\right)/16$

Avec $g_1(x)=(1-x)/2\leq 0$

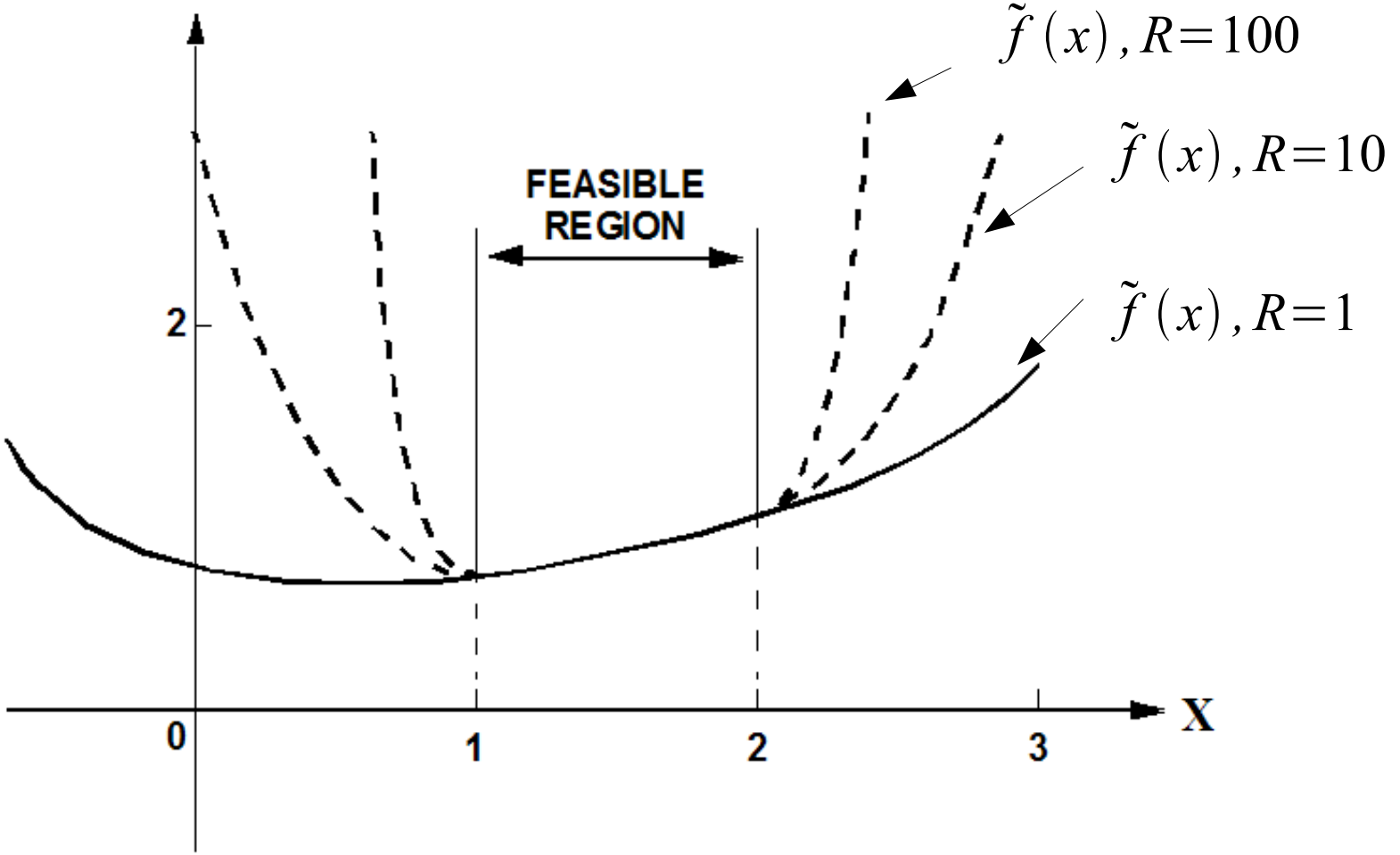
$g_2(x)=(x-2)/2\leq 0$



Exemple illustratif extrait d'une présentation de G. Vanderplaats

Les méthodes à pénalités

Pénalité Extérieure - Exemple



Les méthodes à pénalités

Pénalité Intérieure

- Expression de la pénalité pour les contraintes violées

$$p(x) = R' \sum_{j=1}^{m} -1/g_j(x) + R \sum_{k=m+1}^{p} \left[h_k(x)\right]^2$$

- Algorithme

1. Commencer avec une petite valeur de R et une grande valeur de R'
2. Minimiser $\tilde{f}(x)$
3. Augmenter R et diminuer R' (d'un facteur 10 par exemple)
4. Tant que la convergence n'est pas assurée, repartir à l'étape 2

Notes:

Toujours utiliser une pénalité extérieure pour les contraintes d'égalité

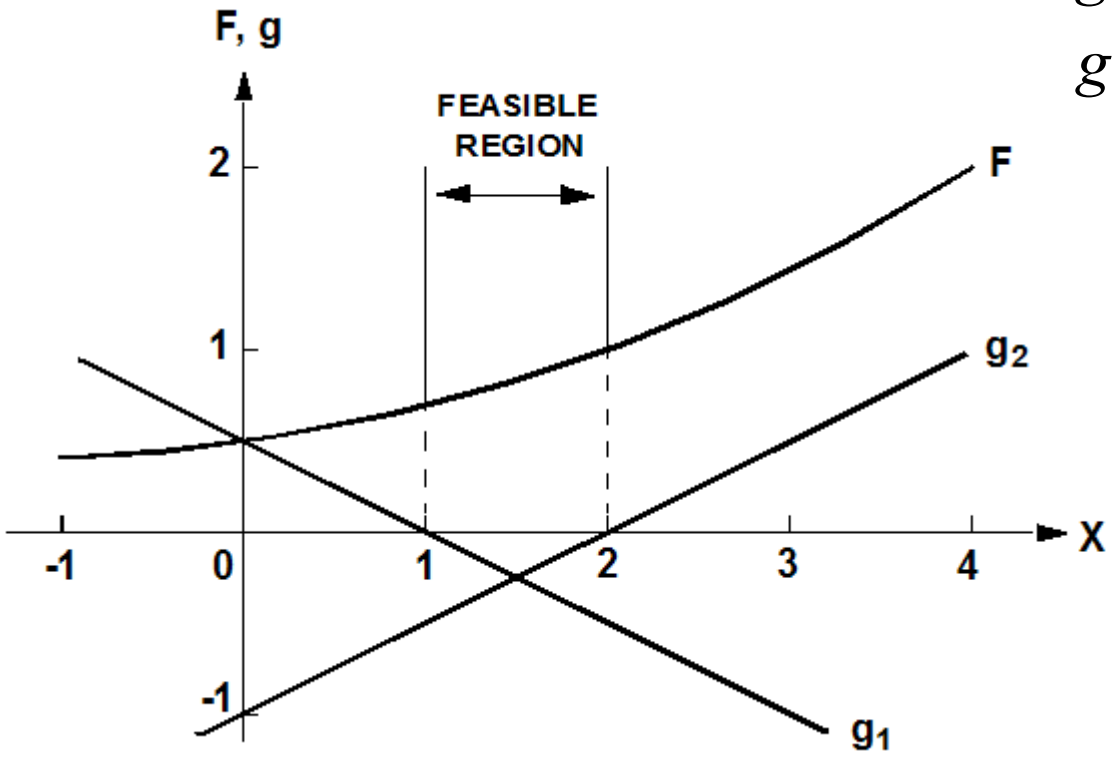
Les méthodes à pénalités

Pénalité Intérieure - Exemple

Minimiser $f(x)=\left(x^2+2x+8\right)/16$

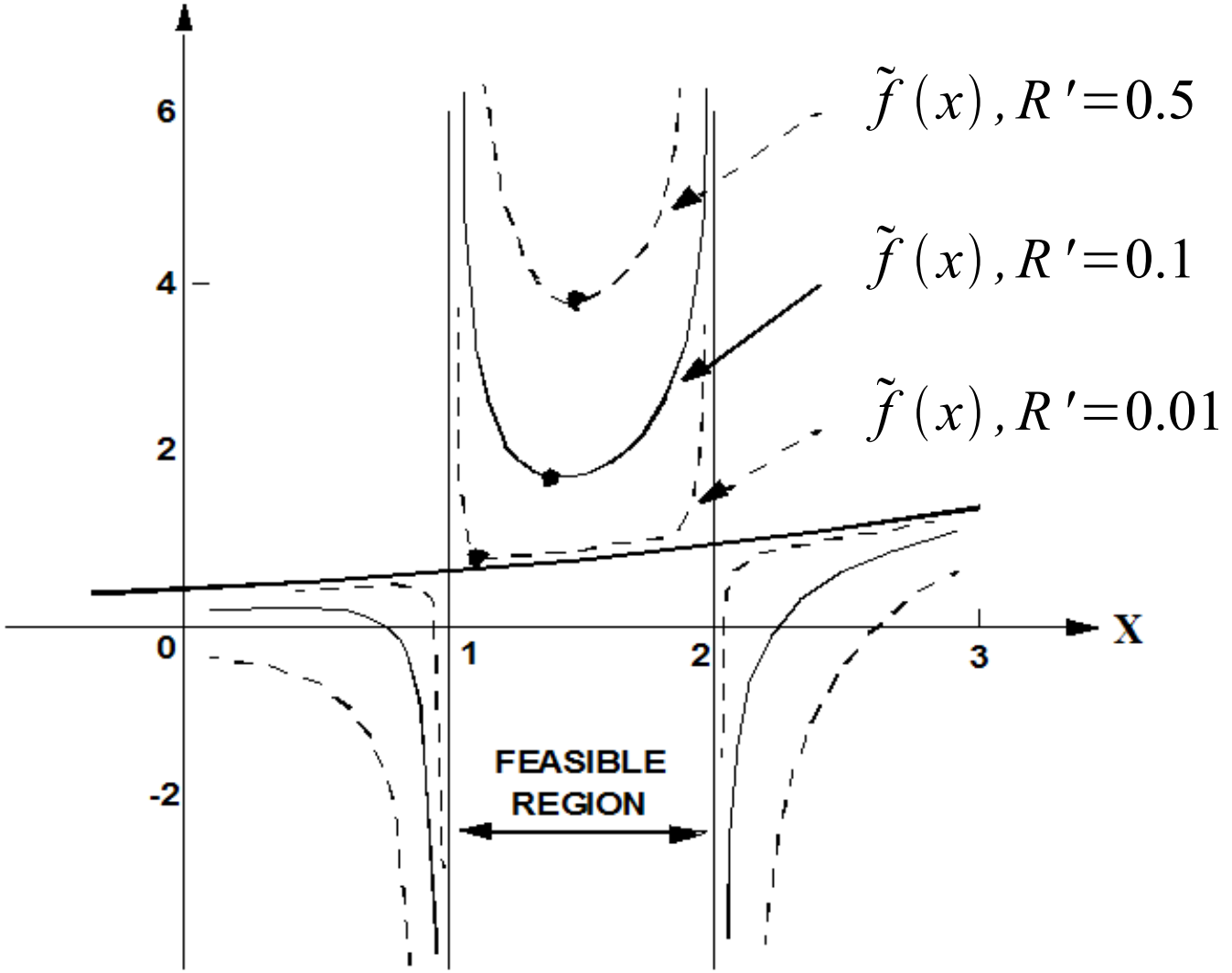
Avec $g_1(x)=(1-x)/2\leq 0$

$g_2(x)=(x-2)/2\leq 0$



Les méthodes à pénalités

Pénalité Intérieure - Exemple



Les méthodes à pénalités

Pénalité Intérieure - Autres

Formes alternatives:

- Fonction log:

$$p(x) = R' \sum_{j=1}^{m} -\log(-g_j(x))$$

- Fonction log décalée:

$$p(x) = R' \sum_{j=1}^{m} -\lambda_j \log\left(1 - \frac{g_j(x)}{R'}\right)$$

Pénalités

Pénalité extérieure:
Si les contraintes sont normalisées via la formule:

$$c_j = \frac{|\nabla f(x_0)|}{|\nabla g_j(x_0)|}$$

On normalise de façon à minimiser autant qu'on améliore les contraintes

Alors le coefficient k=1 est un bon paramètre de normalisation

Pénalité intérieure:
Si on normalise les contraintes en utilisant l'expression c_j, alors r_0' = 1 est un bon choix.
Sinon, choisir r_p' tel que $f(x_0) = -r_p \cdot P(x_0)$

$$r_0' = \frac{|f(x_0)|}{P(x_0)}$$

On s'arrange pour que la fonction objectif et la pénalisation soit équivalente

Les méthodes à pénalités

Lagrangien Augmenté

Minimiser le Lagrangien augmenté:

$$A(x,\lambda,R)=f(x)+\sum_{i=1}^{m}\left[\lambda_i \Psi_i+R\Psi_i^2\right]+\sum_{j=m+1}^{p}\lambda_j h_j(x)+R\left[h_i(x)^2\right]$$

Où:

$$\Psi_j=max\left[g_j(x),\frac{-\lambda_j}{2R}\right]$$

Les méthodes à pénalités

Lagrangien Augmenté

Si on connait λ*, une optimisation du Lagrangien fournit une solution optimale !

But: trouver un algorithme tel que, en partant de λ arbitraire (habituellement 0 ou 1), on converge rapidement vers l'optimum.
Solution partielle: on peut considérer λ_k comme des variables supplémentaires du problème.

Problème: ça augmente le nombre de variables du problème !

Lagrangien augmenté:
Soit on commence avec $\lambda_k=0$, sinon:

$$\lambda_k = 1.0 \text{ si } \nabla h_k(x).\nabla f(x) < 0$$
$$\lambda_k = -1.0 \text{ si } \nabla h_k(x).\nabla f(x) > 0$$

λ et r_p sont maintenus constant lors de la séquence d'optimisation.
Ensuite, on met les λ_k à jours.

Les méthodes à pénalités

Lagrangien Augmenté

Algorithme

1. Commencer avec une petite valeur de R et $\lambda_i = 0$
2. Minimiser $A(x, \lambda, R)$
3. Mise à jour des coefficients de Lagrange

$$\lambda_j = \lambda_j^{\text{Old}} + 2 \cdot R \cdot \left\{ max\left(g_j, \frac{-\lambda_j^{\text{Old}}}{2 \cdot R} \right) \right\} \qquad j = 1, \ldots, m$$

$$\lambda_j = \lambda_j^{\text{Old}} + 2 \cdot R \cdot h_j(x) \qquad j = m+1, \ldots, p$$

