

Controle Dinâmico da Inviabilidade em Problemas com Restrições de Desigualdade

Abel Soares Siqueira
Orientador: Francisco A. M. Gomes

23 de Novembro de 2012

Projeto financiado pela Fapesp. Processo 2009/17273-4



Introdução

- Objetivos:
 - Estender o método de Controle Dinâmico da Infactibilidade para Desigualdades
 - Implementar uma interface em C++ com suporte ao repositório CUTer.
 - Esta interface deve ser livre, e disponível facilmente online.



DCI para igualdades

$$\begin{array}{ll}\min & f(x) \\ \text{s.a} & h(x) = 0,\end{array}$$

$$f : \mathbb{R}^n \rightarrow \mathbb{R}, h : \mathbb{R}^n \rightarrow \mathbb{R}^m, f, h \in C^2.$$



DCI para igualdades

$$\mathcal{C}(\rho) = \{x \in \mathbb{R}^n : \|h(x)\| \leq \rho\}$$

$$g_p(x) = \nabla f(x) + \nabla h(x)^T \lambda_{LS}(x)$$

$$\lambda_{LS}(x) = \arg \min_{\lambda} \{\|\nabla h(x)^T \lambda + \nabla f(x)\|^2\}$$

$$\rho^k = \mathcal{O}(\|g_p(x_c^k)\|)$$

$$L(x, \lambda) = f(x) + \sum_{i=1}^m h_i(x) \lambda_i$$

DCI para igualdades

Método

Passo Horizontal

$$\begin{aligned} \min \quad & q_k(\delta) = \frac{1}{2} \delta^T B^k \delta + \delta^T g_p(x_c^k), \\ & \nabla h(x_c^k) \delta = 0, \|\delta\| \leq \Delta_H \end{aligned}$$

Passo Vertical

$$\min \frac{1}{2} \|h(x)\|^2$$

DCI para igualdades

Método

Algorithm 1 Iteração k do Algoritmo DCI

- 1: Dado x^{k-1} ,
 - 2: Calcule ρ^k e obtenha $x_c^k \in \mathcal{C}(\rho^k)$, minimizando aproximadamente $\|h(x)\|$,
 - 3: **se** $\|g_p(x_c^k)\| < \varepsilon$ **então**
 - 4: PARE com $x^* = x_c^k$.
 - 5: **fim do se**
 - 6: Obtenha λ^k
 - 7: Obtenha δ_t resolvendo $\min q_k(\delta)$ sujeito a $\nabla h(x_c^k)\delta = 0$, $\|\delta\| \leq \Delta_H$.
 - 8: Caso $x_c^k + \delta_t \notin \mathcal{C}(2\rho^k)$, ou não houve decréscimo suficiente, diminua δ_H e repita o passo anterior
 - 9: Caso necessário, faça uma correção de segunda ordem δ_{soc} .
 - 10: Defina $x^k = x_c^k + \delta_t + \delta_{\text{soc}}$.
-

DCI para igualdades

Exemplo

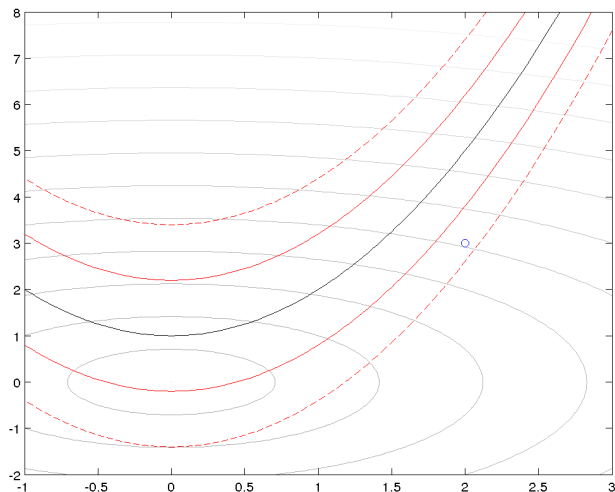
Exemplo:

$$\begin{array}{ll} \min & f(x) = \frac{1}{2}(x_1^2 + x_2^2) \\ \text{s.a} & x_2 = x_1^2 + 1 \end{array}$$

