



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**

sid.inpe.br/mtc-m21b/2016/09.15.14.04-TDI

## **MÉTODO FQ - G PARA OTIMIZAÇÃO GLOBAL DE PROBLEMAS DE GRANDE PORTE**

Marluce da Cruz Scarabello

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Fernando Manuel Ramos, Roberto Luiz Galski, e Aline Cristina Soterroni, aprovada em 01 de setembro de 2016.

URL do documento original:

<<http://urlib.net/8JMKD3MGP3W34P/3MEGGBH>>

INPE  
São José dos Campos  
2016

## **PUBLICADO POR:**

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3208-6923/6921

Fax: (012) 3208-6919

E-mail: pubtc@inpe.br

## **COMISSÃO DO CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELECTUAL DO INPE (DE/DIR-544):**

### **Presidente:**

Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação (CPG)

### **Membros:**

Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

Dr. André de Castro Milone - Coordenação de Ciências Espaciais e Atmosféricas (CEA)

Dra. Carina de Barros Melo - Coordenação de Laboratórios Associados (CTE)

Dr. Evandro Marconi Rocco - Coordenação de Engenharia e Tecnologia Espacial (ETE)

Dr. Hermann Johann Heinrich Kux - Coordenação de Observação da Terra (OBT)

Dr. Marley Cavalcante de Lima Moscati - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Silvia Castro Marcelino - Serviço de Informação e Documentação (SID)

### **BIBLIOTECA DIGITAL:**

Dr. Gerald Jean Francis Banon

Clayton Martins Pereira - Serviço de Informação e Documentação (SID)

### **REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:**

Simone Angélica Del Ducca Barbedo - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

### **EDITORAÇÃO ELETRÔNICA:**

Marcelo de Castro Pazos - Serviço de Informação e Documentação (SID)

André Luis Dias Fernandes - Serviço de Informação e Documentação (SID)



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**

sid.inpe.br/mtc-m21b/2016/09.15.14.04-TDI

## **MÉTODO FQ - G PARA OTIMIZAÇÃO GLOBAL DE PROBLEMAS DE GRANDE PORTE**

Marluce da Cruz Scarabello

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Fernando Manuel Ramos, Roberto Luiz Galski, e Aline Cristina Soterroni, aprovada em 01 de setembro de 2016.

URL do documento original:

<<http://urlib.net/8JMKD3MGP3W34P/3MEGGBH>>

INPE  
São José dos Campos  
2016

Dados Internacionais de Catalogação na Publicação (CIP)

---

Scarabello, Marluce da Cruz.

Sc71m Método Fq - G para otimização global de problemas de grande porte / Marluce da Cruz Scarabello. – São José dos Campos : INPE, 2016.

xx + 113 p. ; (sid.inpe.br/mtc-m21b/2016/09.15.14.04-TDI)

Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2016.

Orientadores : Drs. Fernando Manuel Ramos, Roberto Luiz Galski, e Aline Cristina Soterroni.

1. Otimização global de grande porte. 2. Cálculo quântico.  
3. Vetor q-gradiente. 4. Perturbações simultâneas. 5. Método Fq-G.  
I.Título.

CDU 004.4'416

---



Esta obra foi licenciada sob uma Licença [Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada](https://creativecommons.org/licenses/by-nc/3.0/).

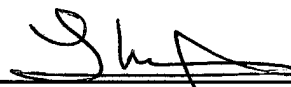
This work is licensed under a [Creative Commons Attribution-NonCommercial 3.0 Unported License](https://creativecommons.org/licenses/by-nc/3.0/).

Aluno (a): **Marluce da Cruz Scarabello**

Título: " MÉTODO Fq - G PARA OTIMIZAÇÃO GLOBAL DE PROBLEMAS DE GRANDE PORTE".

Aprovado (a) pela Banca Examinadora em cumprimento ao requisito exigido para obtenção do Título de **Doutor(a)** em **Computação Aplicada**

Dr. Ezzat Selim Chalhoub



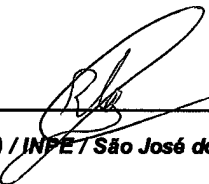
Presidente / INPE / S.JCampos - SP

Dr. Fernando Manuel Ramos



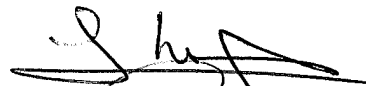
Orientador(a) / INPE / S.JCampos - SP

Dr. Roberto Luiz Galski



Orientador(a) / INPE / São José dos Campos - SP

Dra. Aline Cristina Soterroni



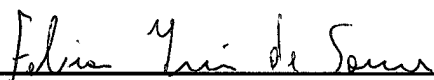
Orientador(a) / INPE / São José dos Campos - SP

Dr. Haroldo Fraga de Campos Velho



Membro da Banca / INPE / São José dos Campos - SP

Dr. Fabiano Luis de Sousa



Membro da Banca / INPE / S.JCampos - SP

Dr. Antônio Augusto Chaves



Convidado(a) / UNIFESP / São José dos Campos - SP

Dr. Luís Felipe Cesar da Rocha Bueno



Convidado(a) / UNIFESP / São José dos Campos - SP

Este trabalho foi aprovado por:

( ) maioria simples

() unanimidade



*“I’m a great believer in luck,  
and I find the harder I work  
the more I have of it.”*

COLEMAN COX





## AGRADECIMENTOS

Primeiramente, agradeço a Deus por ser o meu sustento e por permitir que eu percorresse este caminho.

Agradeço de forma especial aos meus orientadores Dr. Fernando Manuel Ramos, Dra. Aline Cristina Soterroni e Dr. Roberto Luiz Galski por todo o tempo dedicado ao meu trabalho. Obrigada por serem exemplo de pesquisador(a) e de pessoa, por todas as ideias, pelo apoio e por não desanimarem quando as coisas não iam bem.

Eu agradeço ao Instituto Nacional de Pesquisas Espaciais (INPE) pela oportunidade de prosseguir meus estudos, por toda infraestrutura e também agradeço aos professores e funcionários do programa de pós-graduação em Computação Aplicada do INPE.

A todos os amigos da CAP, especialmente aos mais próximos: Eduardo, Érica, Felipe, Heloisa, Ivana, Pedro, Vitor e Wanderson. E às amigas de sempre: Mari e Ana.

Agradeço aos membros da banca examinadora por todas as sugestões e comentários.

Agradeço imensamente a meus pais e a minha irmã pelo amor e apoio incondicional, a meu sobrinho por ser minha alegria, e a meu marido Rafael, por entender minha ausência, por ser meu maior incentivador e por todo amor.

A todos que contribuíram na elaboração deste trabalho.

E finalmente, à CAPES pelo suporte financeiro.



## RESUMO

Recentemente foram propostos métodos de otimização global baseados no conceito de  $q$ -gradiente, tais como o método  $q$ -G, o método  $q$ -GC e os métodos  $q$ -quase-Newton que são generalizações, respectivamente, dos algoritmos clássicos do método da máxima descida, método dos gradientes conjugados e métodos quase-Newton. De forma análoga aos métodos baseados em gradiente, os métodos baseados em  $q$ -gradiente, de modo geral, necessitam de pelo menos  $N + 1$  avaliações da função objetivo por iteração, onde  $N$  é a dimensão do problema a ser otimizado. Devido a esta característica, os métodos baseados em  $q$ -gradiente têm o seu desempenho deteriorado com o aumento da dimensionalidade dos problemas. Com o intuito de solucionar problemas de otimização com um grande número de variáveis de decisão, uma versão rápida do método  $q$ -G, denominada método  $Fq$ -G, é apresentada neste trabalho. Este novo método é baseado no uso de perturbações simultâneas para calcular uma aproximação do vetor  $q$ -gradiente, abordagem que exige apenas duas avaliações da função objetivo por iteração, independentemente do valor de  $N$ . Assim como nos métodos baseados em  $q$ -gradiente, no algoritmo do  $Fq$ -G o processo de busca muda gradualmente de global, no início do procedimento iterativo, para local no final. Além disso, são utilizadas perturbações gaussianas para garantir a convergência do método  $Fq$ -G para o mínimo global em um sentido probabilístico. O método  $Fq$ -G foi aplicado em 27 funções testes com 1000 variáveis de decisão propostas na competição para problemas de otimização de grande porte do *IEEE Congress on Evolutionary Computation* (CEC) de 2008 e 2010. A comparação foi realizada, no total, com 14 Algoritmos Evolutivos participantes das duas competições, e o  $Fq$ -G alcançou o primeiro ou o segundo lugar dependendo da competição e da métrica de comparação utilizada. Os resultados apontam para o potencial deste novo método na resolução de problemas de otimização de alta dimensionalidade.

Palavras-chave: otimização global de grande porte. cálculo quântico. vetor  $q$ -gradiente. perturbações simultâneas. Método  $Fq$ -G.



# THE F $q$ -G METHOD FOR LARGE SCALE GLOBAL OPTIMIZATION

## ABSTRACT

Recently, global optimization methods based on the concept of the  $q$ -gradient vector have been proposed, such as the  $q$ -G method, the  $q$ -GC method and  $q$ -versions of the quasi-Newton methods, a  $q$ -analog of the classic steepest descent, conjugate gradient and quasi-Newton algorithms, respectively. Similar to the most gradient-based optimization algorithms that use finite difference gradients, the  $q$ -gradient-based methods require at least  $N + 1$  function evaluations per iteration, where  $N$  is the dimension of the function to be optimized. This feature implies that their performance quickly deteriorates as the dimensionality of the problem increases. Here we introduce a fast variant of the  $q$ -G method. Called the F $q$ -G method, it is based on the use of simultaneous perturbations to compute an approximation of the  $q$ -gradient, an approach that requires only two function evaluations per iteration, regardless the value of  $N$ . A remarkable feature of the F $q$ -G algorithm is that its search process gradually shifts from global in the beginning to local in the end of the optimization procedure. Moreover, gaussian perturbations are used to guarantee the convergence of the F $q$ -G to the global minimum in a probabilistic sense. The F $q$ -G method was performed to 27 test functions of  $N = 1000$  variables proposed at the 2008 and 2010 IEEE Congress on Evolutionary Computation (CEC'2008 and CEC'2010) competitions on large scale global optimization. We compared the performance of the F $q$ -G method with 14 evolutionary algorithms. Our approach achieved the first or second position depending on the competition and comparison metric applied. Results show the potential of this new method for solving high-dimensional global optimization problems.

Keywords: large scale global optimization. quantum calculus.  $q$ -gradient vector. simultaneous perturbations. F $q$ -G method.



## LISTA DE FIGURAS

	<u>Pág.</u>
2.1 Exemplos de mínimos locais ( $M_1$ , $M_2$ e $M_4$ ) e mínimo global ( $M_3$ ). . . . .	6
2.2 Exemplo de função (a) convexa, (b) côncava e (c) nem convexa e nem côncava. . . . .	8
3.1 Direções de busca definidas pelo gradiente clássico (semi-reta número 1 em vermelho) e pelo $q$ -gradiente (semi-retas de 2(2') a 9(9') ) para diferentes valores do parâmetro $q$ . . . . .	20
4.1 Comparação da direção de busca do gradiente clássico com a aproximação do gradiente utilizada no SPSA com $N = 2$ . (a) Apresenta as quatro perturbações de $\mathbf{x}^{(k)}$ , $\mathbf{x}^{(k)} \pm c^{(k)}$ , usando a distribuição de probabilidade simétrica de Bernoulli $\pm 1$ ; e o vetor gradiente $\nabla f(\mathbf{x})$ . (b) Ilustra duas possíveis direções de busca $-\widehat{\nabla} f(\mathbf{x}^{(k)})$ . (c) Exibe a média das direções de busca exibidas em (b). . . . .	30
4.2 Possíveis 100 direções do vetor Fq-G para uma função quadrática bidimensional em $\mathbf{x} = (11, 11)$ obtidas para valores do parâmetro $q$ gerados a partir de uma distribuição de probabilidade gaussiana com média em $\mathbf{1}$ e desvio padrão (a) grande e (b) pequeno. O segmento em vermelho indica a direção de máxima descida da função no ponto $\mathbf{x}$ . . . . .	32
4.3 Função Elipsoidal. . . . .	35
4.4 Função Rastrigin. . . . .	36
4.5 Função Ackley. . . . .	36
4.6 Ilustração da função Rastrigin Deslocada para o caso bidimensional. . . . .	40
4.7 Número médio de avaliação da função objetivo ( $\text{NAFO}_{\text{Médio}}$ ) necessários para atingir o critério de parada pelos métodos $q$ -G e Fq-G ao solucionarem a função Rastrigin Deslocada para as dimensões $N = 10, 100$ e $1000$ . . . . .	41
5.1 Ilustração das funções teste do CEC'2008 para o caso bidimensional. . . . .	54
5.2 Gráfico de convergência para as funções teste F1 a F6 do CEC'2008 e diferentes métodos. . . . .	61
5.3 Gráfico de convergência para a função teste Rosenbrock Deslocada (F7) do CEC'2008 e diferentes métodos. . . . .	62
5.4 Gráfico de convergência para as funções teste G2, G5, G10 e G15 do CEC'2010 e diferentes métodos. . . . .	68
5.5 Gráfico de convergência para a função teste Rosenbrock e suas variações e diferentes métodos. . . . .	71





## LISTA DE TABELAS

	<u>Pág.</u>
4.1 Parâmetros usados pelos métodos $q$ -G e $Fq$ -G. . . . .	38
4.2 Desempenho comparativo, em termos de NAFO, $\overline{f(\mathbf{x}_{melhor})}$ , número médio de iterações e tempo gasto, entre os métodos $q$ -G e $Fq$ -G. . . . .	38
4.3 Desempenho comparativo, em termos de NAFO, número médio de iterações e tempo gasto, entre os métodos $q$ -G e $Fq$ -G com perturbações gaussianas. . . . .	41
5.1 CEC'2008: $f(\mathbf{x}_{melhor}) - f(\mathbf{x}^*)$ para F1 a F6, e $f(\mathbf{x}_{melhor})$ para F7, média de 25 execuções independentes, para todos os algoritmos; os melhores resultados estão em negrito; $N = 100$ . . . . .	55
5.2 CEC'2008: $f(\mathbf{x}_{melhor}) - f(\mathbf{x}^*)$ para F1 a F6, e $f(\mathbf{x}_{melhor})$ para F7, média de 25 execuções independentes, para todos os algoritmos; os melhores resultados estão em negrito; $N = 500$ . . . . .	55
5.3 CEC'2008: $f(\mathbf{x}_{melhor}) - f(\mathbf{x}^*)$ para F1 a F6, e $f(\mathbf{x}_{melhor})$ para F7, média de 25 execuções independentes, para todos os algoritmos; os melhores resultados estão em negrito; $N = 1000$ . . . . .	56
5.4 CEC'2008: Valores do erro $f(\mathbf{x}_{melhor}) - f(\mathbf{x}^*)$ para F1 a F6, e $f(\mathbf{x}_{melhor})$ para F7; $N = 100$ . . . . .	57
5.5 CEC'2008: Valores do erro $f(\mathbf{x}_{melhor}) - f(\mathbf{x}^*)$ para F1 a F6, e $f(\mathbf{x}_{melhor})$ para F7; $N = 500$ . . . . .	58
5.6 CEC'2008: Valores do erro $f(\mathbf{x}_{melhor}) - f(\mathbf{x}^*)$ para F1 a F6, e $f(\mathbf{x}_{melhor})$ para F7; $N = 1000$ . . . . .	59
5.7 CEC'2010: $f(\mathbf{x}_{melhor})$ , média de 25 execuções independentes, para todos os algoritmos e todas as funções teste; $f(\mathbf{x}^*) = 0$ é o mínimo global para G1 a G20; os melhores resultados estão em negrito. . . . .	67
5.8 CEC'2010: $f(\mathbf{x}_{melhor})$ para as funções G1 a G10 com $N = 1000$ . . . . .	69
5.9 CEC'2010: $f(\mathbf{x}_{melhor})$ para as funções G11 a G20 com $N = 1000$ . . . . .	70
5.10 CEC'2008: classificação baseada em 15 estatísticas. . . . .	72
5.11 CEC'2010: classificação baseada em 15 estatísticas. . . . .	72
5.12 CEC'2008: classificação baseada nos resultados da Tabela 5.3 . . . . .	72
5.13 CEC'2010: classificação baseada nos resultados da Tabela 5.7. . . . .	72
A.1 DEwSAcc (ZAMUDA et al., 2008); CEC'2008; $f(\mathbf{x}) - f(\mathbf{x}^*)$ para F1 a F6; $f(\mathbf{x})$ para F7; $N = 1000$ . . . . .	93
A.2 DMS-PSO (ZHAO et al., 2008); CEC'2008; $f(\mathbf{x}) - f(\mathbf{x}^*)$ para F1 a F6; $f(\mathbf{x})$ para F7; $N = 1000$ . . . . .	94

A.3	MLCC (YANG et al., 2008b); CEC'2008; $f(\mathbf{x}) - f(\mathbf{x}^*)$ para F1 a F6; $f(\mathbf{x})$ para F7; $N = 1000$ . . . . .	95
A.4	jDEdynNP-F (BREST et al., 2008); CEC'2008; $f(\mathbf{x}) - f(\mathbf{x}^*)$ para F1 a F6; $f(\mathbf{x})$ para F7; $N = 1000$ . . . . .	96
A.5	LSEDA-lg (WANG; LI, 2008); CEC'2008; $f(\mathbf{x}) - f(\mathbf{x}^*)$ para F1 a F6; $f(\mathbf{x})$ para F7; $N = 1000$ . . . . .	97
A.6	MTS (TSENG; CHEN, 2008); CEC'2008; $f(\mathbf{x}) - f(\mathbf{x}^*)$ para F1 a F6; $f(\mathbf{x})$ para F7; $N = 1000$ . . . . .	98
A.7	EPUS-PSO (HSIEH et al., 2008); CEC'2008; $f(\mathbf{x}) - f(\mathbf{x}^*)$ para F1 a F6; $f(\mathbf{x})$ para F7; $N = 1000$ . . . . .	99
A.8	jDElsgo (BREST et al., 2010); CEC'2010; $f(\mathbf{x})$ para G1 a G10. . . . .	100
A.9	jDElsgo (BREST et al., 2010); CEC'2010; $f(\mathbf{x})$ para G11 a G20. . . . .	101
A.10	DECC-DML (OMIDVAR et al., 2010); CEC'2010; $f(\mathbf{x})$ para G1 a G10. . . . .	102
A.11	DECC-DML (OMIDVAR et al., 2010); CEC'2010; $f(\mathbf{x})$ para G11 a G20. . . . .	103
A.12	DMS-PSO-SHS (ZHAO et al., 2010); CEC'2010; $f(\mathbf{x})$ para G1 a G10. . . . .	104
A.13	DMS-PSO-SHS (ZHAO et al., 2010); CEC'2010; $f(\mathbf{x})$ para G11 a G20. . . . .	105
A.14	MA-SW-Chain (MOLINA et al., 2010); CEC'2010; $f(\mathbf{x})$ para G1 a G10. . . . .	106
A.15	MA-SW-Chain (MOLINA et al., 2010); CEC'2010; $f(\mathbf{x})$ para G11 a G20. . . . .	107
A.16	DASA (KOROŠEC et al., 2010); CEC'2010; $f(\mathbf{x})$ para G1 a G10. . . . .	108
A.17	DASA (KOROŠEC et al., 2010); CEC'2010; $f(\mathbf{x})$ para G11 a G20. . . . .	109
A.18	SDENS (WANG et al., 2010); CEC'2010; $f(\mathbf{x})$ para G1 a G10. . . . .	110
A.19	SDENS (WANG et al., 2010); CEC'2010; $f(\mathbf{x})$ para G11 a G20. . . . .	111
A.20	EOEA (WANG; LI, 2010); CEC'2010; $f(\mathbf{x})$ para G1 a G10. . . . .	112
A.21	EOEA (WANG; LI, 2010); CEC'2010; $f(\mathbf{x})$ para G11 a G20. . . . .	113

## LISTA DE ABREVIATURAS E SIGLAS

CQ	–	Cálculo Quântico
$q$ -G	–	Método $q$ -Gradiente
GC	–	Método dos Gradientes Conjugados
$q$ -GC	–	Método $q$ -Gradientes Conjugados
DFP	–	Davidon-Fletcher-Powell
BFGS	–	Broyden-Fletcher-Goldfarb-Shanno
$q$ -DFP	–	Método $q$ -Davidon-Fletcher-Powell
$q$ -BFGS	–	Método $q$ -Broyden-Fletcher-Goldfarb-Shanno
AE	–	Algoritmo Evolutivo
F $q$ -G	–	Método F $q$ -G
SPSA	–	<i>Simultaneous Perturbation Stochastic Approximation</i>
CEC	–	<i>IEEE Congress on Evolutionary Computation</i>
KKT	–	Karush-Kuhn-Tucker
BB	–	Barzilai-Borwein
SPG	–	<i>Spectral Projected Gradient</i>
IPOPT	–	<i>Interior Point Optimizer</i>
SNOPT	–	<i>Sparse Nonlinear OPTimizer</i>
LANCELOT	–	<i>Large And Nonlinear Constrained Extended Lagrangian Optimization Techniques</i>
CC	–	Coevolução Cooperativa
CCGA	–	<i>Cooperative Coevolutionary Genetic Algorithm</i>
PSO	–	<i>Particle Swarm Optimizer</i>
CPSO	–	<i>Cooperative PSO</i>
CCDE	–	<i>Cooperative Coevolutionary Differential Evolution</i>
CABC	–	<i>Cooperative Artificial Bee Colony</i>
CBFO	–	<i>Cooperative Bacterial Foraging Optimization</i>
SaNSDE	–	<i>Self-adaptive neighbourhood search differential evolution</i>
MLCC	–	<i>Multilevel Cooperative Coevolution</i>
CCEA-AVP	–	<i>Cooperative Coevolutionary Algorithm with Adaptive Variable Partitioning</i>
CCVIL	–	<i>Cooperative Coevolution with Variable Interaction Learning</i>
CMA-ES	–	<i>Covariance Matrix Adaptation Evolution Strategy</i>
CC-CMA-ES	–	<i>Cooperative Coevolution Covariance Matrix Adaptation Evolution Strategy</i>
MiVD	–	<i>Min-Variance decomposition</i>
MaVD	–	<i>Max-Variance decomposition</i>
EPUS-PSO	–	<i>Efficient Population Utilization Strategy for Particle Swarm Optimizer</i>
DMS-PSO	–	<i>Dynamic Multi-Swarm Particle Swarm Optimizer</i>
ED	–	Evolução Diferencial

SDENS	–	<i>Differential Evolution Enhanced by Neighborhood Search</i>
EOEA	–	<i>Two-stage based Ensemble Optimization Evolutionary Algorithm</i>
BL	–	<i>Busca Local</i>
MTS	–	<i>Multiple Trajectory Search</i>
FDSA	–	<i>Finite Differences Stochastic Approximation</i>
NAFO	–	<i>Número de Avaliações da Função Objetivo</i>
GARS	–	<i>Generalized Adaptive Random Search</i>
DEwSAcc	–	<i>Differential Evolution with Self-adaptation and Cooperative Co-evolution</i>
UEP	–	<i>Unbiased Evolutionary Programming</i>
DMS-PSO-SHS	–	<i>Dynamic Multi-Swarm Particle Swarm Optimizer with Sub-regional Harmony Search</i>
DASA	–	<i>Differential Ant-Stigmergy Algorithm</i>

## SUMÁRIO

	<u>Pág.</u>
<b>1 Introdução</b> . . . . .	<b>1</b>
1.1 Contextualização . . . . .	1
1.2 Objetivos e Contribuições . . . . .	3
1.3 Organização da Tese . . . . .	4
<b>2 Conceitos Fundamentais de Otimização</b> . . . . .	<b>5</b>
2.1 Problemas de Otimização . . . . .	5
2.2 Condições de Otimalidade . . . . .	6
2.3 Problemas de Otimização de Grande Porte . . . . .	9
2.3.1 Métodos para otimização de problemas de grande porte . . . . .	9
<b>3 Métodos baseados no vetor <math>q</math>-gradiente</b> . . . . .	<b>17</b>
3.1 $q$ -cálculo . . . . .	17
3.1.1 $q$ -derivada . . . . .	18
3.1.2 Vetor $q$ -gradiente . . . . .	18
3.2 Método $q$ -G . . . . .	20
3.3 Métodos baseados em $q$ -gradiente com perturbações gaussianas . . . . .	22
<b>4 Método F<math>q</math>-G</b> . . . . .	<b>27</b>
4.1 Perturbações Simultâneas . . . . .	27
4.2 Vetor F $q$ -gradiente . . . . .	30
4.3 Estratégia para escolha do tamanho do passo . . . . .	33
4.4 Algoritmo do método F $q$ -G . . . . .	34
4.5 Convergência do método F $q$ -G para otimização global . . . . .	42
4.5.1 Prova de convergência . . . . .	42
<b>5 Resultados Numéricos</b> . . . . .	<b>51</b>
5.1 Configuração dos Parâmetros de Ajuste . . . . .	51
5.2 Comparações com Algoritmos Evolutivos . . . . .	51
5.2.1 Primeiro conjunto de funções teste . . . . .	52
5.2.2 Segundo conjunto de funções teste . . . . .	60
5.3 Classificação . . . . .	68
<b>6 Conclusões</b> . . . . .	<b>73</b>

<b>REFERÊNCIAS BIBLIOGRÁFICAS . . . . .</b>	<b>75</b>
<b>APÊNDICE A - Definições . . . . .</b>	<b>87</b>
<b>APÊNDICE B - Breve descrição dos Algoritmos Evolutivos que participaram do CEC'2008 e do CEC'2010. . . . .</b>	<b>89</b>
B.1 CEC'2008 . . . . .	89
B.2 CEC'2010 . . . . .	90
<b>ANEXO A - Resultados obtidos pelos Algoritmos Evolutivos que participaram do CEC'2008 e do CEC'2010. . . . .</b>	<b>93</b>

# 1 Introdução

## 1.1 Contextualização

Otimização é uma importante ferramenta na ciência da tomada de decisões. Para conseguir utilizá-la é preciso primeiro identificar algum objetivo, uma medida que possa quantificar o desempenho e a qualidade da decisão. O objetivo pode ser expresso, matematicamente, por meio de uma ou várias funções objetivo sujeitas às variáveis de decisão. São as variáveis de decisão que devem ser modificadas para que o objetivo seja otimizado. Também é necessário verificar se as variáveis possuem algum tipo de restrição (NOCEDAL; WRIGHT, 2006).

