

# Variações do método de máxima descida em otimização irrestrita

Momoe Sakamori\*      Márcia A. Gomes–Ruggiero †

## Resumo

Neste trabalho propomos um novo algoritmo para otimização irrestrita onde a direção resulta de uma combinação linear das direções de máxima descida das duas iterações anteriores e os parâmetros que compõem esta combinação são os passos de Barzilai e Borwein nas versões direta e inversa. Algumas formas de realizar o controle de passo são também propostas e analisadas. Um único algoritmo resultante da nova direção com a melhor opção para controle de passo é testado com algoritmos clássicos para otimização irrestrita na resolução de um conjunto de problemas de quadrados mínimos. A ferramenta empregada no processo de comparação é o perfil de desempenho (*performance profile*) através da qual comprovamos o bom desempenho do método proposto frente aos demais processos que usam apenas informações de primeira ordem.

**Palavras Chave:** otimização irrestrita, direção de descida, controle de passo, método de máxima descida, método de Newton.

## 1 Introdução

O enfoque neste trabalho é dado para métodos propostos para resolução de problemas de otimização irrestrita:

$$\min f(x), \quad f : \mathbb{R}^n \rightarrow \mathbb{R}. \quad (1)$$

---

\*DMA–IMECC–UNICAMP, email: momoe\_sakamori@yahoo.com.br. Apoio FAPESP 02/13522-0.

†DMA–IMECC–UNICAMP, email: marcia@ime.unicamp.br. Apoio FAPESP 2000-00375-4, PRONEX, CNPq 307167/2003-8.

Nestes algoritmos, a nova aproximação  $x_{k+1}$ , é obtida a partir de  $x_k$  através de:

$$x_{k+1} = x_k + \lambda_k d_k, \quad (2)$$

onde  $d_k$  deve ser uma direção de descida para  $f$  em  $x_k$  e  $\lambda_k \in \mathbb{R}$ ,  $\lambda_k > 0$  é o tamanho ou controle do passo, determinado de modo a garantir decréscimo no valor de  $f$ .

Neste trabalho realizamos uma análise computacional da direção de máxima descida com tamanho do passo obtido por: busca linear exata e através das propostas de Barzilai e Borwein direta e inversa, [1]. Além disto, elaboramos um novo algoritmo, no qual a direção resulta de uma combinação linear das direções de máxima descida das duas iterações anteriores:

$$d_{k+1} = \alpha d_{k-1} + \beta d_k, \quad (3)$$

onde os parâmetros  $\alpha$  e  $\beta$  são dados pelos passos de Barzilai e Borwein nas versões direta (BB1) e inversa (BB2). O algoritmo resultante incorpora salvaguardas para o caso de  $d_{k+1}$  não resultar em uma direção de descida. Propomos também 4 opções para o cálculo do tamanho do passo de modo a se obter decréscimo no valor de  $f$  baseadas em processo de busca linear através da bissecção ou através de valores para  $\lambda$  relacionados às seções áureas:  $(1 + \sqrt{5})/2$ ,  $2/(1 + \sqrt{5})$  e  $(3 - \sqrt{5})/2$ .

Os teste computacionais foram realizados com um amplo conjunto de problemas de quadrados mínimos extraídos da coleção proposta em [5]. Inicialmente foram compostos 4 algoritmos com direção de descida dada pela nova direção e cada um com uma das novas propostas para o controle do passo. A avaliação dos 4 algoritmos foi realizada através da ferramenta de comparação denominada perfil de desempenho, proposta em [3] que essencialmente aponta os métodos mais eficientes em termos de rapidez de convergência e eficiência na capacidade de resolver um grande número de problemas. Por este processo de avaliação o algoritmo, entre os 4, com melhor desempenho foi então comparado com os algoritmos clássicos de descida: máxima descida, Barzilai e Borwein 1 e 2. Os critérios usados para realizar a comparação foram o número de iterações e o número de avaliações de funções.

Todos estes métodos foram programados e executados usando o Matlab, v6.1, numa máquina Pentium III. Os gráficos resultantes da avaliação de desempenho apontam o novo algoritmo como um processo eficiente pelo seu desempenho na resolução da totalidade de problemas.

