Otimização Contínua: Aspectos teóricos e computacionais

Ademir Alves Ribeiro
Elizabeth Wegner Karas

Cap. 6 - Implementação Computacional

CENGAGE Learning

Banco de funções

Considere $f : \mathbb{R}^n \to \mathbb{R}$ uma função de classe $\mathcal{C}^2$.

- Entrada:
  - um ponto $x \in \mathbb{R}^n$
  - um parâmetro $\texttt{ordem} \in \{0, 1, 2\}$
- Saída:
  - valor de $f(x)$, se $\texttt{ordem} = 0$
  - vetor gradiente $\nabla f(x)$, se $\texttt{ordem} = 1$
  - Hessiana $\nabla^2 f(x)$, se $\texttt{ordem} = 2$.

Banco de funções

Considere $f : \mathbb{R}^n \to \mathbb{R}$ uma função de classe $\mathcal{C}^2$.

- Entrada:
  - um ponto $x \in \mathbb{R}^n$
  - um parâmetro `ordem` $\in \{0, 1, 2\}$
- Saída:
  - valor de $f(x)$, se `ordem`$= 0$
  - vetor gradiente $\nabla f(x)$, se `ordem`$= 1$
  - Hessiana $\nabla^2 f(x)$, se `ordem`$= 2$.

CENGAGE Learning

Banco de funções

Considere $f : \mathbb{R}^n \to \mathbb{R}$ uma função de classe $\mathcal{C}^2$.

- Entrada:
  - um ponto $x \in \mathbb{R}^n$
  - um parâmetro `ordem` $\in \{0, 1, 2\}$
- Saída:
  - valor de $f(x)$, se `ordem`$= 0$
  - vetor gradiente $\nabla f(x)$, se `ordem`$= 1$
  - Hessiana $\nabla^2 f(x)$, se `ordem`$= 2$.

Banco de funções

Dados de entrada: $x \in \mathbb{R}^n$, $\texttt{ordem} \in \{0, 1, 2\}$

SE $\texttt{ordem} = 0$

$y = f(x)$

SENÃO, SE $\texttt{ordem} = 1$

$y = \nabla f(x)$

SENÃO, SE $\texttt{ordem} = 2$

$y = \nabla^2 f(x)$

SENÃO

MENSAGEM: Erro na variável `ordem`!

Banco de funções - Exemplo

Implemente a rotina para $f(x) = x_1^2 + 4x_1x_2 + 6x_2^2$ e use-a para avaliar a função, o gradiente e a Hessiana no ponto $x = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.

Rotina

Dados de entrada: $x \in \mathbb{R}^2$, $\texttt{ordem} \in \{0, 1, 2\}$

SE $\texttt{ordem} = 0$

$$y = x_1^2 + 4x_1x_2 + 6x_2^2$$

SENÃO, SE $\texttt{ordem} = 1$

$$y = \begin{pmatrix} 2x_1 + 4x_2 \\ 4x_1 + 12x_2 \end{pmatrix}$$

SENÃO, SE $\texttt{ordem} = 2$

$$y = \begin{pmatrix} 2 & 4 \\ 4 & 12 \end{pmatrix}$$

SENÃO

MENSAGEM: Erro na variável $\texttt{ordem}$!

Saída

- $y = 33$, se $\texttt{ordem} = 0$.
- $y = \begin{pmatrix} 10 \\ 28 \end{pmatrix}$, se $\texttt{ordem} = 1$.
- $y = \begin{pmatrix} 2 & 4 \\ 4 & 12 \end{pmatrix}$, se $\texttt{ordem} = 2$.

Banco de funções - Exemplo

Implemente a rotina para $f(x) = x_1^2 + 4x_1x_2 + 6x_2^2$ e use-a para avaliar a função, o gradiente e a Hessiana no ponto $x = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.

Rotina

Dados de entrada: $x \in \mathbb{R}^2$, ordem $\in \{0, 1, 2\}$

SE ordem $= 0$

$$y = x_1^2 + 4x_1x_2 + 6x_2^2$$

SENÃO, SE ordem $= 1$

$$y = \begin{pmatrix} 2x_1 + 4x_2 \\ 4x_1 + 12x_2 \end{pmatrix}$$

SENÃO, SE ordem $= 2$

$$y = \begin{pmatrix} 2 & 4 \\ 4 & 12 \end{pmatrix}$$

SENÃO

MENSAGEM: Erro na variável ordem!

Saída

- $y = 33$, se ordem $= 0$.
- $y = \begin{pmatrix} 10 \\ 28 \end{pmatrix}$, se ordem $= 1$.
- $y = \begin{pmatrix} 2 & 4 \\ 4 & 12 \end{pmatrix}$, se ordem $= 2$.

CENGAGE Learning

Banco de funções - Exemplo

Implemente a rotina para $f(x) = x_1^2 + 4x_1x_2 + 6x_2^2$ e use-a para avaliar a função, o gradiente e a Hessiana no ponto $x = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.

Rotina

Dados de entrada: $x \in \mathbb{R}^2$, $\texttt{ordem} \in \{0,1,2\}$

SE $\texttt{ordem} = 0$

$$y = x_1^2 + 4x_1x_2 + 6x_2^2$$

SENÃO, SE $\texttt{ordem} = 1$

$$y = \begin{pmatrix} 2x_1 + 4x_2 \\ 4x_1 + 12x_2 \end{pmatrix}$$

SENÃO, SE $\texttt{ordem} = 2$

$$y = \begin{pmatrix} 2 & 4 \\ 4 & 12 \end{pmatrix}$$

SENÃO

MENSAGEM: Erro na variável `ordem`!

Saída

- $y = 33$, se $\texttt{ordem} = 0$.
- $y = \begin{pmatrix} 10 \\ 28 \end{pmatrix}$, se $\texttt{ordem} = 1$.
- $y = \begin{pmatrix} 2 & 4 \\ 4 & 12 \end{pmatrix}$, se $\texttt{ordem} = 2$.

CENGAGE Learning

Banco de funções - Exemplo

Implemente a rotina para $f(x) = x_1^2 + 4x_1x_2 + 6x_2^2$ e use-a para avaliar a função, o gradiente e a Hessiana no ponto $x = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.

Rotina

Dados de entrada: $x \in \mathbb{R}^2$, $\texttt{ordem} \in \{0, 1, 2\}$

SE $\texttt{ordem} = 0$

$$y = x_1^2 + 4x_1x_2 + 6x_2^2$$

SENÃO, SE $\texttt{ordem} = 1$

$$y = \begin{pmatrix} 2x_1 + 4x_2 \\ 4x_1 + 12x_2 \end{pmatrix}$$

SENÃO, SE $\texttt{ordem} = 2$

$$y = \begin{pmatrix} 2 & 4 \\ 4 & 12 \end{pmatrix}$$

SENÃO

MENSAGEM: Erro na variável `ordem`!

Saída

- $y = 33$, se $\texttt{ordem} = 0$.
- $y = \begin{pmatrix} 10 \\ 28 \end{pmatrix}$, se $\texttt{ordem} = 1$.
- $y = \begin{pmatrix} 2 & 4 \\ 4 & 12 \end{pmatrix}$, se $\texttt{ordem} = 2$.

Banco de funções - Exemplo

Implemente a rotina para $f(x) = x_1^2 + 4x_1x_2 + 6x_2^2$ e use-a para avaliar a função, o gradiente e a Hessiana no ponto $x = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.

Rotina

Dados de entrada: $x \in \mathbb{R}^2$, $\texttt{ordem} \in \{0, 1, 2\}$

SE $\texttt{ordem} = 0$

$$y = x_1^2 + 4x_1x_2 + 6x_2^2$$

SENÃO, SE $\texttt{ordem} = 1$

$$y = \begin{pmatrix} 2x_1 + 4x_2 \\ 4x_1 + 12x_2 \end{pmatrix}$$

SENÃO, SE $\texttt{ordem} = 2$

$$y = \begin{pmatrix} 2 & 4 \\ 4 & 12 \end{pmatrix}$$

SENÃO

MENSAGEM: Erro na variável `ordem`!

Saída

- $y = 33$, se $\texttt{ordem} = 0$.
- $y = \begin{pmatrix} 10 \\ 28 \end{pmatrix}$, se $\texttt{ordem} = 1$.
- $y = \begin{pmatrix} 2 & 4 \\ 4 & 12 \end{pmatrix}$, se $\texttt{ordem} = 2$.

CENGAGE Learning

Banco de funções - Função Quadrática

Implemente a rotina para a função dada por $f(x) = \dfrac{1}{2}x^T Ax$ considerando uma matriz simétrica arbitrária $A \in \mathbb{R}^{n\times n}$. Denomine esta rotina como `quadratica`, teste-a com os dados do exemplo anterior e compare os resultados. Teste-a com uma matriz em $\mathbb{R}^{4\times 4}$ e um vetor em $\mathbb{R}^4$, arbitrários.

Rotina quadratica

Variável global: $A \in \mathbb{R}^{n\times n}$ simétrica

Dados de entrada: $x \in \mathbb{R}^n$, $\texttt{ordem} \in \{0,1,2\}$

SE $\texttt{ordem} = 0$

$$y = \frac{1}{2}x^T Ax$$

SENÃO, SE $\texttt{ordem} = 1$

$$y = Ax$$

SENÃO, SE $\texttt{ordem} = 2$

$$y = A$$

SENÃO

MENSAGEM: Erro na variável `ordem`!