O processo de identificar o objetivo, as variáveis de decisão e as restrições do problema é chamado de modelagem. Problemas nas áreas de matemática, ciência aplicada, engenharia, economia, medicina e estatística podem ser modelados como problemas de otimização (GILL et al., 1981). Uma vez que o problema de otimização é formulado, é preciso encontrar sua solução. A resolução desses problemas consiste na busca das melhores soluções para determinados objetivos. Existem diversas técnicas destinadas a tarefa de resolver problemas de otimização. A escolha do algoritmo deve ser feita de acordo com as características do problema a ser otimizado, pois não existe nenhum método que seja ideal para solucionar todos os tipos de problemas (*No Free Lunch Theorem* (WOLPERT; MACREADY, 1997)), mas sim métodos mais ou menos indicados para determinados tipos de problemas. Dessa forma, o desenvolvimento de novos algoritmos é sempre oportuno.

Recentemente, Soterroni et al. (2012) desenvolveram um novo método de otimização denominado método do  $q$ -gradiente ou método  $q$ -G. Esse método é uma generalização do método da máxima descida e utiliza o conceito de  $q$ -derivada proveniente do  $q$ -cálculo, um dos ramos do Cálculo Quântico (CQ). O CQ também conhecido como cálculo sem limites, tem origens que remetem aos séculos XVIII e XIX, com trabalhos de Euler, Gauss e Heine (ERNST, 2012). Em vez de considerar infinitesimais e tomar o limite (para se obter uma derivada, por exemplo), o CQ se concentra no comportamento de funções ao sofrerem determinadas deformações finitas. Considere o quociente  $\frac{f(y)-f(x)}{y-x}$ . Tomando  $y = x + h$ , com  $h \neq 0$ , tem-se o  $h$ -cálculo ou o cálculo por diferenças finitas. Esse ramo do CQ foi estudado por George Boole (BOOLE, 1880; BOOLE, 1958), Milne-Thomson (MILNE-THOMSON, 1951) e outros, e é útil em várias áreas, entre elas mecânica dos fluidos e combinatória. Definindo  $y = qx$ , com  $q \neq 1$ , tem-se o  $q$ -cálculo. Foi o reverendo inglês Frank Hilton Jackson que desenvolveu o  $q$ -cálculo de forma sistemática (KAC; CHEUNG, 2002). Em 1909,

ele reintroduziu o conceito de  $q$ -derivada, também conhecida como derivada de Jackson, e propôs uma  $q$ -versão da integral, que é naturalmente o operador inverso da  $q$ -derivada (JACKSON, 1909; JACKSON, 1910a; JACKSON, 1910b). Jackson também gerou  $q$ -versões dos principais teoremas de cálculo, tais como o teorema fundamental do cálculo e séries de Taylor. Ao longo dos últimos cem anos, o  $q$ -cálculo auxiliou no desenvolvimento da teoria dos números, análise combinatória, polinômios ortogonais, funções básicas hiper-geométricas e aplicações em mecânica quântica, relatividade e mecânica estatística.

Soterroni et al. (2012) estenderam a aplicabilidade do  $q$ -cálculo na área de otimização contínua por meio da definição e do uso do vetor  $q$ -gradiente no método da máxima descida. Logo, para funções contínuas e irrestritas, o método  $q$ -G utiliza como direção de busca a mesma direção e sentido contrário ao do vetor  $q$ -gradiente. O  $q$ -gradiente, por sua vez, é o vetor das  $q$ -derivadas parciais de primeira ordem da função objetivo. O método  $q$ -G retorna para a sua versão clássica à medida que o parâmetro  $q$  tende a 1. Quando  $q \neq 1$  a direção de busca do método  $q$ -G pode ser tanto de descida quanto de subida, característica que permite ao método escapar de extremos locais. Aplicando estratégias simples para determinar o parâmetro  $q$ , necessário para o cálculo do  $q$ -gradiente, e para o tamanho do passo dado na direção de busca, tem-se um algoritmo estocástico para o método  $q$ -G destinado a problemas de otimização global. De maneira semelhante, Gouvêa (2016) e Gouvêa et al. (2016) desenvolveram uma  $q$ -versão do método dos gradientes conjugados de Fletcher e Reeves, denominado  $q$ -GC, e duas  $q$ -versões de métodos quase-Newton,  $q$ -BFGS e  $q$ -DFP, generalizações dos métodos de Broyden-Fletcher-Goldfarb-Shanno e Davidon-Fletcher-Powell, respectivamente. Os métodos baseados no vetor  $q$ -gradiente foram extensivamente comparados com Algoritmos Evolutivos (AEs) e exibiram bons resultados, sobretudo em funções multimodais identificadas pela existência de inúmeros extremos locais e com no máximo 30 dimensões. No entanto, quando aplicados a problemas de alta dimensionalidade, o desempenho desses métodos se deteriora rapidamente.

A resolução de problemas de grande porte é custosa para métodos baseados em gradiente que usam diferenças finitas. Normalmente, o cálculo do gradiente utiliza no mínimo  $N + 1$  avaliações da função objetivo para determinar a direção de busca, sendo  $N$  a dimensão do problema. E quanto maior é o valor de  $N$ , mais avaliações da função objetivo são necessárias para o cálculo dessa direção. Isto afeta diretamente o desempenho de algoritmos desse tipo, principalmente quando o número de avaliações da função objetivo é utilizado como critério de parada. Problemas desse tipo aparecem em muitas aplicações de ciência e engenharia, tais como o projeto ótimo



de sistemas eletrônicos integrados, roteamento e programação de recursos em redes complexas, reconhecimento de gene em bioinformática, mecânica estrutural, configuração molecular, e projeto de banco de dados (HAGER et al., 1994; MAHDAVI et al., 2015). Alguns fatores dificultam a resolução de problemas de grande porte. Primeiro, o espaço de busca aumenta exponencialmente com o tamanho do problema, exigindo uma estratégia de busca eficiente para explorar todas as regiões promissoras em um dado tempo. Segundo, as propriedades do espaço de busca podem mudar com o aumento do número de variáveis de decisão. Além disso, a avaliação de problemas de grande porte é usualmente custosa (OMIDVAR et al., 2015; TANG et al., 2007).

## 1.2 Objetivos e Contribuições

Dentro deste contexto, esta tese se propõe a contribuir com o desenvolvimento do  $q$ -cálculo na área da otimização, gerando uma versão eficiente ou rápida (*Fast*) do método  $q$ -G para a resolução de problemas de otimização de grande porte, chamado de método F $q$ -G.

A direção de busca desse novo algoritmo é obtida através de uma aproximação do vetor  $q$ -gradiente (denotado por  $\widehat{\nabla}_{\mathbf{q}}f$ ), desenvolvida especialmente para o método F $q$ -G, que exige apenas duas avaliações da função objetivo por iteração, independente do valor de  $N$ . Esta aproximação é inspirada no método SPSA (SPALL, 2012), sigla de *Simultaneous Perturbation Stochastic Approximation*, na medida em que se baseia no uso de perturbações simultâneas para estimar o vetor  $q$ -gradiente da função objetivo. Com a economia no número de avaliações da função objetivo realizada pelo novo método para determinar a direção de busca, espera-se que o algoritmo do F $q$ -G consiga explorar o espaço de busca de problemas de otimização com muitas variáveis de decisão.

A estratégia desenvolvida para determinar o tamanho do passo dado na direção indicada pelo vetor  $\widehat{\nabla}_{\mathbf{q}}f$  é baseada em interpolação parabólica e no valor do parâmetro  $\mathbf{q}$ . Além disso, perturbações gaussianas são utilizadas em algumas iterações para assegurar a convergência do método F $q$ -G para o mínimo global no sentido probabilístico.

O principal objetivo deste trabalho é gerar um método baseado em  $q$ -gradiente competitivo e eficiente na resolução de problemas de otimização de grande porte. Para tanto, espera-se que o método F $q$ -G apresente um bom desempenho quando aplicado em 27 funções escaláveis propostas na competição de otimização de grande porte do IEEE *Congress on Evolutionary Computation* de 2008 (CEC'2008) e de

2010 (CEC'2010) e comparado com 14 AEs participantes das duas competições.

### **1.3 Organização da Tese**

Este trabalho está estruturado em seis capítulos, sendo este o primeiro. No Capítulo 2 são apresentados os conceitos fundamentais sobre os problemas de otimização e os métodos utilizados para resolver problemas de grande porte. Os fundamentos do  $q$ -cálculo na área da otimização, além dos métodos baseados no vetor  $q$ -gradiente são descritos no Capítulo 3.

Em seguida, no Capítulo 4 é introduzido o conceito do vetor  $Fq$ -gradiente e o algoritmo do método  $Fq$ -G, juntamente com a prova de convergência do método no sentido probabilístico. A avaliação do desempenho do método  $Fq$ -G é apresentada no Capítulo 5. No último capítulo, tem-se as conclusões desta Tese, seguido de sugestões para trabalhos futuros.

## 2 Conceitos Fundamentais de Otimização

Este capítulo apresenta conceitos fundamentais sobre otimização de problemas com uma única função objetivo e exibe algumas técnicas utilizadas para encontrar o ótimo de problemas que possuem um grande número de variáveis de decisão.

Por uma questão de notação, ao longo deste texto o problema de otimização será tratado como um problema de minimização, uma vez que sempre é possível transformar um problema de maximização em um problema de minimização, e vice-versa.

### 2.1 Problemas de Otimização

A formulação matemática para um problema de otimização com apenas uma função objetivo pode ser dada por

$$\begin{aligned} \text{Minimizar: } & f(\mathbf{x}), \\ \text{Sujeito a: } & \mathbf{x} \in \mathcal{D}. \end{aligned} \tag{2.1}$$

Quando  $\mathcal{D} = \mathbb{R}^N$ , o problema de otimização é dito irrestrito. No caso de  $\mathcal{D} \subset \mathbb{R}^N$  ser descrito por restrições de desigualdade, igualdade e limites laterais sobre as variáveis de decisão, isto é,

$$\mathcal{D} = \{\mathbf{x} : g_j(\mathbf{x}) \leq 0, j = 1, \dots, J; h_k(\mathbf{x}) = 0, k = 1, \dots, K; x_{\min_i} \leq x_i \leq x_{\max_i}, i = 1, \dots, N\}, \tag{2.2}$$

diz-se que o problema de otimização é restrito e sua formulação é dada por

$$\begin{aligned} \text{Minimizar: } & f(\mathbf{x}), \\ \text{Sujeito a: } & g_j(\mathbf{x}) \leq 0, & j = 1, \dots, J, \\ & h_k(\mathbf{x}) = 0, & k = 1, \dots, K, \\ & x_{\min_i} \leq x_i \leq x_{\max_i}, & i = 1, \dots, N. \end{aligned} \tag{2.3}$$

Além de serem classificados como irrestritos ou restritos, os problemas de otimização (2.1) também podem ser classificados de acordo com a natureza da função objetivo e das funções de restrições, com número de variáveis de decisão, entre outros. Por exemplo, se todas as funções em (2.3) são lineares têm-se um problema linear, caso contrário, (2.3) é chamado de problema não linear.

Resolver o problema de otimização (2.3) é procurar pelo ponto  $\mathbf{x}^*$  que respeite todas as restrições impostas pela formulação do problema, enquanto minimiza a função objetivo  $f$ . A solução  $\mathbf{x}^*$  pode ser um mínimo local ou um mínimo global de  $f$ , de acordo com a definição abaixo.

**Definição 1.** Seja  $f : \mathcal{D} \subset \mathbb{R}^N \rightarrow \mathbb{R}$ . Se  $\mathbf{x}^* \in \mathcal{D}$  e se existe uma vizinhança em torno de  $\mathbf{x}^*$ ,  $\mathbf{V}(\mathbf{x}^*)$ , tal que  $f(\mathbf{x}^*) \leq f(\mathbf{x})$ , para cada  $\mathbf{x} \in \mathcal{D} \cap \mathbf{V}(\mathbf{x}^*)$ , então  $\mathbf{x}^*$  é chamado mínimo local. Se  $f(\mathbf{x}^*) < f(\mathbf{x})$ , para cada  $\mathbf{x} \in \mathcal{D} \cap \mathbf{V}(\mathbf{x}^*)$  com  $\mathbf{x}^* \neq \mathbf{x}$ , o ponto  $\mathbf{x}^*$  é chamado mínimo local estrito de  $f$ . Um ponto  $\mathbf{x}^* \in \mathcal{D}$  é mínimo global de  $f$ , se  $f(\mathbf{x}^*) \leq f(\mathbf{x})$ , para cada  $\mathbf{x} \in \mathcal{D}$ . Se  $f(\mathbf{x}^*) < f(\mathbf{x})$ , para cada  $\mathbf{x} \in \mathcal{D}$  com  $\mathbf{x} \neq \mathbf{x}^*$ ,  $\mathbf{x}^*$  é chamado mínimo global estrito.

A Figura 2.1 ilustra mínimos locais e o mínimo global para uma função  $f : \mathbb{R} \rightarrow \mathbb{R}$ .  $M_1$  é um mínimo local de  $f$ , enquanto  $M_2$  e  $M_4$  são mínimos locais estritos e  $M_3$  é o mínimo global da função.

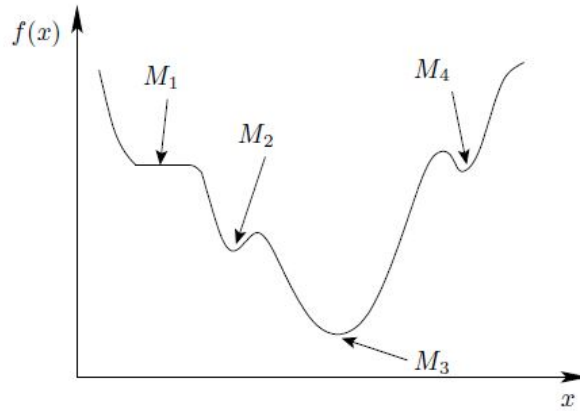


Figura 2.1 - Exemplos de mínimos locais ( $M_1$ ,  $M_2$  e  $M_4$ ) e mínimo global ( $M_3$ ).  
Fonte : (COLLETTE; SIARRY, 2003).

## 2.2 Condições de Otimalidade

Nesta seção são apresentadas condições para que o ponto  $\mathbf{x}^*$  seja um mínimo local ou global de problemas de otimização irrestritos e com restrições. As demonstrações dos Teoremas e Proposição abaixo podem ser encontradas em Bazarraa et al. (2013).

As condições de otimalidade para problemas restritos são uma extensão das condições para problemas sem restrições (BAZARAA et al., 2013).

**Proposição 1** (Condição Necessária de 1ª ordem). *Suponha que  $f : \mathbb{R}^N \rightarrow \mathbb{R}$  é diferenciável em  $\mathbf{x}^*$ . Se  $\mathbf{x}^*$  é um mínimo local, então*

$$\nabla f(\mathbf{x}^*)^T = \left[ \frac{\partial f(\mathbf{x}^*)}{\partial x_1} \quad \frac{\partial f(\mathbf{x}^*)}{\partial x_2} \quad \cdots \quad \frac{\partial f(\mathbf{x}^*)}{\partial x_N} \right] = \mathbf{0}.$$

O ponto  $\mathbf{x}^*$  que cumpre a condição  $\nabla f(\mathbf{x}^*) = \mathbf{0}$  é dito ponto crítico ou estacionário da função  $f$ . A Proposição 1 utiliza a informação do vetor gradiente, por isto é chamada condição necessária de primeira ordem. Outra condição necessária de otimalidade também pode ser determinada em termos da matriz Hessiana  $\mathbf{H}$ .

**Teorema 1** (Condição Necessária de 2ª ordem). *Suponha que  $f : \mathbb{R}^N \rightarrow \mathbb{R}$  é duas vezes diferenciável em  $\mathbf{x}^*$ . Se  $\mathbf{x}^*$  é um mínimo local, então  $\nabla f(\mathbf{x}^*) = \mathbf{0}$  e*

$$\mathbf{H}(\mathbf{x}^*) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x}^*)}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{x}^*)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x}^*)}{\partial x_1 \partial x_N} \\ \frac{\partial^2 f(\mathbf{x}^*)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\mathbf{x}^*)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(\mathbf{x}^*)}{\partial x_2 \partial x_N} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 f(\mathbf{x}^*)}{\partial x_N \partial x_1} & \frac{\partial^2 f(\mathbf{x}^*)}{\partial x_N \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x}^*)}{\partial x_N^2} \end{bmatrix}$$

é positiva semidefinida, isto é,

$$\mathbf{z}^T \mathbf{H}(\mathbf{x}^*) \mathbf{z} \geq 0,$$

para todo  $\mathbf{z} \in \mathbb{R}^N$ .

As condições dadas pela Proposição 1 e pelo Teorema 1 são requisitos que todo mínimo local deve cumprir, no entanto, um ponto pode satisfazer as condições acima e não ser um mínimo local da função. O Teorema abaixo apresenta uma condição suficiente para  $\mathbf{x}^*$  ser um mínimo local.

**Teorema 2** (Condição Suficiente de 2ª ordem). *Suponha que  $f : \mathbb{R}^N \rightarrow \mathbb{R}$  é duas vezes diferenciável em  $\mathbf{x}^*$ . Se  $\nabla f(\mathbf{x}^*) = \mathbf{0}$  e  $\mathbf{H}(\mathbf{x}^*)$  é positiva definida, isto é,*

$$\mathbf{z}^T \mathbf{H}(\mathbf{x}^*) \mathbf{z} > 0,$$

para todo  $\mathbf{z} \neq \mathbf{0} \in \mathbb{R}^N$ , então  $\mathbf{x}^*$  é um mínimo local estrito de  $f$ .

Os Teoremas e a Proposição anteriores versam sobre mínimos locais. No entanto, para problemas convexos, as condições de otimalidade garantem que o mínimo local é um mínimo global. Em particular, se  $\nabla f(\mathbf{x}^*) = \mathbf{0}$ ,  $\mathbf{H}(\mathbf{x})$  é positiva semidefinida para todo  $\mathbf{x}$ , então,  $\mathbf{x}^*$  é um mínimo global.

O conceito de convexidade é importante na área de otimização. O conjunto  $\mathcal{D} \subset \mathbb{R}^N$  é convexo se todo segmento de reta que liga dois pontos quaisquer de  $\mathcal{D}$  está completamente contido no conjunto (LUENBERGER; YE, 2008). Ou seja,

$$\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D} \text{ e } \forall \lambda \in [0, 1], \quad \text{tem-se que } \lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2 \in \mathcal{D}.$$

**Definição 2.** Seja  $f : \mathcal{D} \rightarrow \mathbb{R}$ , onde  $\mathcal{D} \subset \mathbb{R}^N$  é um conjunto convexo não vazio. A função  $f$  é convexa em  $\mathcal{D}$  se

$$f(\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1 - \lambda) f(\mathbf{x}_2), \quad \forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D} \text{ e } \lambda \in [0, 1].$$

A função  $f : \mathcal{D} \rightarrow \mathbb{R}$  é côncava em  $\mathcal{D}$  se  $-f$  é convexa em  $\mathcal{D}$ .

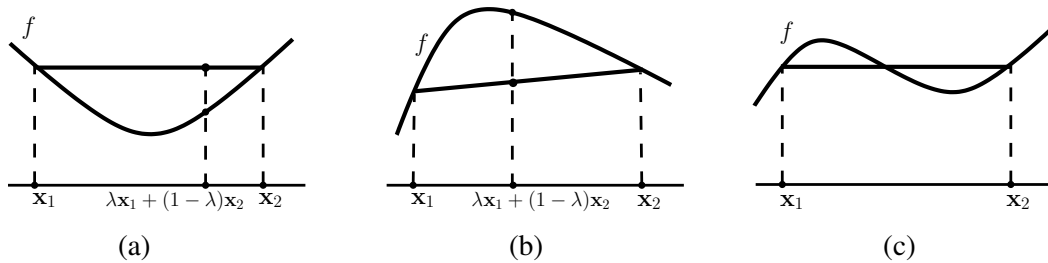


Figura 2.2 - Exemplo de função (a) convexa, (b) côncava e (c) nem convexa e nem côncava. Fonte : Bazaraa et al. (2013).

Quando o problema de otimização possui restrições laterais, de igualdade e de desigualdade (2.3), um ponto de mínimo deverá satisfazer as condições de Karush-Kuhn-Tucker (KKT). As condições de otimalidade foram desenvolvidas independentemente por Karush (1939) e por Kuhn e Tucker (1951).

**Teorema 3** (Condições Necessárias de Karush-Kuhn-Tucker (KKT)). *Sejam  $f : \mathbb{R}^N \rightarrow \mathbb{R}$ ,  $g_j : \mathbb{R}^N \rightarrow \mathbb{R}$ ,  $j = 1, \dots, J$  e  $h_k : \mathbb{R}^N \rightarrow \mathbb{R}$ ,  $k = 1, \dots, K$ . Considere o problema de minimização (2.3). Seja  $\mathbf{x}^*$  uma solução viável, e seja  $\mathcal{I} = \{j : g_j(\mathbf{x}^*) = 0\}$ . Suponha que  $f$  e  $g_j$  para  $j \in \mathcal{I}$  são diferenciáveis em  $\mathbf{x}^*$ , que cada  $g_j$  para  $j \notin \mathcal{I}$  é contínuo em  $\mathbf{x}^*$ , e que cada  $h_k$ ,  $k = 1, \dots, K$  é continuamente diferenciável em  $\mathbf{x}^*$ . Além disso, suponha que  $\nabla g_j(\mathbf{x}^*)$  para  $j \in \mathcal{I}$  e  $\nabla h_k(\mathbf{x}^*)$  para  $k = 1, \dots, K$  são linearmente independentes. Se  $\mathbf{x}^*$  resolve o problema (2.3) localmente, existe escalares únicos  $u_j$ ,  $j \in \mathcal{I}$  e  $v_k$ ,  $k = 1, \dots, K$  tais que*

$$\nabla f(\mathbf{x}^*) + \sum_{j \in \mathcal{I}} u_j \nabla g_j(\mathbf{x}^*) + \sum_{k=1}^K v_k \nabla h_k(\mathbf{x}^*) = \mathbf{0},$$

$$u_j \geq 0 \quad \text{para } j \in \mathcal{I}.$$

Ainda, se  $g_j$  para  $j \notin \mathcal{I}$  também é diferenciável em  $\mathbf{x}^*$ , as condições de KKT podem ser escritas como

$$\nabla f(\mathbf{x}^*) + \sum_{j=1}^J u_j \nabla g_j(\mathbf{x}^*) + \sum_{k=1}^K v_k \nabla h_k(\mathbf{x}^*) = \mathbf{0},$$

$$\begin{aligned} u_j g_j(\mathbf{x}^*) &= 0 \quad \text{para } j = 1, \dots, J, \\ u_j &\geq 0 \quad \text{para } j = 1, \dots, J. \end{aligned}$$

Os escalares  $u_j, v_k$  são os multiplicadores de Lagrange.

Qualquer ponto  $\mathbf{x}^*$  para qual exista os multiplicadores de Lagrange  $u_j$  e  $v_k$  tal que  $(\mathbf{x}^*, \mathbf{u}, \mathbf{v})$  satisfazem as condições de KKT é chamado ponto de KKT.

## 2.3 Problemas de Otimização de Grande Porte

Não existe um consenso sobre quantas variáveis de decisão um problema deve ter para ser considerado de grande porte. Bazarraa et al. (2013) considera que a dimensão deve ser maior do que 100, enquanto Cheng et al. (2014) e Roma (2001) definem que a função objetivo deve ter mais de 1000 variáveis. Já Benson et al. (2003) consideram que um problema é de alta dimensionalidade quando a soma do número de variáveis de decisão com o número de restrições é no mínimo 1000.

### 2.3.1 Métodos para otimização de problemas de grande porte

Nesta seção são destacados métodos destinados à resolução de problemas com um grande número de variáveis de decisão, dentre eles estão os métodos clássicos, modificações de algoritmos já existentes e também os que foram desenvolvidos especialmente para solucionar este tipo de problema.

Uma estratégia utilizada por muitos métodos de otimização para resolução de problemas de pequeno, médio e grande porte é o procedimento iterativo dado por meio da expressão

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)}, \quad (2.4)$$

em que  $\mathbf{d}^{(k)} \in \mathbb{R}^N$  é a direção de busca,  $\alpha^{(k)} \in \mathbb{R}$  é o tamanho do passo dado ao longo da direção  $\mathbf{d}^{(k)}$  e  $\mathbf{x}^{(k)} \in \mathbb{R}^N$  representa o vetor de variáveis de decisão do problema na iteração  $k$ .

Os algoritmos de otimização que utilizam o processo iterativo (2.4) se diferenciam pela maneira como a direção de busca  $\mathbf{d}^{(k)}$  e o tamanho do passo  $\alpha^{(k)}$  são determinados. O método dos gradientes conjugados (GC), utilizado para solucionar problemas de grande porte, obtêm sua direção de busca através de informações do vetor gradiente. A direção de busca na primeira iteração  $\mathbf{d}^{(0)}$  é dada pela mesma direção e sentido oposto ao do vetor gradiente no ponto  $\mathbf{x}^{(0)}$ ,  $-\nabla f(\mathbf{x}^{(0)})$ , e para as iterações seguintes, a direção é dada por

$$\mathbf{d}^{(k+1)} = -\nabla f(\mathbf{x}^{(k+1)}) + \delta^{(k)} \mathbf{d}^{(k)}, \quad (2.5)$$

onde  $\delta^{(k)}$  é um parâmetro de deflexão (BAZARAA et al., 2013). Diferentes escolhas para o parâmetro  $\delta^{(k)}$  dão origem a diversos tipos de métodos GC.

Uma das primeiras versões dos métodos GC foi desenvolvida por Hestenes e Stiefel (1952) para resolver sistemas lineares, considerando a matriz é simétrica e definida positiva. Em 1964, Fletcher e Reeves (FLETCHER; REEVES, 1964) estenderam a aplicabilidade dos métodos GC para funções não lineares, onde o tamanho do passo  $\alpha^{(k)}$  é obtido através de uma busca linear na direção  $\mathbf{d}^{(k)}$  e o parâmetro  $\delta^{(k)}$  é definido por

$$\delta_{FR}^{(k)} = \frac{\nabla f(\mathbf{x}^{(k+1)})^T \nabla f(\mathbf{x}^{(k+1)})}{\nabla f(\mathbf{x}^{(k)})^T \nabla f(\mathbf{x}^{(k)})}. \quad (2.6)$$

Entre as vantagens dos métodos GC estão os modestos requisitos de memória e sua rápida convergência. De fato, observe que com a definição de  $\delta^{(k)}$  da Equação (2.6), o método GC de Fletcher e Reeves precisa armazenar apenas três vetores  $N$ -dimensionais ( $\mathbf{d}^{(k)}$ ,  $\nabla f(\mathbf{x}^{(k)})$ ,  $\nabla f(\mathbf{x}^{(k+1)})$ ) para determinar a direção  $\mathbf{d}^{(k+1)}$ . Devido a essas características, várias versões do método dos gradientes conjugados foram propostas, considerando diferentes maneiras de determinar  $\alpha^{(k)}$  e  $\delta^{(k)}$ , para solucionar problemas de alta dimensionalidade (DAI; NI, 2003; YU et al., 2007; ANDREI, 2013; DENG; WAN, 2015).