4. Augmenter R (R = min(10R, 1000))
5. Si pas de convergence, recommencer à l'étape 2

Les méthodes à pénalités

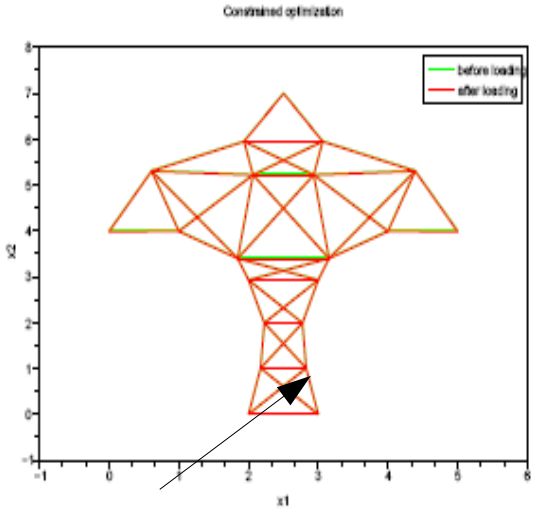
Lagrangien Augmenté

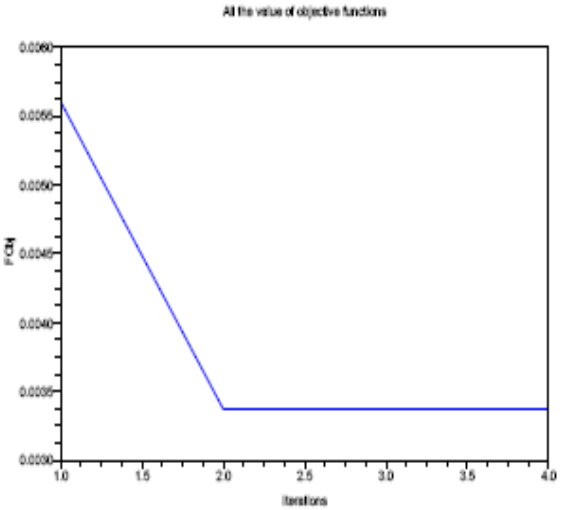
- Facile à programmer
- Considéré comme la meilleure fonction de pénalité
- A utiliser si la fonction objectif n'est pas coûteuse à évaluer
- La fonction de pénalité log décalée est considérée comme la meilleure fonction

Demo

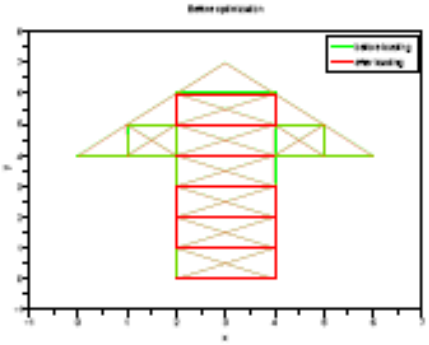
Les méthodes à pénalités

Lagrangien Augmenté





Solution avec codage symétrique de la structure du pylône



Structure de départ

Méthode des Directions Réalisables

Introduction

- Zoutendijk – 1960
- Utilisée par Conmin (optimiseur utilisé par la NASA)
- Trouve rapidement une solution
- Ne traite que des contraintes d'inégalité