Na Seção 2 descrevemos um resumo dos métodos de máxima descida e Barzilai e Borwein. Um novo algoritmo para otimização irrestrita é apresentado na Seção 3, onde são também avaliadas algumas novas propostas para o tamanho do passo. Na Seção 4 apresentamos os teste numéricos e na Seção 5 tecemos algumas conclusões.

## 2 Algoritmo básico e métodos clássicos de descida

Considerando o problema (1), um minimizador local para  $f$  é um ponto  $x^* \in \mathbb{R}^n$  se e somente se existe  $\varepsilon > 0$  tal que  $f(x^*) \leq f(x)$  para todo  $x \in \mathbb{R}^n$  tal que  $\|x - x^*\| < \varepsilon$ . Denotamos o gradiente da função  $f$  avaliado em  $x_k$  por  $g_k$  e a matriz Hessiana de  $f$  em  $x_k$  por  $H_k$ . Se  $x_0$  é uma aproximação inicial para (1), um algoritmo básico para a resolução deste problema pode ser escrito como:

Algoritmo 1:

Passo 1: Faça  $k = 0$ .

Passo 2: Enquanto  $g_k \neq 0$ :

passo 2.1: escolha uma direção de descida  $d_k$ ;

passo 2.2: obtenha  $\lambda_k > 0$  tal que:

$$f(x_k + \lambda_k d_k) < f(x_k) \quad (4)$$

passo 2.3: faça  $x_{k+1} = x_k + \lambda_k d_k$  e  $k = k + 1$ .

Os algoritmos para otimização irrestrita se distinguem pela forma como é obtida a direção  $d_k$ , e, também pela forma como realizam o passo 2.2: obtida a direção de busca, é necessário um controle para o tamanho de passo. É claro que este passo pode ser executado de modo que  $\lambda_k$  seja o minimizador de  $f$  ao longo da direção  $d_k$  a partir de  $x_k$ , e neste caso dizemos que o processo é o de Busca Linear Exata. Se  $f$  é uma função quadrática da forma:  $f(x) = x^t H x + b^t x + c$ , podemos determinar  $\lambda_k$  através de uma fórmula, [4]:

$$\lambda_k = -\frac{d_k^t g_k}{d_k^t H d_k}. \quad (5)$$

Contudo, para  $f$  geral, não linear, a minimização ao longo de uma direção de descida pode ser bastante trabalhosa e as alternativas consistem em pro-

cedimentos simples de busca através da redução do parâmetro  $\lambda$  até que o decréscimo (4) seja verificado.

## 2.1 Método de máxima descida

No Método de Máxima Descida, a direção escolhida no passo  $k$  é:  $d_k = -g_k$ . Seguem os passos com busca linear exata e os de Barzilai e Borwein [1]:

- Passo com Busca Linear Exata

Desde que  $d_k = -g_k$ , a expressão para  $\lambda_k$  dada por (5) quando  $f$  é uma quadrática é a seguinte:  $\lambda_k = g_k^t g_k / g_k^t H g_k$ .

- Passo de Barzilai e Borwein

Em 1988, Barzilai e Borwein, [1], apresentaram uma nova escolha de tamanho de passo para o Método de Máxima Descida. Esta proposta consiste em obter  $\lambda_k = 1/\alpha_k$  onde:

$$\alpha_k = \frac{s_{k-1}^t y_{k-1}}{s_{k-1}^t s_{k-1}} \quad \text{ou} \quad \alpha_k = \frac{y_{k-1}^t y_{k-1}}{s_{k-1}^t y_{k-1}}$$

onde  $s_{k-1} = x_k - x_{k-1}$  e  $y_{k-1} = g_k - g_{k-1}$ , resultando em duas fórmulas para o tamanho do passo aqui denominadas respectivamente por BB1 e BB2.

## 3 Proposta de nova direção de busca

Nesta seção descrevemos as duas direções propostas neste trabalho e apresentamos quatro opções para o controle de passo resultando em um novo algoritmo para otimização irrestrita para funções gerais. As direções são obtidas a partir de uma combinação linear dos gradientes das duas últimas iterações.