Os métodos GC são tipicamente menos eficientes e robustos que os métodos quase-Newton. No entanto, para problemas de grande porte os quase-Newton tornam-se quase impraticáveis devido ao uso da aproximação da matriz Hessiana (BAZARAA et al., 2013). Com o intuito de tornar viável o uso dos métodos quase-Newton para otimização desse tipo de problema, Nocedal (1980) desenvolveu o L-BFGS, uma versão limitada do conhecido método quase-Newton BFGS (Broyden-Fletcher-Goldfarb-Shanno) (SHANNO, 1970). Em vez de armazenar a inversa matriz Hessiana, o L-BFGS propõe a reconstrução da matriz utilizando apenas informa-



ções das últimas  $m$  iterações, sendo  $m$  um número pequeno (NOCEDAL, 1980), o que reduz o custo computacional do método. O método L-BFGS é considerado como estado da arte dos métodos de otimização baseados em gradiente para solucionar problemas de grande porte (LOSHCHILOV, 2014).

Assim como os métodos GC e L-BFGS, o método Barzilai-Borwein (BB) (BARZILAI; BORWEIN, 1988) também utiliza a informação do gradiente para determinar a direção de busca. O método BB, que foi estendido para solucionar problemas irrestritos de grande porte por Raydan (1997), se difere do conhecido método da máxima descida pela maneira como o tamanho do passo é determinado. O método BB escolhe o tamanho do passo  $\alpha^{(k)}$  de forma que  $\alpha^{(k)}\mathbf{d}^{(k)}$  se aproxime de  $[\mathbf{H}^{(k)}]^{-1}\mathbf{d}^{(k)}$ . Essa estratégia de passo também é utilizada em outro método destinado à solução de problemas de otimização convexa de grande porte, o método *Spectral Projected Gradient* (SPG), que foi desenvolvido por Birgin et al. (2000). O SPG combina o uso do passo de Barzilai-Borwein e da estratégia de busca linear não monótona desenvolvida por Grippo et al. (1986) com método clássico do gradiente projetado.

Existem ainda alguns *softwares* baseados no método de ponto interior que são reconhecidos por solucionarem problemas não lineares com restrições e um grande número de variáveis de decisão, dentre eles estão o IPOPT (WÄCHTER; BIEGLER, 2006), o KNITRO (BYRD et al., 2006) e o LOQO (VANDERBEI; SHANNO, 1999). E também o pacote SNOPT (GILL et al., 1999) que implementa o método de Programação Quadrática Sequencial, o LANCELOT (CONN et al., 1992) e o ALGENCAN (ANDREANI et al., 2007) que implementam o método do Lagrangiano Aumentado.

Além destes métodos tradicionais, inúmeras meta-heurísticas têm sido destinadas à resolução de problemas de grande porte. Segundo (MAHDAVI et al., 2015), essas meta-heurísticas podem ser divididas em duas categorias: os métodos de coevolução cooperativa, que são baseados em estratégias de decomposição, e os algoritmos que resolvem o problema sem decomposição. Naturalmente existem outras maneiras de classificar as meta-heurísticas, veja, por exemplo, (BOUSSAÏD et al., 2013).

Os algoritmos de coevolução cooperativa (CC) consistem em dividir o problema de otimização em subproblemas com poucas variáveis de decisão ou, até mesmo, em subproblemas unidimensionais; evoluir a população em cada subproblema e, por fim, unir as soluções de todos os subproblemas para construir a solução final. A aptidão do indivíduo de um subproblema é calculada através do vetor de  $N$  dimensões construído pela integração do próprio indivíduo com indivíduos selecionados dos outros subproblemas (MAHDAVI et al., 2015).

Os primeiros métodos que utilizaram a estratégia de decomposição CC foram propostos por Potter e Jong (1994). Desenvolvidos para solucionar problemas complexos, os métodos CCGA-1 e CCGA-2 utilizam o algoritmo genético para evoluir a população de cada subproblema. Cada população evolui independentemente, interagindo apenas para obter o valor de aptidão dos indivíduos.

De forma semelhante à Potter e Jong (1994), Bergh e Engelbrecht (2004) integraram o método CC ao otimizador PSO, sigla de *Particle Swarm Optimizer*. O CPSO- $H_k$  e o CPSO- $S_k$  foram comparados ao PSO, ao algoritmo genético e ao CCGA-1, apresentando desempenho superior aos desses métodos. A principal diferença entre as duas versões do CPSO e as duas versões do CCGA, além do método evolutivo, é como a divisão das variáveis é realizada. Enquanto nos métodos CCGA o problema de  $N$  dimensões é dividido em  $N$  subproblemas unidimensionais, nos CPSO o vetor de  $N$  dimensões é decomposto em  $p$  partes, resultando em  $p$  subproblemas.

Vários autores utilizaram a estratégia de decomposição com outros métodos evolutivos, como por exemplo *Cooperative Co-evolutionary Differential Evolution* (CCDE) (SHI et al., 2005), *Cooperative Artificial Bee Colony* (CABC) (EL-ABD, 2010), *Cooperative Bacterial Foraging Optimization* (CBFO) (CHEN et al., 2008), entre outros.

Nas meta-heurísticas citadas acima, a divisão em subproblemas é sempre realizada no começo do processo de busca e as variáveis permanecem no mesmo grupo até o fim do processo. Por esta razão, são denominados métodos baseados em CC com agrupamento estático. Os resultados obtidos por esses métodos mostraram um desempenho fraco em problemas cujas funções objetivo são não separáveis. Uma função de  $N$  variáveis é não separável se não pode ser reescrita como a soma de  $N$  funções de apenas uma variável (TANG et al., 2009).

Com intuito de determinar o mínimo de funções não separáveis de maneira eficiente, foram desenvolvidos métodos cujas divisões em subproblemas são realizadas dinamicamente. Assim, as variáveis podem ser reagrupadas de acordo com a evolução do algoritmo. Classificadas como métodos baseados em CC com agrupamento dinâmico, essas meta-heurísticas também podem ser categorizadas de acordo com a forma que as variáveis são agrupadas, aleatoriamente ou baseada em aprendizagem (MAHDAVI et al., 2015).

Proposto por Yang et al. (2008a), o DECC-G é um dos métodos baseados em CC que tem suas variáveis agrupadas dinamicamente e aleatoriamente. Nesse algoritmo, a cada novo ciclo as variáveis do vetor objetivo  $N$ -dimensional são divididas aleatoriamente

em grupos de tamanhos pré-definidos ( $s$ ), formando  $p$  subproblemas (assumindo  $N = p \cdot s$ ). Todos os subproblemas são evoluídos através do algoritmo de evolução diferencial autoadaptativo SaNSDE (YANG et al., 2008c). Uma estratégia de ponderação é aplicada em cada um dos subproblemas e os pesos de cada subproblema são otimizados. O DECC-G apresentou desempenho superior ao de dois métodos que utilizam a estrutura original dos algoritmos de CC, principalmente para as funções não separáveis.

Como a decomposição dinâmica permite a interação entre as variáveis de decisão, o bom desempenho do DECC-G na resolução de funções não separáveis é esperado. Entretanto, determinar o tamanho apropriado de cada subproblema não é uma tarefa fácil. Subproblemas com poucas variáveis, podem ser vantajosos para funções separáveis e subproblemas com tamanhos grandes são interessantes para problemas não separáveis, pois permitem uma maior interação entre as variáveis. No início do processo de busca, subproblemas com poucas variáveis de decisão podem ajudar a localizar rapidamente regiões promissoras, porém no final da busca, grupos de tamanho grande fornecem informações globais (YANG et al., 2008b).

A fim de fornecer um método que conseguisse adaptar o tamanho apropriado de cada subproblema, dependendo da função objetivo, estágio de evolução e características de recurso, Yang et al. (2008b) propuseram o MLCC (*Multilevel Cooperative Coevolution*). O MLCC gera, de maneira aleatória, um conjunto com variados tamanhos de subproblemas ( $s_i$ 's), onde cada tamanho implica em diferentes níveis de interação entre as variáveis de decisão. No início de cada ciclo, um tamanho de subproblema é selecionado no conjunto com base nos seus níveis de desempenho. Depois, o vetor objetivo é decomposto de acordo com o tamanho selecionado e então cada subproblema é evoluído. No final de cada ciclo, o desempenho do tamanho de grupo selecionado é atualizado e salvo. Com este mecanismo, o MLCC é capaz de se autoadaptar ao nível da interação adequada.

Além dos métodos de CC com agrupamento dinâmico aleatório, existem os que são baseados em aprendizagem. Essas meta-heurísticas utilizam estratégias que permitem identificar as interações entre as variáveis durante o processo de otimização (MAHDAVI et al., 2015).

O método CCEA-AVP (RAY; YAO, 2009) é uma das meta-heurísticas de CC com agrupamento dinâmico baseado em aprendizagem. Nas primeiras  $M$  gerações do algoritmo, todas as variáveis de decisão pertencem a um mesmo grupo e evoluem com um AE. Na  $(M + 1)$ -ésima geração, uma matriz de correlação é calculada

baseada nos 50% melhores indivíduos e as variáveis são particionadas em múltiplos grupos da seguinte forma: as variáveis que têm um coeficiente de correlação maior do que um limiar são agrupadas juntas, respeitando o tamanho máximo de cada grupo. O particionamento das variáveis baseadas em correlação é repetido a cada geração subsequente. Outra meta-heurística deste tipo foi apresentada por [Chen et al. \(2010\)](#), o CCVIL inicialmente considera todas as variáveis de forma independente e coloca cada uma delas em um grupo separado. Iterativamente, o método descobre a relação entre as variáveis e as agrupa de acordo com estas relações.

O método CC-CMA-ES ([LIU; TANG, 2013](#)) aplica a estratégia de CC ao eficiente CMA-ES ([HANSEN et al., 2003](#)) com o propósito de solucionar problemas de grande porte. Com o objetivo de fazer um balanço entre busca global e local, esse método utiliza três diferentes estratégias de decomposição: MiVD (*Min-Variance decomposition*), MaVD (*Max-Variance decomposition*) e decomposição aleatória.

Existem ainda as meta-heurísticas que não fazem uso de qualquer tipo de decomposição. Esses métodos utilizam versões melhoradas de operadores clássicos, aqueles usados para problemas de baixa dimensionalidade, e desenvolvem novas estratégias de busca.

O EPUS-PSO foi desenvolvido por [Hsieh et al. \(2008\)](#) para solucionar o problema de grande porte sem utilizar decomposições. O algoritmo possui um gerenciador da população responsável pela eliminação de partículas redundantes, entrada de novas partículas ou manutenção do número de partículas na população de acordo com o estado da solução na busca para tornar o processo mais eficiente. No DMS-PSO ([ZHAO et al., 2008](#)), um outro método baseado no PSO, a população é dividida em um grande número de subenxames; esses subenxames são frequentemente reagrupados e informações são trocadas entre as partículas em todo o enxame.

Além de versões do PSO para gerar otimizadores de problemas de grande porte, o algoritmo de Evolução Diferencial (ED) também se tornou muito popular para gerar versões que resolvem este tipo de problema devido a sua simplicidade e fácil implementação ([MAHDAVI et al., 2015](#)). O jDEdynNP-F ([BREST et al., 2008](#)) utiliza mecanismos de autoadaptação para determinar os parâmetros de controle próprios do ED, além de estratégias para reduzir o tamanho da população durante o processo de otimização. Uma versão estendida do jDEdynNP-F, jDElsgo, pode ser encontrada em [Brest et al. \(2010\)](#). O SDENS ([WANG et al., 2010](#)) é um algoritmo sequencial DE reforçado com busca na vizinhança. Já o método JADE ([ZHANG; SANDERSON, 2009](#)), além de atualizar os parâmetros de controle do ED de maneira adaptativa, adota

uma estratégia de mutação que utiliza um arquivo externo opcional. Essas operações diversificam a população e melhoram o desempenho.

O EOEa (WANG; LI, 2010) é uma outra alternativa para resolver problemas de alta dimensionalidade e o seu procedimento de busca é dividido em duas etapas: a fase de encolhimento global, que procura diminuir o espaço de busca a procura de regiões promissoras o mais rápido possível; e a etapa de exploração local, que tem como objetivo explorar intensamente a área que foi limitada na primeira fase para encontrar a melhor solução possível.

O MA-SW-Chain (MOLINA et al., 2010) é um algoritmo memético que utiliza o método de Busca Local (BL) *Solis Wets* (SOLIS; WETS, 1981). O método foi projetado para adaptar a intensidade do BL, com o propósito de explorar mais a região dos indivíduos mais promissores. Métodos de BL também são utilizados no algoritmo MTS (TSENG; CHEN, 2008). MTS usa múltiplos agentes para encontrar as melhores soluções dentro do espaço de busca. Cada agente utiliza um dos três métodos de BL disponíveis. O BL escolhido é o que melhor se ajusta ao formato da função na vizinhança de uma solução.

Esses são alguns dos inúmeros otimizadores de problemas de grande porte, dentre eles há algoritmos que utilizam a informação do vetor gradiente, algoritmos bioinspirados, algoritmos populacionais, mas não há nenhum na área do  $q$ -cálculo. Os únicos métodos de otimização na área do  $q$ -cálculo são os que foram desenvolvidos por Soterroni et al. (2011) e Gouvêa et al. (2016), que serão descritos no próximo capítulo, destinados à solução de problemas com até 30 dimensões.



### 3 Métodos baseados no vetor $q$ -gradiente

O método  $q$ -G foi o primeiro a utilizar conceitos do  $q$ -cálculo na área de otimização. A partir da definição da  $q$ -derivada, [Soterroni et al. \(2011\)](#) definiu o vetor  $q$ -gradiente e utilizou-o para determinar a direção de busca do método. Esse método foi destinado à resolução de problemas de otimização global e demonstrou eficiência e competitividade quando comparado aos AEs.

Modificações no método  $q$ -G foram realizadas por [Gouvêa et al. \(2016\)](#) para obter um algoritmo com prova de convergência no sentido probabilístico. Além disso, foram geradas  $q$ -versões convergentes do método dos gradientes conjugados,  $q$ -GC, e de dois métodos quasi-Newton,  $q$ -BFGS e  $q$ -DFP. Essas  $q$ -versões convergentes apresentaram bom desempenho quando aplicadas em funções multimodais.

Uma introdução sobre o  $q$ -cálculo, características do método  $q$ -G original e das  $q$ -versões convergentes de métodos clássicos são apresentados neste capítulo.

#### 3.1 $q$ -cálculo

O Cálculo Quântico (CQ) é a variação do cálculo que trabalha sem o conceito de limite, baseado principalmente na ideia de reescalonamento das diferenças finitas ([AHMAD et al., 2016](#)). Considere a expressão

$$\frac{f(y) - f(x)}{y - x},$$

quando  $y$  se aproxima de  $x$ , o limite, quando existe, é dado pela conhecida definição de derivada de uma função  $f$  no ponto  $x$ . No entanto, sem tomar o limite e fazendo  $y = qx$  e  $y = x + h$ , onde  $q$  é um número fixo diferente de 1 e  $h$  é um número fixo diferente de 0, tem-se, respectivamente, expressões do  $q$ -cálculo e do  $h$ -cálculo ([KAC; CHEUNG, 2002](#)).

O  $q$ -cálculo utiliza o parâmetro multiplicativo  $q$  para generalizar expressões matemáticas. As versões análogas das expressões matemáticas, também conhecidas por  $q$ -versões, retornam a sua forma original, quando  $q$  tende a 1.

A história do  $q$ -cálculo tem origem no século XVIII, quando Euler introduziu o parâmetro  $q$  nas séries infinitas de Newton ([ERNST, 2012](#)). Identidades de Euler para funções  $q$ -exponenciais e fórmula  $q$ -binomiais foram desenvolvidas por Gauss no século XIX. Essas definições levaram a descoberta da fórmula de Heine para a função  $q$ -hipergeométrica como uma generalização para uma série hipergeométrica

(AHMAD et al., 2016). Mas foi no começo do século XX que F. H. Jackson desenvolveu o  $q$ -cálculo de uma maneira sistemática. Ele reintroduziu o operador  $q$ -derivada, amplamente conhecido como derivada de Jackson e definiu a  $q$ -integral (JACKSON, 1909; JACKSON, 1910a; JACKSON, 1910b).

Nos últimos anos, o  $q$ -cálculo tem sido consideravelmente explorado, tendo em conta suas numerosas aplicações em funções especiais, análise combinatória, mecânica quântica, teoria dos operadores, cálculo  $q$ -variacional, relatividade, entre outros (AHMAD et al., 2016).

### 3.1.1 $q$ -derivada

Dada uma função diferenciável de uma única variável  $f(x)$  e  $q \in \mathbb{R}$ , a  $q$ -derivada de  $f(x)$ , também conhecida como derivada de Jackson, pode ser definida como (KOEKOEK; KOEKOEK, 1993):

$$D_q f(x) = \frac{f(qx) - f(x)}{qx - x} \quad (3.1)$$

para  $q \neq 1$  e  $x \neq 0$ . Quando  $q = 1$  ou  $x = 0$ , a  $q$ -derivada retorna à derivada clássica. Observe que na  $q$ -derivada (3.1) a variável independente sofre deformações, ou seja,  $x$  pode ser dilatada ou contraída dependendo do valor de  $q$ .

Para uma função diferenciável de  $N$  variáveis  $f(\mathbf{x})$ , a  $q$ -derivada de primeira ordem de  $f$  com relação à variável  $x_i$  é dada por (SOTERRONI et al., 2011) :

$$D_{q,x_i} f(\mathbf{x}) = \begin{cases} \frac{f(x_1, \dots, qx_i, \dots, x_N) - f(x_1, \dots, x_i, \dots, x_N)}{qx_i - x_i}, & \text{se } q \neq 1 \text{ e } x_i \neq 0, \\ \frac{\partial f(\mathbf{x})}{\partial x_i}, & \text{caso contrário,} \end{cases}$$

$$= \begin{cases} \frac{f(\mathbf{x} + (q-1)x_i \mathbf{e}^{(i)}) - f(\mathbf{x})}{(q-1)x_i}, & \text{se } q \neq 1 \text{ e } x_i \neq 0, \\ \frac{\partial f(\mathbf{x})}{\partial x_i}, & \text{caso contrário,} \end{cases}$$

onde  $\mathbf{e}^{(i)}$  é a  $i$ -ésima coluna da matriz identidade  $\mathbf{I}_N$ .

### 3.1.2 Vetor $q$ -gradiente

O vetor  $q$ -gradiente é definido como o vetor das  $N$   $q$ -derivadas parciais de primeira ordem de  $f$  (SOTERRONI et al., 2011):

$$\nabla_{\mathbf{q}} f(\mathbf{x}) = [D_{q_1, x_1} f(\mathbf{x}) \dots D_{q_i, x_i} f(\mathbf{x}) \dots D_{q_N, x_N} f(\mathbf{x})]^T, \quad (3.2)$$



onde o parâmetro  $q$  é um vetor  $\mathbf{q} = (q_1, \dots, q_i, \dots, q_N) \in \mathbb{R}^N$ . Assim como acontece com a  $q$ -derivada, o vetor  $q$ -gradiente retorna ao gradiente clássico no limite  $q_i \rightarrow 1$ , para todo  $i = 1, \dots, N$ .

É intuitivo considerar a direção determinada pelo vetor gradiente da função objetivo no ponto atual da busca como direção de busca de um método de otimização, uma vez que esse vetor fornece a direção e sentido de máximo crescimento da função no ponto de busca e, naturalmente, a mesma direção e sentido contrário determinam o máximo decréscimo da função. Logo, para os métodos de otimização baseados em  $q$ -gradiente, deve-se utilizar o vetor  $q$ -gradiente para determinar a direção de busca (SOTERRONI, 2012).

Na Figura 3.1 está representado a mesma direção e sentido contrário ao vetor gradiente clássico, ou seja, a direção de máxima descida (semi-reta número 1 em vermelho), e a mesma direção e sentido contrário ao vetor  $q$ -gradiente (semi-retas de 2(2') a 10(10')) para diferentes valores do parâmetro  $\mathbf{q}$ , juntamente com as curvas de nível de uma função bidimensional ( $N = 2$ )  $f : \mathbb{R}^N \rightarrow \mathbb{R}$  dada por<sup>1</sup>

$$f(\mathbf{x}) = \sum_{i=1}^N (x_i - o_i)^2, \quad (3.3)$$

com  $\mathbf{o} = (10, 10)$  e  $\mathbf{x} = (12, 12)$ . Para traçar as semi-retas numeradas de 2 a 9 na Figura 3.1, foram tomados  $q_1$  e  $q_2$  simétricos em relação a 1 ( $\mathbf{q} = (q_1, q_2)$ ) e, respectivamente, iguais a  $(0.97, 1.03)$ ,  $(0.9, 1.1)$ ,  $(0.8, 1.2)$ ,  $(0.7, 1.3)$ ,  $(0.6, 1.4)$ ,  $(0.5, 1.5)$ ,  $(-1.2, -0.8)$  e  $(-1.4, -0.6)$ . As semi-retas numeradas de 2' a 9' usam os mesmos valores de  $q_1$  e  $q_2$ , mas com posições invertidas  $\mathbf{q} = (q_2, q_1)$ . Conforme os valores de  $q_i$  se aproximam de 1 as direções determinadas pelo  $q$ -gradiente se aproximam da direção de máxima descida, como demonstram as semi-retas de 2(2') a 7(7'). Enquanto que as semi-retas 8(8') e 9(9') mostram que a direção e sentido contrário ao vetor  $q$ -gradiente pode definir uma direção de subida.

Esse exemplo é capaz de mostrar que a utilização do vetor  $q$ -gradiente (3.2) para determinar a direção de busca em métodos de otimização oferece mecanismos para escapar de mínimos locais.

---

<sup>1</sup>Função quadrática  $f(\mathbf{x}) = \sum_{i=1}^N x_i^2$  deslocada e com mínimo global em  $\mathbf{o} = (0, 0)$ .

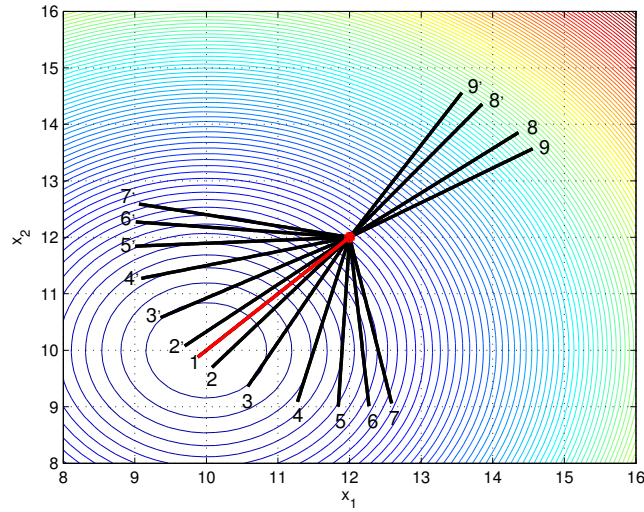


Figura 3.1 - Direções de busca definidas pelo gradiente clássico (semi-reta número 1 em vermelho) e pelo  $q$ -gradiente (semi-retas de 2(2') a 9(9')) para diferentes valores do parâmetro  $q$ .

Fonte : Soterroni et al. (2015).

### 3.2 Método $q$ -G

O método  $q$ -G (SOTERRONI et al., 2012) é uma  $q$ -versão do método da máxima descida desenvolvida para solucionar problemas de otimização global, contínuos e irrestritos. Esse método utiliza o procedimento iterativo (2.4), em que a direção de busca  $\mathbf{d}^{(k)} \in \mathbb{R}^N$  na iteração  $k$  é dada por

$$\mathbf{d}^{(k)} = -\frac{\nabla_{\mathbf{q}}f(\mathbf{x}^{(k)})}{\|\nabla_{\mathbf{q}}f(\mathbf{x}^{(k)})\|},$$

onde  $\nabla_{\mathbf{q}}f(\mathbf{x}^{(k)})$  é o vetor  $q$ -gradiente de  $f$  no ponto  $\mathbf{x}^{(k)}$  dado por (3.2).

Como observado na Figura 3.1 o uso do vetor  $q$ -gradiente para determinar a direção de busca permite que os métodos baseados em  $q$ -gradiente escapem de mínimos locais e, para isso, uma boa escolha do parâmetro  $q$  é fundamental.

Em Soterroni et al. (2012) os valores do parâmetro  $\mathbf{q}$  são obtidos por meio do sorteio de  $q_i^{(k)}$  ( $i = 1, \dots, N$ ) segundo uma distribuição gaussiana com função de densidade de probabilidade dada por

$$f(q_i, \sigma, \mu) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(q_i - \mu)^2}{2\sigma^2}\right),$$

com média  $\mu = 1$  e desvio padrão  $\sigma = \sigma^{(k)}/|x_i^{(k)}|$ , quando  $x_i^{(k)} \neq 0$ . O desvio padrão inicial é diferente de zero ( $\sigma^{(0)} > 0$ ) e tende a zero ao longo do procedimento iterativo por meio da expressão

$$\sigma^{(k+1)} = \beta \cdot \sigma^{(k)},$$

em que  $\beta \in (0, 1)$  é o fator de redução. Quando  $\sigma^{(k)} \neq 0$  (e não próximo de zero), os valores  $q_i^{(k)}$  podem ser quaisquer números reais e, associados aos valores da função objetivo, permitem que a direção de busca seja tanto de subida quanto de descida. Esta propriedade confere ao método  $q$ -G a capacidade de realizar uma busca global (*exploration*). Quando  $\sigma^{(k)} \rightarrow 0$ , tem-se  $q_i^{(k)} \rightarrow 1, \forall i, i = 1, 2, \dots, N$ , dessa forma, direções de busca de descida são obtidas com maior probabilidade e o método  $q$ -G realiza uma busca local (*exploitation*). E quando  $q_i^{(k)}$  é numericamente igual a 1, a direção de busca é dada exatamente pela direção de máxima descida. Assim, a influência que o desvio padrão  $\sigma^{(k)}$  tem sobre o comportamento do método  $q$ -G pode ser comparada à influência da temperatura no Recozimento Simulado (KIRKPATRICK et al., 1983), pois valores altos de  $\sigma^{(k)}$  implicam em maior probabilidade de geração de direções de busca de subida, ou seja, probabilidade de aceitar soluções piores é maior. Enquanto que para valores baixos de  $\sigma^{(k)}$ , a probabilidade de geração de direções de busca de descida é maior, isto é, a probabilidade de aceitar soluções piores é menor (SOTERRONI et al., 2011; SOTERRONI, 2012).