Méthode des Directions Réalisables

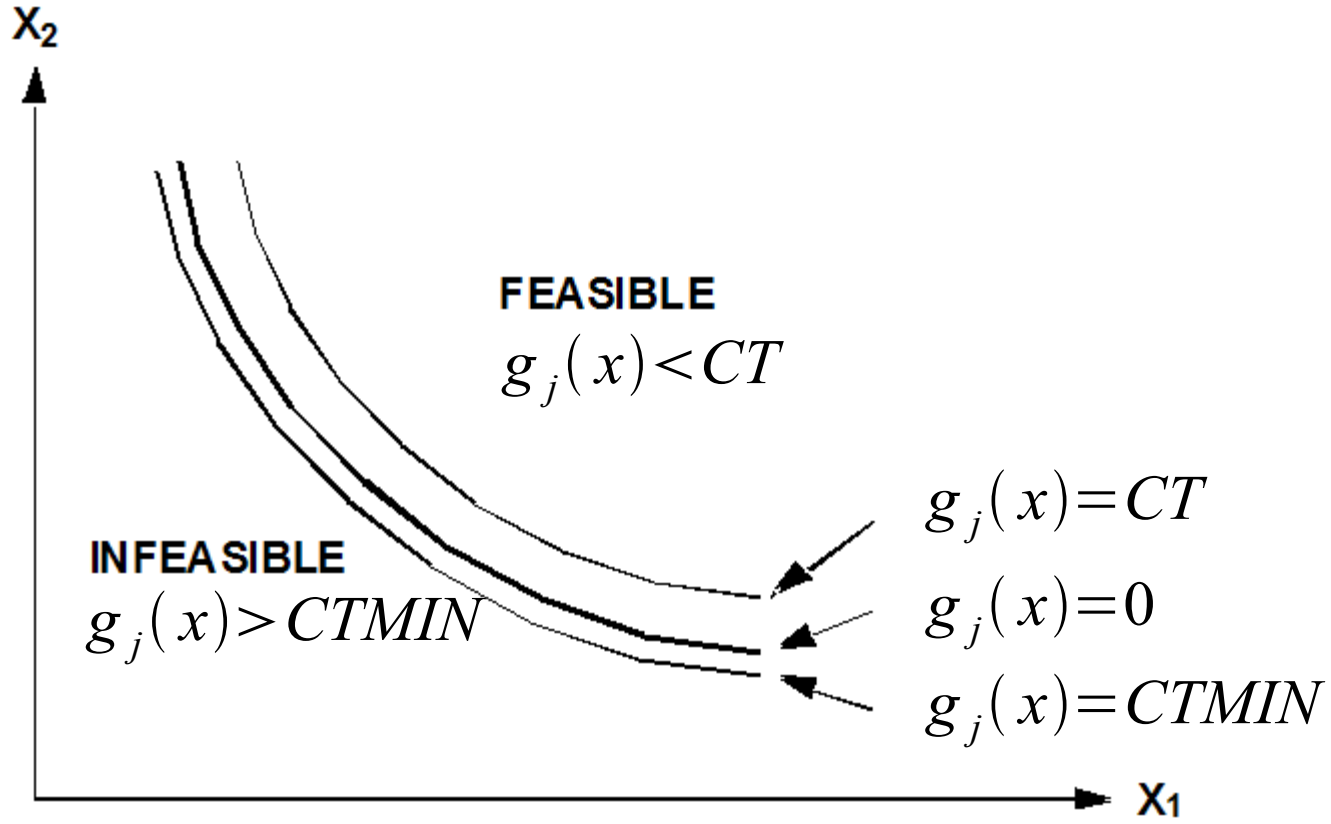
Algorithme

Algorithme

1. x_0 solution initial , compteur d'itérations q = 0
2. Evaluer $f(x), g_j(x) \forall j = 1, m$
3. q=q+1, évaluer $\nabla f(x), \nabla g_j(x)$ où $j \in J$ où J est l'ensemble des contraintes actives et des contraintes violées
   - $g_j(x)$ est active si $g_j(x) \geq CT$ (typiquement CT=-0.05)
   - $g_j(x)$ est violée si $g_j(x) \geq CTMIN$ (typiquement CTMIN=0.001)
4. Calculer la direction de recherche S_q
5. Effectuer une recherche unidirectionnelle dans la direction S_q
6. Recommencer à l'étape 3 jusqu'à convergence

Méthode des Directions Réalisables

Illustration des paramètres



Méthode des Directions Réalisables

Détails

Si aucune contrainte n'est active ni violée

- Si n=1, on utilise la méthode Steepest Descent

$$S_n = -\nabla f(x_{n-1})$$

- Si n>1, on utilise la méthode de Fletcher-Reeves

$$S_n = -\nabla f(x_{n-1}) + \beta \cdot S_{n-1} \text{ où } \beta = \frac{|\nabla f_{(x_{n-1})}|^2}{|\nabla f(x_{n-2})|^2}$$

- Recommencer avec la méthode Steepest-Descent chaque n+1 itérations ou quand la méthode ralenti

Méthode des Directions Réalisables

Détails

Si des contraintes sont actives

- Résoudre le sous-problème suivant: cherche et tels que