$$x^{k-1} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

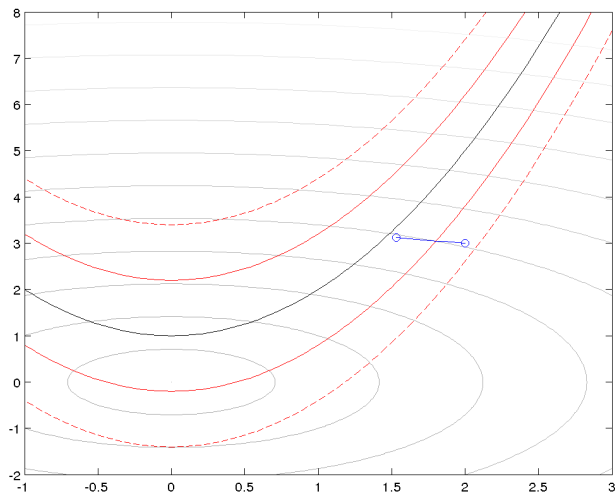
DCI para igualdades

Exemplo



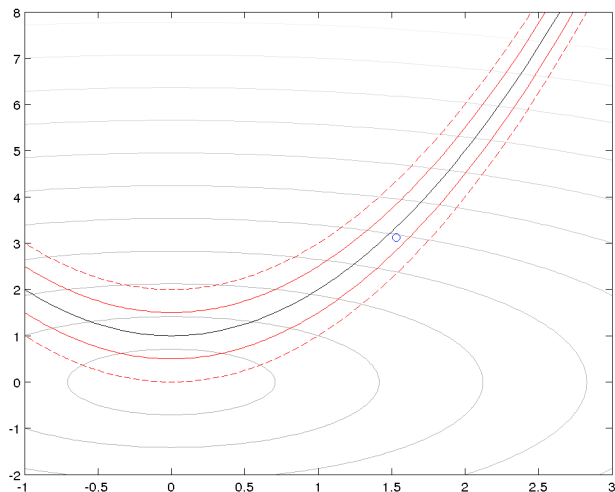
DCI para igualdades

Exemplo



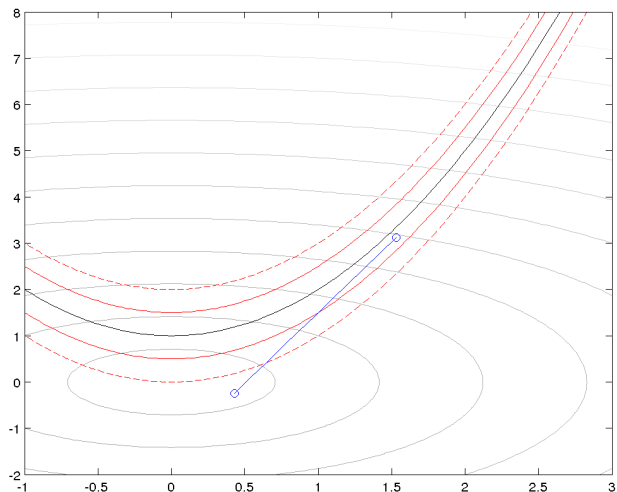
DCI para igualdades

Exemplo



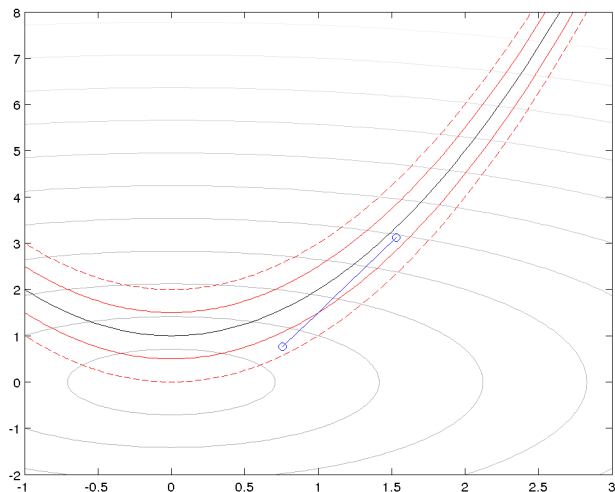
DCI para igualdades

Exemplo



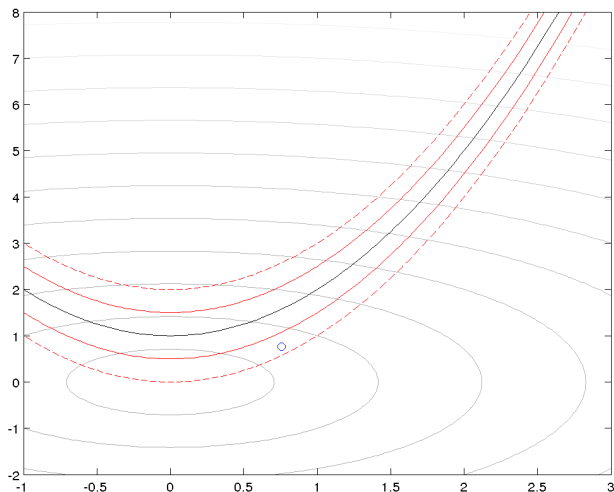
DCI para igualdades

Exemplo



DCI para igualdades

Exemplo



DCI para igualdades

Convergência

Hipótesis

H1 $f, h \in C^2$

H2 As sequências $\{x_c^k\}$, $\{x^k\}$, as aproximações B^k e os multiplicadores $\{\lambda^k\}$ permanecem uniformemente limitados.

H3 A restauração nunca falha e $Z = \{x_c^k\}$ permanece afastado do conjunto singular de h , no sentido que h é regular no fecho de Z . Além disso, se a sequência $\{x_c^k\}$ é infinita, ela satisfaz

$$\|x_c^{k+1} - x^k\| = \mathcal{O}(\|h(x^k)\|).$$

$$\text{H4} \quad \|\delta_{soc}^k\| = \mathcal{O}(\|\delta_t^k\|^2).$$

DCI para igualdades

Convergência

Teorema: Sob as hipóteses H1-H4, DCI para em um ponto estacionário em um número finito de iterações, ou gera uma sequência com pontos estacionários em seu conjunto de acumulação. Além disso, se

$$\begin{aligned}\|x^k - x_c^k\| &= \mathcal{O}(\|g_p(x_c^k)\|), \\ \|\lambda^k - \lambda_{LS}(x_c^k)\| &= \mathcal{O}(\|g_p(x_c^k)\|),\end{aligned}$$

então todo ponto de acumulação de x_c^k é estacionário.



DCI para igualdades

Convergência

Suponha que $\{x_c^k\}$ e $\{\lambda^k\}$ convergem a x^* , ponto estacionário com multiplicador $\lambda^* = \lambda_{LS}(x^*)$, e que existe $\theta_1 > 0$, tal que

$$\theta_1 \|y\|^2 \leq y^T \nabla_{xx}^2 L(x^*, \lambda^*) y,$$

para $y \in \mathcal{N}(\nabla h(x^*))$. Também defina