Uma vez definida a direção de busca é necessário determinar o tamanho do passo a ser dado nesta direção. Estratégias para gerar o tamanho do passo foram testadas por Soterroni (2012) e colaboradores, e a que apresentou melhores resultados foi a “Controle do passo por redução”. Essa estratégia consiste no uso de uma sequência decrescente de valores do comprimento do passo  $\alpha^{(k)}$ , partindo de um valor inicial  $\alpha^{(0)} > 0$ , que é reduzido ao longo do procedimento iterativo por meio da equação

$$\alpha^{(k+1)} = \beta \cdot \alpha^{(k)},$$

em que  $\beta \in (0, 1)$  é o fator de redução. Por simplicidade, o fator de redução  $\beta$  é o mesmo utilizado no decrescimento do desvio padrão  $\sigma^{(k)}$ .

O algoritmo do método  $q$ -G para funções diferenciáveis  $f : \mathbb{R}^N \rightarrow \mathbb{R}$  é apresentado a seguir. Os critérios de parada do algoritmo podem ser o número máximo de avaliações da função objetivo ou a acurácia desejada. O ponto inicial  $\mathbf{x}^{(0)} \in \mathbb{R}^N$  pode ser obtido por meio de sorteio uniforme no espaço de busca. Note que há apenas três parâmetros de ajuste  $\sigma^{(0)}$ ,  $\alpha^{(0)}$  e  $\beta$ . O desvio padrão inicial  $\sigma^{(0)}$  determina o quanto a busca é local ou global e o fator de redução  $\beta$  controla a velocidade desta

transição.

---

**Algoritmo 1:** Método  $q$ -G para problemas de otimização irrestritos (SOTERRONI et al., 2013)

---

**Entradas:**

- (1)  $f : \mathbb{R}^N \rightarrow \mathbb{R}$  contínua e diferenciável
- (2)  $\mathbf{x}^{(0)} \in \mathbb{R}^N$  (ponto inicial)
- (3)  $\sigma^{(0)}$  (desvio padrão inicial da distribuição gaussiana para gerar o parâmetro  $\mathbf{q}$ )
- (4)  $\alpha^{(0)}$  (tamanho do passo inicial)
- (5)  $0 < \beta < 1$  (fator que decresce o tamanho inicial do passo e o desvio padrão de uma distribuição gaussiana que gera o parâmetro  $\mathbf{q}$ )

**Passo 0:** Faça  $k = 0$  e  $\mathbf{x}_{melhor} = \mathbf{x}^{(0)}$

**Passo 1:** Para  $i = 1$  até  $N$ , se  $x_i^{(k)} \neq 0$ , então sorteie  $q_i^{(k)}$  a partir de uma distribuição gaussiana com média 1 e desvio padrão  $\sigma^{(k)} / |x_i^{(k)}|$ ; senão, faça  $q_i^{(k)} = 1$

**Passo 2:**  $\mathbf{d}^{(k)} = -\nabla_{\mathbf{q}} f(\mathbf{x}^{(k)}) / \|\nabla_{\mathbf{q}} f(\mathbf{x}^{(k)})\|$

**Passo 3:**  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)}$

**Passo 4:** Se  $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}_{melhor})$  então  $\mathbf{x}_{melhor} = \mathbf{x}^{(k+1)}$

**Passo 5:** Faça  $\sigma^{(k+1)} = \beta \cdot \sigma^{(k)}$  e  $\alpha^{(k+1)} = \beta \cdot \alpha^{(k)}$

**Passo 6:** Se o critério de parada é atingido, retorne  $\mathbf{x}_{melhor}$  e  $f(\mathbf{x}_{melhor})$ , senão incremente  $k = k + 1$ , e vá para o Passo 1.

---

O método  $q$ -G se mostrou competitivo em comparação com 11 AEs participantes da competição do *IEEE Congress on Evolutionary Computation* (CEC) 2005, sobretudo para funções multimodais de 10 e 30 dimensões. Mais detalhes sobre o método  $q$ -G podem ser obtidos em Soterroni et al. (2011), Soterroni et al. (2012), Soterroni et al. (2013), Soterroni et al. (2015).

### 3.3 Métodos baseados em $q$ -gradiente com perturbações gaussianas

Gouvêa et al. (2016) realizaram duas modificações no método  $q$ -G (Algoritmo 1) necessárias para garantir a convergência do método para o mínimo global no sentido probabilístico. A primeira delas foi a inserção de restrições laterais nas variáveis de decisão para que o espaço de busca seja limitado. A segunda modificação foi a inclusão de perturbações gaussianas em algumas iterações do algoritmo, permitindo assim que qualquer ponto do espaço de busca possa ser potencialmente alcançado.

O Algoritmo 2 apresenta a versão convergente do método  $q$ -G. Seja  $\mathcal{D}$  um conjunto compacto, uma função determinística  $\rho_{\mathcal{D}} : \mathbb{R}^N \rightarrow \mathcal{D}$ , tal que  $\rho_{\mathcal{D}}(\mathbf{x}) = \mathbf{x}$  para todo  $\mathbf{x} \in \mathcal{D}$ , é inserida para garantir que todos os pontos de busca estejam dentro do espaço de busca limitado. Essa função é denominada transformação de

absorção e pode ser definida de diversas formas. Se  $\mathcal{D}$  é um conjunto compacto e convexo,  $\rho_{\mathcal{D}}(\mathbf{x})$  pode ser tomado como o ponto de  $\mathcal{D}$  mais próximo de  $\mathbf{x}$ , ou seja,  $\|\mathbf{x} - \rho_{\mathcal{D}}(\mathbf{x})\| = \inf_{\mathbf{y} \in \mathcal{D}} \|\mathbf{x} - \mathbf{y}\|$ . No caso em que  $\mathcal{D} = [\mathbf{a}, \mathbf{b}] \in \mathbb{R}^N$  é um hipercubo,  $\rho_{\mathcal{D}}$  pode ser tomado como o ponto da fronteira da seguinte maneira  $\rho_{\mathcal{D}}(\mathbf{x}) = \min(\max(\mathbf{a}, \mathbf{x}), \mathbf{b})$  (REGIS, 2010). Para os métodos baseados em  $q$ -gradiente, a transformação de absorção  $\rho_{\mathcal{D}}$  utiliza o critério de reflexão sobre a fronteira de  $\mathcal{D}$  para cada componente da variável de decisão  $\mathbf{x}$  (GOUVÊA et al., 2016).

Devido a inclusão de perturbações gaussianas, quatro parâmetros de ajuste são acrescentados ao método:  $m$ ,  $r$ ,  $\theta^{(k)}$  e  $\theta_{\min}$ . As iterações gaussianas ocorrem a cada  $m$  iterações. São sorteados  $r$  vetores  $\mathbf{z}$  segundo uma distribuição gaussiana com média  $\mathbf{0}$  e matriz de covariância  $(\theta^{(k)})^2 \mathbf{I}_N$ . O ponto atual da busca  $\mathbf{x}$  é perturbado  $r$  vezes pelos vetores  $\mathbf{z}$ . O novo ponto da busca é definido como sendo o melhor entre o ponto atual e os  $r$  pontos perturbados. O desvio padrão  $\theta^{(k)}$  é reduzido gradativamente toda vez que a iteração gaussiana não melhora o ponto atual da busca. O menor valor que  $\theta^{(k)}$  pode assumir é fixado em  $\theta_{\min} > 0$ .

Além do método  $q$ -G com perturbações gaussianas, Gouvêa et al. (2016) desenvolveram uma  $q$ -versão do método dos gradientes conjugados, baseada na versão dos métodos dos gradientes conjugados introduzida por Fletcher e Reeves na década de 60 (FLETCHER; REEVES, 1964). No método  $q$ -GC a primeira direção de busca é dada pela mesma direção e sentido contrário ao vetor  $q$ -gradiente e as outras direções são combinações lineares desse vetor no ponto atual da busca com as direções anteriores. Generalizações dos métodos quase-Newton BFGS e DFP (Davidon-Fletcher-Powell) também foram desenvolvidas em (GOUVÊA, 2016). Assim como no método  $q$ -G, nos métodos  $q$ -GC,  $q$ -BFGS e  $q$ -DFP o  $q$  é um parâmetro chave e, à medida que  $q$  tende a 1, a  $q$ -versão do método dos gradientes conjugados e as  $q$ -versões dos métodos quase-Newton retomam suas respectivas versões clássicas.

Da mesma forma que Soterroni et al. (2012), Gouvêa et al. (2016) e Gouvêa (2016) consideraram para a geração do parâmetro  $\mathbf{q}$  o sorteio dos  $q_i$  de acordo com uma distribuição gaussiana centrada em 1 com desvio padrão  $\sigma/|x_i|$ ,  $x_i \neq 0$ , e a estratégia para determinar o tamanho do passo “Controle do passo por redução”.

Os métodos  $q$ -G,  $q$ -GC,  $q$ -BFGS e  $q$ -DFP com perturbações gaussianas foram comparados com as suas versões clássicas e com 5 métodos amplamente utilizados na literatura de otimização. As comparações foram realizadas para 27 funções testes de 10 dimensões. Os resultados mostraram que os métodos baseados no vetor  $q$ -gradiente são competitivos e promissores, principalmente quando aplicados em problemas de

---

**Algoritmo 2:** Método  $q$ -G com perturbações gaussianas para otimização global e com restrições laterais (GOUVÊA et al., 2016)

---

**Entradas**

- (1)  $f : \mathcal{D} \rightarrow \mathbb{R}$ , onde  $\mathcal{D} = [\mathbf{x}_{\min}, \mathbf{x}_{\max}] \subseteq \mathbb{R}^N$  (função objetivo)
- (2)  $\mathbf{x}^{(0)} \in \mathcal{D}$  (ponto inicial)
- (3)  $\rho_{\mathcal{D}} : \mathbb{R}^N \rightarrow \mathcal{D}$  tal que  $\rho_{\mathcal{D}}(\mathbf{x}) = \mathbf{x}$  para todo  $\mathbf{x} \in \mathcal{D}$  (transformação de absorção determinística)
- (4)  $\sigma^{(0)} > 0$  (desvio padrão inicial de uma distribuição gaussiana para gerar o parâmetro  $q$ )
- (5)  $\alpha^{(0)} > 0$  (tamanho do passo inicial)
- (6)  $0 < \beta < 1$  (fator que decresce o tamanho do passo e o desvio padrão de uma distribuição gaussiana que gera o parâmetro  $q$ )
- (7)  $\theta^{(0)}, \theta_{\min} > 0$  (desvio padrão inicial e mínimo da perturbação gaussiana)
- (8)  $m > 1$  e  $m$  é um inteiro (intervalo entre iterações gaussianas)
- (9)  $r > 0$  e  $r$  é um inteiro (número de perturbações gaussianas dentro de cada iteração gaussiana; valor padrão utilizado é  $r = N + 1$ )
- (10)  $\xi > 0$  (coeficiente para derivadas por diferenças finitas)

**Passo 1:** Faça  $k = 0$ ,  $\mathbf{x}_{\text{melhor}} = \mathbf{x}^{(0)}$  e  $\theta^{(k)} = \theta^{(0)}$

**Passo 2:** Se  $k \neq 0$  e  $\text{mod}(k, m) = 0$ , então vá para o Passo 9 para executar a iteração gaussiana; senão, execute a iteração do vetor  $q$ -gradiente e selecione  $\mathbf{q}^{(k)} = (q_1^{(k)}, \dots, q_N^{(k)})$  como segue: Para  $i = 1$  até  $N$ ,

(2a) Se  $x_i^{(k)} \neq 0$ , então sorteie  $q_i^{(k)}$  a partir de uma distribuição gaussiana com média 1 e desvio padrão  $\sigma^{(k)} / |x_i^{(k)}|$ ; senão, faça  $q_i^{(k)} = 1$

(2b) Se  $q_i^{(k)} x_i^{(k)} \notin [x_{\min_i}, x_{\max_i}]$ , então faça  $q_i^{(k)} x_i^{(k)}$  como sendo o ponto mais próximo dentro do espaço de busca e atualize o valor de  $q_i^{(k)}$

**Passo 3:** Para  $i = 1$  até  $N$ ,

(3a) Se  $q_i^{(k)} \neq 1$ , faça  $\mathbf{y}^{(k,i)} = \mathbf{x}^{(k)} + (q_i^{(k)} - 1)x_i^{(k)}\mathbf{e}^{(i)}$ ; senão, faça  $\mathbf{y}^{(k,i)} = \mathbf{x}^{(k)} + \xi\mathbf{e}^{(i)}$ . Aqui,  $\mathbf{e}^{(i)}$  é a  $i$ -ésima coluna da matriz identidade  $I_N$

(3b) Avalie  $f(\mathbf{y}^{(k,i)})$

(3c) Se  $f(\mathbf{y}^{(k,i)}) < f(\mathbf{x}_{\text{melhor}})$ , atualize  $\mathbf{x}_{\text{melhor}} = \mathbf{y}^{(k,i)}$

Use  $f(\mathbf{x}^{(k)}), f(\mathbf{y}^{(k,1)}), \dots, f(\mathbf{y}^{(k,N)})$  para calcular  $\nabla_{\mathbf{q}} f(\mathbf{x}^{(k)})$

**Passo 4:** Calcule  $\mathbf{d}^{(k)} = -\nabla_{\mathbf{q}} f(\mathbf{x}^{(k)}) / \|\nabla_{\mathbf{q}} f(\mathbf{x}^{(k)})\|$

**Passo 5:** Calcule  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \cdot \mathbf{d}^{(k)}$

**Passo 6:** Faça  $\mathbf{x}^{(k+1)} = \rho_{\mathcal{D}}(\mathbf{x}^{(k+1)})$  e avalie  $f(\mathbf{x}^{(k+1)})$

**Passo 7:** Se  $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}_{\text{melhor}})$  faça  $\mathbf{x}_{\text{melhor}} = \mathbf{x}^{(k+1)}$

**Passo 8:** Faça  $\sigma^{(k+1)} = \beta \cdot \sigma^{(k)}$  e  $\alpha^{(k+1)} = \beta \cdot \alpha^{(k)}$  e vá para o Passo 11

**Passo 9:** Execute a iteração gaussiana: Para  $i = 1$  até  $r$

(9a) Sorteie  $\mathbf{z}^{(k,i)}$  por uma distribuição gaussiana com média 0 e matriz de covariância  $(\theta^{(k)})^2 I_N$

(9b) Faça  $\mathbf{y}^{(k,i)} = \rho_{\mathcal{D}}(\mathbf{x}^{(k)} + \mathbf{z}^{(k,i)})$  e avalie  $f(\mathbf{y}^{(k,i)})$

(9c) Se  $f(\mathbf{y}^{(k,i)}) < f(\mathbf{x}_{\text{melhor}})$ , atualize  $\mathbf{x}_{\text{melhor}} = \mathbf{y}^{(k,i)}$

**Passo 10:** Faça  $\hat{i} = \text{argmin}_{1 \leq i \leq r} f(\mathbf{y}^{(k,i)})$ . Se  $f(\mathbf{y}^{(k,\hat{i})}) < f(\mathbf{x}^{(k)})$ , então

$\mathbf{x}^{(k+1)} = \mathbf{y}^{(k,\hat{i})}$  e  $\theta^{(k+m)} = \theta^{(k)}$ ; Senão,  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$  e  $\theta^{(k+m)} = \max(\theta^{(k)}/2, \theta_{\min})$

**Passo 11:** Se o critério de parada é atingido, retorne  $\mathbf{x}_{\text{melhor}}$  e  $f(\mathbf{x}_{\text{melhor}})$ . Caso contrário, incremente  $k = k + 1$  e vá para o Passo 2

---

otimização multimodal (GOUVÊA, 2016).

O bom desempenho dos métodos baseados em  $q$ -gradiente para resolver problemas multimodais com até 30 dimensões não se repete quando os métodos são aplicados a problemas de grande porte. A solução de problemas desse tipo é custosa para os métodos baseados em  $q$ -gradiente, pois para determinar a direção de busca são utilizadas  $N + 1$  avaliações da função objetivo, e quando  $N$  aumenta, o desempenho desses métodos é deteriorado.

Desta forma, foi gerada uma versão do método  $q$ -G com o intuito de solucionar problemas de otimização com um grande número de variáveis de decisão. Este novo método, que é apresentado no próximo capítulo, reduz drasticamente o número de avaliações da função objetivo utilizado para determinar a direção de busca.





## 4 Método Fq-G

Este capítulo apresenta o método Fq-G, uma versão eficiente e competitiva do método q-G para otimização de problemas de grande porte. Para determinar a direção de busca deste novo método foi definido o vetor Fq-gradiente, uma aproximação do vetor q-gradiente que perturba simultaneamente todas as coordenadas do ponto de busca. Essa aproximação foi inspirada na aproximação do vetor gradiente realizada por Spall (1992), que apresenta a vantagem de utilizar apenas duas avaliações da função objetivo, independente da dimensão do problema.

Além da definição e do uso do vetor Fq-gradiente foi desenvolvida uma estratégia, apresentada neste capítulo, para determinar o tamanho do passo no método Fq-G baseada em interpolação parabólica e no valor do parâmetro  $\mathbf{q}$ .

Por fim, é mostrado que com algumas modificações é possível obter uma versão do método Fq-G com perturbações gaussianas que possui prova de convergência no sentido probabilístico.

### 4.1 Perturbações Simultâneas

O SPSA, sigla de *Simultaneous Perturbation Stochastic Approximation* (SPALL, 1992), é um otimizador local estocástico criado para solucionar problemas contínuos com ruído. Métodos deste tipo são denominados *aproximações estocásticas* e utilizam aproximações do vetor gradiente da função objetivo com ruído. SPSA é uma versão da clássica aproximação estocástica baseada em diferenças finitas FDSA (KIEFER et al., 1952), que utiliza apenas duas avaliações da função objetivo na aproximação do vetor gradiente, independente do número de dimensões do problema de otimização. Spall (1992) apresentou resultados que demonstraram que o algoritmo SPSA foi mais eficiente do que o FDSA na resolução de um problema de 20 dimensões, além de condições para a convergência local do método. Posteriormente, Maryak e Chin (2004) demonstraram que com a introdução de ruído, o SPSA pode convergir para o mínimo global no sentido probabilístico.

Diversas versões do método têm sido desenvolvidas, inclusive para solucionar problemas com variáveis discretas, e testadas em aplicações como, por exemplo, sistemas de filas (HILL; FU, 1995; BHATNAGAR, 2005; MISHRA et al., 2007), reconhecimento de padrões, melhoria da qualidade industrial, *design* de aeronaves (XING; DAMODARAN, 2002; XING; DAMODARAN, 2005), otimização baseada em simulação, treinamento da rede neural (ABDULSADDA; IQBAL, 2011; DONG; CHEN, 2012), controle

de processo químico, detecção de falhas (ALESSANDRI; PARISINI, 1997), a interação homem-máquina (XU et al., 2002; VITSENTIY, 2002), configuração de sensor, gestão de tráfego de veículos, entre outras.

A versão básica do SPSA utiliza o procedimento iterativo definido pela Equação (2.4) em que  $\mathbf{d}^{(k)}$  é dada por uma aproximação do vetor gradiente de  $f$  no ponto  $\mathbf{x}^{(k)}$  através de perturbações simultâneas. A aproximação do vetor gradiente calculada através de diferenças finitas centradas é dada pela equação (SPALL, 1998b):

$$\widehat{\nabla} f(\mathbf{x}^{(k)}) \cong \begin{bmatrix} \frac{f(\mathbf{x}^{(k)} + c^{(k)} \mathbf{e}_1) - f(\mathbf{x}^{(k)} - c^{(k)} \mathbf{e}_1)}{2c^{(k)}} \\ \vdots \\ \frac{f(\mathbf{x}^{(k)} + c^{(k)} \mathbf{e}_i) - f(\mathbf{x}^{(k)} - c^{(k)} \mathbf{e}_i)}{2c^{(k)}} \\ \vdots \\ \frac{f(\mathbf{x}^{(k)} + c^{(k)} \mathbf{e}_N) - f(\mathbf{x}^{(k)} - c^{(k)} \mathbf{e}_N)}{2c^{(k)}} \end{bmatrix}, \quad (4.1)$$

em que  $\mathbf{e}_i$  denota a  $i$ -ésima coluna da matriz identidade  $\mathbf{I}_N$  e  $c^{(k)}$  é um escalar positivo.

Na equação (4.1) é possível observar que cada coordenada da variável de decisão  $\mathbf{x}^{(k)}$  é deslocada em duas direções e a função é avaliada duas vezes para determinar cada componente do vetor  $\widehat{\nabla} f(\mathbf{x}^{(k)})$ , totalizando  $2 \cdot N$  avaliações da função objetivo necessárias para determinar a aproximação do vetor gradiente.

Spall (1992) propôs um novo modo para o cálculo da aproximação do vetor gradiente, que consiste em perturbar simultaneamente todas as componentes de  $\mathbf{x}^{(k)}$  de maneira aleatória, exigindo apenas duas avaliações da função objetivo. Isto permite uma significativa diminuição do custo computacional, especialmente em problemas com um grande número de variáveis a serem otimizadas (SPALL, 1998b). Matematicamente, a aproximação do gradiente de  $f$  no ponto  $\mathbf{x}^{(k)}$  por perturbações simultâneas é dada por

$$\widehat{\nabla} f(\mathbf{x}^{(k)}) = \begin{bmatrix} \frac{f(\mathbf{x}^{(k)} + c^{(k)} \Delta^{(k)}) - f(\mathbf{x}^{(k)} - c^{(k)} \Delta^{(k)})}{2c^{(k)} \Delta_1^{(k)}} \\ \vdots \\ \frac{f(\mathbf{x}^{(k)} + c^{(k)} \Delta^{(k)}) - f(\mathbf{x}^{(k)} - c^{(k)} \Delta^{(k)})}{2c^{(k)} \Delta_i^{(k)}} \\ \vdots \\ \frac{f(\mathbf{x}^{(k)} + c^{(k)} \Delta^{(k)}) - f(\mathbf{x}^{(k)} - c^{(k)} \Delta^{(k)})}{2c^{(k)} \Delta_N^{(k)}} \end{bmatrix}, \quad (4.2)$$

em que  $\Delta^{(k)}$  é o vetor de perturbação aleatória de  $N$  dimensões e  $c^{(k)}$ , assim como em

(4.1), é um escalar positivo. Observe que o numerador em todas as coordenadas é o mesmo, enquanto o denominador varia de acordo com o valor de  $\Delta_i^{(k)}$ ,  $i = 1, \dots, N$ .

Cada uma das  $N$  componentes do vetor  $\Delta^{(k)}$  são independentemente geradas a partir de uma distribuição de probabilidade com média zero que satisfaz as seguintes condições: simetria, média zero, variância finita, e momentos inversos finitos (SPALL, 1992). A distribuição de probabilidade simétrica de Bernoulli  $\pm 1$ , onde cada componente de  $\Delta^{(k)}$  assume os valores  $-1$  ou  $+1$  com probabilidade de  $1/2$ , é uma distribuição simples que satisfaz as condições sobre  $\Delta^{(k)}$ . Enquanto que as distribuições de probabilidade simétricas uniforme e normal com média zero não satisfazem a condição de momentos inversos finitos.

O desempenho do SPSA depende da correta seleção dos seus parâmetros de ajuste. O par  $c^{(k)}, \alpha^{(k)}$  é chamado de sequências de ganho, as duas sequências devem ser monótonas decrescentes para cumprir a condição de convergência apresentada em (SPALL, 1992; SPALL, 2005). O par é alterado a cada iteração  $k$  por meio das seguintes expressões sugeridas por (SPALL, 2005):

$$c^{(k)} = \frac{c}{(k+1)^\gamma}, \quad (4.3)$$

$$\alpha^{(k)} = \frac{a}{(k+1+A)^\eta}, \quad (4.4)$$

onde  $c, \gamma, A, a$  e  $\eta$  são constantes estritamente positivas. O parâmetro  $c$  assume um valor pequeno quando  $f$  não apresenta ruído, quando tem-se uma função com ruído, pode ser tomado como o desvio padrão do ruído. A constante  $A$  é denominada constante de estabilidade, pois permite que  $a$  assumam valores altos, sem o risco de comportamento instável do algoritmo nas iterações iniciais. E nas iterações finais, o valor de  $A$  se torna insignificante em comparação com  $k$ , enquanto o valor alto de  $a$  no numerador ajuda a manter um passo significativo (SPALL, 2005). Uma boa escolha para  $A$  é 10% (ou menos) do número máximo de iterações (SPALL, 1998a). O parâmetro  $a$  pode ser definido como a mudança de magnitude desejada nas primeiras iterações, isto é,  $a$  deve ser tomado de forma que  $\alpha^{(k)} \widehat{\nabla} f(\mathbf{x}^{(k)})$  possua a magnitude desejada para as primeiras iterações do algoritmo. Os expoentes  $\gamma$  e  $\eta$  controlam a velocidade de decrescimento das sequências de ganho, e possuem  $1/6$  e  $1$  como seus valores ótimos, respectivamente. Mas os valores que se mostraram mais eficazes, e por isso indicados por Spall (1998a) são  $\eta = 0.602$  e  $\gamma = 0.101$ .

A Figura 4.1 ilustra as direções de busca do método SPSA para casos em que o vetor perturbação  $\Delta^{(k)}$  é gerado segundo uma distribuição de probabilidade simé-

trica de Bernoulli  $\pm 1$ . A Figura 4.1 (a) exibe os quatro pontos ao redor de  $\mathbf{x}^{(k)}$ , estes são os únicos pontos que  $\mathbf{x}^{(k)} \pm c^{(k)} \Delta^{(k)}$  podem assumir, considerando os quatro possíveis valores de  $\Delta^{(k)}$ . Além disso, está representada a direção de máxima descida  $-\nabla f(\mathbf{x}^{(k)})$ . A Figura 4.1(b) ilustra duas possíveis aproximações do gradiente  $-\widehat{\nabla} f(\mathbf{x}^{(k)})$ . Note que apesar de não apresentarem direções de máxima descida, ambas direções são de descida. Finalmente, a Figura 4.1(c) mostra a média de duas possíveis aproximações do gradiente da Figura 4.1(b). A média não corresponde ao gradiente clássico, mas é bem próxima.