Banco de funções - Função Quadrática

Implemente a rotina para a função dada por $f(x) = \frac{1}{2}x^T Ax$ considerando uma matriz simétrica arbitrária $A \in \mathbb{R}^{n\times n}$. Denomine esta rotina como `quadratica`, teste-a com os dados do exemplo anterior e compare os resultados. Teste-a com uma matriz em $\mathbb{R}^{4\times 4}$ e um vetor em $\mathbb{R}^4$, arbitrários.

Rotina quadratica

Variável global: $A \in \mathbb{R}^{n\times n}$ simétrica

Dados de entrada: $x \in \mathbb{R}^n$, $\texttt{ordem} \in \{0,1,2\}$

SE $\texttt{ordem} = 0$

$$y = \frac{1}{2}x^T Ax$$

SENÃO, SE $\texttt{ordem} = 1$

$$y = Ax$$

SENÃO, SE $\texttt{ordem} = 2$

$$y = A$$

SENÃO

MENSAGEM: Erro na variável `ordem`!

Implemente uma rotina que, dados a dimensão n do espaço e dois reais $\lambda_1 < \lambda_n$, forneça uma matriz simétrica $A \in \mathbb{R}^{n\times n}$ cujos autovalores estejam uniformemente distribuídos entre λ_1 e λ_n. Use a rotina implementada para gerar uma matriz simétrica 4×4 com autovalores entre $\lambda_1 = 1$ e $\lambda_n = 100$.

Rotina

Dados de entrada: $n \in \mathbb{N}$, $\lambda_1 < \lambda_n$

$u = \mathrm{rand}(n,1)$ (vetor randômico com componentes entre 0 e 1)

$$v = \lambda_1 + \frac{\lambda_n - \lambda_1}{\max(u) - \min(u)}(u - \min(u)\, e)$$

Obtenha uma matriz $Q \in \mathbb{R}^{n\times n}$ ortogonal

$A = Q^T \mathrm{diag}(v) Q$

Um matriz ortogonal $Q \in \mathbb{R}^{n\times n}$ pode ser gerada pela decomposição QR de uma matriz arbitrária em $\mathbb{R}^{n\times n}$.

Implemente uma rotina que, dados a dimensão n do espaço e dois reais $\lambda_1 < \lambda_n$, forneça uma matriz simétrica $A \in \mathbb{R}^{n \times n}$ cujos autovalores estejam uniformemente distribuídos entre λ_1 e λ_n. Use a rotina implementada para gerar uma matriz simétrica 4×4 com autovalores entre $\lambda_1 = 1$ e $\lambda_n = 100$.

Rotina

Dados de entrada: $n \in \mathbb{N}$, $\lambda_1 < \lambda_n$

$u = \text{rand}(n, 1)$ (vetor randômico com componentes entre 0 e 1)

$$v = \lambda_1 + \frac{\lambda_n - \lambda_1}{\max(u) - \min(u)}(u - \min(u)\, e)$$

Obtenha uma matriz $Q \in \mathbb{R}^{n \times n}$ ortogonal

$A = Q^T \text{diag}(v) Q$

Um matriz ortogonal $Q \in \mathbb{R}^{n \times n}$ pode ser gerada pela decomposição QR de uma matriz arbitrária em $\mathbb{R}^{n \times n}$.

Implemente uma rotina que, dados a dimensão n do espaço e dois reais $\lambda_1 < \lambda_n$, forneça uma matriz simétrica $A \in \mathbb{R}^{n\times n}$ cujos autovalores estejam uniformemente distribuídos entre λ_1 e λ_n. Use a rotina implementada para gerar uma matriz simétrica 4×4 com autovalores entre $\lambda_1 = 1$ e $\lambda_n = 100$.

Rotina

Dados de entrada: $n \in \mathbb{N}$, $\lambda_1 < \lambda_n$

$u = \mathrm{rand}(n,1)$ (vetor randômico com componentes entre 0 e 1)

$$v = \lambda_1 + \frac{\lambda_n - \lambda_1}{\max(u) - \min(u)}(u - \min(u)\, e)$$

Obtenha uma matriz $Q \in \mathbb{R}^{n\times n}$ ortogonal

$A = Q^T \mathrm{diag}(v) Q$

Um matriz ortogonal $Q \in \mathbb{R}^{n\times n}$ pode ser gerada pela decomposição QR de uma matriz arbitrária em $\mathbb{R}^{n\times n}$.

Família de 35 funções dadas como somatório de quadrados, da forma

$$f(x) = \sum_{i=1}^{m} (f_i(x))^2,$$

onde $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$, são funções dadas.

O código em Matlab e em Fortran deste banco de funções está disponível em
http://www.mat.univie.ac.at/∼neum/glopt/test.html#test_unconstr

Referência

J. J. Moré, B. S. Garbow, and K. E. Hillstrom.
Testing unconstrained optimization software.
ACM Transactions on Mathematical Software, 7(1):17–41, 1981.

Família de 35 funções dadas como somatório de quadrados, da forma

$$f(x) = \sum_{i=1}^{m} (f_i(x))^2,$$

onde $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$, são funções dadas.

O código em Matlab e em Fortran deste banco de funções está disponível em http://www.mat.univie.ac.at/∼neum/glopt/test.html#test_unconstr

Referência

J. J. Moré, B. S. Garbow, and K. E. Hillstrom.
Testing unconstrained optimization software.
ACM Transactions on Mathematical Software, 7(1):17–41, 1981.

Banco de funções de Moré, Garbow e Hillstrom

Família de 35 funções dadas como somatório de quadrados, da forma

$$f(x) = \sum_{i=1}^{m} (f_i(x))^2 \, ,$$

onde $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$, são funções dadas.

O código em Matlab e em Fortran deste banco de funções está disponível em http://www.mat.univie.ac.at/∼neum/glopt/test.html#test_unconstr

Referência

J. J. Moré, B. S. Garbow, and K. E. Hillstrom.
Testing unconstrained optimization software.
ACM Transactions on Mathematical Software, 7(1):17–41, 1981.

CENGAGE Learning

Banco de funções de Moré, Garbow e Hillstrom

Cada função tem quatro dados de entrada:

- dimensão n do espaço;
- o número m de funções usadas para definir a função objetivo;
- o ponto $x \in \mathbb{R}^n$ onde se deseja calculá-la;
- um parâmetro `opt` $\in \{1, 2, 3\}$.

A versão implementada de cada função fornece:

- o vetor `fvec` $\in \mathbb{R}^m$ cuja i-ésima componente é o valor $f_i(x)$, se $\mathtt{opt} = 1$.
- a matriz Jacobiana de $(f_1, f_2, \ldots, f_m)$, isto é, uma matriz `J`, cuja i-ésima linha é $\nabla f_i(x)^T$, se $\mathtt{opt} = 2$.
- vetor `fvec` e a matriz `J`, se $\mathtt{opt} = 3$.

Note que nesta notação a função $f(x) = \sum_{i=1}^{m} (f_i(x))^2$ pode ser escrita como

$$f(x) = \mathtt{fvec}^T \mathtt{fvec}$$

e o gradiente de f pode ser calculado como

$$\nabla f(x) = 2 \sum_{i=1}^{m} f_i(x) \nabla f_i(x) = 2\mathtt{J}^T \mathtt{fvec}$$

Cada função tem quatro dados de entrada:

- dimensão n do espaço;
- o número m de funções usadas para definir a função objetivo;
- o ponto $x \in \mathbb{R}^n$ onde se deseja calculá-la;
- um parâmetro `opt` $\in \{1, 2, 3\}$.

A versão implementada de cada função fornece:

- o vetor `fvec` $\in \mathbb{R}^m$ cuja i-ésima componente é o valor $f_i(x)$, se `opt` $= 1$.
- a matriz Jacobiana de $(f_1, f_2, \ldots, f_m)$, isto é, uma matriz `J`, cuja i-ésima linha é $\nabla f_i(x)^T$, se `opt` $= 2$.
- vetor `fvec` e a matriz `J`, se `opt` $= 3$.

Note que nesta notação a função $f(x) = \sum_{i=1}^{m} (f_i(x))^2$ pode ser escrita como

$$f(x) = \texttt{fvec}^T \texttt{fvec}$$

e o gradiente de f pode ser calculado como

$$\nabla f(x) = 2\sum_{i=1}^{m} f_i(x)\nabla f_i(x) = 2J^T \texttt{fvec}$$

Banco de funções de Moré, Garbow e Hillstrom

Cada função tem quatro dados de entrada:

- dimensão n do espaço;
- o número m de funções usadas para definir a função objetivo;
- o ponto $x \in \mathbb{R}^n$ onde se deseja calculá-la;
- um parâmetro `opt` $\in \{1, 2, 3\}$.

A versão implementada de cada função fornece:

- o vetor `fvec` $\in \mathbb{R}^m$ cuja i-ésima componente é o valor $f_i(x)$, se `opt` $= 1$.
- a matriz Jacobiana de $(f_1, f_2, \ldots, f_m)$, isto é, uma matriz `J`, cuja i-ésima linha é $\nabla f_i(x)^T$, se `opt` $= 2$.
- vetor `fvec` e a matriz `J`, se `opt` $= 3$.

Note que nesta notação a função $f(x) = \sum_{i=1}^{m} (f_i(x))^2$ pode ser escrita como

$$f(x) = \texttt{fvec}^T\texttt{fvec}$$

e o gradiente de f pode ser calculado como

$$\nabla f(x) = 2\sum_{i=1}^{m} f_i(x)\nabla f_i(x) = 2\texttt{J}^T\texttt{fvec}$$

Banco de funções de Moré, Garbow e Hillstrom

Cada função tem quatro dados de entrada:

- dimensão n do espaço;
- o número m de funções usadas para definir a função objetivo;
- o ponto $x \in \mathbb{R}^n$ onde se deseja calculá-la;
- um parâmetro `opt` $\in \{1,2,3\}$.