$P(x) = I - \nabla h(x)^T [\nabla h(x) \nabla h(x)^T]^{-1} \nabla h(x)$ numa vizinhança de x^* .

A1 $\|\lambda^k - \lambda_{LS}(x_c^k)\| = \mathcal{O}(\|g_p(x_c^k)\|)$.

A2 B^k é assintoticamente uniformemente positiva-definida no espaço tangente às restrições, isto é, em alguma vizinhança de x^* , podemos redefinir θ_1 , e definir θ_2 tais que

$$\theta_1 \|y\|^2 \leq y^T B^k y \leq \theta_2 \|y\|^2,$$

para $y \in \mathcal{N}(\nabla h(x_c^k))$.

DCI para igualdades

Convergência

A3 Seja δ_{HN}^k o minimizador $q_k(\delta)$ com $\nabla h(x_c^k)\delta = 0$. Suponha que, sempre que $\|\delta_{HN}^k\| \leq \Delta_H$, ele é o primeiro passo horizontal tentado pelo algoritmo. Além disso, suponha também que

$$P(x_c^k)[B^k - \nabla_{xx}^2 L(x^*, \lambda^*)]\delta_{HN}^k = o(\|\delta_{HN}^k\|).$$

A4 Todo passo vertical não-nulo $\delta_V^{k+1} = x_c^{k+1} - x^k$, é calculado tomando um ou mais passos da forma

$$\delta_V^+ = -A^T(AA^T)^{-1}h(x_c),$$

onde A satisfaz

$$\|A - \nabla h(x_c)\| = \mathcal{O}(\|g_p(x_c^k)\|).$$

DCI para igualdades

Convergência

Teorema: Sob as hipóteses H1-H4 e A1-A4, x^k e x_c^k são superlinearmente convergentes a x^* de 2 passos. Se uma restauração for feita à cada x^k , então x^k converge superlinearmente a x^* .



Extensão

Problema do CUTer:

$$\begin{array}{ll} \min & f(x) \\ \text{s.a} & c_E(x) = 0, \\ & c_L \leq c_I(x) \leq c_U, \\ & b_L \leq x \leq b_U \end{array} \quad (1)$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $c_E : \mathbb{R}^n \rightarrow \mathbb{R}^{m_E}$, $c_I : \mathbb{R}^n \rightarrow \mathbb{R}^{m_I}$,
 $f, c_E, c_I \in C^2$, $c_{U_i}, b_{U_i} \in \mathbb{R} \cup \{\infty\}$, $c_{L_i}, b_{L_i} \in \mathbb{R} \cup \{-\infty\}$.