### 3.1 Direção 1: proposta para funções quadráticas

No método de máxima descida para funções quadráticas de duas variáveis os termos da sequência  $x_k$  alternam-se entre duas retas, conforme demonstrado em [2], sendo que estas retas possuem como ponto de intersecção o ponto de mínimo da quadrática. Resumimos a seguir este resultado.

Seja  $f$  uma função quadrática bidimensional na forma  $f(x) = 0.5 x^t H x$ , com  $x = (x_1, x_2)^t$  e  $H = [1 \ 0; 0 \ \alpha]$ , com  $1 < \alpha \in \mathbb{R}$ . O gradiente de  $f$  é

dado por  $g(x) = (x_1, \alpha x_2)^t$  e a direção de máxima descida é  $d_k = -g(x_k) = -(x_1^k, \alpha x_2^k)^t$ .

Sejam  $x^0 \neq (0, 0)^t$  uma aproximação inicial e a constante  $C$  obtida por  $C = x_1^0/x_2^0$ . Considerando que  $x^{k+1}$  é obtido por  $x^{k+1} = x^k + \lambda_k d_k$  com  $\lambda_k$  obtido por busca linear exata obtém-se as seguintes taxas entre as componentes de  $x^{k+1}$  e  $x^{k+2}$  para  $k = 0, 1, \dots$ :

$$x_1^{k+1}/x_2^{k+1} = -\alpha^2/C$$

e

$$x_1^{k+2}/x_2^{k+2} = C.$$

Isto significa que, geometricamente, a sequência  $\{x^k\}$  alterna entre duas retas: os termos pares da sequência pertencem à reta  $x_2 = (1/C)x_1$  e os ímpares à reta  $x_2 = -(C/\alpha^2)x_1$ , como mostra a Figura 1(a).

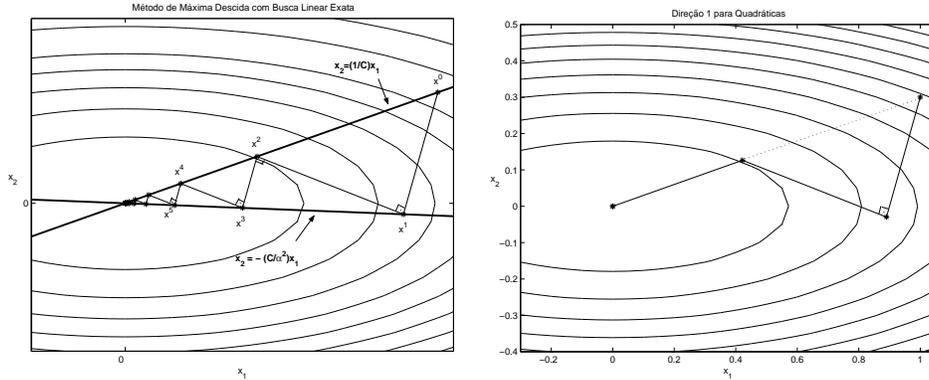


Figura 1. (a) Método de Máxima Descida com Busca Linear Exata e (b) Direção 1.

Com esta motivação geométrica, propomos a seguinte direção que é a combinação linear dos gradientes dos dois últimos pontos encontrados pelo Método de Máxima Descida com busca linear exata:

$$d_k = -\lambda_{k-2}g_{k-2} - \lambda_{k-1}g_{k-1}.$$

Utilizamos a busca linear exata para o tamanho do passo da nova direção aqui denominada de Direção 1, e desta forma, teremos que  $x^3$  será o minimizador de  $f$ , conforme podemos verificar na Figura 1(b). A seguir está proposto o algoritmo básico para o Método da Direção 1 para Quadráticas.

Algoritmo 2:

Passo 1: Faça  $k = 1$ .