$$\begin{array}{lll} \text{Maximiser} & \beta & \\ \text{Avec} & \nabla f(x_{n-1})\cdot S_n+\beta\le 0 & S_n \text{ est utilisable} \\ & \nabla g_j(x_{n-1})\cdot S_n+\theta_j\cdot\beta\le 0,\ j\in J & S_n \text{ est réalisable} \\ & S_n^t\cdot S_n\le 1 & s_n \text{ est borné} \end{array}$$

- Où J est l'ensemble des contraintes actives et θ_j est appelé facteur de répulsion (push-off factor)

Méthode des Directions Réalisables

Détails

Si des contraintes sont violées

- Résoudre le sous-problème suivant: chercher β et S_n tels que

$$\begin{array}{lll} \text{Minimiser} & \nabla f(x)^t \cdot S_n - \Phi \cdot \beta & S_n \text{ est utilisable} \\ \text{Avec} & \nabla g_j(x^{n-1}) \cdot S_n + \theta \cdot \beta \le 0,\ j \in J & S_n \text{ est réalisable} \\ & S_q^t \cdot S_n \le 1 & s_n \text{ est borné} \end{array}$$

- Où J est l'ensemble des contraintes actives et θ_j est appelé facteur de répulsion (push-off factor) et Φ est un grand nombre positif

Méthode des Directions Réalisables

Facteur de répulsion

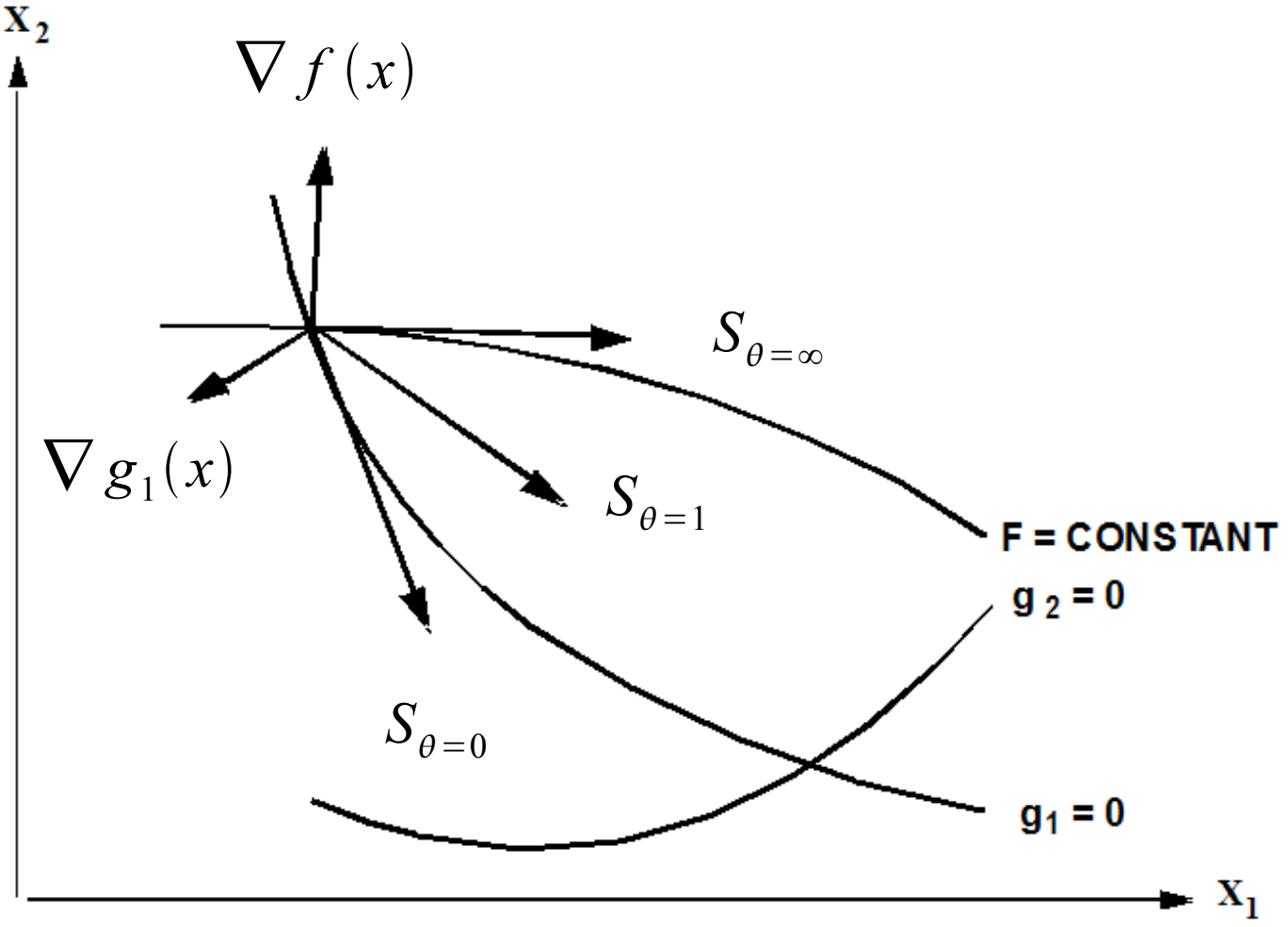
- Si une contrainte devient juste active, autoriser la méthode de recherche à suivre la contrainte
- Plus la contrainte devient active ou violée, plus on pousse à suivre la contrainte
- Si $g_j(x) \geq CT$