Extensão

$$\begin{array}{ll} \min & f(x) \\ \text{s.a} & c_E(x) = 0, \\ & c_I(x) \geq 0, \end{array} \quad (2)$$

$$\begin{array}{ll} \min & f(x) \\ \text{s.a} & c_E(x) = 0, \\ & c_I(x) - s = 0, \\ & s \geq 0. \end{array} \quad (3)$$

Extensão

Definições

$$z = \begin{bmatrix} x \\ s \end{bmatrix}$$

$$\varphi(z, \mu) = f(x) - \mu \sum_{i=1}^{m_I} \ln(s_i)$$

$$h(z) = \begin{bmatrix} c_E(x) \\ c_I(x) - s \end{bmatrix}$$

$$\begin{array}{ll} \min & \varphi(z, \mu) \\ \text{s.a} & h(z) = 0. \end{array} \quad (4)$$

Extensão

Definições

$$\Lambda(z) = \begin{bmatrix} I & 0 \\ 0 & S \end{bmatrix},$$

$$g(z, \mu) = \begin{bmatrix} \nabla f(x) \\ -\mu e \end{bmatrix} = \Lambda(z) \nabla_z \varphi(z, \mu),$$

$$A(z) = \begin{bmatrix} \nabla c_E(x) & 0 \\ \nabla c_I(x) & -S \end{bmatrix} = \nabla h(z) \Lambda(z),$$

$$\Gamma(z, \mu) = \begin{bmatrix} \nabla^2 f(x) & 0 \\ 0 & \mu I \end{bmatrix} = \Lambda(z) \nabla_{zz}^2 \varphi(z, \mu) \Lambda(z),$$

$$\begin{aligned} W(z, \lambda, \mu) &= \Gamma(z, \mu) + \sum_{i=1}^m \lambda_i \begin{bmatrix} \nabla^2 c_i(x) & 0 \\ 0 & 0 \end{bmatrix}, \\ &= \Lambda(z) \nabla_{zz}^2 L(z, \lambda, \mu) \Lambda(z) \end{aligned}$$

Extensão

Definições

$$\lambda_{LS}(z, \mu) = \arg \min_{\lambda} \left\{ \frac{1}{2} \|g(z, \mu) + A(z)^T \lambda\|^2 \right\}$$

$$\lambda_i^k = \begin{cases} \lambda_{LS}(z_c^k, \mu_c^k)_i, & \text{se } i \in E \\ \min\{\lambda_{LS}(z_c^k, \mu_c^k)_i, \alpha(\mu_c^k)^n\}, & \text{se } i \in I \end{cases}$$

$$g_p(z, \mu) = g(z, \mu) + A(z)^T \lambda_{LS}(z, \mu).$$

e

$$g_p^k = g(z_c^k, \mu_c^k) + A(z_c^k)^T \lambda^k$$

Método

Algorithm 2 Iteração k do Algoritmo DCI estendido

- 1: Dado z^{k-1}
 - 2: Faça o passo vertical, obtendo z_c^k e ρ^k tais que $z_c^k \in \mathcal{C}(\rho^k)$.
 - 3: Calcule λ^k e $\mu_c^k = \min\{\mu^{k-1}, \theta_\rho \rho^k, \theta_\rho (\rho^k)^2\}$.
 - 4: **se** $\|g_p^k\| < \varepsilon$ e $\|h(z_c^k)\| < \varepsilon$ e $\mu_c^k < \varepsilon$. **então**
 - 5: PARE com $z^* = z_c^k$.
 - 6: **fim do se**
 - 7: Atualize ρ_{\max}^k
 - 8: Obtenha δ_t resolvendo $\min q_k(\delta)$ sujeito a $A(z_c^k)\delta = 0$, $\|\delta\| \leq \Delta_H$.
 - 9: Caso $z_c^k + \delta_t \notin \mathcal{C}(2\rho^k)$, ou não houve decréscimo suficiente, diminua Δ_H e repita o passo anterior.
 - 10: Caso necessário, faça uma correção de segunda ordem δ_{soc} .
 - 11: Defina $z^k = z_c^k + \Lambda(z_c^k)(\delta_t + \delta_{\text{soc}})$
 - 12: Defina $\mu^k = \min\{\mu_c^k, (\lambda_I^k)^T s_c^k / m_I, \alpha_h \|h(z^k)\|\}$.
-