Passo 2: Enquanto  $g_k \neq 0$ , faça;

passo 2.1: se  $(k \bmod 3) = 1$  ou  $(k \bmod 3) = 2$

passo 2.1.1: faça  $d_k = -g_k$ .

passo 2.2: se  $(k \bmod 3) = 0$ :

passo 2.2.1: faça  $d_k = -\lambda_{k-2}g_{k-2} - \lambda_{k-1}g_{k-1}$

passo 2.3: faça  $\lambda_k = -\frac{d_k^t g_k}{d_k^t H d_k}$

passo 2.4: faça  $x_{k+1} = x_k + \lambda_k d_k$  e  $k = k + 1$ .

Analisamos o desempenho computacional deste processo na resolução de problemas de minimização de quadráticas em  $\mathbb{R}^2$  e em dimensões maiores  $n = 3, 4$ , e  $5$ . Comparamos os resultados com o desempenho do Método de Máxima Descida com Busca Linear Exata, BLE, e com os passos Barzilai e Borwein, BB1 e BB2. O resultado de convergência em 3 iterações do Algoritmo 2 para  $n = 2$  foi comprovado mesmo em exemplos onde BLE não convergiu devido ao mal condicionamento da matriz Hessiana. Para os problemas com dimensões maiores os processos BB1 e BB2 tiveram desempenho superior demonstrando que a proposta da Direção 1 é eficiente apenas para  $n = 2$ .

No entanto, a idéia de se obter uma direção a partir de uma combinação linear das direções de máxima descida das iterações anteriores deve ser aproveitada bastando pesquisar os pesos adequados para esta combinação bem como uma escolha para o tamanho do passo.

## 3.2 Direção 2: nova proposta para funções gerais

Buscando uma alternativa, mantivemos a mesma idéia de combinação linear dos gradientes, resultando, como proposta para a nova direção, a seguinte fórmula, com dois parâmetros a serem determinados:

$$d_k = -\beta_1 g_{k-1} - \beta_2 g_k, \quad \beta_1, \beta_2 \in \mathbb{R}. \quad (6)$$

Considerando que os passos de Barzilai e Borwein resultam em métodos com bom desempenho, uma opção seria construir  $d_k$  como em (6) usando para  $\beta_1$  e  $\beta_2$  os passos encontrados em BB1 e BB2. Como esta direção pode não ser de descida, o algoritmo inclui uma salvaguarda para prevenir este caso. A seguir está proposto o algoritmo básico para o Método da Direção 2.

Algoritmo 3:

Passo 1: Faça  $k = 0$ .

Passo 2: Faça  $x_{k+1} = x_k - g_k$  e  $k = k + 1$ .

Passo 3: Enquanto  $g_k \neq 0$ :

passo 3.1: se  $(k \bmod 3) = 1$

passo 3.1.1 faça  $d_k = -g_k$

passo 3.1.2 faça  $\alpha_k = \frac{s_{k-1}^t y_{k-1}}{s_{k-1}^t s_{k-1}}$  e  $\lambda_k = 1/\alpha_k$  BB1

passo 3.1.3 faça  $x_{k+1} = x_k + \lambda_k d_k$  e  $k = k + 1$ .

passo 3.2: se  $(k \bmod 3) = 2$

passo 3.2.1 faça  $d_k = -g_k$

passo 3.2.2 faça  $\alpha_k = \frac{y_{k-1}^t y_{k-1}}{s_{k-1}^t y_{k-1}}$  e  $\lambda_k = 1/\alpha_k$  BB2

passo 3.2.3 faça  $x_{k+1} = x_k + \lambda_k d_k$  e  $k = k + 1$ .

passo 3.3: se  $(k \bmod 3) = 0$

passo 3.3.1 faça  $d_k = -\lambda_{k-2} g_{k-1} - \lambda_{k-1} g_k$

passo 3.3.2 se  $d_k^t g_k > 0$

passo 3.3.2.1 faça  $d_k = -d_k$

passo 3.3.3 obter  $\bar{\lambda} > 0$  tal que  $f(x_k + \bar{\lambda} d_k) < f(x_k)$

passo 3.3.4 faça  $x_{k+1} = x_k + \lambda_k d_k$  e  $k = k + 1$ .