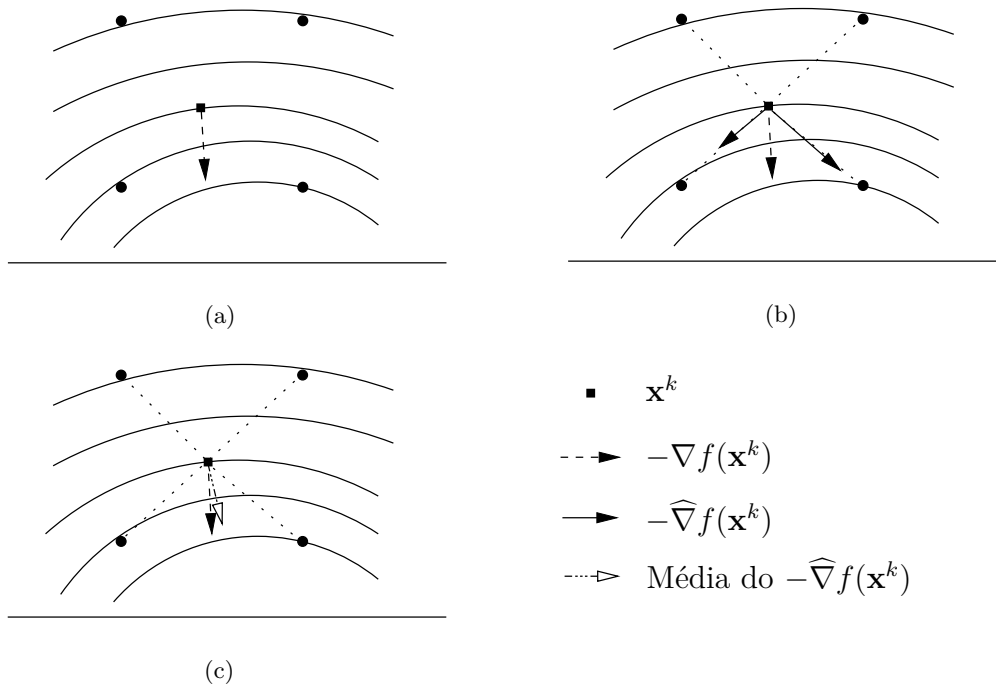


Figura 4.1 - Comparação da direção de busca do gradiente clássico com a aproximação do gradiente utilizada no SPSA com  $N = 2$ . (a) Apresenta as quatro perturbações de  $\mathbf{x}^{(k)}$ ,  $\mathbf{x}^{(k)} \pm c^{(k)}$ , usando a distribuição de probabilidade simétrica de Bernoulli  $\pm 1$ ; e o vetor gradiente  $\nabla f(\mathbf{x})$ . (b) Ilustra duas possíveis direções de busca  $-\widehat{\nabla} f(\mathbf{x}^{(k)})$ . (c) Exibe a média das direções de busca exibidas em (b).

Fonte: Spall (2005).

O Algoritmo 3 do método SPSA é apresentado a seguir.

## 4.2 Vetor Fq-gradiente

Como foi visto na seção 3.2, o método  $q$ -G utiliza o vetor  $q$ -gradiente para determinar a direção de busca  $\mathbf{d}^{(k)}$ , estratégia que necessita de  $N + 1$  avaliações da função objetivo  $f$  a cada nova iteração  $k$ .

---

**Algoritmo 3:** Método SPSA (SPALL, 2005)

---

**Entrada**

- (1)  $f : \mathbb{R}^N \rightarrow \mathbb{R}$  contínua e diferenciável
- (2)  $\mathbf{x}^{(0)} \in \mathbb{R}^N$
- (3)  $a > 0$
- (4)  $c > 0$
- (5)  $A > 0$  (constante de estabilidade; valor padrão utilizado é 10% do número máximo de iterações )
- (6)  $\gamma > 0$  (expoente de controle da velocidade do decrescimento da sequência  $a^{(k)}$ ; valor padrão utilizado é 0.101)
- (7)  $\eta > 0$  (expoente de controle da velocidade do decrescimento da sequência  $a^{(k)}$ ; valor padrão utilizado é 0.602 )

**Passo 0:** Faça  $k = 0$

**Passo 1:**  $\alpha^{(k)} = a/(A + k + 1)^\eta$

**Passo 2:**  $c^{(k)} = c/(k + 1)^\gamma$

**Passo 3:** Obtenha o vetor de perturbação aleatória  $\Delta^{(k)}$

**Passo 4:**  $\mathbf{d}^{(k)} = -\widehat{\nabla} f(\mathbf{x}^{(k)})$  (Equação 4.2 )

**Passo 5:**  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)}$

**Passo 6:** Se o critério de parada é atingido, retorne  $\mathbf{x}^{(k)}$  e  $f(\mathbf{x}^{(k)})$ , senão incremente  $k = k + 1$ , e vá para o Passo 1.

---

Naturalmente, quanto maior a dimensão do problema, mais avaliações da função objetivo são necessárias para determinar a direção de busca, o que afeta o desempenho do algoritmo  $q$ -G. Desta forma, com o intuito de evitar a degradação do desempenho computacional que ocorre quando o valor de  $N$  cresce e inspirado nas perturbações simultâneas realizadas por Spall (1992) no cálculo da aproximação do gradiente, propõe-se a seguinte aproximação para o vetor  $q$ -gradiente, chamada F $q$ -gradiente:

$$\widehat{\nabla}_{\mathbf{q}} f(\mathbf{x}) = [\widehat{D}_{\mathbf{q},x_1} f(\mathbf{x}) \dots \widehat{D}_{\mathbf{q},x_i} f(\mathbf{x}) \dots \widehat{D}_{\mathbf{q},x_N} f(\mathbf{x})]^T, \quad (4.5)$$

onde

$$\widehat{D}_{\mathbf{q},x_i} f(\mathbf{x}) = \begin{cases} \frac{f(q_1 x_1, \dots, q_i x_i, \dots, q_N x_N) - f(x_1, \dots, x_i, \dots, x_N)}{q_i x_i - x_i}, & \text{se } q_i \neq 1 \text{ e } x_i \neq 0, \\ \frac{\partial f(\mathbf{x})}{\partial x_i}, & \text{caso contrário,} \end{cases}$$
$$= \begin{cases} \frac{f(\mathbf{q} \odot \mathbf{x}) - f(\mathbf{x})}{(q_i - 1)x_i}, & \text{se } q_i \neq 1 \text{ e } x_i \neq 0, \\ \frac{\partial f(\mathbf{x})}{\partial x_i}, & \text{caso contrário.} \end{cases} \quad (4.6)$$

Nessa aproximação, o vetor  $\mathbf{q}$  perturba todas as coordenadas do ponto  $\mathbf{x}$  de maneira simultânea. Quando  $\mathbf{x} \neq \mathbf{0}$  e  $\mathbf{q} \neq \mathbf{1}$ , o cálculo do vetor  $Fq$ -gradiente utiliza apenas duas avaliações da função objetivo, por iteração, para determinar todos os  $N$  componentes de  $\widehat{\nabla}_{\mathbf{q}}f(\mathbf{x})$ , independentemente do tamanho do problema. A estratégia de geração do vetor  $\mathbf{q}$  é a mesma utilizada por Soterroni et al. (2015), Gouvêa et al. (2016) e descrita na seção 3.2.

As Figuras 4.2(a) e 4.2(b) mostram 100 possíveis direções do vetor  $Fq$ -gradiente no ponto  $\mathbf{x} = (11, 11)$ , para uma função quadrática bidimensional, geradas por dois valores de  $\sigma^{(0)}$ . A linha em vermelho representa a direção de máxima descida. Note que quando  $\sigma^{(0)}$  é suficientemente grande (esquerda), geralmente no início do processo de busca, inúmeras direções são possíveis, inclusive direções de subida. Essa característica é semelhante à do vetor  $q$ -gradiente, e permite ao método que utiliza o vetor  $Fq$ -gradiente realizar uma busca global (*exploration*). Para  $\sigma^{(0)}$  pequenos (direita), as direções geradas são todas de descida, permitindo que uma busca local (*exploitation*) seja realizada. No limite  $k \rightarrow \infty$ , tem-se uma direção de descida.

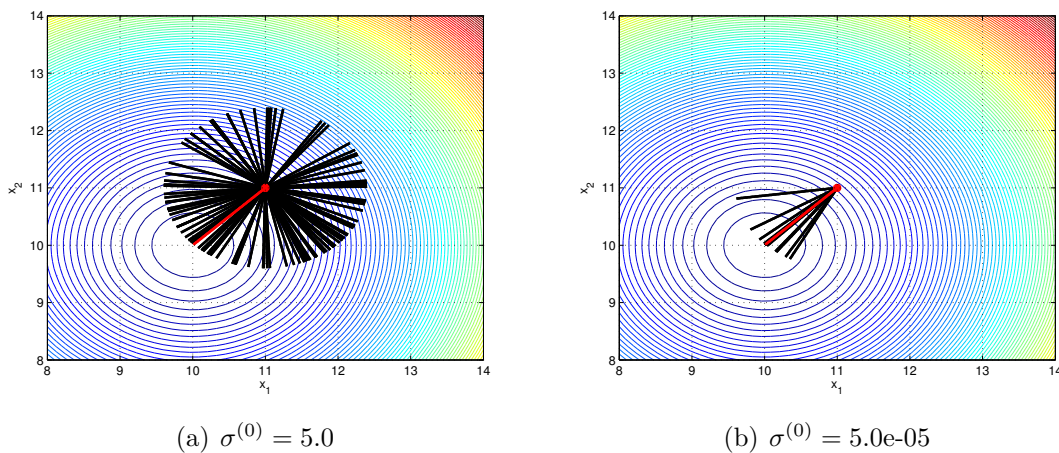


Figura 4.2 - Possíveis 100 direções do vetor  $Fq$ -G para uma função quadrática bidimensional em  $\mathbf{x} = (11, 11)$  obtidas para valores do parâmetro  $q$  gerados a partir de uma distribuição de probabilidade gaussiana com média em  $\mathbf{1}$  e desvio padrão (a) grande e (b) pequeno. O segmento em vermelho indica a direção de máxima descida da função no ponto  $\mathbf{x}$ .

Para gerar uma versão eficiente do método  $q$ -G destinada à otimização de problemas de grande porte, o vetor  $Fq$ -gradiente é utilizado para determinar a direção de busca

deste novo método. Dessa forma considerando o procedimento iterativo (2.4) tem-se

$$\mathbf{d}^{(k)} = -\frac{\widehat{\nabla}_{\mathbf{q}} f(\mathbf{x}^{(k)})}{\|\widehat{\nabla}_{\mathbf{q}} f(\mathbf{x}^{(k)})\|}. \quad (4.7)$$

### 4.3 Estratégia para escolha do tamanho do passo

Depois de definir como a direção de busca deve ser calculada é preciso determinar qual o tamanho do passo  $\alpha^{(k)}$  será dado nessa direção. Obviamente, a primeira técnica considerada para calcular o tamanho do passo foi a “Controle do Passo por Redução” utilizada pelos métodos baseados em  $q$ -gradiente. No entanto, exaustivos testes numéricos demonstraram a ineficiência da combinação dessa estratégia de determinação do passo com a direção de busca dada pelo vetor  $Fq$ -gradiente.

Determinar o tamanho do passo em uma direção de busca pode ser considerado um problema de minimização unidimensional

$$\min \varphi(\alpha^{(k)}) = f(\mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)}), \text{ sujeito a } \alpha^{(k)} > 0. \quad (4.8)$$

Existem diversas técnicas utilizadas para resolver esse problema de busca em linha. O método da seção áurea considera que a função está definida em um intervalo de busca que cerca o menor valor da função. Pontos no interior deste intervalo são obtidos segundo a proporção áurea, e substituições são realizadas com o propósito de reduzir progressivamente o comprimento do intervalo. Semelhante ao método da seção áurea, o método da biseção e o método de Fibonacci também realizam sucessivas reduções do intervalo de busca com o objetivo de cercar o mínimo da função. Em geral, esses métodos que particionam o intervalo até encontrar o ponto de mínimo utilizam muitas avaliações da função objetivo e se tornam uma estratégia computacionalmente custosa.

Uma técnica que precisa de um número bem menor de avaliações da função objetivo é a Regra de Armijo, a ideia não é determinar o mínimo e sim uma boa redução da função  $f$  ao longo da direção  $\mathbf{d}^{(k)}$ . Assim, a regra de Armijo encontra  $\alpha^{(k)} > 0$  tal que

$$f(\mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)}) \leq f(\mathbf{x}^{(k)}) + \eta \alpha^{(k)} \nabla f(\mathbf{x}^{(k)}) \mathbf{d}^{(k)}, \quad (4.9)$$

onde  $\eta \in (0, 1)$  (RIBEIRO; KARAS, 2013).

Os métodos citados acima exigem que  $\mathbf{d}^{(k)}$  seja uma direção de descida. No entanto, as direções do vetor  $q$ -gradiente e do vetor  $Fq$ -gradiente podem ser tanto de subida

quanto de descida, com isto só é possível utilizar essas técnicas de busca em linha depois de realizar algumas adaptações.

Outra estratégia utilizada para resolver o problema de minimização (4.8) é a interpolação polinomial. A ideia é aproximar a função  $\varphi$  através de um polinômio utilizando valores de  $\varphi$ . Sucessivas aproximações são realizadas para que o mínimo do polinômio esteja cada vez mais próximo do mínimo de  $\varphi$  (IZMAILOV; SOLODOV, 2007). Na interpolação parabólica por exemplo, a partir de três pontos  $a < b < c$ , tais que

$$\varphi(a) > \varphi(b) \quad \text{e} \quad \varphi(b) < \varphi(c), \quad (4.10)$$

$\varphi$  é aproximada por um polinômio quadrático. O ponto de mínimo desta parábola substitui um desses três pontos e uma nova aproximação de  $\varphi$  é realizada até que uma determinada tolerância seja alcançada. Uma das vantagens da interpolação parabólica é que o seu ponto de mínimo é determinado analiticamente. A desvantagem é o grande número de avaliações da função objetivo utilizados para determinar o mínimo de  $\varphi$ .

A estratégia escolhida para definir o tamanho do passo  $\alpha^{(k)}$  do método Fq-G combina interpolação parabólica e o valor do parâmetro  $\mathbf{q}$  na iteração  $k$ , e será denotada “Passo Parabólico”. Dado o ponto atual da busca  $\mathbf{x}^{(k)}$  e o vetor  $\mathbf{q}^{(k)}$ , o tamanho do passo  $\alpha^{(k)}$  será a distância entre  $\mathbf{x}^{(k)}$  e o ponto de mínimo da parábola que passa pela tripla  $(\mathbf{x}^{(k)} - \delta^{(k)}\mathbf{d}^{(k)}, \mathbf{x}^{(k)}, \mathbf{x}^{(k)} + \delta^{(k)}\mathbf{d}^{(k)})$ , onde  $\delta^{(k)} = \max(\epsilon, \|\mathbf{q}^{(k)}\mathbf{x}^{(k)} - \mathbf{x}^{(k)}\|_2)$ , e  $\epsilon$  é um real positivo. Observe que não há garantias de que  $f(\mathbf{x}^{(k)} - \delta^{(k)}\mathbf{d}^{(k)})$ ,  $f(\mathbf{x}^{(k)})$  e  $f(\mathbf{x}^{(k)} + \delta^{(k)}\mathbf{d}^{(k)})$  cumpram a condição (4.10), desta forma quando a parábola possui um ponto de máximo ou os pontos de avaliação da tripla são colineares, o tamanho do passo  $\alpha^{(k)}$  é definido como  $\delta^{(k)}$ .

No início do processo de busca, quando os valores assumidos por  $\sigma^{(k)}$  são grandes e portanto, os valores de  $\mathbf{q}^{(k)}$  estão provavelmente bem distantes de  $\mathbf{1}$ , o tamanho do passo  $\alpha^{(k)}$  é consideravelmente grande, característica que permite ao método explorar o espaço de busca. Quando  $\sigma^{(k)} \rightarrow 0$ , conseqüentemente  $\|\mathbf{q}^{(k)}\mathbf{x}^{(k)} - \mathbf{x}^{(k)}\|_2 \rightarrow 0$ , a aproximação parabólica é realizada localmente, permitindo a busca local. O valor de  $\epsilon$  é tomado como um real positivo para garantir que  $\alpha^{(k)} > 0$ .

#### 4.4 Algoritmo do método Fq-G

A seguir, o algoritmo para o método Fq-G, desenvolvido para solucionar problemas de otimização globais e irrestritos de grande porte é apresentado.



---

**Algoritmo 4:** Método Fq-G para otimização global de problemas de grande porte irrestritos

---

**Entradas:**

- (1)  $f : \mathbb{R}^N \rightarrow \mathbb{R}$  contínua e diferenciável
- (2)  $\mathbf{x}^{(0)} \in \mathbb{R}^N$  (ponto inicial)
- (2)  $\sigma^{(0)} > 0$  (desvio padrão inicial da distribuição gaussiana para gerar o parâmetro  $\mathbf{q}$ )
- (4)  $0 < \beta < 1$  (fator que decresce o desvio padrão de uma distribuição gaussiana que gera o parâmetro  $\mathbf{q}$ )

**Passo 0:** Faça  $k = 0$  e  $\mathbf{x}_{melhor} = \mathbf{x}^{(0)}$

**Passo 1:** Para  $i = 1$  até  $N$ , se  $x_i^{(k)} \neq 0$ , então sorteie  $q_i^{(k)}$  a partir de uma distribuição gaussiana com média 1 e desvio padrão  $\sigma^{(k)} / |x_i^{(k)}|$ ; senão, faça  $q_i^{(k)} = 1$

**Passo 2::**  $\mathbf{d}^{(k)} = -\widehat{\nabla}_{\mathbf{q}} f(\mathbf{x}^{(k)}) / \|\widehat{\nabla}_{\mathbf{q}} f(\mathbf{x}^{(k)})\|$

**Passo 3:** Calcule  $\alpha^{(k)}$  utilizando o Passo Parabólico

**Passo 4:**  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)}$

**Passo 5:** Se  $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}_{melhor})$  então  $\mathbf{x}_{melhor} = \mathbf{x}^{(k+1)}$

**Passo 6:** Faça  $\sigma^{(k+1)} = \beta \cdot \sigma^{(k)}$

**Passo 7:** Se o critério de parada é atingido, retorne  $\mathbf{x}_{melhor}$  e  $f(\mathbf{x}_{melhor})$ , senão incremente  $k = k + 1$ , e vá para o Passo 1.

---

Como foi visto, o método Fq-G é uma variação do método q-G para problemas de otimização global com muitas variáveis de decisão. Com o propósito de comparar o desempenho do método Fq-G apresentado no Algoritmo 4 com a versão do Método q-G para problemas irrestritos, os dois métodos foram aplicados em três funções teste  $f : \mathbb{R}^N \rightarrow \mathbb{R}$ . A função Elipsoidal é dada por

$$f(\mathbf{x}) = \sum_{i=1}^N ix_i^2. \quad (4.11)$$

Essa é uma função unimodal e o caso bidimensional está ilustrado na Figura 4.4.

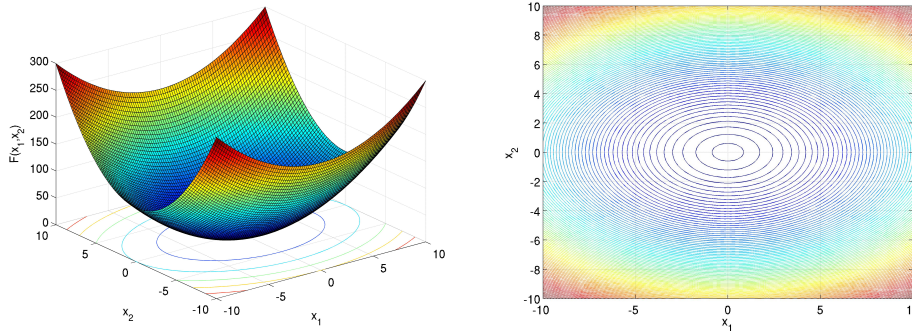


Figura 4.3 - Função Elipsoidal.

A função Rastrigin é dada pela expressão

$$f(\mathbf{x}) = 10N + \sum_{i=1}^N (x_i^2 - 10\cos(2\pi x_i)) \quad (4.12)$$

e possui um envoltório parabólico com múltiplos mínimos locais, como pode ser observado na Figura 4.4.

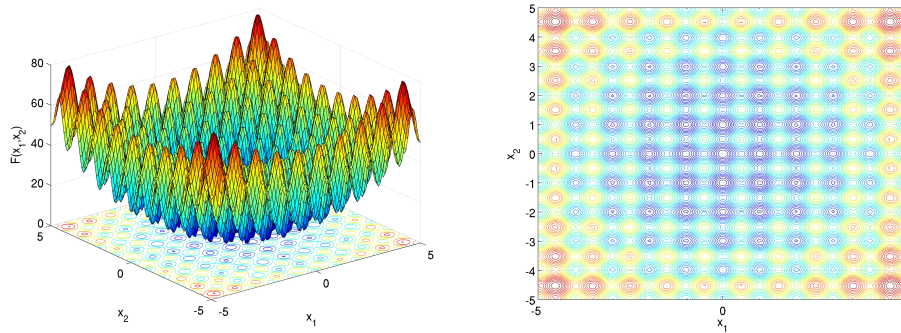


Figura 4.4 - Função Rastrigin.

Por fim, a função multimodal Ackley, ilustrada na Figura 4.4, é dada por

$$f(\mathbf{x}) = 20 + e - 20\exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^N x_i^2}\right) - \exp\left(\frac{1}{N}\sum_{i=1}^N \cos(2\pi x_i)\right). \quad (4.13)$$

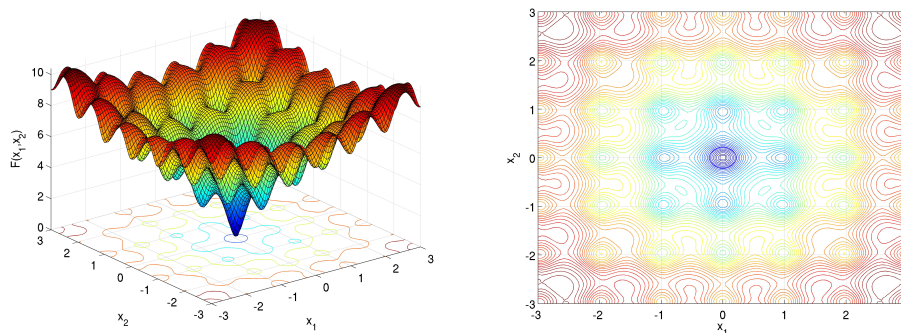


Figura 4.5 - Função Ackley.

O método  $q$ -G utilizado nesta comparação é semelhante ao do Algoritmo 1, se diferenciando apenas por utilizar a estratégia Passo Parabólico para determinar o tamanho do passo. A partir de 25 pontos iniciais distintos gerados uniformemente no espaço de busca de cada uma das três funções teste, foram realizadas 25 execuções independentes de ambos os algoritmos. Como as três funções teste são irrestritas, para sortear os pontos iniciais de cada execução considerou-se os espaços de busca  $[-10, 10]^N$ ,  $[-5.12, 5.12]^N$  e  $[-30, 30]^N$  para a função Elipsoidal, Rastrigin e Ackley, respectivamente. Três diferentes dimensões foram consideradas:  $N = 10, 100$  e  $1000$ . Os critérios de parada foram: atingir o número máximo de avaliação da função objetivo ( $\text{NAFO}_{\max}$ ) de  $10000 \cdot N$  ou atingir a acurácia desejada de  $(1.0\text{e-}08)$ , isto é,  $|f(\mathbf{x}_{\text{melhor}}) - f(\mathbf{x}^*)| \leq 1.0\text{e-}08$ . Para as três funções teste tem-se o mínimo global  $\mathbf{x}^* = \mathbf{0}$  com  $f(\mathbf{x}^*) = 0$ . Os dois algoritmos foram implementados em Fortran 90 para o compilador FORTRAN (*Intel Fortran Compiler*) e executado em uma máquina com processador Intel Core i5 com 8GB RAM e sistema operacional MAC OS X.

Nessa comparação os dois parâmetros de ajuste dos métodos,  $\sigma^{(0)}$  e  $\beta$ , foram obtidos a partir de rodadas preliminares combinando valores de ambos os parâmetros para selecionar os melhores para cada método, dimensão e função. A Tabela 4.1 apresenta os melhores valores de  $\sigma^{(0)}$  e  $\beta$ , o símbolo  $L = \|\mathbf{x}_{\max} - \mathbf{x}_{\min}\|_2$  que aparece na tabela indica a maior distância dentro do espaço de busca de cada função. A Tabela 4.2 exhibe o NAFO mínimo, mediano e máximo necessários para atingir o critério de parada, além do número médio de iterações utilizado por cada um dos métodos. O valor médio de  $f(\mathbf{x}_{\text{melhor}})$  em 25 execuções,  $\overline{f(\mathbf{x}_{\text{melhor}})}$ , também é exibido. E a última coluna mostra o tempo gasto (em segundos) para realizar as 25 execuções dos algoritmos.

Como esperado, os resultados mostram que o aumento das dimensões influenciou o desempenho do método  $q$ -G, enquanto que o método F $q$ -G conseguiu atingir a acurácia desejada nas três funções teste e para as três dimensões consideradas. Os resultados obtidos pelo método F $q$ -G são superiores aos obtidos pelo  $q$ -G principalmente para os problemas com dimensão igual a 1000, tanto em relação ao NAFO utilizado quanto ao  $\overline{f(\mathbf{x}_{\text{melhor}})}$  alcançado. Para  $N = 1000$  o número iterações usado pelo método F $q$ -G é no mínimo três vezes maior do utilizado pelo  $q$ -G. Essa diferença é explicada pelo fato do método F $q$ -G utilizar um número menor de avaliação da função objetivo do que o algoritmo do método  $q$ -G para determinar a direção de busca, e com isso conseguir usar o NAFO disponível para visitar um número maior de pontos no espaço de busca. O tempo gasto na realização de 25 execuções pelos dois algoritmos não é proporcional a diferença de NAFO utilizado por cada um dos

métodos.

Tabela 4.1 - Parâmetros usados pelos métodos  $q$ -G e F $q$ -G.