A versão implementada de cada função fornece:

- o vetor `fvec` $\in \mathbb{R}^m$ cuja i-ésima componente é o valor $f_i(x)$, se `opt` $= 1$.
- a matriz Jacobiana de $(f_1, f_2, \dots, f_m)$, isto é, uma matriz `J`, cuja i-ésima linha é $\nabla f_i(x)^T$, se `opt` $= 2$.
- vetor `fvec` e a matriz `J`, se `opt` $= 3$.

Note que nesta notação a função $f(x) = \sum_{i=1}^{m} (f_i(x))^2$ pode ser escrita como

$$f(x) = \texttt{fvec}^T\texttt{fvec}$$

e o gradiente de f pode ser calculado como

$$\nabla f(x) = 2\sum_{i=1}^{m} f_i(x)\nabla f_i(x) = 2\texttt{J}^T\texttt{fvec}$$

Cada função tem quatro dados de entrada:

- dimensão n do espaço;
- o número m de funções usadas para definir a função objetivo;
- o ponto $x \in \mathbb{R}^n$ onde se deseja calculá-la;
- um parâmetro `opt` $\in \{1,2,3\}$.

A versão implementada de cada função fornece:

- o vetor `fvec` $\in \mathbb{R}^m$ cuja i-ésima componente é o valor $f_i(x)$, se `opt` $= 1$.
- a matriz Jacobiana de $(f_1, f_2, \ldots, f_m)$, isto é, uma matriz `J`, cuja i-ésima linha é $\nabla f_i(x)^T$, se `opt` $= 2$.
- vetor `fvec` e a matriz `J`, se `opt` $= 3$.

Note que nesta notação a função $f(x) = \sum_{i=1}^{m} (f_i(x))^2$ pode ser escrita como

$$f(x) = \texttt{fvec}^T \texttt{fvec}$$

e o gradiente de f pode ser calculado como

$$\nabla f(x) = 2 \sum_{i=1}^{m} f_i(x) \nabla f_i(x) = 2 \texttt{J}^T \texttt{fvec}$$

CENGAGE Learning

Banco de funções de Moré, Garbow e Hillstrom

Cada função tem quatro dados de entrada:

- dimensão n do espaço;
- o número m de funções usadas para definir a função objetivo;
- o ponto $x \in \mathbb{R}^n$ onde se deseja calculá-la;
- um parâmetro `opt` $\in \{1,2,3\}$.

A versão implementada de cada função fornece:

- o vetor `fvec` $\in \mathbb{R}^m$ cuja i-ésima componente é o valor $f_i(x)$, se `opt` $= 1$.
- a matriz Jacobiana de $(f_1, f_2, \ldots, f_m)$, isto é, uma matriz `J`, cuja i-ésima linha é $\nabla f_i(x)^T$, se `opt` $= 2$.
- vetor `fvec` e a matriz `J`, se `opt` $= 3$.

Note que nesta notação a função $f(x) = \sum_{i=1}^{m} (f_i(x))^2$ pode ser escrita como

$$f(x) = \texttt{fvec}^T \texttt{fvec}$$

e o gradiente de f pode ser calculado como

$$\nabla f(x) = 2 \sum_{i=1}^{m} f_i(x) \nabla f_i(x) = 2\texttt{J}^T \texttt{fvec}.$$

CENGAGE Learning

Banco de funções de Moré, Garbow e Hillstrom

Para as seguintes funções, calcule o vetor `fvec`, a matriz `J`, o valor de f e seu gradiente no ponto x^0 fornecido

(a) Rosenbrock, numerada como (1) no artigo.

(b) Jennrich e Sampson, numerada como (6), com $m = 10$.

(c) Extended Rosenbrock, numerada como (21), com $n = 4$ e $m = n$.

Indicamos abaixo o ponto x^0, o valor de f e do seu gradiente neste ponto.

$$(a) \quad x^0 = \begin{pmatrix} -1.2 \\ 1 \end{pmatrix}, \qquad f(x^0) = 24.2 \quad \text{e} \quad \nabla f(x^0) = \begin{pmatrix} -215.6 \\ -88.0 \end{pmatrix};$$

$$(b) \quad x^0 = \begin{pmatrix} 0.3 \\ 0.4 \end{pmatrix}, \qquad f(x^0) = 4171.3 \quad \text{e} \quad \nabla f(x^0) = \begin{pmatrix} 33796.6 \\ 87402.1 \end{pmatrix};$$

$$(c) \quad x^0 = \begin{pmatrix} -1.2 \\ 1 \\ -1.2 \\ 1 \end{pmatrix}, \qquad f(x^0) = 48.4 \quad \text{e} \quad \nabla f(x^0) = \begin{pmatrix} -215.6 \\ -88.0 \\ -215.6 \\ -88.0 \end{pmatrix}.$$

Banco de funções de Moré, Garbow e Hillstrom

Para as seguintes funções, calcule o vetor `fvec`, a matriz `J`, o valor de f e seu gradiente no ponto x^0 fornecido

(a) Rosenbrock, numerada como (1) no artigo.

(b) Jennrich e Sampson, numerada como (6), com $m = 10$.

(c) Extended Rosenbrock, numerada como (21), com $n = 4$ e $m = n$.

Indicamos abaixo o ponto x^0, o valor de f e do seu gradiente neste ponto.

$$(a) \quad x^0 = \begin{pmatrix} -1.2 \\ 1 \end{pmatrix}, \quad f(x^0) = 24.2 \quad \text{e} \quad \nabla f(x^0) = \begin{pmatrix} -215.6 \\ -88.0 \end{pmatrix};$$

$$(b) \quad x^0 = \begin{pmatrix} 0.3 \\ 0.4 \end{pmatrix}, \quad f(x^0) = 4171.3 \quad \text{e} \quad \nabla f(x^0) = \begin{pmatrix} 33796.6 \\ 87402.1 \end{pmatrix};$$

$$(c) \quad x^0 = \begin{pmatrix} -1.2 \\ 1 \\ -1.2 \\ 1 \end{pmatrix}, \quad f(x^0) = 48.4 \quad \text{e} \quad \nabla f(x^0) = \begin{pmatrix} -215.6 \\ -88.0 \\ -215.6 \\ -88.0 \end{pmatrix}.$$

Banco de funções de Moré, Garbow e Hillstrom

Para as seguintes funções, calcule o vetor `fvec`, a matriz `J`, o valor de f e seu gradiente no ponto x^0 fornecido

(a) Rosenbrock, numerada como (1) no artigo.

(b) Jennrich e Sampson, numerada como (6), com $m = 10$.

(c) Extended Rosenbrock, numerada como (21), com $n = 4$ e $m = n$.

Indicamos abaixo o ponto x^0, o valor de f e do seu gradiente neste ponto.

$$(a)\quad x^0 = \begin{pmatrix} -1.2 \\ 1 \end{pmatrix}, \quad f(x^0) = 24.2 \quad \text{e} \quad \nabla f(x^0) = \begin{pmatrix} -215.6 \\ -88.0 \end{pmatrix};$$

$$(b)\quad x^0 = \begin{pmatrix} 0.3 \\ 0.4 \end{pmatrix}, \quad f(x^0) = 4171.3 \quad \text{e} \quad \nabla f(x^0) = \begin{pmatrix} 33796.6 \\ 87402.1 \end{pmatrix};$$

$$(c)\quad x^0 = \begin{pmatrix} -1.2 \\ 1 \\ -1.2 \\ 1 \end{pmatrix}, \quad f(x^0) = 48.4 \quad \text{e} \quad \nabla f(x^0) = \begin{pmatrix} -215.6 \\ -88.0 \\ -215.6 \\ -88.0 \end{pmatrix}.$$

Banco de funções de Moré, Garbow e Hillstrom

Para as seguintes funções, calcule o vetor `fvec`, a matriz `J`, o valor de f e seu gradiente no ponto x^0 fornecido

(a) Rosenbrock, numerada como (1) no artigo.

(b) Jennrich e Sampson, numerada como (6), com $m = 10$.

(c) Extended Rosenbrock, numerada como (21), com $n = 4$ e $m = n$.

Indicamos abaixo o ponto x^0, o valor de f e do seu gradiente neste ponto.

(a) $x^0 = \begin{pmatrix} -1.2 \\ 1 \end{pmatrix}, \quad f(x^0) = 24.2$ e $\nabla f(x^0) = \begin{pmatrix} -215.6 \\ -88.0 \end{pmatrix};$

(b) $x^0 = \begin{pmatrix} 0.3 \\ 0.4 \end{pmatrix}, \quad f(x^0) = 4171.3$ e $\nabla f(x^0) = \begin{pmatrix} 33796.6 \\ 87402.1 \end{pmatrix};$

(c) $x^0 = \begin{pmatrix} -1.2 \\ 1 \\ -1.2 \\ 1 \end{pmatrix}, \quad f(x^0) = 48.4$ e $\nabla f(x^0) = \begin{pmatrix} -215.6 \\ -88.0 \\ -215.6 \\ -88.0 \end{pmatrix}.$

Banco de funções de Moré, Garbow e Hillstrom

Para as seguintes funções, calcule o vetor `fvec`, a matriz `J`, o valor de f e seu gradiente no ponto x^0 fornecido

(a) Rosenbrock, numerada como (1) no artigo.