Destacamos que este algoritmo aciona uma sequência de direções: para  $k = 3, 6, 9, \dots$ :

na iteração  $k - 2$ , teremos  $d_{k-2} = -g_{k-2}$  e passo  $\lambda_{k-2}$  dado por BB1;

na iteração  $k - 1$ ,  $d_{k-1} = -g_{k-1}$  e o passo  $\lambda_{k-1}$  dado por BB2;

e finalmente

$$d_k = -\lambda_{k-2} g_{k-1} - \lambda_{k-1} g_k \quad (7)$$

é a nova direção proposta, denominada por Direção 2; o passo  $\lambda_k$  será obtido por um procedimento de busca, conforme proposto a seguir.

### 3.3 Controle do passo para a direção 2

Propomos nesta seção algumas escolhas para o cálculo do tamanho do passo. As propostas consistem em realizar uma busca linear ao longo da direção  $d_k$ , iniciando com valores 1 e  $(1 + \sqrt{5})/2$  para  $\lambda$  e reduzindo o valor obtido por um fator  $\sigma$  que pode ser como na bissecção,  $\sigma = 0.5$ , baseado na seção áurea  $2/(1 + \sqrt{5})$  e no seu segmento menor  $(3 - \sqrt{5})/2$ , resultando em 4 alternativas.

As propostas estão descritas a seguir. No Passo 3.3.3 do Algoritmo 3 fazemos:

- CP1: inicia com  $\lambda = 1$  e faz  $\lambda = 0.5\lambda$  até que  $f(x_k + \lambda d_k) < f(x_k)$ ;
- CP2: inicia com  $\lambda = (1 + \sqrt{5})/2$  e faz  $\lambda = 0.5\lambda$  até que  $f(x_k + \lambda d_k) < f(x_k)$ ;
- CP3: inicia com  $\lambda = 1$  e faz  $\lambda = 2/(1 + \sqrt{5})\lambda$  até que  $f(x_k + \lambda d_k) < f(x_k)$ ;
- CP4: inicia com  $\lambda = 1$  e faz  $\lambda = (3 - \sqrt{5})/2\lambda$  até que  $f(x_k + \lambda d_k) < f(x_k)$ .

## 4 Testes numéricos

Nesta seção relatamos e analisamos alguns resultados obtidos com os experimentos numéricos. Os algoritmos foram programados em `MatLab` v6.1. Os critérios de parada utilizados nos algoritmos foram através do teste com a norma do gradiente  $\|g(x_k)\|_2 < 10^{-8}$ , o número máximo de iteração como sendo 1000 e a diferença entre duas aproximações  $\|x_{k+1} - x_k\|_2 < 10^{-16}$ . Quando a execução é interrompida devido a um dos dois últimos critérios temos a ocorrência de fracasso na execução, sendo que o teste com a diferença entre duas aproximações é necessária para prevenir dificuldades numéricas como por exemplo divisão por zero.

A análise do desempenho dos métodos é realizada através da ferramenta denominada o perfil de desempenho que será descrita na Seção 4.1. A seguir estão listados os 18 problemas de quadrados mínimos escolhidos da coleção de problemas propostos em [5]:

linear function - full rank  $n = 10$ ,  $m = 20$ ; linear function - rank 1  $n = 10$ ,  $m = 20$ ; linear function - rank 1 with zero columns and rows  $n = 10$ ,  $m = 20$ ; Rosenbrock function  $n = 2$ ,  $m = 2$ ; helical valley function  $n = 3$ ,  $m = 3$ ; Powell singular function  $n = 4$ ,  $m = 4$ ; Freudenstein and Roth function  $n = 2$ ,  $m = 2$ ; Bard function  $n = 3$ ,  $m = 15$ ; Kowalik and Osborne function  $n = 4$ ,  $m = 11$ ; Meyer function  $n = 3$ ,  $m = 16$ ; Watson function  $n = 9$ ,  $m = 31$ ; box three-dimensional function  $n = 3$ ,  $m = 3$ ; Jennrich and Sampson function  $n = 2$ ,  $m = 10$ ; Brown and Dennis function  $n = 4$ ,  $m = 20$ ; Powell badly scaled function  $n = 2$ ,  $m = 2$ ; Brown almost-linear function  $n = 10$ ,  $m = 10$ ; Osborne 1 function  $n = 5$ ,  $m = 33$ ; Osborne 2 function  $n = 11$ ,  $m = 65$ .