Função	Dimensão	Método	$\sigma^{(0)}$	$\beta$
Elipsoidal	10	F $q$ -G	4.00e-02L	0.992
		$q$ -G	1.00e-05L	0.999
	100	F $q$ -G	2.00e+01L	0.999
		$q$ -G	1.00e-04L	0.9995
	1000	F $q$ -G	7.00e-02L	0.99999
		$q$ -G	1.05e+01L	0.996
Rastrigin	10	F $q$ -G	6.00e+00L	0.9999
		$q$ -G	1.80e+01L	0.992
	100	F $q$ -G	1.95e+01L	0.9995
		$q$ -G	2.00e+00L	0.9999
	1000	F $q$ -G	2.50e+00L	0.99999
		$q$ -G	4.00e-01L	0.9995
Ackley	10	F $q$ -G	7.00e-02L	0.99
		$q$ -G	2.00e-01L	0.99
	100	F $q$ -G	1.00e-02L	0.999
		$q$ -G	2.00e-02L	0.998
	1000	F $q$ -G	1.00e-03L	0.99995
		$q$ -G	1.00e-02L	0.999

Tabela 4.2 - Desempenho comparativo, em termos de NAFO,  $\overline{f(\mathbf{x}_{melhor})}$ , número médio de iterações e tempo gasto, entre os métodos  $q$ -G e F $q$ -G.

Função	Dimensão	Método	NAFO			Iterações	$\overline{f(\mathbf{x}_{melhor})}$	Tempo(seg)
			Mínimo	Mediano	Máximo			
Elipsoidal	10	F $q$ -G	951	1332	1622	262	8.26e-09	8.63e-03
		$q$ -G	22101	26179	31145	1543	7.76e-11	5.91e-02
	100	F $q$ -G	21231	26826	48356	5917	8.45e-09	9.35e-01
		$q$ -G	1000010	1000055	1000134	8954	9.15e-05	3.84e+00
	1000	F $q$ -G	519364	846947	1527246	196366	9.04e-09	2.69e+02
		$q$ -G	10000340	10000630	10001136	7154	1.49e-03	2.72e+02
Rastrigin	10	F $q$ -G	951	1211	1386	238	7.74e-09	9.49e-03
		$q$ -G	2871	3459	4047	248	8.72e-09	1.52e-02
	100	F $q$ -G	14021	14761	15921	2966	9.91e-09	6.92e-01
		$q$ -G	299521	309193	322505	2985	9.91e-09	5.61e+00
	1000	F $q$ -G	158711	162350	165952	32527	9.97e-09	6.72e+01
		$q$ -G	10000006	10000017	10000043	9960	2.91e+02	1.56e+03
Ackley	10	F $q$ -G	3129	3431	3759	683	9.43e-09	2.87e-02
		$q$ -G	9804	10690	11417	759	9.12e-09	5.07e-02
	100	F $q$ -G	34921	36152	38373	6990	9.93e-09	1.63e+00
		$q$ -G	568503	591873	630982	5510	9.94e-09	1.04e+01
	1000	F $q$ -G	408123	412025	418217	75423	9.99e-09	1.57e+02
		$q$ -G	10000041	10000764	10001072	9878	6.60e-05	1.49e+03

Assim como no método  $q$ -G, uma versão convergente para o método F $q$ -G foi gerada, inserindo perturbações gaussianas em algumas iterações do algoritmo além de limites laterais para o problema de otimização. Aqui, o parâmetro  $r$  (ver Algoritmo 2) é

fixado com valor 1 e por isto, não aparece no Algoritmo 5.

---

**Algoritmo 5:** Método Fq-G para otimização global de grande escala com restrições laterais

---

**Entradas**

- (1)  $f : \mathcal{D} \rightarrow \mathbb{R}$ , onde  $\mathcal{D} = [\mathbf{x}_{\min}, \mathbf{x}_{\max}] \subseteq \mathbb{R}^N$  (função objetivo)
- (2)  $\mathbf{x}^{(0)} \in \mathcal{D}$  (ponto inicial)
- (3)  $\rho_{\mathcal{D}} : \mathbb{R}^N \rightarrow \mathcal{D}$  tal que  $\rho_{\mathcal{D}}(\mathbf{x}) = \mathbf{x}$  para todo  $\mathbf{x} \in \mathcal{D}$  (transformação de absorção determinística)
- (4)  $\sigma^{(0)} > 0$  (desvio padrão inicial de uma distribuição gaussiana para gerar o parâmetro  $q$ )
- (5)  $0 < \beta < 1$  (fator que decresce o desvio padrão de uma distribuição gaussiana que gera o parâmetro  $q$ )
- (7)  $\theta^{(0)}, \theta_{\min} > 0$  (desvio padrão inicial e mínimo da perturbação gaussiana)
- (8)  $m > 1$  e  $m$  é um inteiro (intervalo entre iterações gaussianas)

**Passo 0:** Faça  $k = 0$ ,  $\mathbf{x}_{\text{melhor}} = \mathbf{x}^{(0)}$  e  $\theta^m = \theta^{(0)}$ ; avalie a função objetivo  $f(\mathbf{x}^{(0)})$  e faça  $f(\mathbf{x}_{\text{melhor}}) = f(\mathbf{x}^{(0)})$ .

**Passo 1:** Se  $k \neq 0$  e  $\text{mod}(k, m) = 0$ , então vá para o Passo 9; senão, selecione  $\mathbf{q}^{(k)} = (q_1^{(k)}, \dots, q_N^{(k)})$  como segue: Para  $i = 1$  até  $N$ ,

(1a) Se  $x_i^{(k)} \neq 0$ , então sorteie  $q_i^{(k)}$  a partir de uma distribuição gaussiana com média 1 e desvio padrão  $\sigma^{(k)} / |x_i^{(k)}|$ ; senão, faça  $q_i^{(k)} = 1$ .

(1b) Se  $q_i^{(k)} x_i^{(k)} \notin [x_{\min_i}, x_{\max_i}]$ , então faça  $q_i^{(k)} x_i^{(k)}$  como sendo o ponto mais próximo dentro do espaço de busca e atualize o valor de  $q_i^{(k)}$ .

**Passo 2:** Calcule  $\mathbf{s}^{(k)} = \mathbf{q}^{(k)} \odot \mathbf{x}^{(k)}$ , onde  $\odot$  denota a multiplicação vetorial; faça  $\mathbf{s}^{(k)} = \rho_{\mathcal{D}}(\mathbf{s}^{(k)})$  e avalie  $f(\mathbf{s}^{(k)})$ .

**Passo 3:** A partir de  $f(\mathbf{s}^{(k)})$  e  $f(\mathbf{x}^{(k)})$ , calcule  $\widehat{\nabla}_{\mathbf{q}} f(\mathbf{x}^{(k)})$  e faça  $\mathbf{d}^{(k)} = -\widehat{\nabla}_{\mathbf{q}} f(\mathbf{x}^{(k)}) / \|\widehat{\nabla}_{\mathbf{q}} f(\mathbf{x}^{(k)})\|$ .

**Passo 4:** Calcule  $\alpha^{(k)}$  usando o Passo Parabólico.

**Passo 5:** Calcule  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)}$ .

**Passo 6:** Faça  $\mathbf{x}^{(k+1)} = \rho_{\mathcal{D}}(\mathbf{x}^{(k+1)})$  e avalie  $f(\mathbf{x}^{(k+1)})$ .

**Passo 7:** Se  $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}_{\text{melhor}})$  faça  $\mathbf{x}_{\text{melhor}} = \mathbf{x}^{(k+1)}$ .

**Passo 8:** Faça  $\sigma^{(k+1)} = \beta \cdot \sigma^{(k)}$  e vá para o Passo 11.

**Passo 9:** Execute a iteração gaussiana:

(9a) Sorteie  $\mathbf{z}^{(k)}$  a partir da distribuição gaussiana com média 0 e matriz de covariância  $(\theta^{(k)})^2 \mathbf{I}$ .

(9b) Calcule  $\mathbf{w}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{z}^{(k)}$

(9c) Faça  $\mathbf{y}^{(k+1)} = \rho_{\mathcal{D}}(\mathbf{w}^{(k+1)})$  e avalie  $f(\mathbf{y}^{(k+1)})$ .

(9d) Se  $f(\mathbf{y}^{(k+1)}) < f(\mathbf{x}_{\text{melhor}})$ , atualize  $\mathbf{x}_{\text{melhor}} = \mathbf{y}^{(k+1)}$ .

**Passo 10:** Se  $f(\mathbf{y}^{(k+1)}) < f(\mathbf{x}^{(k)})$ , então  $\mathbf{x}^{(k+1)} = \mathbf{y}^{(k+1)}$  e  $\theta^{(k+m)} = \theta^{(k)}$ ; senão,  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$  e  $\theta^{(k+m)} = \max(\theta^{(k)}/2, \theta_{\min})$ .

**Passo 11:** Se o critério de parada é atingido (acurácia desejada ou o número máximo de avaliação da função objetivo), retorne  $\mathbf{x}_{\text{melhor}}$  e  $f(\mathbf{x}_{\text{melhor}})$ ; caso contrário, incremente  $k = k + 1$  e vá para o Passo 1.

---

Para comparar o desempenho dos métodos  $q$ -G e F $q$ -G com perturbações gaussianas ambos os algoritmos foram aplicados na função teste Rastrigin Deslocada, descrita em Tang et al. (2007), dada pela expressão

$$f(\mathbf{x}) = \sum_{i=1}^N (z_i - 10\cos(2\pi z_i) + 10) - 330, \quad (4.14)$$

onde  $\mathbf{z} = \mathbf{x} - \mathbf{o}$  e  $\mathbf{x} \in [-5, 5]^N$ . O mínimo global é  $\mathbf{x}^* = \mathbf{o}$  com  $f(\mathbf{x}^*) = -330$ . O gráfico da função Rastrigin Deslocada é ilustrado na Figura 4.4.

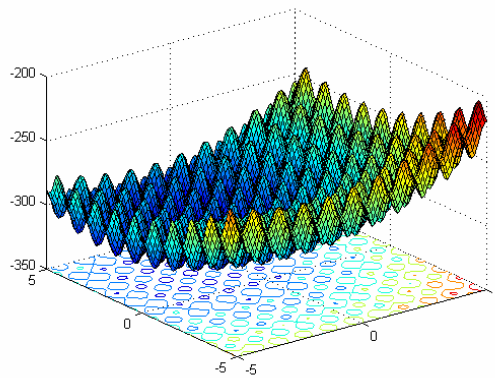


Figura 4.6 - Ilustração da função Rastrigin Deslocada para o caso bidimensional.  
Fonte: Tang et al. (2007).

Foram realizadas 25 execuções independentes de ambos os algoritmos a partir de 25 pontos iniciais uniformemente gerados dentro do espaço de busca dessa função. Além disso, considerou-se as dimensões  $N = 10$ ,  $N = 100$  e  $N = 1000$  e os seguintes critérios de parada: atingir o NAFO máximo de  $5000 \cdot N$  ou  $|f(\mathbf{x}_{melhor}) - f(\mathbf{x}^*)| \leq 1.0e-02$ . Nessa comparação, ambos os métodos utilizaram a estratégia de passo baseada em interpolação parabólica descrita na Seção 4.3. Os parâmetros de ajuste utilizados pelo método  $q$ -G foram selecionados com base na heurística descrita por Soterroni et al. (2015), e os parâmetros utilizados pelo método F $q$ -G baseados na heurística que será explicada no Capítulo 5. Assim como na comparação anterior, os dois algoritmos foram implementados em Fortran 90 para o compilador FORTRAN (*Intel Fortran Compiler*) e executado em uma máquina com processador Intel Core i5 com 8GB RAM e sistema operacional MAC OS X.

Na Figura 4.7 o número médio de avaliação da função objetivo ( $\text{NAFO}_{\text{Médio}}$ ) neces-

sário para que ambos os métodos atinjam um dos critérios de parada é apresentado. A Tabela 4.4 mostra o NAFO e o número de iterações médios usados por cada um dos métodos, além do tempo gasto (em segundos) nas 25 execuções dos algoritmos. Observe que para  $N = 10$ , o método  $q$ -G atinge a acurácia desejada com um número menor de avaliações da função objetivo em comparação com o  $Fq$ -G, comportamento que não se mantém para  $N = 100$  e  $N = 1000$ . O desempenho do  $Fq$ -G supera rapidamente o do método  $q$ -G quando  $N$  aumenta.

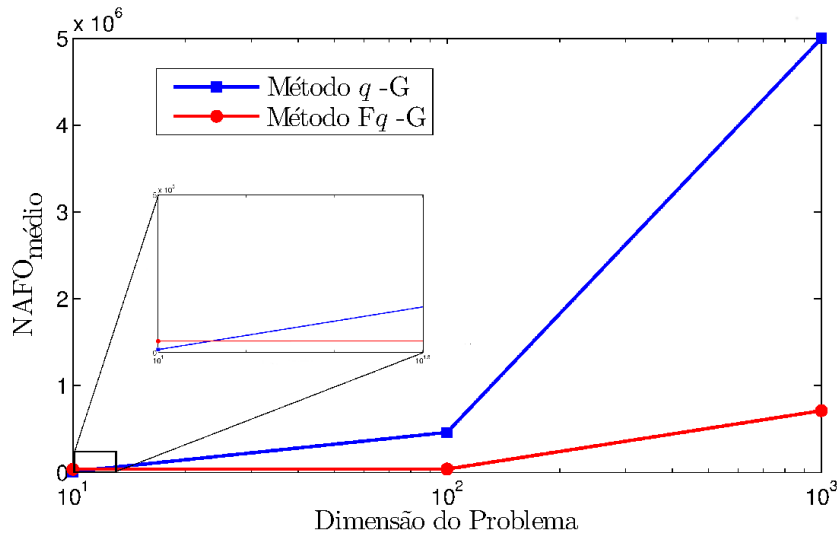


Figura 4.7 - Número médio de avaliação da função objetivo ( $NAFO_{M\u00e9dio}$ ) necess\u00e1rios para atingir o crit\u00e9rio de parada pelos m\u00e9todos  $q$ -G e  $Fq$ -G ao solucionarem a fun\u00e7\u00e3o Rastrigin Deslocada para as dimens\u00f5es  $N = 10, 100$  e  $1000$ .

Tabela 4.3 - Desempenho comparativo, em termos de NAFO, n\u00famero m\u00e9dio de itera\u00e7\u00f5es e tempo gasto, entre os m\u00e9todos  $q$ -G e  $Fq$ -G com perturba\u00e7\u00f5es gaussianas.

M\u00e9todo	Dimens\u00e3o	$NAFO_{M\u00e9dio}$	Itera\u00e7\u00f5es	Tempo (seg)
$q$ -G	10	9833	666	6.43e-02
	100	460733	4400	1.69e+01
	1000	5000571	4746	1.73e+03
$Fq$ -G	10	36509	7098	3.70e-01
	100	39069	7797	2.58e+00
	1000	710637	142081	4.47e+02

## 4.5 Convergência do método Fq-G para otimização global

Esta seção apresenta a prova de convergência do algoritmo Fq-G para um mínimo global no sentido probabilístico. Em geral, encontrar um mínimo global de uma função de alta dimensionalidade envolvendo milhares de variáveis de decisão é impossível a menos que o problema tenha uma estrutura especial. Por exemplo, encontrar o mínimo global de uma função com 1000 variáveis de decisão sujeita apenas às restrições laterais, o espaço de busca tem  $2^{1000} \approx 1.07 \times 10^{301}$  pontos de canto. Assim, mesmo com um orçamento computacional de bilhões de avaliações da função, o método não pode examinar mais de uma fração destes pontos de canto, e muito menos determinar o mínimo global. Contudo, continua a ser matematicamente interessante determinar se o algoritmo irá convergir para o mínimo global, caso pudesse ser executado indefinidamente. Além disso, o método proposto pode também ser aplicado a problemas de baixa dimensionalidade, onde o mínimo global pode ser encontrado dentro de um tempo razoável.

### 4.5.1 Prova de convergência

Suponha que o algoritmo Fq-G é aplicado a uma função objetivo determinística e mensurável  $f$  definida sobre o conjunto  $\mathcal{D} = [\mathbf{x}_{\min}, \mathbf{x}_{\max}] \subseteq \mathbb{R}^N$ . Além disso, assuma que  $f$  é contínua sobre o conjunto compacto  $\mathcal{D}$  para garantir a existência do seu mínimo global sobre  $\mathcal{D}$ . Uma vez que Fq-G é um algoritmo estocástico e a função objetivo  $f$  é determinística, sua convergência será estabelecida mostrando que segue a estrutura de Busca Aleatória Adaptativa Generalizada, do inglês, *Generalized Adaptive Random Search* (GARS) (REGIS, 2010) assim como foi feito para os métodos q-G e q-GC (GOUVÊA et al., 2016).

Antes de fornecer a prova de convergência do algoritmo Fq-G com perturbações gaussianas e com restrições laterais, a estrutura do GARS será apresentada. Além disso serão reproduzidos alguns teoremas presentes em (REGIS, 2010) que facilitam a compreensão da prova de convergência do Fq-G. Algumas definições utilizadas nesta seção são apresentadas no Apêndice A.

Considere um algoritmo estocástico cuja sequência de vetores aleatórios  $\{Y^{(k)}\}_{k \geq 1}$  são definidos no espaço de probabilidade  $(\Omega, \mathcal{B}, P)$ , onde o vetor aleatório  $Y^{(k)} : (\Omega, \mathcal{B}) \rightarrow (\mathcal{D}, \mathcal{B}(\mathcal{D}))$  representa o  $k$ -ésimo ponto de avaliação da função. Aqui,  $\Omega$  é o espaço amostral,  $\mathcal{B}$  é uma  $\sigma$ -álgebra dos subconjuntos de  $\Omega$ ,  $\mathcal{B}(\mathcal{D})$  são conjuntos de Borel em  $\mathcal{D}$ , e  $P$  é uma medida de probabilidade, isto é,  $P : \mathcal{B} \rightarrow [0, 1]$  tal que: (i)  $P(B) \geq 0$  para todo  $B \in \mathcal{B}$ ; (ii)  $P$  é  $\sigma$ -aditivo: Se  $\{B_k, k \geq 1\}$  são eventos



em  $\mathcal{B}$  que são disjuntos, então  $P(\cup_{k=1}^{\infty} B_k) = \sum_{k=1}^{\infty} P(B_k)$ ; e (iii)  $P(\Omega) = 1$ . Diz-se que esse algoritmo segue a estrutura de GARS para otimização global se apresentar a seguinte forma:

---

**Algoritmo 6:** Estrutura GARS (REGIS, 2010)

---

**Entradas:**

- (1) Uma função objetivo  $f : \mathcal{D} \rightarrow \mathbb{R}$ , onde  $\mathcal{D} \subseteq \mathbb{R}^N$ .
- (2) Uma transformação de absorção determinística  $\rho_{\mathcal{D}} : \mathbb{R}^N \rightarrow \mathcal{D}$ , isto é  $\rho_{\mathcal{D}}(\mathbf{x}) = \mathbf{x}$  para todo  $\mathbf{x} \in \mathcal{D}$ .
- (3) Uma coleção de elementos aleatórios intermediários  $\{\Lambda_{i,j} : (\Omega, \mathcal{B}) \rightarrow (\Omega_{i,j}, \mathcal{B}_{i,j}) : i \geq 0 \text{ and } j = 0, 1, \dots, r_i\}$  que são usados para determinar os vetores aleatórios experimentais. Esses  $\Lambda_{i,j}$ 's podem ser variáveis aleatórias, vetores aleatórios ou outros tipos de elementos aleatórios definidos no mesmo espaço de probabilidade  $(\Omega, \mathcal{B}, P)$ .

**Passo 0.** Faça  $k = 0$ .

**Passo 1.** Gere uma realização de  $Y^{(k)} : (\Omega, \mathcal{B}) \rightarrow (\mathbb{R}^N, \mathcal{B}(\mathbb{R}^N))$  como segue:

(1a) Para cada  $j = 0, \dots, r_k$ , gere uma realização de elementos aleatórios intermediários  $\Lambda_{k,j} : (\Omega, \mathcal{B}) \rightarrow (\Omega_{k,j}, \mathcal{B}_{k,j})$  de acordo com alguma distribuição de probabilidade .

(1b) Faça  $Y^{(k)} = \Phi_k(\mathcal{E}_k)$  para alguma função determinística  $\Phi_k$ , onde  $\mathcal{E}_k := \{\Lambda_{i,j} : i = 0, 1, \dots, k; j = 0, 1, \dots, r_i\}$  é a coleção de todos os elementos aleatórios intermediários até a iteração atual.

**Passo 2.** Faça  $X^{(k)} = \rho_{\mathcal{D}}(Y^{(k)})$  e avalie  $f(X^{(k)})$ .

**Passo 3.** Incremente  $k = k + 1$  e volte para o Passo 1.

---

Nos algoritmos que seguem a estrutura GARS, a sequência de pontos de avaliação da função é dada por  $\{X^{(k)}\}_{k \geq 1}$ . Se a realização de  $Y^{(k)}$  pertence a  $\mathcal{D}$ , então  $X^{(k)} = Y^{(k)}$ . E,  $\{X^{(k)*}\}_{k \geq 1}$  é sequência dos melhores pontos encontrados pelo algoritmo, onde  $X^{(1)*} = X^{(1)}$ , e para  $k > 1$ ,  $X^{(k)*} = X^{(k)}$  se  $f(X^{(k)}) < f(X^{(k-1)*})$ , enquanto  $X^{(k)*} = X^{(k-1)*}$  caso contrário.

Regis (2010) afirma que os algoritmos que seguem a estrutura de GARS convergem para o mínimo global de  $f$  sobre  $\mathcal{D}$  quase certamente se  $\{f(X^{(k)*})\}_{k \geq 1}$  converge para  $f^* := \inf_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x})$  quase certamente. Segundo Resnick (1999) a sequência de variáveis aleatórias  $\{X^{(k)}\}_{k \geq 1}$  converge para  $X^*$  quase certamente se

$$P\{w : \lim_{k \rightarrow \infty} X^{(k)}(w) = X^*(w)\} = 1.$$

A seguir são reproduzidos alguns dos teoremas apresentados por Regis (2010), que

forneem as condições necessárias para a prova de convergência de algoritmos que seguem a estrutura GARS para o mínimo global no sentido probabilístico.

Para cada  $k \geq 0$ , Regis (2010) define  $\sigma(\mathcal{E}_k)$  sendo a  $\sigma$ -álgebra gerada por elementos aleatórios em  $\mathcal{E}_k$ . Pode-se pensar que  $\sigma(\mathcal{E}_k)$  como representando todas as informações que podem ser derivadas a partir dos elementos aleatórios  $\mathcal{E}_k$ .

**Teorema 4.** (REGIS, 2010) *Seja  $f$  uma função em  $\mathcal{D} \subseteq \mathbb{R}^N$  tal que  $f^* := \inf_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x}) > -\infty$ . Suponha que o algoritmo GARS satisfaça a seguinte propriedade. Para todo  $\epsilon > 0$ , existe  $0 < L(\epsilon) < 1$  tal que*

$$P[X^{(k_t)} \in \mathcal{D} : f(X^{(k_t)}) < f^* + \epsilon \mid \sigma(\mathcal{E}_{(k_t)-1})] \geq L(\epsilon) \quad (4.15)$$

para alguma subsequência  $\{k_t\}_{t \geq 1}$ . Então,  $f(X^{(k)^*}) \rightarrow f^*$  quase certamente (q.c.).

*Demonstração.* Fixe  $\epsilon > 0$  e defina  $\mathcal{S}_\epsilon := \{\mathbf{x} \in \mathcal{D} : f(\mathbf{x}) < f^* + \epsilon\}$ . Por hipótese,

$$P[X^{(k_t)} \in \mathcal{S}_\epsilon \mid \sigma(\mathcal{E}_{(k_t)-1})] \geq L(\epsilon), \text{ para algum } t \geq 1.$$

Para cada  $t \geq 1$ , tem-se

$$\begin{aligned} P[X^{(k_1)} \notin \mathcal{S}_\epsilon, X^{(k_2)} \notin \mathcal{S}_\epsilon, \dots, X^{(k_t)} \notin \mathcal{S}_\epsilon] = \\ \prod_{i=1}^t P[X^{(k_i)} \notin \mathcal{S}_\epsilon \mid X^{(k_1)} \notin \mathcal{S}_\epsilon, X^{(k_2)} \notin \mathcal{S}_\epsilon, \dots, X^{(k_{i-1})} \notin \mathcal{S}_\epsilon]. \end{aligned}$$

Pelo condicionamento dos elementos aleatórios em  $\mathcal{E}_{(k_i)-1}$ , é possível verificar que para cada  $\epsilon > 0$ , tem-se  $P[X^{(k_i)} \in \mathcal{S}_\epsilon \mid X^{(k_1)} \notin \mathcal{S}_\epsilon, X^{(k_2)} \notin \mathcal{S}_\epsilon, \dots, X^{(k_{i-1})} \notin \mathcal{S}_\epsilon] \geq L(\epsilon)$ . Então,

$$\begin{aligned} P[X^{(k_1)} \notin \mathcal{S}_\epsilon, X^{(k_2)} \notin \mathcal{S}_\epsilon, \dots, X^{(k_t)} \notin \mathcal{S}_\epsilon] \\ = \prod_{i=1}^t P[X^{(k_i)} \notin \mathcal{S}_\epsilon \mid X^{(k_1)} \notin \mathcal{S}_\epsilon, X^{(k_2)} \notin \mathcal{S}_\epsilon, \dots, X^{(k_{i-1})} \notin \mathcal{S}_\epsilon] \\ \stackrel{P[A|B]=1-P[A^c|B]}{=} \prod_{i=1}^t (1 - P[X^{(k_i)} \in \mathcal{S}_\epsilon \mid X^{(k_1)} \notin \mathcal{S}_\epsilon, X^{(k_2)} \notin \mathcal{S}_\epsilon, \dots, X^{(k_{i-1})} \notin \mathcal{S}_\epsilon]) \\ \leq (1 - L(\epsilon))^t. \end{aligned} \quad (4.16)$$

Observe que se  $i$  é o menor índice tal que  $X^{(i)} \in \mathcal{S}_\epsilon$ , segue que  $X^{(i)^*} = X^{(i)}$  e  $X^{(k)^*} \in \mathcal{S}_\epsilon$  para todo  $k \geq i$ . Consequentemente, se  $X^{(k_i)^*} \notin \mathcal{S}_\epsilon$ , então  $X^{(k_1)} \notin$

$\mathcal{S}_\epsilon, X^{(k_2)} \notin \mathcal{S}_\epsilon, \dots, X^{(k_t)} \notin \mathcal{S}_\epsilon$ . Portanto, para cada  $t \geq 1$ , tem-se

$$\begin{aligned} P[f(X^{(k_t)^*}) - f^* \geq \epsilon] &= P[f(X^{(k_t)^*}) \geq f^* + \epsilon] \\ &= P[X^{(k_t)^*} \notin \mathcal{S}_\epsilon] \leq P[X^{(k_1)} \notin \mathcal{S}_\epsilon, X^{(k_2)} \notin \mathcal{S}_\epsilon, \dots, X^{(k_t)} \notin \mathcal{S}_\epsilon] \stackrel{\text{Eq. (4.16)}}{\leq} (1 - L(\epsilon))^t, \end{aligned}$$

então,

$$\lim_{t \rightarrow \infty} P[f(X^{(k_t)^*}) - f^* \geq \epsilon] = 0,$$

isto é,  $f(X^{(k_t)^*}) \rightarrow f^*$  em probabilidade. Pelo Teorema 3.2.1b<sup>1</sup> de (RESNICK, 1999) segue que  $f(X^{(k_{t(i)})^*}) \rightarrow f^*$  quase certamente (q. c.) quando  $i \rightarrow \infty$  para alguma subsequência  $\{k_{t(i)}\}_{i \geq 1}$ .