$$\theta_j = \left[1 - \frac{g_j(x_{n-1})}{CT} \right] \cdot \theta_0$$

- θ_j est une fonction quadratique de la valeur de la contrainte

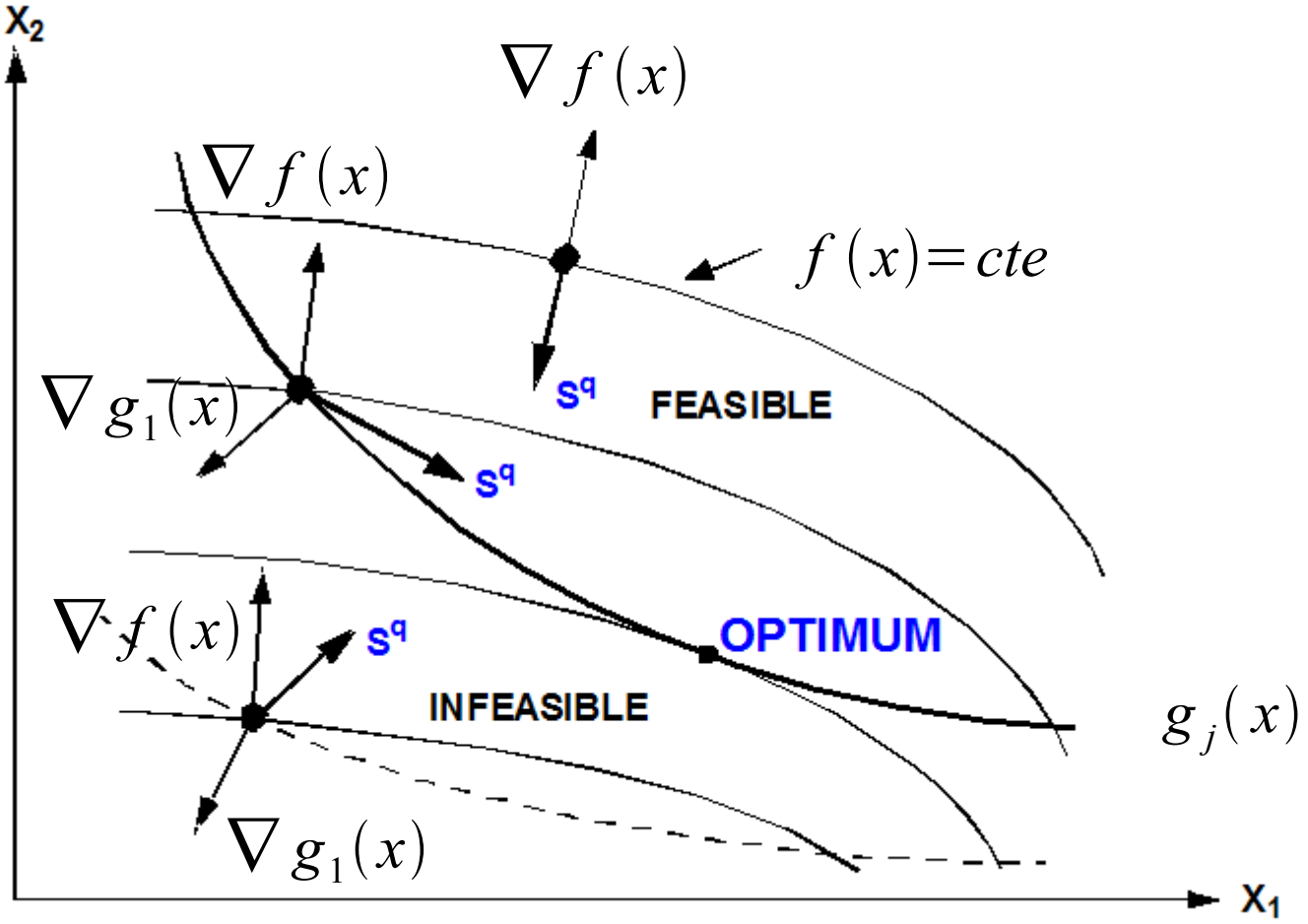
Méthode des Directions Réalisables

Calcule de la direction de recherche

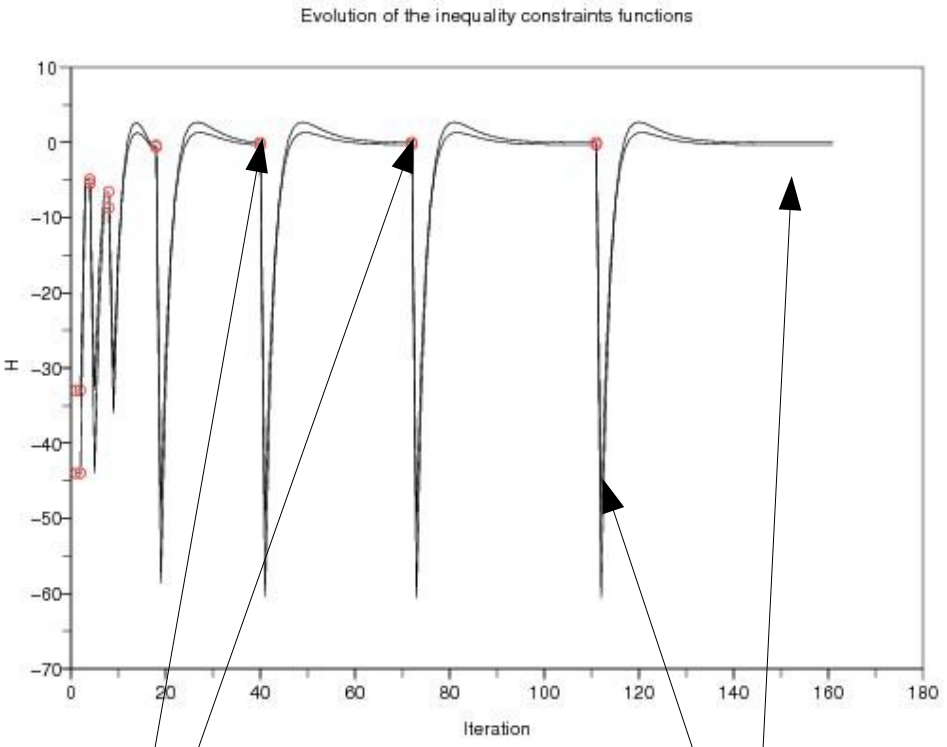


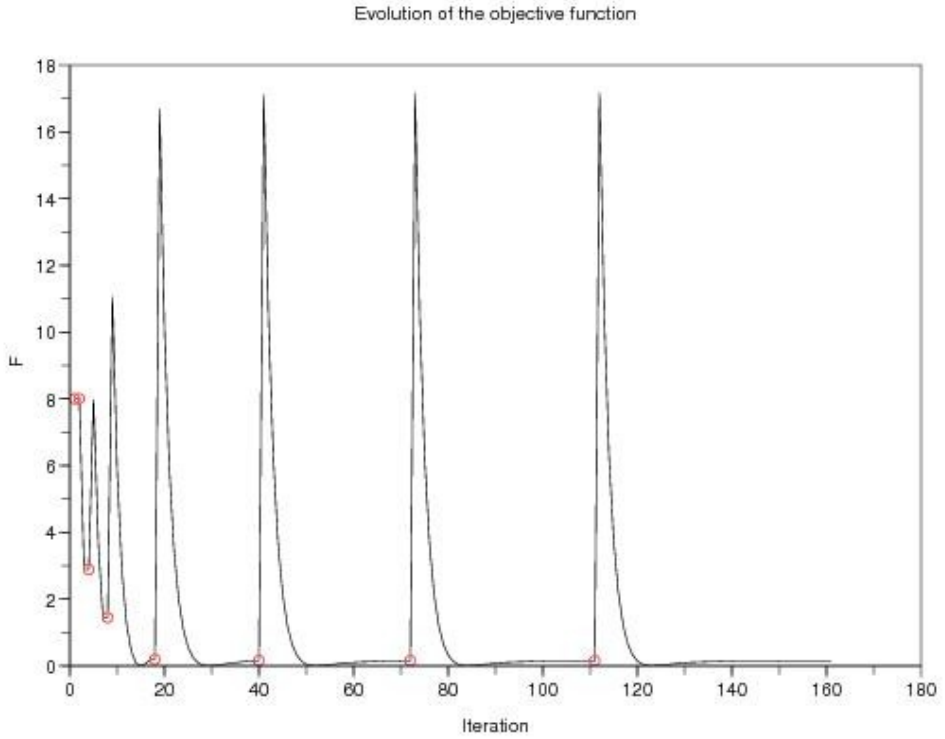
Méthode des Directions Réalisables

Calcule de la direction de recherche



Méthode des Directions Réalisables





Meilleure solution trouvée à chaque itération

Comportement de la recherche en ligne

$$\min f(x) = x_1^2 + x_2^2$$

$$\text{avec} \quad -\begin{bmatrix} 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 3 & 4 \end{bmatrix} \cdot \begin{bmatrix} x_1^2 \\ x_2^2 \end{bmatrix} + 1 \leq 0$$

$$-\begin{bmatrix} 1 & 4 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 4 & 5 \end{bmatrix} \cdot \begin{bmatrix} x_1^2 \\ x_2^2 \end{bmatrix} + 2 \leq 0$$