(b) Jennrich e Sampson, numerada como (6), com $m = 10$.

(c) Extended Rosenbrock, numerada como (21), com $n = 4$ e $m = n$.

Indicamos abaixo o ponto x^0, o valor de f e do seu gradiente neste ponto.

(a) $x^0 = \begin{pmatrix} -1.2 \\ 1 \end{pmatrix}$, $\quad f(x^0) = 24.2$ e $\nabla f(x^0) = \begin{pmatrix} -215.6 \\ -88.0 \end{pmatrix}$;

(b) $x^0 = \begin{pmatrix} 0.3 \\ 0.4 \end{pmatrix}$, $\quad f(x^0) = 4171.3$ e $\nabla f(x^0) = \begin{pmatrix} 33796.6 \\ 87402.1 \end{pmatrix}$;

(c) $x^0 = \begin{pmatrix} -1.2 \\ 1 \\ -1.2 \\ 1 \end{pmatrix}$, $\quad f(x^0) = 48.4$ e $\nabla f(x^0) = \begin{pmatrix} -215.6 \\ -88.0 \\ -215.6 \\ -88.0 \end{pmatrix}$.

Para as seguintes funções, calcule o vetor `fvec`, a matriz `J`, o valor de f e seu gradiente no ponto x^0 fornecido

(a) Rosenbrock, numerada como (1) no artigo.

(b) Jennrich e Sampson, numerada como (6), com $m = 10$.

(c) Extended Rosenbrock, numerada como (21), com $n = 4$ e $m = n$.

Indicamos abaixo o ponto x^0, o valor de f e do seu gradiente neste ponto.

(a) $x^0 = \begin{pmatrix} -1.2 \\ 1 \end{pmatrix}, \quad f(x^0) = 24.2$ e $\nabla f(x^0) = \begin{pmatrix} -215.6 \\ -88.0 \end{pmatrix};$

(b) $x^0 = \begin{pmatrix} 0.3 \\ 0.4 \end{pmatrix}, \quad f(x^0) = 4171.3$ e $\nabla f(x^0) = \begin{pmatrix} 33796.6 \\ 87402.1 \end{pmatrix};$

(c) $x^0 = \begin{pmatrix} -1.2 \\ 1 \\ -1.2 \\ 1 \end{pmatrix}, \quad f(x^0) = 48.4$ e $\nabla f(x^0) = \begin{pmatrix} -215.6 \\ -88.0 \\ -215.6 \\ -88.0 \end{pmatrix}.$

Para as seguintes funções, calcule o vetor `fvec`, a matriz `J`, o valor de f e seu gradiente no ponto x^0 fornecido

(a) Rosenbrock, numerada como (1) no artigo.

(b) Jennrich e Sampson, numerada como (6), com $m = 10$.

(c) Extended Rosenbrock, numerada como (21), com $n = 4$ e $m = n$.

Indicamos abaixo o ponto x^0, o valor de f e do seu gradiente neste ponto.

$$(a)\quad x^0 = \begin{pmatrix} -1.2 \\ 1 \end{pmatrix}, \quad f(x^0) = 24.2 \quad \text{e} \quad \nabla f(x^0) = \begin{pmatrix} -215.6 \\ -88.0 \end{pmatrix};$$

$$(b)\quad x^0 = \begin{pmatrix} 0.3 \\ 0.4 \end{pmatrix}, \quad f(x^0) = 4171.3 \quad \text{e} \quad \nabla f(x^0) = \begin{pmatrix} 33796.6 \\ 87402.1 \end{pmatrix};$$

$$(c)\quad x^0 = \begin{pmatrix} -1.2 \\ 1 \\ -1.2 \\ 1 \end{pmatrix}, \quad f(x^0) = 48.4 \quad \text{e} \quad \nabla f(x^0) = \begin{pmatrix} -215.6 \\ -88.0 \\ -215.6 \\ -88.0 \end{pmatrix}.$$

Interface - banco de funções de Moré, Garbow e Hillstrom

Defina variáveis globais `FUNC` e `mm` que devem receber o nome da função que se quer avaliar e o valor de m correspondente. Dados $x \in \mathbb{R}^n$ e `ordem` $\in \{0, 1\}$, a rotina deve fornecer o valor da função `FUNC` ou do seu gradiente no ponto x, dependendo se `ordem` é 0 ou 1.

Rotina de interface

Variáveis globais: `FUNC` e `mm`
Dados de entrada: x, `ordem` $\in \{0, 1\}$
Defina n como a dimensão de x
SE `ordem` $= 0$
  Calcule `fvec` avaliando `FUNC` em x com `opt` $= 1$
  $y = \texttt{fvec}^T \texttt{fvec}$
SENÃO, SE `ordem` $= 1$
  Calcule `fvec` e J avaliando `FUNC` em x com `opt` $= 3$
  $y = 2\mathrm{J}^T \texttt{fvec}$
SENÃO
  MENSAGEM: Erro na variável `ordem`!

Interface - Banco de funções de Moré, Garbow e Hillstrom

Teste a rotina anterior para calcular o valor da função e do seu gradiente no ponto x^0 fornecido, para as funções abaixo e compare com os resultados obtidos anteriormente.

(a) Rosenbrock, numerada como (1) no artigo.

(b) Jennrich e Sampson, numerada como (6), com $m = 10$.

(c) Extended Rosenbrock, numerada como (21), com $n = 4$ e $m = n$.

Solução

Defina variáveis globais: $\mathtt{FUNC}$ e $\mathtt{mm}$

(a) Rosenbrock: $\mathtt{FUNC} = \mathtt{rosen}$, $\mathtt{mm} = 2$,
$f(x^0) = \mathtt{mgh}(x^0, 0)$, $\nabla f(x^0) = \mathtt{mgh}(x^0, 1)$;

(b) Jennrich e Sampson: $\mathtt{FUNC} = \mathtt{jensam}$, $\mathtt{mm} = 10$,
$f(x^0) = \mathtt{mgh}(x^0, 0)$, $\nabla f(x^0) = \mathtt{mgh}(x^0, 1)$;

(c) Extended Rosenbrock: $\mathtt{FUNC} = \mathtt{rosex}$, $\mathtt{mm} = 4$,
$f(x^0) = \mathtt{mgh}(x^0, 0)$, $\nabla f(x^0) = \mathtt{mgh}(x^0, 1)$.

Teste a rotina anterior para calcular o valor da função e do seu gradiente no ponto x^0 fornecido, para as funções abaixo e compare com os resultados obtidos anteriormente.

(a) Rosenbrock, numerada como (1) no artigo.

(b) Jennrich e Sampson, numerada como (6), com $m = 10$.

(c) Extended Rosenbrock, numerada como (21), com $n = 4$ e $m = n$.

Solução

Defina variáveis globais: FUNC e mm

(a) Rosenbrock: FUNC = rosen, mm = 2,
$f(x^0) = \mathtt{mgh}(x^0, 0)$, $\nabla f(x^0) = \mathtt{mgh}(x^0, 1)$;

(b) Jennrich e Sampson: FUNC = jensam, mm = 10,
$f(x^0) = \mathtt{mgh}(x^0, 0)$, $\nabla f(x^0) = \mathtt{mgh}(x^0, 1)$;

(c) Extended Rosenbrock: FUNC = rosex, mm = 4,
$f(x^0) = \mathtt{mgh}(x^0, 0)$, $\nabla f(x^0) = \mathtt{mgh}(x^0, 1)$.

Interface - Banco de funções de Moré, Garbow e Hillstrom

Teste a rotina anterior para calcular o valor da função e do seu gradiente no ponto x^0 fornecido, para as funções abaixo e compare com os resultados obtidos anteriormente.

(a) Rosenbrock, numerada como (1) no artigo.

(b) Jennrich e Sampson, numerada como (6), com $m = 10$.

(c) Extended Rosenbrock, numerada como (21), com $n = 4$ e $m = n$.

Solução

Defina variáveis globais: FUNC e mm

(a) Rosenbrock: $\texttt{FUNC} = \texttt{rosen}$, $\texttt{mm} = 2$,
$f(x^0) = \texttt{mgh}(x^0, 0)$, $\nabla f(x^0) = \texttt{mgh}(x^0, 1)$;

(b) Jennrich e Sampson: $\texttt{FUNC} = \texttt{jensam}$, $\texttt{mm} = 10$,
$f(x^0) = \texttt{mgh}(x^0, 0)$, $\nabla f(x^0) = \texttt{mgh}(x^0, 1)$;

(c) Extended Rosenbrock: $\texttt{FUNC} = \texttt{rosex}$, $\texttt{mm} = 4$,
$f(x^0) = \texttt{mgh}(x^0, 0)$, $\nabla f(x^0) = \texttt{mgh}(x^0, 1)$.

Interface - Banco de funções de Moré, Garbow e Hillstrom

Teste a rotina anterior para calcular o valor da função e do seu gradiente no ponto x^0 fornecido, para as funções abaixo e compare com os resultados obtidos anteriormente.

(a) Rosenbrock, numerada como (1) no artigo.

(b) Jennrich e Sampson, numerada como (6), com $m = 10$.

(c) Extended Rosenbrock, numerada como (21), com $n = 4$ e $m = n$.