Como  $f^* > -\infty$  e  $\{f(X^{(k)^*})\}_{k \geq 1}$  é monotonicamente não crescente, segue que existe  $\lim_{k \rightarrow \infty} f(X^{(k)^*}(\omega))$  para cada ponto de amostragem  $\omega$ . Finalmente, visto que a subsequência  $f(X^{(k_{t(i)})^*}) \rightarrow f^*$  q.c., segue que  $f(X^{(k)^*}) \rightarrow f^*$  q.c.  $\square$

No Teorema 4, a condição da Equação (4.15) é exigida apenas para uma subsequência de iterações para garantir a convergência q. c. para o mínimo global. O teorema seguinte diz que se para um ponto qualquer do espaço de busca  $\mathcal{D}$ , a probabilidade de  $Y^{(k_t)}$  ser próxima deste ponto for positiva, então existe uma subsequência  $f(X^{(k)^*}) \rightarrow f^*$ .

**Teorema 5.** (REGIS, 2010) *Seja  $f$  uma função sobre  $\mathcal{D} \subseteq \mathbb{R}^N$  tal que  $f^* := \inf_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x}) > -\infty$ . Além disso, suponha que exista  $\mathbf{x}^*$ , um mínimo global de  $f$  sobre  $\mathcal{D}$  e que  $f$  é contínua em  $\mathbf{x}^*$ . Suponha também que o algoritmo que segue a estrutura GARS satisfaça a seguinte propriedade:*

*Para todo  $\mathbf{w} \in \mathcal{D}$  e  $\delta > 0$ , existe  $0 < \nu(\mathbf{w}, \delta) < 1$  tal que*

$$P[Y^{(k_t)} \in B(\mathbf{w}, \delta) \cap \mathcal{D} \mid \sigma(\mathcal{E}_{(k_t)-1})] \geq \nu(\mathbf{w}, \delta), \quad (4.17)$$

*para alguma subsequência  $\{k_t\}_{t \geq 1}$ .  $B(\mathbf{w}, \delta)$  é a bola aberta centrada em  $\mathbf{w}$  com raio  $\delta$ . Então,  $f(X^{(k)^*}) \rightarrow f^*$  quase certamente (q.c.).*

*Demonstração.* Fixe  $\epsilon > 0$  e  $t \geq 1$ . Como  $f$  é contínua em  $\mathbf{x}^*$ , existe  $\delta(\epsilon) > 0$  tal que  $|f(\mathbf{x}) - f(\mathbf{x}^*)| < \epsilon$  sempre que  $\|\mathbf{x} - \mathbf{x}^*\| < \delta(\epsilon)$ . Então, o evento

$$[X^{(k_t)} \in \mathcal{D} : f(X^{(k_t)}) < f(\mathbf{x}^*) + \epsilon] = [X^{(k_t)} \in \mathcal{D} : f(X^{(k_t)}) - f(\mathbf{x}^*) < \epsilon]$$

<sup>1</sup>Suponha que  $\{X^{(k)}\}_{k \geq 1}$  sejam variáveis aleatórias reais.  $X^{(k)} \rightarrow X$  em probabilidade se e somente se cada subsequência  $X^{(k_t)}$  contém uma subsequência  $X^{(k_{t(i)})}$  que converge quase certamente para  $X$ .

$$\supseteq [X^{(k_t)} \in \mathcal{D} : \|X^{(k_t)} - \mathbf{x}^*\| < \delta(\epsilon)],$$

e assim,

$$\begin{aligned} & P[X^{(k_t)} \in \mathcal{D} : f(X^{(k_t)}) < f(\mathbf{x}^*) + \epsilon \mid \sigma(\mathcal{E}_{(k_t)-1})] \\ & \geq P[X^{(k_t)} \in \mathcal{D} : \|X^{(k_t)} - \mathbf{x}^*\| < \delta(\epsilon) \mid \sigma(\mathcal{E}_{(k_t)-1})] \\ & = P[X^{(k_t)} \in B(\mathbf{x}^*, \delta(\epsilon)) \cap \mathcal{D} \mid \sigma(\mathcal{E}_{(k_t)-1})] \geq P[Y^{(k_t)} \in B(\mathbf{x}^*, \delta(\epsilon)) \cap \mathcal{D} \mid \sigma(\mathcal{E}_{(k_t)-1})] \\ & \geq \nu(\mathbf{x}^*, \delta(\epsilon)) =: L(\epsilon). \end{aligned}$$

Claramente,  $L(\epsilon) > 0$  desde que  $\delta(\epsilon) > 0$ . Pelo Teorema 4, segue que  $f(X^{(k)^*}) \rightarrow f^*$  q.c.  $\square$

**Teorema 6.** (REGIS, 2010) *Seja  $\mathcal{D}$  um subconjunto de  $\mathbb{R}^N$  tal que*

$$\psi_{\mathcal{D}}(\delta) := \inf_{\mathbf{w} \in \mathcal{D}} \mu(B(\mathbf{w}, \delta) \cap \mathcal{D}) > 0$$

para cada  $\delta > 0$ , onde  $\mu$  é a medida de Lebesgue em  $\mathbb{R}^N$ . Seja  $f$  uma função real em  $\mathcal{D}$  tal que  $f^* := \inf_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x}) > -\infty$ . Além disso, suponha que exista  $\mathbf{x}^*$ , um mínimo global, em  $f$  contínua sobre  $\mathcal{D}$ . Considere o algoritmo GARS e suponha que existe uma subsequência  $\{k_t\}_{t \geq 1}$  tal que para cada  $t \geq 1$ ,  $Y_{k_t}$  tem uma densidade condicional  $g_{k_t}(\mathbf{y} \mid \sigma(\mathcal{E}_{(k_t)-1}))$  satisfazendo a seguinte condição:

$$\mu(\{\mathbf{y} \in \mathcal{D} : h(\mathbf{y}) = 0\}) = 0, \text{ onde } h(\mathbf{y}) := \inf_{t \geq 1} g_{k_t}(\mathbf{y} \mid \sigma(\mathcal{E}_{(k_t)-1})).$$

Então  $f(X^{(k)^*}) \rightarrow f^*$  q. c.

*Demonstração.* Fixe  $\delta > 0$  e  $\mathbf{w} \in \mathcal{D}$ . Para todo  $t \geq 1$ , tem-se

$$\begin{aligned} P[Y^{(k_t)} \in B(\mathbf{w}, \delta) \cap \mathcal{D} \mid \sigma(\mathcal{E}_{(k_t)-1})] &= \int_{B(\mathbf{w}, \delta) \cap \mathcal{D}} g_{k_t}(\mathbf{y} \mid \sigma(\mathcal{E}_{(k_t)-1})) dy \\ &\geq \int_{B(\mathbf{w}, \delta) \cap \mathcal{D}} \inf g_{k_t}(\mathbf{y} \mid \sigma(\mathcal{E}_{(k_t)-1})) dy = \int_{B(\mathbf{w}, \delta) \cap \mathcal{D}} h(\mathbf{y}) dy =: \nu(\mathbf{w}, \delta). \end{aligned}$$

Como  $h(\mathbf{y})$  é uma função não negativa em  $\mathcal{D}$ ,  $\mu(\{\mathbf{y} \in \mathcal{D} : h(\mathbf{y}) = 0\}) = 0$  e  $\mu(B(\mathbf{w}, \delta) \cap \mathcal{D}) \geq \psi_{\mathcal{D}}(\delta) > 0$ , segue que  $\nu(\mathbf{w}, \delta) > 0$ . Pelo Teorema 5,  $f(X^{(k)^*}) \rightarrow f^*$  q.c.  $\square$

Como em Gouvêa et al. (2016), um dos teoremas usados em (REGIS, 2010) é usado para provar a convergência do algoritmo Fq-G que usa perturbações gaussianas. A seguir está um caso especial do Teorema 6 em Regis (2010) que pode ser aplicado

em um algoritmo que usa distribuição gaussiana para gerar alguns dos seus vetores aleatórios de  $N$  dimensões.

**Teorema 7.** *Suponha que  $\mathcal{D}$  é um subconjunto limitado de  $\mathbb{R}^N$  tal que*

$$\psi_{\mathcal{D}}(\delta) := \inf_{\mathbf{w} \in \mathcal{D}} \mu(B(\mathbf{w}, \delta) \cap \mathcal{D}) > 0, \quad \text{para todo } \delta > 0, \quad (4.18)$$

onde  $B(\mathbf{w}, \delta)$  é a bola aberta centrada em  $\mathbf{w}$  com raio  $\delta$  e  $\mu$  é a medida de Lebesgue em  $\mathbb{R}^N$ . Seja  $f$  uma função mensurável real definida em  $\mathcal{D}$  tal que  $f^* := \inf_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x}) > -\infty$  e assumamos que exista um mínimo global  $\mathbf{x}^*$  de  $f$  sobre  $\mathcal{D}$  e que  $f$  é contínua em  $\mathbf{x}^*$ . Considere o algoritmo GARS cujos vetores aleatórios são  $\{Y^{(k)} : k \geq 0\}$  e cuja sequência de melhores vetores aleatórios são  $\{Y^{(k)*} : k \geq 0\}$ . Suponha que exista uma subsequência  $\{k_t\}_{t \geq 1}$  tal que para cada  $t \geq 1$ , tenha-se  $Y^{(k_t)} = U^{(t)} + Z^{(t)}$ , onde  $U^{(t)} = \Phi_t(\mathcal{E}_{(k_t)-1})$  para alguma função determinística  $\Phi_t$  e  $Z^{(t)}$  é um vetor aleatório cuja distribuição condicional é dada por  $\sigma(\mathcal{E}_{(k_t)-1})$  (a  $\sigma$ -álgebra gerada por  $\mathcal{E}_{(k_t)-1}$ ) é uma distribuição gaussiana com média  $0_{N \times 1}$  e matriz de covariância  $V_t$ , i.e., sua densidade condicional é dada por

$$g_t(\mathbf{z} \mid \sigma(\mathcal{E}_{(k_t)-1})) = (2\pi)^{-N/2} [\det(V_t)]^{-1/2} \exp\left(-\frac{1}{2} \mathbf{z}^T V_t^{-1} \mathbf{z}\right), \quad \mathbf{z} \in \mathbb{R}^N. \quad (4.19)$$

Para cada  $t \geq 1$ , sejam  $\lambda_t$  e  $\lambda_t^*$ , respectivamente, o menor e o maior autovalor da matriz de covariância  $V_t$ . Além disso, suponhamos que  $\inf_{t \geq 1} \lambda_t > 0$  e  $\sup_{t \geq 1} \lambda_t^* < \infty$ . Então  $f(X^{(k_t)*}) \rightarrow f^*$  quase certamente (q.c.).

Observe que na prática tem-se  $U^{(t)} = \Phi_t(\mathcal{E}_{(k_t)-1}) = X^{((k_t)-1)*}$ , que é o vetor aleatório representando a melhor solução encontrada depois de  $(k_t) - 1$  avaliações da função.

*Demonstração.* Para cada  $t \geq 1$ , observe que a distribuição condicional de  $Y^{(k_t)}$  dada por  $\sigma(\mathcal{E}_{(k_t)-1})$  é uma distribuição gaussiana com densidade condicional

$$g_t(\mathbf{y} \mid \sigma(\mathcal{E}_{(k_t)-1})) = (2\pi)^{-N/2} [\det(V_t)]^{-1/2} \exp\left(-\frac{1}{2} (\mathbf{y} - \mathbf{u}^{(t)})^T V_t^{-1} (\mathbf{y} - \mathbf{u}^{(t)})\right), \quad \mathbf{y} \in \mathbb{R}^N,$$

onde  $V_t$  é uma matriz definida positiva e simétrica e  $\mathbf{u}^{(t)} = \Phi_t(\{\lambda_{i,j} : i = 1, 2, \dots, (k_t) - 1; j = 1, 2, \dots, t_i\})$  e os  $\lambda_{i,j}$  são as realizações dos elementos aleatórios em  $\mathcal{E}_{(k_t)-1}$ . Agora, para cada  $t \geq 1$  e  $\mathbf{y} \in \mathcal{D}$ , tem-se

$$\begin{aligned} (\mathbf{y} - \mathbf{u}^{(t)})^T V_t^{-1} (\mathbf{y} - \mathbf{u}^{(t)}) &\leq \| \mathbf{y} - \mathbf{u}^{(t)} \|_2 \| V_t^{-1} (\mathbf{y} - \mathbf{u}^{(t)}) \|_2 \\ &\leq \| \mathbf{y} - \mathbf{u}^{(t)} \|_2^2 \| V_t^{-1} \|_2, \end{aligned}$$

como  $V_t^{-1}$  é uma matriz simétrica, então  $\|V_t^{-1}\|_2 = \frac{1}{\lambda_t}$ , logo

$$(\mathbf{y} - \mathbf{u}^{(t)})^T V_t^{-1} (\mathbf{y} - \mathbf{u}^{(t)}) \leq \|\mathbf{y} - \mathbf{u}^{(t)}\|_2^2 \|V_t^{-1}\|_2 \leq \frac{\text{diam}(\mathcal{D})^2}{\lambda_t}, \quad (4.20)$$

onde  $\text{diam}(\mathcal{D}) = \sup_{\mathbf{x}, \mathbf{y} \in \mathcal{D}} \|\mathbf{x} - \mathbf{y}\|$ . Como  $\mathcal{D}$  é limitado, segue que  $\text{diam}(\mathcal{D}) < \infty$ . Além disso, como  $\exp^{-y/2}$  é monotonicamente não crescente, então

$$\exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{u}^{(t)})^T V_t^{-1} (\mathbf{y} - \mathbf{u}^{(t)})\right) \geq \exp\left(-\frac{\text{diam}(\mathcal{D})^2}{2\lambda_t}\right). \quad (4.21)$$

Como o determinante de uma matriz é o produto dos seus autovalores, segue que

$$\det(V_t) \leq (\lambda_t^*)^N, \quad (4.22)$$

onde  $\lambda_t^*$  é o maior autovalor de  $V_t$ . Das Eq. (4.20), (4.21) e (4.22) resulta que para cada  $\mathbf{y} \in \mathcal{D}$ ,

$$\begin{aligned} g_{k_t}(\mathbf{y} \mid \sigma(\mathcal{E}_{(k_t)-1})) &\geq (2\pi)^{-N/2} (\lambda_t^*)^{-N/2} \exp\left(-\frac{\text{diam}(\mathcal{D})^2}{2\lambda_t}\right) \\ &\geq (2\pi)^{-N/2} \left(\sup_{t \geq 1} \lambda_t^*\right)^{-N/2} \exp\left(-\frac{\text{diam}(\mathcal{D})^2}{\inf_{t \geq 1} \lambda_t}\right) > 0. \end{aligned}$$

Consequentemente, tem-se

$$h(\mathbf{y}) := \inf_{t \geq 1} g_{k_t}(\mathbf{y} \mid \sigma(\mathcal{E}_{(k_t)-1})) \geq (2\pi)^{-N/2} \left(\sup_{t \geq 1} \lambda_t^*\right)^{-N/2} \exp\left(-\frac{\text{diam}(\mathcal{D})^2}{\inf_{t \geq 1} \lambda_t}\right) > 0.$$

Pelo Teorema 6, segue que  $f(X^{(k)^*}) \rightarrow f^*$ . □

A convergência do método Fq-G será provada no teorema a seguir.

**Teorema 8.** *Suponha que Fq-G é aplicado a uma função real mensurável  $f$  sobre  $\mathcal{D} = [\mathbf{x}_{\min}, \mathbf{x}_{\max}] \subseteq \mathbb{R}^N$  tal que  $f^* = \inf_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x}) > -\infty$ . Além disso, suponha que exista um mínimo global  $\mathbf{x}^*$  de  $f$  sobre  $\mathcal{D}$  e que  $f$  é contínua em  $\mathbf{x}^*$ . Então  $f(\mathbf{x}_{\text{melhor}}) \rightarrow f^*$  q.c.*

*Demonstração.* É possível verificar que o espaço de busca  $\mathcal{D} = [\mathbf{x}_{\min}, \mathbf{x}_{\max}]$  é um subconjunto limitado de  $\mathbb{R}^N$  que satisfaz a condição  $\psi_{\mathcal{D}}(\delta) > 0$  para todo  $\delta > 0$  no Teorema 7 (Proposição 1<sup>2</sup> em Regis (2010)).

---

<sup>2</sup>Seja  $\mathcal{D} = [a, b] \subseteq \mathbb{R}^N$  um hipercubo fechado e  $\ell(\mathcal{D})$  o comprimento de um dos lados de

A mesma notação na descrição do algoritmo Fq-G com perturbações gaussianas (Algoritmo 5) é utilizada, exceto pelas letras maiúsculas utilizados em alguns vetores para enfatizar que eles são vetores aleatórios (por exemplo,  $X^{(k)}$  e  $S^{(k)}$  em vez de  $\mathbf{x}^{(k)}$  e  $\mathbf{s}^{(k)}$ ). Lembrando que depois de avaliar  $f$  em  $X^{(0)}$ , Fq-G realiza uma série de iterações onde o  $q$ -gradiente aproximado  $\widehat{\nabla}_{\mathbf{q}}f(X^{(k)})$  é usado para determinar a próxima iteração  $X^{(k+1)}$ , e depois da  $m$ -ésima iteração uma iteração gaussiana é gerada. Em particular, na Iteração 1,  $k = 0$ ,  $S^{(0)}$  e  $X^{(1)}$  são gerados e a função é avaliada nestes pontos. Na Iteração 2,  $k = 1$ ,  $S^{(1)}$  e  $X^{(2)}$  são gerados e a função é avaliada nestes pontos, e assim por diante até  $k = m - 1$  (Iteração  $m$ ). Quando  $k = m$ ,  $Z^{(m)}$  é gerado usando uma distribuição gaussiana com média  $\mathbf{0}$  e matriz de covariância  $(\theta^{(k)})^2\mathbf{I}$  e então  $W^{(m+1)}$  é computado. Depois deste passo gaussiano, o algoritmo volta a usar a aproximação  $q$ -gradiente para gerar os vetores aleatórios  $N$ -dimensionais. Em geral, para qualquer iteração  $t > 1$  tal que  $k = t - 1$  não é um múltiplo de  $m$ ,  $S^{(t-1)}$  e  $X^{(t)}$  são gerados e a função é avaliada nestes pontos. Além disso, para qualquer iteração  $t > 1$  tal que  $k = t - 1$  é múltiplo de  $m$ ,  $Z^{(t-1)}$  é gerado usando a distribuição gaussiana e então  $W^{(t)}$  é calculado.

Por conveniência, define-se  $\{U^{(j)}\}_{j \geq 0}$  como sendo a sequência de pontos de avaliação da função gerados pelo algoritmo *antes* da função absorção  $\rho_{\mathcal{D}}$  ser aplicada (por exemplo,  $U^{(j)} = S^{(k)} = \mathbf{q}^{(k)} \odot X^{(k)}$  para algum  $k$  ou  $U^{(j)} = X^{(k+1)} = X^{(k)} + \alpha^{(k)}\mathbf{d}^{(k)}$  para algum  $k$ ). Isto é,

$$\begin{aligned} U^{(0)} &= X^{(0)}, U^{(1)} = S^{(0)}, U^{(2)} = X^{(1)}, U^{(3)} = S^{(1)}, U^{(4)} = X^{(2)}, \dots, \\ U^{(2m-1)} &= S^{(m-1)}, U^{(2m)} = X^{(m)}, U^{(2m+1)} = W^{(m+1)}, \\ U^{(2m+2)} &= S^{(m+1)}, U^{(2m+3)} = X^{(m+2)}, \dots \end{aligned}$$

e assim por diante. Note que  $\{\rho_{\mathcal{D}}(U^{(j)})\}_{j \geq 0}$  é o conjunto de todos os pontos onde a função objetivo é avaliada se o algoritmo for executado indefinidamente. Além disso, seja  $\mathcal{E}_j$  o conjunto de todos os vetores aleatórios que foram gerados *até*  $U^{(j)}$  (isto inclui os vetores aleatórios da forma  $\mathbf{q}^{(i)}$ , e os vetores aleatórios gaussianos anteriores).

Seja  $\{U^{(j_h)}\}_{h \geq 1}$  uma subsequência  $\{U^{(j)}\}_{j \geq 0}$  que consiste de todos os passos gaussianos. É possível verificar que  $j_h = (2m + 1) + (h - 1)(2m - 1) = 2 + h(2m - 1)$  para

---

$\mathcal{D}$ . Se  $0 < \delta \leq \frac{1}{2}\ell(\mathcal{D})$ , então  $\psi_{\mathcal{D}}(\delta) = \left(\frac{\delta}{2}\right)^N \frac{\pi^{N/2}}{\Gamma(\frac{N}{2}+1)}$ . Aqui,  $\Gamma$  é a função Gama definida por  $\Gamma(k) := \int_0^\infty x^{k-1}e^{-x}dx$ .

todo  $h \geq 1$ . A partir da descrição do algoritmo,

$$U^{(j_h)} = W^{(hm+1)} = X^{(hm)} + Z^{(hm)}, \quad \text{para todo } h \geq 1, \quad (4.23)$$

onde  $Z^{(hm)}$  é um vetor aleatório cuja distribuição condicional é dada por  $\sigma(\mathcal{E}_{j_{h-1}})$  (a  $\sigma$ -álgebra é gerada pelo vetor aleatório em  $\mathcal{E}_{j_{h-1}}$ ) é uma distribuição gaussiana com média no vetor  $0_{N \times 1}$  e matriz de covariância  $V_h = (\theta^{(hm)})^2 I_N$ . Na Equação (4.23), note que  $X^{(hm)} = \Phi_h(\mathcal{E}_{j_{h-1}})$  para alguma função determinística  $\Phi_h$ . Além disso, note que  $(\theta^{(hm)})^2$  é o único autovalor de  $V_h$  (e tem multiplicidade  $N$ ). Como  $\inf_{h \geq 1} \theta^{(hm)} \geq \theta_{\min} > 0$  e  $\sup_{h \geq 1} \theta^{(hm)} \leq \theta^{(0)} < \infty$ , segue da Proposição 7 que  $f(\mathbf{x}_{\text{melhor}}) \rightarrow f^*$  q.c.  $\square$



## 5 Resultados Numéricos

Experimentos numéricos foram realizados com o objetivo de avaliar a eficiência do método Fq-G na resolução de problemas de otimização de grande porte. Esses experimentos consistem na resolução de 27 funções teste com até 1000 dimensões. Os principais resultados desses experimentos são apresentados nesse capítulo, juntamente com a comparação entre o Fq-G e 14 algoritmos evolutivos desenvolvidos para solucionar problemas com um grande número de variáveis. Além disto, são exibidos os valores fixos dos parâmetros de ajuste definidos a partir de rodadas preliminares do algoritmo do Fq-G.

### 5.1 Configuração dos Parâmetros de Ajuste

Os dois principais parâmetros do método Fq-G são  $\sigma^{(0)}$  and  $\beta$ , pois é através deles que o algoritmo consegue determinar a direção de busca e o tamanho do passo. A configuração ideal desses parâmetros depende do problema. Rodados preliminares demonstraram que  $\sigma^{(0)} = 1.5L$ , onde  $L = \|\mathbf{x}_{\max} - \mathbf{x}_{\min}\|_2$  é a maior distância dentro do espaço de busca, fornece uma heurística simples para configurar este parâmetro. O valor de  $\beta$ , que controla a velocidade com que o algoritmo muda de busca global para busca local, depende da dimensão do problema  $N$ . Como uma regra, para  $N = 100, 500$  e  $1000$ , pode-se assumir  $\beta = 0.9999, 0.99995$  e  $0.99999$ , respectivamente. Naturalmente, a configuração final de  $\beta$  pode ser balanceada com o número máximo de avaliação da função objetivo disponível. O algoritmo Fq-G é mais robusto com relação aos parâmetros  $\theta^{(0)}, \theta_{\min}$  e  $m$ . Em todas as simulações foram usados  $m = N$ ,  $\theta^{(0)} = 0.2L$ , e  $\theta_{\min} = 0.0125L$ .

### 5.2 Comparações com Algoritmos Evolutivos

Nesta seção são apresentadas comparações entre o método Fq-G e alguns AEs na resolução de dois conjuntos de funções teste utilizados nas competições organizadas no *IEEE Congress on Evolutionary Computation* de 2008 (TANG et al., 2007) e de 2010 (TANG et al., 2009) (CEC'2008 e CEC'2010). Uma breve descrição desses AEs pode ser vista no Apêndice B. Assim como em Tang et al. (2007) e em Tang et al. (2009), para ambos os conjuntos de problemas, os pontos iniciais foram gerados por uma distribuição uniforme dentro do espaço de busca, além disso, todas as funções têm os seus respectivos mínimos deslocados do zero a fim de evitar a exploração das simetrias do espaço de busca pelos algoritmos. O critério de parada utilizado foi o número máximo de avaliação da função objetivo ( $\text{NAFO}_{\max}$ ) igual a  $5000 \cdot N$ , para as funções do CEC'2008, e  $3000 \cdot N$ , para os problemas do CEC'2010. O algoritmo do

Fq-G foi implementado em Fortran 90 para o compilador FORTRAN (*Intel Fortran Compiler*) e executado em uma máquina com processador Intel Core i5 com 8GB RAM e sistema operacional MAC OS X.

### 5.2.1 Primeiro conjunto de funções teste

O CEC'2008 (TANG et al., 2007) consiste de 7 funções para as dimensões 100, 500 e 1000, cujas expressões matemáticas são apresentadas a seguir. O caso bidimensional da função Rastrigin Deslocada foi mostrado na Seção 4.4, o caso bidimensional das outras 6 funções são ilustrados na Figura 5.2.1. O mínimo global da F1 a F6 é  $\mathbf{x}^* = \mathbf{o}$  com  $f(\mathbf{x}^*)$  diferente para cada uma das funções teste e o mínimo global da F7 é desconhecido.