### 4.1 Perfil de desempenho

Introduzido em 2002 por Dolan e Moré, [3], o perfil de desempenho (*performance profile*) é uma ferramenta poderosa para comparar o desempenho de  $n_s$  métodos de um conjunto  $S$  quando aplicado para resolver  $n_p$  problemas

de um conjunto  $P$ , usando algumas medidas como o número de iterações ou o número de avaliações de função. Seja  $m_{s,p}$  quantidade da medida  $m$  que o método  $s$  usou para resolver o problema  $p$ . Para cada problema  $p$  e método  $s$ , é calculada a taxa de desempenho  $r_{s,p}$ :  $r_{s,p} = m_{s,p} / \min\{m_{s,p} \mid \forall s \in S\}$ , se o problema  $p$  é resolvido pelo método  $s$ ; caso contrário,  $r_{s,p} = r_M$  onde  $r_M$  é um parâmetro fixado suficientemente grande. Então, para cada  $s \in S$ , a função de distribuição acumulada

$$\rho_s : \mathbb{R}^n \rightarrow [0, 1],$$

para a taxa de desempenho  $r_{s,t}$  é construída:

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

Esta função representa o desempenho do método  $s$ . Para a análise do método  $s$ , os valores  $\rho_s(1)$  e  $\bar{t}$ , tal que  $\rho_s(\bar{t}) = 1$ , fornecem informações importantes. Seja  $\bar{s}$  o método  $s$  que dá o valor máximo para a função  $\rho_s(1)$ . Este método pode resolver o maior número de problemas usando o menor número da medida  $m$ . A eficiência do método  $s$  em termos do número de problemas que foram resolvidos é avaliado pelo menor valor de  $t$ , denotado por  $\bar{t}_s$ , tal que  $\rho_s(\bar{t}_s) = 1$ , se existe tal valor para  $t < r_M$ . Portanto, o melhor método em termos de eficiência será o método  $\hat{s}$  para o qual  $\bar{t}_{\hat{s}} = \min\{\bar{t}_s, \forall s \in S\}$ .

## 4.2 Direção 2 com escolhas de controle de passo

Na Seção 3.3 foram descritas 4 propostas de controle de passo estudadas para a Direção 2. Testamos essas propostas para o conjunto de problemas de quadrados mínimos com o objetivo de selecionar somente aquela com melhor desempenho. Os resultados estão na Tabela 1, onde **Feval** representa o número de avaliações de função; **Iter** representa o número de iterações; o símbolo \*\* indica que houve interrupção da execução mas a aproximação estava próxima da solução. Os problemas assinalados com X representam os casos onde a execução foi interrompida devido a alguma dificuldade numérica.

Para analisar o perfil de desempenho dos controles de passo para a Direção 2 eliminamos os problemas assinalados com X. Portanto, consideramos 4 métodos e 12 problemas: Funções 1-9, 12, 14 e 16. Os critérios utilizados foram o número de iterações e o número de avaliações de função.

Func.	CP1		CP2		CP3		CP4	
	Iter	Feval	Iter	Feval	Iter	Feval	Iter	Feval
1	2	2	2	2	2	2	2	2
2	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3
4	107	307	122	362	140	438	114	269
5	146	298	133	306	161	383	98	179
6	1000**	2348**	489	1376	959	3159	695	1746
7	107	226	92	238	1000**	2476**	1000**	1858**
8	26	161	32	230	26	221	26	127
9	453	999	419	993	674	1838	505	986
10	X	X	X	X	X	X	X	X
11	X	X	X	X	X	X	X	X
12	131	379	119	599	143	964	125	478
13	X	X	X	X	X	X	X	X
14	427**	769**	796**	1690**	1000**	2217**	1000**	1715**
15	X	X	X	X	X	X	X	X
16	2	2	2	2	2	2	2	2
17	X	X	X	X	X	X	X	X
18	X	X	X	X	X	X	X	X