Solução

Defina variáveis globais: FUNC e mm

(a) Rosenbrock: $\texttt{FUNC} = \texttt{rosen}$, $\texttt{mm} = 2$,
$f(x^0) = \texttt{mgh}(x^0, 0)$, $\nabla f(x^0) = \texttt{mgh}(x^0, 1)$;

(b) Jennrich e Sampson: $\texttt{FUNC} = \texttt{jensam}$, $\texttt{mm} = 10$,
$f(x^0) = \texttt{mgh}(x^0, 0)$, $\nabla f(x^0) = \texttt{mgh}(x^0, 1)$;

(c) Extended Rosenbrock: $\texttt{FUNC} = \texttt{rosex}$, $\texttt{mm} = 4$,
$f(x^0) = \texttt{mgh}(x^0, 0)$, $\nabla f(x^0) = \texttt{mgh}(x^0, 1)$.

Teste a rotina anterior para calcular o valor da função e do seu gradiente no ponto x^0 fornecido, para as funções abaixo e compare com os resultados obtidos anteriormente.

(a) Rosenbrock, numerada como (1) no artigo.

(b) Jennrich e Sampson, numerada como (6), com $m = 10$.

(c) Extended Rosenbrock, numerada como (21), com $n = 4$ e $m = n$.

Solução

Defina variáveis globais: `FUNC` e `mm`

(a) Rosenbrock: $\texttt{FUNC} = \texttt{rosen}$, $\texttt{mm} = 2$,
$f(x^0) = \texttt{mgh}(x^0, 0)$, $\nabla f(x^0) = \texttt{mgh}(x^0, 1)$;

(b) Jennrich e Sampson: $\texttt{FUNC} = \texttt{jensam}$, $\texttt{mm} = 10$,
$f(x^0) = \texttt{mgh}(x^0, 0)$, $\nabla f(x^0) = \texttt{mgh}(x^0, 1)$;

(c) Extended Rosenbrock: $\texttt{FUNC} = \texttt{rosex}$, $\texttt{mm} = 4$,
$f(x^0) = \texttt{mgh}(x^0, 0)$, $\nabla f(x^0) = \texttt{mgh}(x^0, 1)$.

Teste a rotina anterior para calcular o valor da função e do seu gradiente no ponto x^0 fornecido, para as funções abaixo e compare com os resultados obtidos anteriormente.

(a) Rosenbrock, numerada como (1) no artigo.

(b) Jennrich e Sampson, numerada como (6), com $m = 10$.

(c) Extended Rosenbrock, numerada como (21), com $n = 4$ e $m = n$.

Solução

Defina variáveis globais: `FUNC` e `mm`

(a) Rosenbrock: $\mathtt{FUNC} = \mathtt{rosen}$, $\mathtt{mm} = 2$,
$f(x^0) = \mathtt{mgh}(x^0, 0)$, $\nabla f(x^0) = \mathtt{mgh}(x^0, 1)$;

(b) Jennrich e Sampson: $\mathtt{FUNC} = \mathtt{jensam}$, $\mathtt{mm} = 10$,
$f(x^0) = \mathtt{mgh}(x^0, 0)$, $\nabla f(x^0) = \mathtt{mgh}(x^0, 1)$;

(c) Extended Rosenbrock: $\mathtt{FUNC} = \mathtt{rosex}$, $\mathtt{mm} = 4$,
$f(x^0) = \mathtt{mgh}(x^0, 0)$, $\nabla f(x^0) = \mathtt{mgh}(x^0, 1)$.

Teste a rotina anterior para calcular o valor da função e do seu gradiente no ponto x^0 fornecido, para as funções abaixo e compare com os resultados obtidos anteriormente.

(a) Rosenbrock, numerada como (1) no artigo.

(b) Jennrich e Sampson, numerada como (6), com $m = 10$.

(c) Extended Rosenbrock, numerada como (21), com $n = 4$ e $m = n$.

Solução

Defina variáveis globais: `FUNC` e `mm`

(a) Rosenbrock: $\texttt{FUNC} = \texttt{rosen}$, $\texttt{mm} = 2$,
$f(x^0) = \texttt{mgh}(x^0, 0)$, $\nabla f(x^0) = \texttt{mgh}(x^0, 1)$;

(b) Jennrich e Sampson: $\texttt{FUNC} = \texttt{jensam}$, $\texttt{mm} = 10$,
$f(x^0) = \texttt{mgh}(x^0, 0)$, $\nabla f(x^0) = \texttt{mgh}(x^0, 1)$;

(c) Extended Rosenbrock: $\texttt{FUNC} = \texttt{rosex}$, $\texttt{mm} = 4$,
$f(x^0) = \texttt{mgh}(x^0, 0)$, $\nabla f(x^0) = \texttt{mgh}(x^0, 1)$.

CENGAGE Learning

Teste a rotina anterior para calcular o valor da função e do seu gradiente no ponto x^0 fornecido, para as funções abaixo e compare com os resultados obtidos anteriormente.

(a) Rosenbrock, numerada como (1) no artigo.

(b) Jennrich e Sampson, numerada como (6), com $m = 10$.

(c) Extended Rosenbrock, numerada como (21), com $n = 4$ e $m = n$.

Solução

Defina variáveis globais: `FUNC` e `mm`

(a) Rosenbrock: $\texttt{FUNC} = \texttt{rosen}$, $\texttt{mm} = 2$,
$f(x^0) = \texttt{mgh}(x^0, 0)$, $\nabla f(x^0) = \texttt{mgh}(x^0, 1)$;

(b) Jennrich e Sampson: $\texttt{FUNC} = \texttt{jensam}$, $\texttt{mm} = 10$,
$f(x^0) = \texttt{mgh}(x^0, 0)$, $\nabla f(x^0) = \texttt{mgh}(x^0, 1)$;

(c) Extended Rosenbrock: $\texttt{FUNC} = \texttt{rosex}$, $\texttt{mm} = 4$,
$f(x^0) = \texttt{mgh}(x^0, 0)$, $\nabla f(x^0) = \texttt{mgh}(x^0, 1)$.

Método da Seção Áurea

Dados $x, d \in \mathbb{R}^n$, o objetivo é minimizar a função $\varphi : [0, \infty) \to \mathbb{R}$ definida por

$$\varphi(t) = f(x + td).$$

Parâmetros

- $\varepsilon = 10^{-5}$
- $\rho = 1$
- limitante superior $b_{\max} = 10^8$, para o valor de b.

Rotina do Método da Seção Áurea

Dados de entrada: variável fun com o nome da função, $x \in \mathbb{R}^n$, $d \in \mathbb{R}^n$

Fixados: $\varepsilon = 10^{-5}$, $\rho = 1$, $b_{\max} = 10^8$, $\theta_1 = (3-\sqrt{5})/2$, $\theta_2 = 1-\theta_1$

Fase 1: Obtenção do intervalo $[a,b]$

$a = 0$, $s = \rho$, $b = 2\rho$, $\mathtt{phib} = \mathtt{feval}(\mathtt{fun}, x+bd, 0)$, $\mathtt{phis} = \mathtt{feval}(\mathtt{fun}, x+sd, 0)$

REPITA enquanto $\mathtt{phib} < \mathtt{phis}$ e $2b < b_{\max}$

$a = s$, $s = b$, $b = 2b$, $\mathtt{phis} = \mathtt{phib}$, $\mathtt{phib} = \mathtt{feval}(\mathtt{fun}, x+bd, 0)$

Fase 2: Obtenção de $t \in [a,b]$

$u = a+\theta_1(b-a)$, $v = a+\theta_2(b-a)$,

$\mathtt{phiu} = \mathtt{feval}(\mathtt{fun}, x+ud, 0)$, $\mathtt{phiv} = \mathtt{feval}(\mathtt{fun}, x+vd, 0)$

REPITA enquanto $(b-a) > \varepsilon$

SE $\mathtt{phiu} < \mathtt{phiv}$

$b = v$, $v = u$, $u = a+\theta_1(b-a)$,

$\mathtt{phiv} = \mathtt{phiu}$, $\mathtt{phiu} = \mathtt{feval}(\mathtt{fun}, x+ud, 0)$

SENÃO

$a = u$, $u = v$, $v = a+\theta_2(b-a)$,

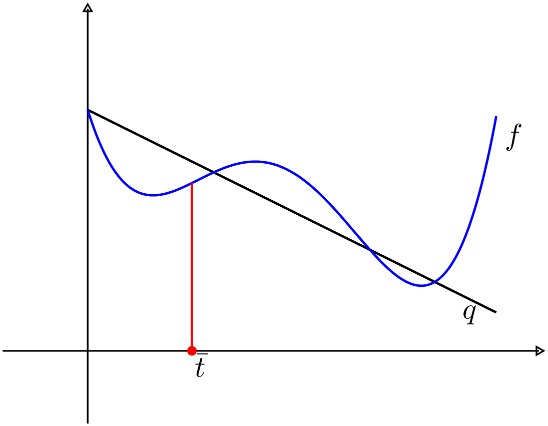
$\mathtt{phiu} = \mathtt{phiv}$, $\mathtt{phiv} = \mathtt{feval}(\mathtt{fun}, x+vd, 0)$