Método

Passo Vertical

A partir de z^{k-1} são feitos passos que resolvem aproximadamente

$$\min \frac{1}{2} \|h(z)\|^2 \quad \text{s.a} \quad s \geq 0$$

Para isso, utilizamos uma modificação do método proposto por Francisco, Krejić e Martínez ([4]), para resolver o problema

$$\min r(z) = \frac{1}{2} \|h(z)\|^2 \quad \text{s. a} \quad l \leq z \leq u$$

Método

Passo Vertical

Algorithm 3 Passo Vertical

- 1: Dado z^{k-1} , defina $z_c = z^{k-1}$ e $\rho = \rho^{k-1}$.
 - 2: **enquanto** $\|h(z_c)\| > \rho$ **faça**
 - 3: Encontre z_c tal que $\|h(z_c)\| < \rho$ pelo **Passo Vertical Interno**.
 - 4: Atualize ρ .
 - 5: **se** $\|h(z_c)\| > \varepsilon$ **e** $\|\nabla h(z_c)^T h(z_c)\| < \varepsilon$ **então**
 - 6: Declare o problema infactível e pare com $z^* = z_c$ ponto estacionário do inviabilidade.
 - 7: **fim do se**
 - 8: **fim do enquanto**
 - 9: Defina $z_c^k = z_c$ e $\rho^k = \rho$.
-

Método

Passo Vertical

Algorithm 4 Passo Vertical Interno

- 1: **enquanto** $\|h(z_c)\| > \rho$ **faça**
- 2: Defina $m(d) = \frac{1}{2}\|\nabla h(z_c)d + h(z_c)\|^2$
- 3: Calcule $g = \nabla m(0) = \nabla h(z_c)^T h(z_c)$
- 4: Defina a matriz $D = \text{diag}(v_1, \dots, v_N)$, onde

$$v_i = \begin{cases} (u_i - z_i)^{-1/2}, & \text{se } g_i < 0 \text{ e } u_i < \infty \\ (z_i - l_i)^{-1/2}, & \text{se } g_i > 0 \text{ e } l_i > -\infty \\ 1, & \text{caso contrário} \end{cases}$$

- 5: Defina $d = -D^{-2}g$
- 6: Defina $l_\varepsilon = l + \varepsilon_\mu(z_c - l) - z_c$ e $u_\varepsilon = u - \varepsilon_\mu(u - z_c) - z_c$.
- 7: Defina $\beta(d) = \arg \max\{t \geq 0 : l_\varepsilon \leq td \leq u_\varepsilon\}$.
- 8: Calcule $\alpha_{CP} = \arg \min_\alpha \{m(\alpha d) : \alpha\|Dd\| \leq \Delta_V\}$

Método

Passo Vertical

-
- 9: Defina $P(d) = \begin{cases} d, & \text{se } \beta(d) > 1 \\ \max\{\theta, 1 - \|d\|\}\beta(d)d, & \text{caso contrário} \end{cases}$
 - 10: Defina $d_{CP} = P(\alpha_{CP}d)$.
 - 11: Defina $\rho_C(d) = \frac{m(0)-m(d)}{m(0)-m(d_{CP})}$ e $\rho_h(d) = \frac{r(z_c)-r(z_c+d)}{m(0)-m(d)}$.
 - 12: Calcule \tilde{d}_N , solução aproximada de $\min_d \{m(d) : \|Dd\| \leq \Delta_V\}$.
 - 13: Defina $d_N = P(\tilde{d}_N)$.
 - 14: Encontre \tilde{d} combinação convexa de d_{CP} e d_N tal que $\rho_C(\tilde{d}) \geq \beta_1$.
 - 15: **se** $\rho_h(\tilde{d}) \geq \beta_2$ **então**
 - 16: $\Delta_V \leftarrow 2\Delta_V$
 - 17: $z_c \leftarrow z_c + \tilde{d}$.
 - 18: **senão**
 - 19: $\Delta_V \leftarrow \Delta_V/4$.
 - 20: **fim do se**
 - 21: **fim do enquanto**
-

Método

Passo Horizontal

$$\begin{aligned}
 \min \quad & q(\delta) = \frac{1}{2} \delta^T B^k \delta + \delta^T g(z_c^k) \\
 \text{s.a} \quad & A(z_c^k) \delta = 0, \\
 & \tilde{l} \leq \Lambda(z_c^k) \delta \leq \tilde{u},
 \end{aligned}$$

com $B^k \approx W(z_c^k, \lambda^k, \mu_c^k)$ e

$$\tilde{l}_i = \begin{bmatrix} -\Delta_H e \\ \max\{-\Delta_H e, (\varepsilon_\mu - 1)s_c^k\} \end{bmatrix} \quad \tilde{u} = \Delta_H e$$

Este método é resolvido com uma modificação do método de Steihaug.