Tabela 1. Algoritmo 3: Direção 2 com as 4 propostas para controle de passo

A Figura 2 mostra os gráficos do perfil de desempenho dos 4 algoritmos: Direção 2 com controle do passo obtido usando as 4 propostas para controle de passo. As medidas usadas foram o número de iterações (figura à esquerda) e o número de avaliações de função (figura à direita). Com relação ao número de iterações, o melhor desempenho foi obtido com o controle de passo 2, CP2, pois este processo resolveu 75% dos problemas com o menor número de iterações e também foi o primeiro que alcançou a taxa 1, isto é, foi o método que resolveu todos os problemas com menor valor para  $t$ . O processo com controle de passo 4, CP4, resolveu 75% dos problemas com o menor número de avaliações de função, enquanto que CP2 resolveu em torno de 50%. Entretanto, temos que CP2 resolveu a totalidade dos problemas com menor valor para  $t$ . Portanto, analisando os gráficos, o controle de passo que obteve melhor desempenho foi CP2. Nos testes seguintes fixamos esta opção para o controle de passo resultando no Algoritmo 3 com passo 3.3.3 executado com a opção CP2, resultando no processo denominado por D2/CP2.

### 4.3 Comparação entre D2/CP2 e outros métodos de descida

Nesta seção apresentamos os resultados e análises dos testes numéricos para os 4 métodos: Método da Máxima Descida com Bisseção (MMD/B), Método da

Máxima Descida com os passos de Barzilai e Borwein (BB1 e BB2) e Direção 2 com Controle de Passo 2 (D2/CP2).

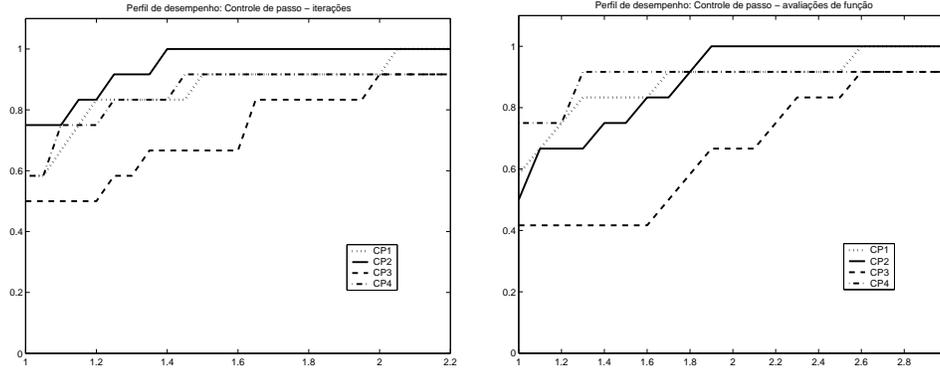


Figura 2. Perfil de desempenho com número de iterações e avaliações de função.

Na Tabela 2 estão os resultados dos testes numéricos, na qual **Feval** indica o número de avaliações de função; **Iter** indica o iterações; X indica ocorrência de algum tipo de erro durante a execução como divisão por zero; nos processos assinalados com o símbolo \* a convergência foi atingida mas provavelmente para um outro ótimo local, pois é diferente da solução indicada em [5]; nos processos assinalados com \*\* e \*\*\* a execução foi interrompida devido a passo muito próximo de zero, sendo que o símbolo \*\* indica que a aproximação está próxima da solução, e, naqueles assinalados com \*\*\* está distante da solução. Dos 18 problemas, apenas na Função 13 a execução foi interrompida para todos os métodos e esta função foi eliminada do conjunto dos problemas para a análise do perfil de desempenho.

Para realizar o perfil de desempenho com os 4 métodos, consideramos 13 problemas: Funções 1-9, 11, 12, 14 e 16, que são aqueles em que todos os métodos atingiram a convergência. A Figura 3 apresenta os gráficos do perfil de desempenho com  $t \in [1, 7]$  e  $t \in [1, 5500]$ . Verificamos que BB2 resolveu 70% dos problemas com menor número de iterações, sendo que D2/CP2 resolveu pouco mais de 50%, porém, D2/CP2 conseguiu resolver todos os problemas com menor valor para o parâmetro  $t$ . Dos gráficos também podemos observar que o método menos eficiente foi MMD/B. Comparando os 4 métodos, D2/CP2 superou MMD/B e BB1 e tem desempenho semelhante ao de BB2. Portanto, para este conjunto de problemas, D2/CP2 foi bem competitivo com os métodos de máxima descida que usam informação de primeira ordem.