$t = (u+v)/2$

Busca de Armijo

Considere $\eta \in (0,1)$. Dados $x, d \in \mathbb{R}^n$, o objetivo da busca de Armijo é encontrar $\bar{t} > 0$ tal que

$$f(x+\bar{t}d) \leq f(x) + \eta \bar{t} \nabla f(x)^T d$$



Dados de entrada:

variável `fun` com o nome da função,

$x \in \mathbb{R}^n$, $d \in \mathbb{R}^n$

Parâmetros: $\gamma = 0.7$, $\eta = 0.45$

$t = 1$

$f = \texttt{feval}(\texttt{fun}, x, 0)$,

$g = \texttt{feval}(\texttt{fun}, x, 1)$

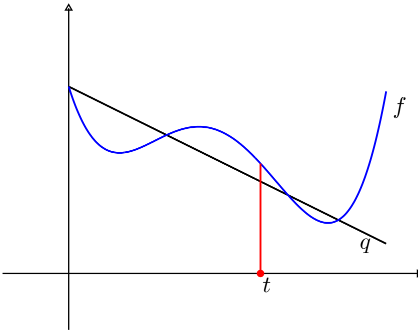
$\texttt{gd} = g^T d$

$\texttt{ft} = \texttt{feval}(\texttt{fun}, x + td, 0)$

REPITA enquanto $\texttt{ft} > f + \eta t \, \texttt{gd}$

$t = \gamma t$

$\texttt{ft} = \texttt{feval}(\texttt{fun}, x + td, 0)$



Dados de entrada:

variável `fun` com o nome da função,

$x \in \mathbb{R}^n$, $d \in \mathbb{R}^n$

Parâmetros: $\gamma = 0.7$, $\eta = 0.45$

$t = 1$

$f = \texttt{feval}(\texttt{fun}, x, 0)$,

$g = \texttt{feval}(\texttt{fun}, x, 1)$

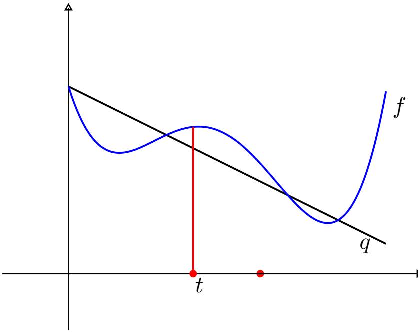
$\texttt{gd} = g^T d$

$\texttt{ft} = \texttt{feval}(\texttt{fun}, x + td, 0)$

REPITA enquanto $\texttt{ft} > f + \eta t\, \texttt{gd}$

$t = \gamma t$

$\texttt{ft} = \texttt{feval}(\texttt{fun}, x + td, 0)$



Dados de entrada:

variável `fun` com o nome da função,

$x \in \mathbb{R}^n$, $d \in \mathbb{R}^n$

Parâmetros: $\gamma = 0.7$, $\eta = 0.45$

$t = 1$

$f = \mathtt{feval}(\mathtt{fun}, x, 0)$,

$g = \mathtt{feval}(\mathtt{fun}, x, 1)$

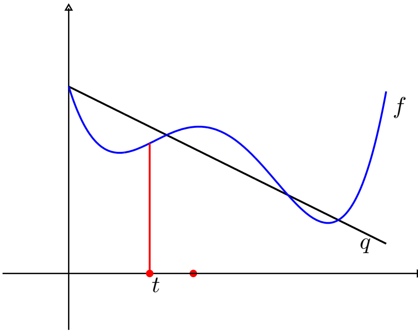
$\mathtt{gd} = g^T d$

$\mathtt{ft} = \mathtt{feval}(\mathtt{fun}, x + td, 0)$

REPITA enquanto $\mathtt{ft} > f + \eta t\, \mathtt{gd}$

$t = \gamma t$

$\mathtt{ft} = \mathtt{feval}(\mathtt{fun}, x + td, 0)$



Rotina da Busca de Armijo

Dados de entrada:

variável `fun` com o nome da função,

$x \in \mathbb{R}^n$, $d \in \mathbb{R}^n$

Parâmetros: $\gamma = 0.7$, $\eta = 0.45$

$t = 1$

$f = \mathtt{feval}(\mathtt{fun}, x, 0)$,

$g = \mathtt{feval}(\mathtt{fun}, x, 1)$

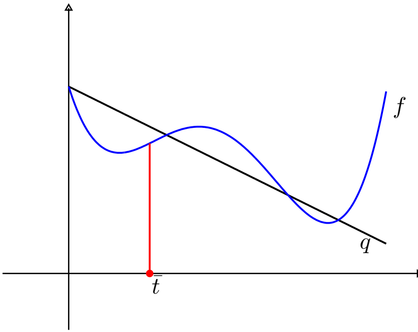
$\mathtt{gd} = g^T d$

$\mathtt{ft} = \mathtt{feval}(\mathtt{fun}, x + td, 0)$

REPITA enquanto $\mathtt{ft} > f + \eta t \, \mathtt{gd}$

$t = \gamma t$

$\mathtt{ft} = \mathtt{feval}(\mathtt{fun}, x + td, 0)$



Busca unidirecional - Exemplo

Considere o problema de minimizar a função $f(x) = x_1^2 + 4x_1x_2 + 6x_2^2$ a partir do ponto $x = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ ao longo da direção $d = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$. Resolva este problema tanto pela busca exata quanto pela de Armijo.

- Rotina da Seção Áurea fornece $t_1 = 5$ e

$$f(x + t_1 d) = 8 < 33 = f(x).$$

- Rotina de Armijo fornece $t_2 = 1$, significando que o passo inicial foi aceito. Além disso,

$$f(x + t_2 d) = 24 < 33 = f(x).$$

Em qualquer caso o valor da função decresceu, como era esperado.

Busca unidirecional - Exemplo

Considere o problema de minimizar a função $f(x) = x_1^2 + 4x_1x_2 + 6x_2^2$ a partir do ponto $x = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ ao longo da direção $d = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$.
Resolva este problema tanto pela busca exata quanto pela de Armijo.

- Rotina da Seção Áurea fornece $t_1 = 5$ e

$$f(x + t_1 d) = 8 < 33 = f(x).$$

- Rotina de Armijo fornece $t_2 = 1$, significando que o passo inicial foi aceito. Além disso,

$$f(x + t_2 d) = 24 < 33 = f(x).$$

Em qualquer caso o valor da função decresceu, como era esperado.

Busca unidirecional - Exemplo

Considere o problema de minimizar a função $f(x) = x_1^2 + 4x_1x_2 + 6x_2^2$ a partir do ponto $x = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ ao longo da direção $d = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$.
Resolva este problema tanto pela busca exata quanto pela de Armijo.

- Rotina da Seção Áurea fornece $t_1 = 5$ e

$$f(x + t_1 d) = 8 < 33 = f(x).$$

- Rotina de Armijo fornece $t_2 = 1$, significando que o passo inicial foi aceito. Além disso,

$$f(x + t_2 d) = 24 < 33 = f(x).$$

Em qualquer caso o valor da função decresceu, como era esperado.

CENGAGE Learning

Considere o problema de minimizar a função $f(x) = x_1^2 + 4x_1x_2 + 6x_2^2$ a partir do ponto $x = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ ao longo da direção $d = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$.
Resolva este problema tanto pela busca exata quanto pela de Armijo.

- Rotina da Seção Áurea fornece $t_1 = 5$ e

$$f(x + t_1 d) = 8 < 33 = f(x).$$

- Rotina de Armijo fornece $t_2 = 1$, significando que o passo inicial foi aceito. Além disso,

$$f(x + t_2 d) = 24 < 33 = f(x).$$

Em qualquer caso o valor da função decresceu, como era esperado.

$k = 0$, $x = x^0$

$g = \texttt{feval}(\texttt{fun}, x, 1)$, $\texttt{normg} = \|g\|$

Para gerar os gráficos

$\quad f = \texttt{feval}(\texttt{fun}, x, 0)$, $F = \{f\}$, $G = \{\texttt{normg}\}$

IMPRIMA NA TELA: k, f, $\texttt{normg}$

REPITA enquanto $\texttt{normg} > \varepsilon$ e $k < k_{\max}$

$\quad$ Calcule t pela busca exata a partir de x na direção $-g$

$\quad x = x - tg$

$\quad g = \texttt{feval}(\texttt{fun}, x, 1)$, $\texttt{normg} = \|g\|$

$\quad k = k + 1$

$\quad$ Para gerar os gráficos

$\quad\quad f = \texttt{feval}(\texttt{fun}, x, 0)$, $F = F \cup \{f\}$, $G = G \cup \{\texttt{normg}\}$

$\quad$ IMPRIMA NA TELA: k, f, $\texttt{normg}$

SE $k = k_{\max}$, MENSAGEM: Atingiu o número máximo de iterações!

$K = \{0, 1, \ldots, k\}$

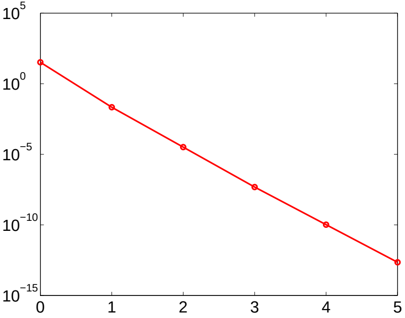
PLOTE OS GRÁFICOS: $K \times F$ e $K \times G$

Método do gradiente com seção áurea - Exemplo

Teste a rotina anterior para minimizar a função a função $f(x) = x_1^2 + 4x_1x_2 + 6x_2^2$ a partir do ponto $x^0 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.
Considere os parâmetros $\varepsilon = 10^{-5}$ e $k_{\max} = 1000$.