Método

Passo Horizontal

Algorithm 5 Passo Horizontal Interno

- 1: Dados $r^0 = g_p^k$, $p^0 = r^0$, $j = 0$, $\delta^0 = 0$, $\theta^0 = \langle r^0, r^0 \rangle$.
- 2: **enquanto** $\theta^j > \varepsilon$ **e** $\theta^k > \varepsilon \theta^0$ **faça**
- 3: **se** $\langle \delta^j, B^k \delta^j \rangle \leq \varepsilon \theta^j$ **então**
- 4: Defina $\delta_t = \delta^j + \nu p^j$ tal que $\tilde{l} \leq \Lambda(z_c^k) \delta_t \leq \tilde{u}$ e ν minimiza $q(\delta^j + \nu p^j)$.
- 5: **fim do se**
- 6: $\alpha^j = \theta^j / \langle \delta^j, B^k \delta^k \rangle$
- 7: **se** $\delta^j + \alpha^j p^j < \tilde{l}$ **ou** $\delta^j + \alpha^j p^j > \tilde{u}$ **então**
- 8: Defina $\delta_t = \delta^j + \bar{\nu} p^j$, onde $\bar{\nu} = \arg \max \{ \nu : \tilde{l} \leq \Lambda(z_c^k) \delta_t \leq \tilde{u} \}$.
- 9: **fim do se**
- 10: $\delta^{j+1} = \delta^j + \alpha^j p^j$.
- 11: $r^{j+1} = \text{proj}_{\mathcal{N}(A(z_c^k))}(r^j - \alpha^j B^k p^j)$
- 12: $\theta^{j+1} = \langle r^{j+1}, r^{j+1} \rangle$; $\beta^{k+1} = \theta^{j+1} / \theta^j$; $p^{j+1} = r^{j+1} - \beta^j p^j$.
- 13: **fim do enquanto**

Convergência

- H1 f , c_E e c_I são C^2 .
- H2 As sequências $\{z_c^k\}$ e $\{z^k\}$, as aproximações B^k e os multiplicadores $\{\lambda^k\}$ permanecem uniformemente limitados.
- H3 A restauração não falha e $\mathcal{Z} = \{z_c^k\}$ permanece longe do conjunto singular de h , no sentido que h é regular no fecho de \mathcal{Z} . Além disso, se a sequência gerada $\{z_c^k\}$ é infinita, então

$$\|z_c^{k+1} - z^k\| = \mathcal{O}(\|h(z^k)\|) \quad (5)$$

H4 $\|\delta_{soc}^k\| = \mathcal{O}(\|\delta_t^k\|^2)$



Convergência

Convergência Global

Teorema *Sob H1-H4, DCI para em um ponto estacionário para (2), em um número finito de iterações, ou gera uma sequência com pontos estacionários em seu conjunto de acumulação. Além disso, se as condições*

$$C1 \quad \|z^k - z_c^k\| = \mathcal{O}(\|g_p(z_c^k, \mu_c^k)\|)$$

$$C2 \quad \|\lambda^k - \lambda_{LS}(z_c^k, \mu_c^k)\| = \mathcal{O}(\|g_p(z_c^k, \mu_c^k)\|)$$

$$C3 \quad \lambda_{LS}(z_c^{k+1}, \mu_c^{k+1})^T (s_c^{k+1} - s^k) = \mathcal{O}(\|g_p(z_c^k, \mu_c^k)\| \rho^k)$$

são satisfeitas, então todo ponto de acumulação de x_c^k é estacionário para (2).



Convergência

Convergência Local

Sejam $\{z^k\}$ e $\{z_c^k\}$ geradas pelo algoritmo, convergentes a z^* , $\{\lambda^k\}$ convergente a $\lambda^* = \lambda_{LS}(z^*, 0)$. Pelo algoritmo, temos

$$\left\{ \begin{array}{rcl} \nabla f(x^*) + \nabla c(x^*)^T \lambda^* & = & 0, \\ c_E(x^*) & = & 0, \\ c_I(x^*) & \geq & 0, \\ c_I(x^*)^T \lambda_I^* & = & 0, \\ \lambda_I^* & \leq & 0. \end{array} \right.$$

Defina $\mathcal{A}(x) = \{i \in E \cup I : c_i(x) = 0\}$, e $\mathcal{A}^* = \mathcal{A}(x^*)$. Defina λ_A^k e λ_A^* como as componentes de λ^k e λ^* , respectivamente, correspondentes às restrições ativas.