Função	MMD/B		BB1	BB2	D2/CP-2	
	Iter	Feval	Iter	Iter	Iter	Feval
1	2	3	2	2	2	3
2	19**	452**	3	3	3	3
3	19**	413**	3	3	3	3
4	1000	9833	1000	175	122	362
5	1000	9333	177	105	133	306
6	1000	7601	195	208	489	1376
7	1000	11819	87	55	92	238
8	1000	4355	63*	399*	32	230
9	1000**	2719**	93*	1000	419	993
10	1000	17676	82*	1000	X	X
11	1000	8347	1000	1000**	X	X
12	1000	1215	50	19	119	599
13	X	X	X	X	X	X
14	629**	14303**	165	153	796**	1690**
15	1000	27491	X	5***	X	X
16	42	378	2	2	2	2
17	1000	15032	X	X	X	X
18	1000	6842	68*	1000*	X	X

Tabela 2. Comparação entre todos os métodos

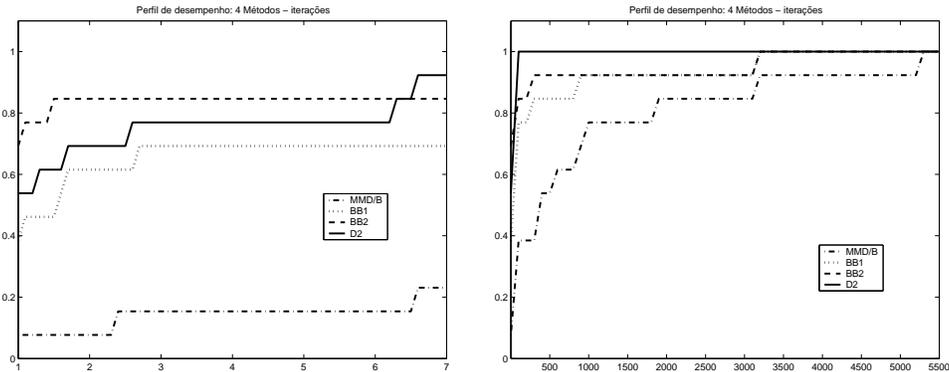


Figura 3. Perfil de desempenho com número de iterações.

## 5 Conclusões finais

Neste trabalho foi proposto um algoritmo para otimização irrestrita com direção obtida a partir de uma combinação linear de direções de máxima descida de iterações anteriores. O algoritmo resultante alterna iterações de Barzilai e Borwein e iterações com a nova direção. É fácil de ser programado

pois requer apenas informação da função e do gradiente e mostrou desempenho superior aos métodos de máxima descida e de Barzilai e Borwein demonstrando ser uma boa proposta na categoria de métodos que usam apenas informações de primeira ordem.

## Referências

- [1] J. Barzilai and J. M. Borwein, Two-point step size gradient methods, *IMA Journal of Numerical Analysis*, Vol. 8, pp. 141–148, 1988.
- [2] M. S. Bazaraa and C. M. Shetty, “Nonlinear Programming - Theory and Algorithms”, John Wiley and Sons, 1979.
- [3] E.D. Dolan, J.J.Moré, Benchmarking optimization software with performance profile, *Math. Program.*, ser. A91, pp. 201-213, 2002.
- [4] A. Friedlander, “Elementos de Programação Não Linear”, Ed. Unicamp, 1994.
- [5] J. J. Moré, B. S. Garbow, K. E. Hillstrom, Testing unconstrained optimization software, *ACM Transactions on Mathematical Software*, V.7, pp.17-41, 1981.
- [6] D. Hanselman and B. Littlefield, “MatLab 6”, Prentice Hall, 2003.
- [7] G. Strang, “Linear Algebra And Its Applications”, Harcourt Brace Jovanovich, 1988.