O problema é resolvido em 5 iterações.

k	f	$\|g\|$
0	33.000000	29.732137
1	0.021607	0.160609
2	0.000032	0.029354
3	0.000000	0.000239
4	0.000000	0.000053
5	0.000000	0.000001



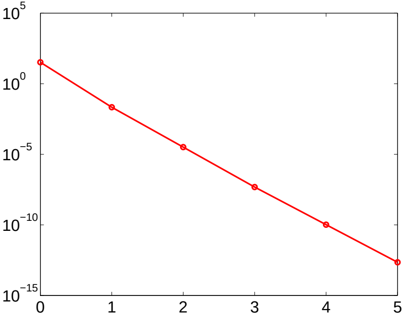
Variação de f

CENGAGE Learning

Teste a rotina anterior para minimizar a função a função $f(x) = x_1^2 + 4x_1x_2 + 6x_2^2$ a partir do ponto $x^0 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.
Considere os parâmetros $\varepsilon = 10^{-5}$ e $k_{\max} = 1000$.

O problema é resolvido em 5 iterações.

k	f	$\|g\|$
0	33.000000	29.732137
1	0.021607	0.160609
2	0.000032	0.029354
3	0.000000	0.000239
4	0.000000	0.000053
5	0.000000	0.000001



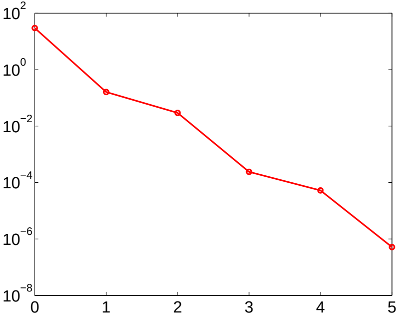
Variação de f

Teste a rotina anterior para minimizar a função a função $f(x) = x_1^2 + 4x_1x_2 + 6x_2^2$ a partir do ponto $x^0 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.

Considere os parâmetros $\varepsilon = 10^{-5}$ e $k_{\max} = 1000$.

O problema é resolvido em 5 iterações.

k	f	$\|g\|$
0	33.000000	29.732137
1	0.021607	0.160609
2	0.000032	0.029354
3	0.000000	0.000239
4	0.000000	0.000053
5	0.000000	0.000001



Variação de $\|\nabla f\|$

Método do gradiente com busca de Armijo

$k = 0$, $x = x^0$

$g = \texttt{feval}(\texttt{fun}, x, 1)$, $\texttt{normg} = \|g\|$

Para gerar os gráficos

$f = \texttt{feval}(\texttt{fun}, x, 0)$, $F = \{f\}$, $G = \{\texttt{normg}\}$

IMPRIMA NA TELA: k, f, $\texttt{normg}$

REPITA enquanto $\texttt{normg} > \varepsilon$ e $k < k_{\max}$

Calcule t pela busca de Armijo a partir de x na direção $-g$

$x = x - tg$

$g = \texttt{feval}(\texttt{fun}, x, 1)$, $\texttt{normg} = \|g\|$

$k = k + 1$

Para gerar os gráficos

$f = \texttt{feval}(\texttt{fun}, x, 0)$, $F = F \cup \{f\}$, $G = G \cup \{\texttt{normg}\}$

IMPRIMA NA TELA: k, f, $\texttt{normg}$

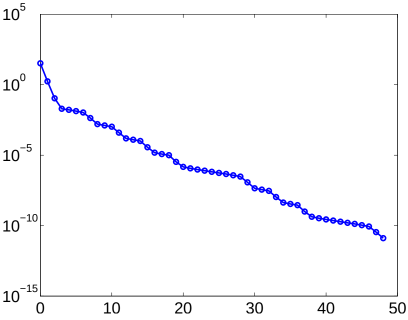
SE $k = k_{\max}$, MENSAGEM: Atingiu o número máximo de iterações!

$K = \{0, 1, \ldots, k\}$

PLOTE OS GRÁFICOS: $K \times F$ e $K \times G$

Teste a rotina anterior para minimizar a função a função $f(x) = x_1^2 + 4x_1x_2 + 6x_2^2$ a partir do ponto $x^0 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$. Considere os parâmetros $\varepsilon = 10^{-5}$ e $k_{\max} = 1000$.

O método gasta 48 iterações para obter uma solução com a mesma precisão para o critério de parada.

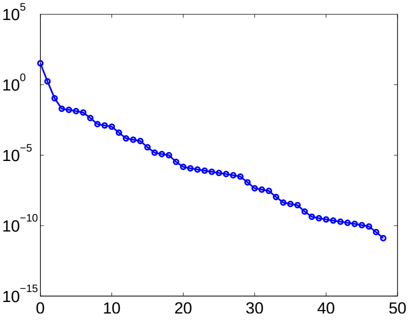


Variação de f

Método do gradiente com busca de Armijo - Exemplo

Teste a rotina anterior para minimizar a função a função $f(x) = x_1^2 + 4x_1x_2 + 6x_2^2$ a partir do ponto $x^0 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.
Considere os parâmetros $\varepsilon = 10^{-5}$ e $k_{max} = 1000$.

O método gasta 48 iterações para obter uma solução com a mesma precisão para o critério de parada.



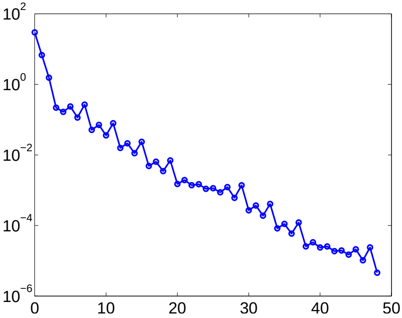
Variação de f

CENGAGE Learning

Método do gradiente com busca de Armijo - Exemplo

Teste a rotina anterior para minimizar a função a função $f(x) = x_1^2 + 4x_1x_2 + 6x_2^2$ a partir do ponto $x^0 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.
Considere os parâmetros $\varepsilon = 10^{-5}$ e $k_{\max} = 1000$.

O método gasta 48 iterações para obter uma solução com a mesma precisão para o critério de parada.



Variação de $\|\nabla f\|$

CENGAGE Learning

Criação de um arquivo que armazena os problemas da coleção Moré, Garbow e Hillstrom, com informações da função e do ponto inicial.

`fun='mgh'`

Variável de entrada: p (representa o problema a ser resolvido)

CASO $\mathrm{p} = 1$

```
    FUNC='rosen', x0=[-1.2;1], mm=2
```

CASO $\mathrm{p} = 2$

```
    FUNC='froth', x0=[0.5;-2], mm=2
```

...

Rotina `problemas_mgh`

Criação de um arquivo que armazena os problemas da coleção Moré, Garbow e Hillstrom, com informações da função e do ponto inicial.

`fun='mgh'`

Variável de entrada: $\mathtt{p}$ (representa o problema a ser resolvido)

CASO $\mathtt{p} = 1$

```
    FUNC='rosen', x0=[-1.2;1], mm=2
```

CASO $\mathtt{p} = 2$

```
    FUNC='froth', x0=[0.5;-2], mm=2
```

...

Rotina - resolve_problemas

Resolva os problemas da coleção Moré, Garbow e Hillstrom pelo método do gradiente com busca exata, com $\varepsilon = 10^{-3}$ e $k_{\max} = 3000$. Calcule o percentual de problemas resolvidos sem atingir o número máximo de iterações.

Variáveis globais: FUNC e mm

Parâmetros: $\varepsilon = 10^{-3}$, $k_{\max} = 3000$

sucesso $= 0$, total $= 0$

PARA p $= 1, 2, \ldots$

    problemas_mgh (chamada do problema)

    grad_aurea (chamada do método)

    SE $k < k_{\max}$

        sucesso $=$ sucesso $+ 1$

    total $=$ total $+ 1$

Total de problemas: total

Problemas resolvidos: sucesso

Percentual de sucesso: sucesso/total

Rotina - `resolve_problemas`

Resolva os problemas da coleção Moré, Garbow e Hillstrom pelo método do gradiente com busca exata, com $\varepsilon = 10^{-3}$ e $k_{\max} = 3000$. Calcule o percentual de problemas resolvidos sem atingir o número máximo de iterações.

Variáveis globais: `FUNC` e `mm`

Parâmetros: $\varepsilon = 10^{-3}$, $k_{\max} = 3000$

$\texttt{sucesso} = 0$, $\texttt{total} = 0$

PARA $\texttt{p} = 1, 2, \ldots$

    problemas_mgh (chamada do problema)

    grad_aurea (chamada do método)

    SE $k < k_{\max}$

        $\texttt{sucesso} = \texttt{sucesso} + 1$

    $\texttt{total} = \texttt{total} + 1$

Total de problemas: `total`

Problemas resolvidos: `sucesso`

Percentual de sucesso: `sucesso/total`

CENGAGE Learning

Comparar o desempenho de diferentes algoritmos

Objetivo: avaliar e comparar o desempenho de um conjunto $\mathcal{S}$ de n_s algoritmos aplicados a um conjunto $\mathcal{P}$ de n_p problemas teste.

Considere, por exemplo,

- $t_{p,s}$: tempo de CPU para resolver o problema $p \in \mathcal{P}$ pelo algoritmo $s \in \mathcal{S}$.
- Se o algoritmo s não resolveu o problema p, faça $t_{p,s} = \infty$.

Índice de desempenho

$$r_{p,s} = \frac{t_{p,s}}{\min\left\{t_{p,j} \mid j \in \mathcal{S}\right\}}.$$

Este índice vale 1 para o algoritmo mais eficiente.
Quanto maior for seu valor, pior será o desempenho do algoritmo.