Convergência

Convergência Local

Suponha que $V = \{\nabla c_i(x^*) : i \in \mathcal{A}^*\}$ é L.I. e que existe $\theta_1 > 0$ tal que

$$y^T \left[\nabla^2 f(x^*) + \sum_{i \in \mathcal{A}^*} \nabla^2 c_i(x^*) \lambda_i^* \right] y \geq \theta_1 \|y\|^2,$$

para $y \in T = \{w : w^T \nabla c_i(x^*) = 0 : i \in E \cup J\}$, onde $J = \{i \in I : \lambda_i^* < 0\}$. Defina a matriz $\nabla c_A(x)$ cujas linhas são os vetores de V . Numa vizinhança de x^* , $\nabla c_A(x)$ tem posto linha completo. Assim, podemos definir

$$\lambda_A(x) = -[\nabla c_A(x) \nabla c_A(x)^T]^{-1} \nabla c_A(x) \nabla f(x),$$

$$g_A(x) = \nabla f(x) + \nabla c_A(x)^T \lambda_A(x),$$

$$H_A(x, \lambda) = \nabla^2 f(x) + \sum_{i \in \mathcal{A}^*} \nabla^2 c_i(x) \lambda_i$$

$$P(x) = I - \nabla c_A(x)^T [\nabla c_A(x) \nabla c_A(x)^T]^{-1} \nabla c_A(x),$$

Convergência

Convergência Local

- A1 $\|\lambda^k - \lambda_{LS}(z_c^k, \mu_c^k)\| = \mathcal{O}(\|g_p(z_c^k, \mu_c^k)\|),$
 $\lambda_{LS}(z_c^{k+1}, \mu_c^{k+1})^T (s_c^{k+1} - s^k) = \mathcal{O}(\|g_p(z_c^k, \mu_c^k)\| \rho^k)$
- A2 B^k é assintoticamente uniformemente definida positiva em $\mathcal{N}(A(x_c^k))$, isto é, em alguma vizinhança de z^* , podemos redefinir θ_1 e definir $\theta_2 > 0$ de modo que

$$\theta_1 \|y\|^2 \leq y^T B^k y \leq \theta_2 \|y\|^2,$$

para $y \in \mathcal{N}(A(z_c^k))$.

- A3 Para k suficientemente grande,

$$\begin{aligned} \|g_A(x_c^k)\| &= \Theta(\|g_p^k\|), \\ \|c_A(x_c^k)\| &= \Theta(\|h(z_c^k)\|), \\ \|c_A(x^k)\| &= \Theta(\|h(z^k)\|), \\ \|x_c^{k+1} - x^k\| &= \mathcal{O}(\|c_A(x^k)\|). \end{aligned}$$

Convergência

Convergência Local

A4 Defina a matriz Z_A^k cujas colunas formam uma base ortonormal para o núcleo de $\nabla c_A(x_c^k)$. Defina

$$\begin{aligned}\delta_x^k &= -Z_A^k [(Z_A^k)^T B_x Z_A^k]^{-1} (Z_A^k)^T g_A(x_c^k), \\ \delta_s^k &= (S_c^k)^{-1} \nabla c_I(x_c^k) \delta_x^k,\end{aligned}$$

e

$$\delta_A^k = \begin{bmatrix} \delta_x^k \\ \delta_s^k \end{bmatrix}.$$

Note que se $s_{c_i}^k \rightarrow 0$, isto é, $i \in \mathcal{A}^*$, então a componente correspondente de δ_s^k é zero, portanto δ_s^k é limitado. Além disso, definimos $s_{\min} > 0$ tal que se $i \notin \mathcal{A}^*$, então $s_{c_i}^k \geq s_{\min}$. Assumimos que δ_A^k é o primeiro passo tentado pelo algoritmo se $\|\delta_A^k\| \leq \Delta$ e $s_c^k + S_c^k \delta_s^k \geq \varepsilon_\mu s_c^k$. Além disso, supomos que

$$P(x_c^k)[B_x^k - H_A(x^*, \lambda^*)]\delta_x^k = o(\|\delta_x^k\|).$$



Convergência

Convergência Local

A5 Cada passo vertical $\delta_V^{k+1} = z_c^{k+1} - z^k$ é calculado tomando um ou mais passos da forma

$$\delta_V^+ = -J^T(JJ^T)^{-1}h(z_c),$$

onde J satisfaz

$$\|J - \nabla h(z_c)\| = \mathcal{O}(\|g_p^k\|).$$

Convergência

Convergência Local

Teorema *Com as hipóteses H1-H4 e A1-A5, x^k e x_c^k são superlinearmente convergentes em dois passos para x^* . Se uma restauração é calculada a cada x^k , então x^k converge superlinearmente para x^* .*