La Méthode LFOPc

Algorithme

Phase 0:
$\mu=10^2$, $\gamma=1$
On applique LFOP à p(x)
On récupère Δt_f et xopt
On note $n_{ineq}^{\ a}$ le nombre des contraintes actives
si $n_{ineq}^{\ a}+n_{eq}=0$ alors STOP
si $n_{ineq}^{\ a}+n_{eq}\neq 0$ ou $\|\text{grad } f(x_{opt})\|\leq 10^2.\varepsilon_x$ allez à la phase 2
sinon allez à la phase 1
Phase 1:
$\mu=10^4$, $\gamma=1$, $\Delta t=\Delta t_f$
On applique LFOP à p(x)
On récupère Δt_f et xopt
On note $n_{ineq}^{\ a}$ le nombre des contraintes actives
si $n_{ineq}^{\ a}+n_{eq}=0$ alors STOP
Phase 2:
On conserve μ et Δt_f et x0 = x_{opt}. $\gamma=0$
On applique LFOP à p(x) avec:

Séquence de pénalisation 1

Séquence de pénalisation 2

Séquence de vérification des contraintes

$$\nabla p(x)=\gamma\cdot\nabla f(x)+\sum_{i=1}^{n_{ineq}} 2\cdot\alpha_i\cdot g_i(x)\cdot\nabla g_i(x)+\sum_{j=1}^{n_{eq}} 2\cdot\beta\cdot h_j(x)\cdot\nabla h_j(x)$$

$$\alpha_i=\begin{cases}\dfrac{\mu\cdot\left(\|\gamma\cdot\nabla f(x)\|+1\right)}{\|\nabla g_i(x)\|^2} & \text{si} \quad g_i(x)>0\\ 0 & \text{si} \quad g_i(x)\leq 0\end{cases}$$

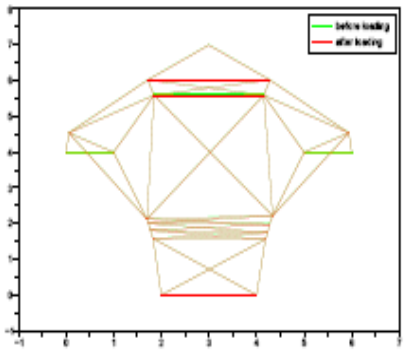
$$\beta_j=\begin{cases}\dfrac{\mu\cdot\left(\|\gamma\cdot\nabla f(x)\|+1\right)}{\|\nabla h_j(x)\|^2} & \text{si} \quad h_j(x)>0\\ 0 & \text{si} \quad h_j(x)\leq 0\end{cases}$$

$$\alpha_i=\begin{cases}\dfrac{\mu}{\|\nabla g_i(x)\|} & \text{si} \quad g_i(x)>0\\ 0 & \text{si} \quad g_i(x)\leq 0\end{cases}$$

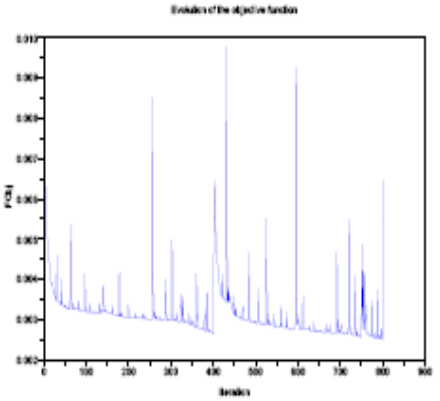
$$\beta_j=\frac{\mu}{\|\nabla h_j(x)\|}$$

La Méthode LFOPc

Exemples

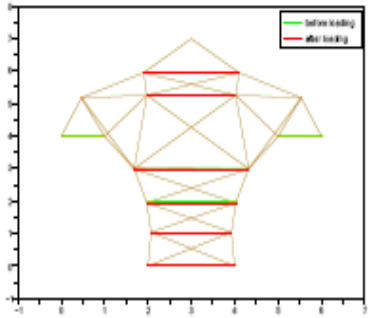


(a) Solution obtenue

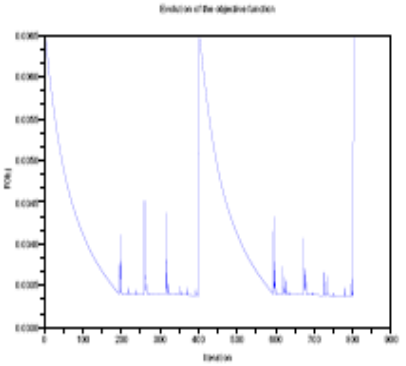


(b) Evolution de la fonction objectif

Mauvais paramétrage de la méthode:
A la fin de l'optimisation, les contraintes ne sont pas respectées



(a) La solution obtenue



(b) L'évolution de la fonction objectif

Bon paramétrage de la méthode:
la descente se fait sans changement de Δt
la fin de l'optimisation voit Δt changer pour mieux localiser l'optimum
les contraintes sont respectées