Para cada algoritmo s consideramos a função de desempenho $\rho_s : [1, \infty) \to [0, 1]$ definida por

$$\rho_s(\tau) = \frac{1}{n_p} \text{card}\,\{p \in \mathcal{P} \mid r_{p,s} \leq \tau\}.$$

$\rho_s(1)$: proporção de problemas que o algoritmo s resolve no menor tempo.

Generalização: considere uma medida de desempenho arbitrária, $\rho_s(\tau)$ é a porcentagem de problemas que o algoritmo s resolve em τ vezes o valor da medida de desempenho do algoritmo mais eficiente.
Para facilitar a visualização, construímos um gráfico de perfil de desempenho, com as funções ρ_s, $s \in \mathcal{S}$.

Comparar o desempenho de diferentes algoritmos

- Variantes do método do gradiente com busca de Armijo, com $\varepsilon = 10^{-3}$, $k_{\max} = 3000$ e os seguintes parâmetros γ e η.
  - A1: $\gamma = 0.7$ e $\eta = 0.45$;
  - A2: $\gamma = 0.5$ e $\eta = 0.25$;
  - A3: $\gamma = 0.7$ e $\eta = 0.25$;
  - A4: $\gamma = 0.8$ e $\eta = 0.45$.
- Problemas: 25 primeiros da coleção Moré, Garbow e Hillstrom.
- Armazene uma matriz $T \in \mathbb{R}^{25\times 4}$ cuja p-ésima linha corresponde ao tempo gasto pelos algoritmos para resolver o problema p. Se o algoritmo s não resolveu o problema p, defina $T(p,s) = 10^8$.
- Implemente uma rotina que forneça o gráfico de perfil de desempenho dos 4 algoritmos.

Rotina para resolução dos problemas pelas variantes do algoritmo

Variáveis globais: `FUNC` e `mm`
Parâmetros: $\varepsilon = 10^{-3}$, $k_{\max} = 3000$
$T = 0$
PARA $p = 1, \ldots, 25$
    `problemas_mgh` (chamada do problema p - Rotina `problemas_mgh`)
    PARA $s = 1, \ldots, 4$
        Escolha γ e η (da variante do Algoritmo s)
        `grad_armijo`
        $T(p,s) = \mathtt{tp}$ (tempo gasto para resolver o problema)

Rotina para gerar o gráfico de perfil de desempenho

Dados: $T \in \mathbb{R}^{25\times 4}$, $\tau_{\max}$

REPITA para $p = 1, \ldots, 25$

$\quad \mathtt{Tmin}(p) = \min\{T(p,s) \mid s = 1, \ldots, 4\}$

$\quad$ REPITA para $s = 1, \ldots, 4$

$$r(p,s) = \frac{T(p,s)}{\mathtt{Tmin}(p)}$$

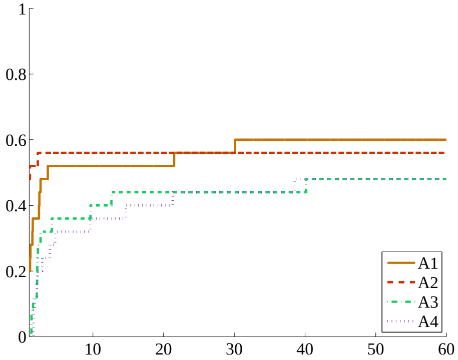
REPITA para $s = 1, \ldots, 4$

$\quad$ REPITA para $1 \leq \tau \leq \tau_{\max}$

$$\rho_s(\tau) = \frac{1}{n_p} \text{card}\{p \in \mathcal{P} \mid r_{p,s} \leq \tau\}$$

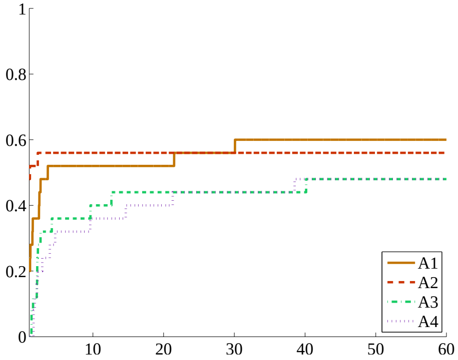
$\quad$ PLOTE O GRÁFICO: $\tau \times \rho_s(\tau)$

Gráfico de perfil de desempenho - Tempo computacional



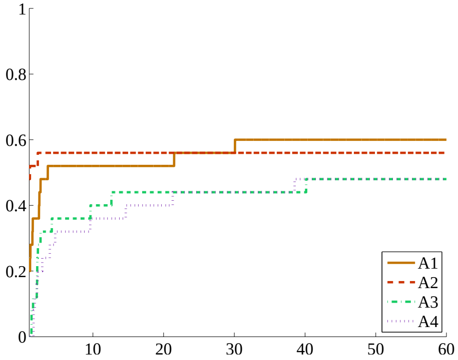
- A1 resolve 60% dos problemas e é mais rápido em 20% deles.
- A2 é o mais rápido em 48% dos problemas e resolve 56% deles.
- A3 resolve 48% dos problemas usando 40 vezes o melhor tempo.
- O pior desempenho foi do algoritmo A4.
- A não resolução de alguns problemas motiva o estudo de outros métodos.

Gráfico de perfil de desempenho - Tempo computacional



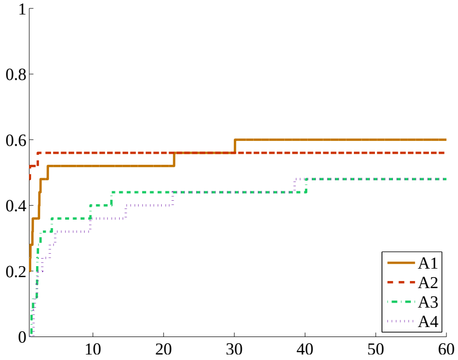
- A1 resolve 60% dos problemas e é mais rápido em 20% deles.
- A2 é o mais rápido em 48% dos problemas e resolve 56% deles.
- A3 resolve 48% dos problemas usando 40 vezes o melhor tempo.
- O pior desempenho foi do algoritmo A4.
- A não resolução de alguns problemas motiva o estudo de outros métodos.

Gráfico de perfil de desempenho - Tempo computacional



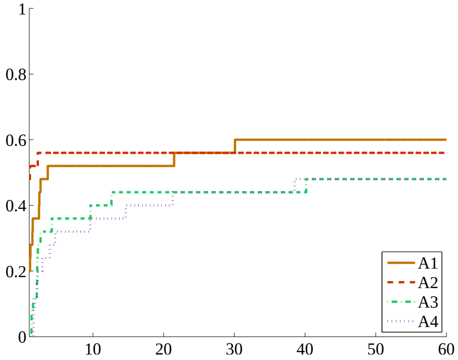
- A1 resolve 60% dos problemas e é mais rápido em 20% deles.
- A2 é o mais rápido em 48% dos problemas e resolve 56% deles.
- A3 resolve 48% dos problemas usando 40 vezes o melhor tempo.
- O pior desempenho foi do algoritmo A4.
- A não resolução de alguns problemas motiva o estudo de outros métodos.

Gráfico de perfil de desempenho - Tempo computacional



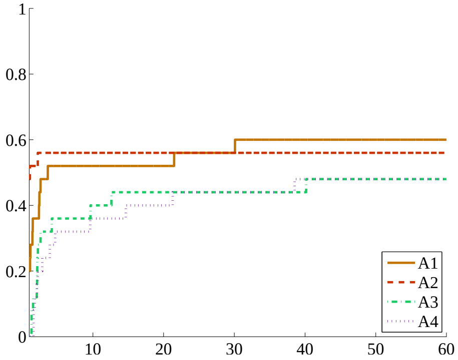
- A1 resolve 60% dos problemas e é mais rápido em 20% deles.
- A2 é o mais rápido em 48% dos problemas e resolve 56% deles.
- A3 resolve 48% dos problemas usando 40 vezes o melhor tempo.
- O pior desempenho foi do algoritmo A4.
- A não resolução de alguns problemas motiva o estudo de outros métodos.

Gráfico de perfil de desempenho - Tempo computacional



- A1 resolve 60% dos problemas e é mais rápido em 20% deles.
- A2 é o mais rápido em 48% dos problemas e resolve 56% deles.
- A3 resolve 48% dos problemas usando 40 vezes o melhor tempo.
- O pior desempenho foi do algoritmo A4.
- A não resolução de alguns problemas motiva o estudo de outros métodos.

Gráfico de perfil de desempenho - Tempo computacional



- A1 resolve 60% dos problemas e é mais rápido em 20% deles.
- A2 é o mais rápido em 48% dos problemas e resolve 56% deles.
- A3 resolve 48% dos problemas usando 40 vezes o melhor tempo.
- O pior desempenho foi do algoritmo A4.
- A não resolução de alguns problemas motiva o estudo de outros métodos.

Gráfico de perfil de desempenho

Medidas de desempenho de algoritmos:

- tempo computacional gasto pelos algoritmos para resolver um conjunto de problemas
- número de iterações
- número de avaliações de função
- ou qualquer outra medida de desempenho de um conjunto de algoritmos, bastando armazenar a informação desejada na matriz T da Rotina `roda_metodos`.

Principais referências

E. D. Dolan e J. J. Moré,
Benchmarking Optimization Software with Performance Profiles,
Mathematical Programming, 91, 201–213, 2002.

J. J. Moré, B. S. Garbow e K. E. Hillstrom,
Testing Unconstrained Optimization Software,
ACM Transactions on Mathematical Software, v. 7, N.1, 17-41, 1981,