Convergência

Problemas Infactíveis

Se o problema for infactível, o passo vertical não conseguirá encontrar um ponto viável. No entanto, podemos mostrar que ele irá encontrar um ponto estacionário da inviabilidade, isto é, um ponto estacionário do problema

$$\min \|c_E(x)\|^2 + \|c_I^-(x)\|^2.$$

O método que utilizamos no passo vertical nos garante isso com as seguintes hipóteses:

- 11 A sequência gerada pelo algoritmo (vertical) é limitada.
- 12 Seja L um conjunto convexo, aberto e limitado que contém toda a sequência gerada pelo algoritmo, e todos os pontos da forma $x^k + P(d^k)$ (definido no passo vertical). Para todo $x, y \in L$, temos

$$\|\nabla h(x) - \nabla h(y)\| \leq 2\gamma_0 \|x - y\|.$$



Resultados Computacionais

- O algoritmo DCI para igualdades foi passado de FORTRAN para C++.
- Implementamos a biblioteca `base_matrices`, um “embrulho” para a biblioteca CHOLMOD, responsável pelo Cholesky.
- Implementamos a biblioteca `dcicpp`.
- Licença GPL, código disponível no Github.
- Utiliza as bibliotecas
 - CHOLMOD [3] (para Cholesky, e é a base das estruturas);
 - MUMPS [1, 2] (alternativa para resolver os sistemas)
 - METIS [6] (requisito de CHOLMOD e MUMPS para uma boa escolha de permutações)
 - GotoBLAS2 [7]
 - `base_matrices` (feita por mim, também disponível no Github)
 - CUTer [5] para testes.

Resultados Computacionais

- 124 problemas de Hock-Schittkowsky
- Máximo de Iterações: 200
- Máximo de Restaurações: 200000
- Máximo de Tempo: 1 minuto

Situação	%
Converge	89,52
Max. Iterações	3,22
Max. Restaurações	4,83
ρ_{\max} muito pequeno	2,42

Resultados Computacionais

- 124 problemas de Hock-Schittkowsky
- Máximo de Iterações: 200000
- Máximo de Restaurações: 200000
- Máximo de Tempo: 10 minutos

Situação	%
Converge	90,32
Max. Iterações	0,81
Max. Restaurações	5,64
ρ_{\max} muito pequeno	3,22

Próximos Passos

- Melhorar o passo horizontal (usando BFGS?)
- Comparar com o IPOPT.
- Trabalhar com Jacobianas singulares.
- Incluir pré-condicionador no gradiente conjugado.
- Tentar fazer funcionar em 64 bits.



Bibliografia



P. R. Amestoy, I. S. Duff, J. Koster, and J.-Y. L'Excellent.

A fully asynchronous multifrontal solver using distributed dynamic scheduling.
SIAM Journal of Matrix Analysis and Applications, 23(1):15–41, 2001.



P. R. Amestoy, I. S. Duff, J. Koster, J.-Y. L'Excellent, and S. Pralet.

Hybrid scheduling for the parallel solution of linear system.
Parallel Computing, 32(2):136–156, 2006.



Y. Chen, T. A. Davis, W. W. Hager, and S. Raamanickam.

Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate.
ACM Transactions on Mathematical Software, 35(3), 887.



J. B. Francisco, N. Krejić, and J. M. Martínez.

An interior-point method for solving box-constrained underdetermined nonlinear system.
Journal of Computational and Applied Mathematics, 177:67–88, 2005.



N. Gould, D. Orban, and Ph. L. Toint.

Cuter, a constrained and unconstrained testing environment, revisited.
Transactions of the American Mathematical Society on Mathematical Software, 29(4):373–394, 2003.



Karypis Lab.

Metis - serial graph partitioning and fill-reducing matrix ordering.
<http://www-users.cs.umn.edu/~karypis/metis>.



TACC: Texas Advanced Computing Center.

GotoBLAS2.
<http://www.tacc.utexas.edu/tacc-projects/gotoblas2/>.