- **Unimodal**

- **F1 - Sphere Deslocada**

$$f(\mathbf{x}) = \sum_{i=1}^N z_i^2 - 450,$$

$$\mathbf{z} = \mathbf{x} - \mathbf{o}, \mathbf{x} \in [-100, 100]^N \text{ e } f(\mathbf{x}^*) = -450.$$

- **F2 - Problema Schwefel 2.21**

$$f(\mathbf{x}) = \max_i \{|z_i|, 1 \leq i \leq N\} - 450,$$

$$\mathbf{z} = \mathbf{x} - \mathbf{o}, \mathbf{x} \in [-100, 100]^N \text{ e } f(\mathbf{x}^*) = -450.$$

- **Multimodal**

- **F3 - Rosenbrock Deslocada**

$$f(\mathbf{x}) = \sum_{i=1}^{N-1} (100(z_i^2 - z_{i+1})^2 + (z_i + 1)^2) + 390,$$

$$\mathbf{z} = \mathbf{x} - \mathbf{o}, \mathbf{x} \in [-100, 100]^N \text{ e } f(\mathbf{x}^*) = 390.$$

- **F4 - Rastrigin Deslocada**

$$f(\mathbf{x}) = \sum_{i=1}^N (z_i - 10 \cos(2\pi z_i) + 10) - 330,$$

$$\mathbf{z} = \mathbf{x} - \mathbf{o}, \mathbf{x} \in [-5, 5]^N \text{ e } f(\mathbf{x}^*) = -330.$$

– **F5 - Griewank Deslocada**

$$f(\mathbf{x}) = \sum_{i=1}^N \frac{z_i^2}{4000} - \prod_{i=1}^N \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 - 180,$$

$$\mathbf{z} = \mathbf{x} - \mathbf{o}, \mathbf{x} \in [-600, 600]^N \text{ e } f(\mathbf{x}^*) = -180.$$

– **F6 - Ackley Deslocada**

$$f(\mathbf{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{N} \sum_{i=1}^N z_i^2}\right) - \exp\left(\frac{1}{N} \sum_{i=1}^N \cos(2\pi z_i)\right) + 20 + e - 140$$

$$\mathbf{z} = \mathbf{x} - \mathbf{o}, \mathbf{x} \in [-32, 32]^N \text{ e } f(\mathbf{x}^*) = -140.$$

– **F7 - FastFractal “DoubleDip”**

$$f(\mathbf{x}) = \sum_{i=1}^N \text{fractal1D}(x_i + \text{twist}(x_{(i \bmod N)+1})), \quad (5.1)$$

onde

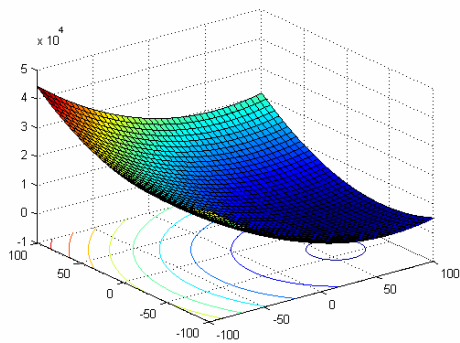
$$\text{twist}(y) = 4(y^4 - 2y^3 + y^2),$$

$$\text{fractal1D}(x) \approx \sum_{k=1}^3 \sum_{o=1}^{2^{k-1} \text{ran2}(o)} \sum_{s=1}^{\text{ran1}(o)} \text{doubledip}(x, \text{ran1}(o), \frac{1}{2^{k-1}(2 - \text{ran1}(o))}),$$

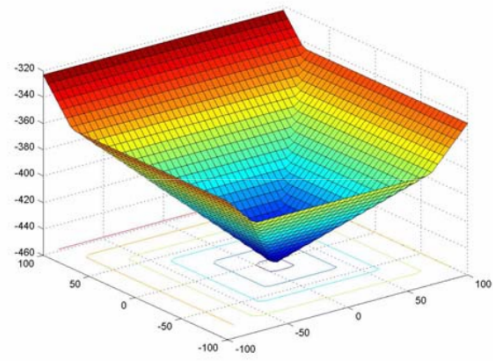
$$\text{doubledip}(x, c, s) = \begin{cases} (-6144(x-c)^6 + 3088(x-c)^4 - 392(x-c)^2 + 1)s, & -0.5 < x < 0.5, \\ 0, & \text{caso contrário} \end{cases}$$

$$\mathbf{x} \in [-1, 1]^N \text{ e } f(\mathbf{x}^*) = \text{desconhecido}.$$

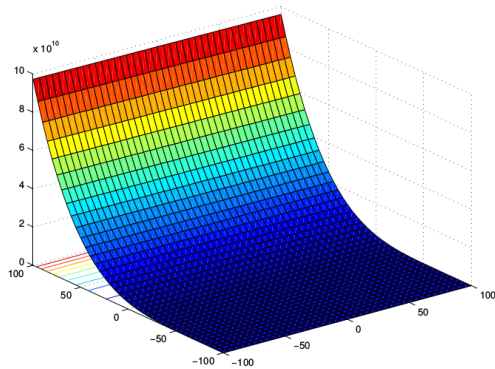
O método Fq-G foi comparado com sete dos oito AEs que participaram da competição promovida pelo CEC’2008: MLCC (YANG et al., 2008b), EPUS-PSO (HSIEH et al., 2008), jDEdynNP-F (BREST et al., 2008), MTS (TSENG; CHEN, 2008), DEw-SAcc (ZAMUDA et al., 2008), DMS-PSO (ZHAO et al., 2008) e LSEDA-gl (WANG; LI, 2008). O algoritmo evolutivo UEP (MACNISH; YAO, 2008) participou da competição do CEC’2008, mas seus autores não disponibilizaram os dados obtidos, por esta razão não foi utilizado nesta comparação. Os resultados para  $N = 100$ ,  $N = 500$  e  $N = 1000$  são apresentados nas Tabelas 5.1, 5.2 e 5.3, respectivamente, e representam os valores médios de  $f(\mathbf{x}_{\text{melhor}}) - f(\mathbf{x}^*)$  após 25 execuções independentes para cada um dos 8 algoritmos e das 7 funções teste. Os melhores resultados estão em negrito.



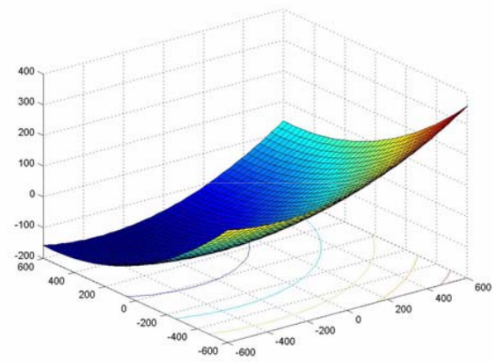
(a) F1 - Sphere Deslocada



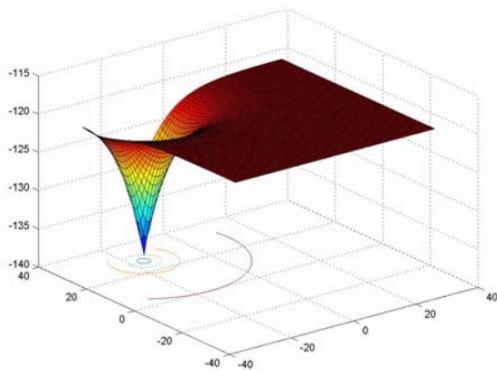
(b) F2 - Problema Schwefel 2.21



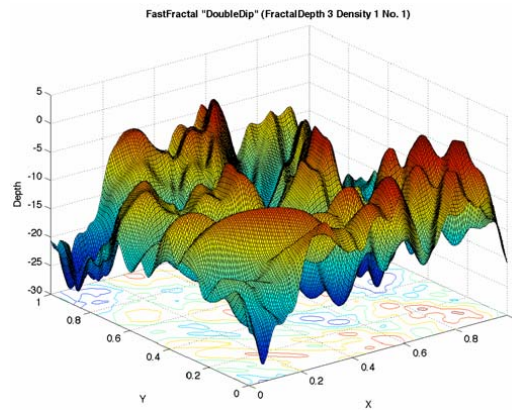
(c) F3 - Rosenbrock Deslocada



(d) F5 - Griewank Deslocada



(e) F6 - Ackley Deslocada



(f) F7 - *Fast Fractal*

Figura 5.1 - Ilustração das funções teste do CEC'2008 para o caso bidimensional.  
Fonte: Tang et al. (2007).

Observe que para  $N = 100$  o algoritmo  $Fq$ -G encontra o mínimo em quatro funções do conjunto teste, duas unimodais e duas multimodais. E para a F6 consegue atingir a bacia de atração do mínimo global. Em uma comparação direta com os AEs, o  $Fq$ -G apresenta resultados levemente inferiores apenas em relação ao MTS. É importante

Tabela 5.1 - CEC'2008:  $f(\mathbf{x}_{melhor}) - f(\mathbf{x}^*)$  para F1 a F6, e  $f(\mathbf{x}_{melhor})$  para F7, média de 25 execuções independentes, para todos os algoritmos; os melhores resultados estão em negrito;  $N = 100$ .

	MTS	LSEDA-gl	jDEdyn-NP-F	MLCC	DMS-PSO	DewSA-cc	EPUS-PSO	F $q$ -G
<b>F1</b>	<b>0.0+00</b>	5.68e-14	5.68e-14	6.82e-14	<b>0.00e+00</b>	5.68e-14	7.47e-01	<b>0.00e+00</b>
<b>F2</b>	1.44e-11	2.21e-13	4.28e-01	2.53e+01	3.65e+00	8.25e+00	1.86e+01	<b>0.00e+00</b>
<b>F3</b>	<b>5.17e-08</b>	2.81e+02	1.12e+02	1.50e+02	2.83e+02	1.45e+02	4.99e+03	5.71e+09
<b>F4</b>	<b>0.00e+00</b>	1.03e+02	5.46e-14	4.39e-13	1.83e+02	4.38e+00	4.71e+02	<b>0.00e+00</b>
<b>F5</b>	<b>0.00e+00</b>	2.84e-14	2.84e-14	3.41e-14	0.00e+00	3.07e-14	3.72e-01	<b>0.00e+00</b>
<b>F6</b>	<b>0.00e+00</b>	9.77e-14	5.68e-14	1.11e-13	<b>0.00e+00</b>	1.13e-13	2.06e+00	9.09e-15
<b>F7</b>	-1.48e+03	-1.46e+03	-1.48e+03	<b>-1.54e+03</b>	-1.14e+03	-1.37e+03	-8.55e+02	-1.38e+03

Tabela 5.2 - CEC'2008:  $f(\mathbf{x}_{melhor}) - f(\mathbf{x}^*)$  para F1 a F6, e  $f(\mathbf{x}_{melhor})$  para F7, média de 25 execuções independentes, para todos os algoritmos; os melhores resultados estão em negrito;  $N = 500$ .

	MTS	LSEDA-gl	jDEdyn-NP-F	MLCC	DMS-PSO	DewSA-cc	EPUS-PSO	F $q$ -G
<b>F1</b>	<b>0.00e+00</b>	2.27e-13	9.32e-14	4.30e-13	<b>0.00e+00</b>	2.10e-09	8.45e+01	<b>0.00e+00</b>
<b>F2</b>	7.32e-06	2.71e-10	8.46e+00	6.67e+01	6.89e+01	7.57e+01	4.35e+01	<b>0.00e+00</b>
<b>F3</b>	<b>5.04e-03</b>	8.67e+02	6.61e+02	9.25e+02	4.67e+07	1.81e+03	5.77e+04	1.48e+10
<b>F4</b>	<b>0.00e+00</b>	8.56e+02	1.47e-12	1.79e-11	1.61e+03	3.64e+02	3.49e+03	2.08e+00
<b>F5</b>	<b>0.00e+00</b>	1.14e-13	4.21e-14	2.13e-13	<b>0.00e+00</b>	6.90e-04	1.64e+00	<b>0.00e+00</b>
<b>F6</b>	2.80e-13	3.13e-13	1.49e-13	5.34e-13	2.00e+00	4.80e-01	6.64e+00	<b>5.68e-14</b>
<b>F7</b>	-7.08e+03	-6.83e+03	-6.88e+03	<b>-7.43e+03</b>	-4.20e+03	-5.75e+03	-3.51e+03	-6.51e+03

notar também que para a função multimodal Rosenbrock (F3), o método baseado em  $q$ -gradiente apresentou o pior resultado entre os algoritmos comparados.

Quando  $N = 500$ , o algoritmo F $q$ -G encontrando os melhores resultados em quatro das sete funções teste. E continua sendo eficiente na resolução das funções F1, F2, F5 e F6; e ineficiente para resolver a F3.

Finalmente, para  $N = 1000$ , o F $q$ -G atinge a bacia de atração do mínimo global em todas as funções, exceto para Rosenbrock Deslocada. Para F2 e F6, o método proposto supera o desempenho de todos os outros algoritmos, enquanto para F1 e F4, sua performance é igual aos melhores encontrados, MTS e DMS-PSO, atingindo o menor valor médio da função objetivo. Para a função F7, não é possível analisar se os métodos atingiram o mínimo da função, já que o mesmo é desconhecido. No entanto, pode-se observar nas Tabelas 5.1, 5.2 e 5.3, que o F $q$ -G conseguiu um desempenho razoável para cada uma das dimensões.

Tabela 5.3 - CEC'2008:  $f(\mathbf{x}_{melhor}) - f(\mathbf{x}^*)$  para F1 a F6, e  $f(\mathbf{x}_{melhor})$  para F7, média de 25 execuções independentes, para todos os algoritmos; os melhores resultados estão em negrito;  $N = 1000$ .

	MTS	LSEDA-gl	jDEdyn-NP-F	MLCC	DMS-PSO	DewSA-cc	EPUS-PSO	Fq-G
<b>F1</b>	<b>0.00e+00</b>	3.22e-13	1.14e-13	8.46e-13	<b>0.00e+00</b>	8.79e-03	5.53e+02	<b>0.00e+00</b>
<b>F2</b>	4.72e-02	1.04e-05	1.95e+01	1.09e+02	9.15e+01	9.61e+01	4.66e+01	<b>5.46e-14</b>
<b>F3</b>	<b>3.41e-04</b>	1.73e+03	1.31e+03	1.80e+03	8.98e+09	9.15e+03	8.37e+05	1.84e+10
<b>F4</b>	<b>0.00e+00</b>	5.45e+02	2.17e-04	1.37e-10	3.84e+03	1.82e+03	7.58e+03	<b>0.00e+00</b>
<b>F5</b>	<b>0.00e+00</b>	1.71e-13	3.98e-14	4.18e-13	<b>0.00e+00</b>	3.58e-03	5.89e+00	2.39e-14
<b>F6</b>	1.24e-11	4.26e-13	1.47e-11	1.06e-12	7.76e+00	2.30e+00	1.89e+01	<b>1.42e-13</b>
<b>F7</b>	-1.40e+4	-1.35e+4	-1.35e+4	<b>-1.47e+4</b>	-7.51e+3	-1.06e+4	-6.62e+3	-1.20e+4

De maneira geral, os resultados apresentados pelo método Fq-G no conjunto de funções teste do CEC'2008 demonstram que este novo algoritmo é eficiente e competitivo em relação aos métodos que participaram da competição. Além disso, o algoritmo Fq-G é ineficaz para determinar o mínimo da função não separável Rosenbrock Deslocada.

Todos os algoritmos participantes do CEC'2008 armazenaram, para cada uma das funções o 1º (melhor), o 7º, o 13º (mediana), o 19º e o 25º (pior) valores do erro  $f(\mathbf{x}_{melhor}) - f(\mathbf{x}^*)$ , média e desvio padrão de 25 execuções, depois de  $50 \cdot N$ ,  $500 \cdot N$  e  $5000 \cdot N$  NAFO. Os resultados obtidos pelos AEs estão reproduzidos no Anexo A e os obtidos pelo método Fq-G para as funções F1 a F7 são apresentados nas Tabelas 5.4, 5.5 e 5.6 para as dimensões  $N = 100$ ,  $N = 500$  e  $N = 1000$ , respectivamente.

Esses resultados demonstram o excelente desempenho do método Fq-G na resolução da função Sphere Deslocada (F1), para as três dimensões testadas, pois com  $500 \cdot N$  NAFO o Fq-G é capaz de terminar o mínimo global da função. Este comportamento é semelhante para F4 quando  $N = 100$  e  $N = 1000$ , e para F5 quando  $N = 100$  e  $N = 500$ . Na resolução da função Ackley Deslocada (F6), o Fq-G precisa de mais de  $500 \cdot N$  NAFO para atingir a bacia de atração no mínimo global. Na resolução do Problema Schwefel 2.21, o algoritmo do método Fq-G consegue atingir a bacia de atração com  $500 \cdot N$  NAFO e determinar o mínimo com  $5000 \cdot N$ , quando  $N = 500$ . O fraco desempenho do método para solucionar a função Rosenbrock Deslocada também pode ser observado nas Tabelas 5.4, 5.5 e 5.6, onde fica evidente que não há diminuição significativa dos valores do  $f(\mathbf{x}_{melhor}) - f(\mathbf{x}^*)$  em nenhuma das dimensões testadas.

Tabela 5.4 - CEC'2008: Valores do erro  $f(\mathbf{x}_{melhor}) - f(\mathbf{x}^*)$  para F1 a F6, e  $f(\mathbf{x}_{melhor})$  para F7;  $N = 100$ .

NAFO		F1	F2	F3	F4	F5	F6	F7
5e3	1° (Melhor)	2.55e-01	3.88e+01	6.42e+09	7.72e-01	1.98e-01	1.70e+01	-8.33e+02
	7°	4.12e-01	4.15e+01	1.04e+10	1.18e+00	2.71e-01	1.74e+01	-8.07e+02
	13° (Mediana)	5.04e-01	4.36e+01	1.21e+10	1.30e+00	3.42e-01	1.77e+01	-8.01e+02
	19°	8.04e-01	4.48e+01	1.33e+10	1.55e+00	4.83e-01	1.80e+01	-7.86e+02
	25° (Pior)	1.22e+00	4.59e+01	1.50e+10	1.77e+00	7.55e-01	1.83e+01	-7.77e+02
	Média	5.91e-01	4.32e+01	1.17e+10	1.33e+00	3.80e-01	1.77e+01	-7.99e+02
	Desvio-padrão	2.53e-01	1.98e+00	2.24e+09	2.65e-02	1.55e-01	4.13e-01	1.56e+01
5e4	1° (Melhor)	0.00e+00	3.24e+01	4.13e+09	0.00e+00	0.00e+00	1.59e+01	-8.59e+03
	7°	0.00e+00	3.58e+01	5.91e+09	0.00e+00	0.00e+00	1.65e+01	-8.37e+03
	13° (Mediana)	0.00e+00	3.80e+01	6.63e+09	0.00e+00	0.00e+00	1.67e+01	-8.26e+03
	19°	0.00e+00	3.95e+01	7.78e+09	0.00e+00	0.00e+00	1.69e+01	-8.19e+03
	25° (Pior)	0.00e+00	4.08e+01	9.55e+09	0.00e+00	0.00e+00	1.72e+01	-8.04e+03
	Média	0.00e+00	3.76e+01	6.83e+09	0.00e+00	0.00e+00	1.67e+01	-8.29e+03
	Desvio Padrão	0.00e+00	2.39e+00	1.37e+09	0.00e+00	0.00e+00	3.21e-01	1.55e+01
5e5	1° (Melhor)	0.00e+00	0.00e+00	4.05e+09	0.00e+00	0.00e+00	0.00e+00	-1.44e+03
	7°	0.00e+00	0.00e+00	5.23e+09	0.00e+00	0.00e+00	0.00e+00	-1.40e+03
	13° (Mediana)	0.00e+00	0.00e+00	5.78e+09	0.00e+00	0.00e+00	0.00e+00	-1.39e+03
	19°	0.00e+00	0.00e+00	6.33e+09	0.00e+00	0.00e+00	2.84e-14	-1.36e+03
	25° (Pior)	0.00e+00	0.00e+00	7.06e+09	0.00e+00	0.00e+00	2.84e-14	-1.27e+03
	Média	0.00e+00	0.00e+00	5.71e+09	0.00e+00	0.00e+00	9.09e-15	-1.38e+03
	Desvio Padrão	0.00e+00	0.00e+00	8.67e+08	0.00e+00	0.00e+00	1.35e-14	4.17e+01

Tabela 5.5 - CEC'2008: Valores do erro  $f(\mathbf{x}_{melhor}) - f(\mathbf{x}^*)$  para F1 a F6, e  $f(\mathbf{x}_{melhor})$  para F7;  $N = 500$ .

NAFO		F1	F2	F3	F4	F5	F6	F7
2.5e4	1° (Melhor)	3.34e-01	4.85e+01	1.83e+10	1.91e+00	6.52e-02	1.66e+01	-3.54e+03
	7°	5.40e-01	5.02e+01	2.54e+10	2.10e+00	1.21e-01	1.72e+01	-3.42e+03
	13° (Mediana)	6.77e-01	5.15e+01	2.76e+10	2.32e+00	1.46e-01	1.73e+01	-3.40e+03
	19°	7.25e-01	5.25e+01	3.01e+10	2.36e+00	1.62e-01	1.74e+01	-3.38e+03
	25° (Pior)	1.08e+00	5.44e+01	3.49e+10	2.56e+00	2.43e-01	1.77e+01	-3.35e+03
	Média	6.54e-01	5.14e+01	2.75e+10	2.27e+00	1.46e-01	1.73e+01	-3.41e+03
	Desvio-padrão	1.72e-01	1.72e+00	4.15e+09	1.87e-01	4.04e-02	2.79e-01	4.25e+01
2.5e5	1° (Melhor)	0.00e+00	8.60e-03	1.15e+10	1.76e+00	0.00e+00	1.61e+01	-3.54e+03
	7°	0.00e+00	2.35e-02	1.32e+10	1.95e+00	0.00e+00	1.66e+01	-3.49e+03
	13° (Mediana)	0.00e+00	3.45e-02	1.45e+10	2.12e+00	0.00e+00	1.68e+01	-3.46e+03
	19°	0.00e+00	5.25e-02	1.63e+10	2.17e+00	0.00e+00	1.70e+01	-3.44e+03
	25° (Pior)	0.00e+00	1.08e-01	1.84e+10	2.42e+00	0.00e+00	1.73e+01	-3.42e+03
	Média	0.00e+00	3.91e-02	1.48e+10	2.08e+00	0.00e+00	1.68e+01	-3.46e+03
	Desvio Padrão	0.00e+00	2.22e-02	2.02e+09	1.75e-01	0.00e+00	2.83e-01	3.10e+01
2.5e6	1° (Melhor)	0.00e+00	0.00e+00	1.15e+10	1.76e+00	0.00e+00	5.68e-14	-6.66e+03
	7°	0.00e+00	0.00e+00	1.32e+10	1.95e+00	0.00e+00	5.68e-14	-6.61e+03
	13° (Mediana)	0.00e+00	0.00e+00	1.45e+10	2.12e+00	0.00e+00	5.68e-14	-6.52e+03
	19°	0.00e+00	0.00e+00	1.63e+10	2.17e+00	0.00e+00	5.68e-14	-6.42e+03
	25° (Pior)	0.00e+00	0.00e+00	1.84e+10	2.42e+00	0.00e+00	5.68e-14	-6.33e+03
	Média	0.00e+00	0.00e+00	1.48e+10	2.08e+00	0.00e+00	5.68e-14	-6.51e+03
	Desvio Padrão	0.00e+00	0.00e+00	2.02e+09	1.76e-01	0.00e+00	0.00e+00	1.03e+02



Tabela 5.6 - CEC'2008: Valores do erro  $f(\mathbf{x}_{melhor}) - f(\mathbf{x}^*)$  para F1 a F6, e  $f(\mathbf{x}_{melhor})$  para F7;  $N = 1000$ .

NAFO		F1	F2	F3	F4	F5	F6	F7
5e4	1° (Melhor)	5.16e-01	5.07e+01	3.31e+10	1.26e+00	4.95e-02	2.00e+01	-6.59e+03
	7°	5.94e-01	5.25e+01	3.95e+10	1.30e+00	7.24e-02	2.00e+01	-6.52e+03
	13° (Mediana)	6.41e-01	5.36e+01	4.19e+10	1.35e+00	7.59e-02	2.00e+01	-6.48e+03
	19°	6.99e-01	5.44e+01	4.51e+10	1.40e+00	9.32e-02	2.00e+01	-6.45e+03
	25° (Pior)	8.37e-01	5.56e+01	5.17e+10	1.46e+00	1.26e-01	2.00e+01	-6.42e+03
	Média	6.46e-01	5.34e+01	4.22e+10	1.35e+00	8.12e-02	2.00e+01	-6.49e+03
	Desvio-padrão	8.69e-02	1.45e+00	4.32e+09	6.04e-02	1.83e-02	6.86e-05	4.89e+01
5e5	1° (Melhor)	0.00e+00	4.36e+01	1.68e+10	0.00e+00	2.84e-14	1.61e+01	-6.62e+03
	7°	0.00e+00	4.79e+01	2.39e+10	0.00e+00	2.84e-14	1.63e+01	-6.58e+03
	13° (Mediana)	0.00e+00	4.85e+01	2.58e+10	0.00e+00	2.84e-14	1.65e+01	-6.57e+03
	19°	0.00e+00	5.03e+01	2.66e+10	0.00e+00	2.84e-14	1.66e+01	-6.55e+03
	25° (Pior)	0.00e+00	5.17e+01	3.19e+10	0.00e+00	2.84e-14	1.69e+01	-6.52e+03
	Média	0.00e+00	4.87e+01	2.50e+10	0.00e+00	2.84e-14	1.65e+01	-6.57e+03
	Desvio Padrão	0.00e+00	1.93e+00	3.27e+09	0.00e+00	0.00e+00	1.98e-01	2.42e+01
5e6	1° (Melhor)	0.00e+00	0.00e+00	1.58e+10	0.00e+00	0.00e+00	1.42e-13	-1.23e+04
	7°	0.00e+00	5.68e-14	1.75e+10	0.00e+00	2.84e-14	1.42e-13	-1.21e+04
	13° (Mediana)	0.00e+00	5.68e-14	1.85e+10	0.00e+00	2.84e-14	1.42e-13	-1.21e+04
	19°	0.00e+00	5.68e-14	1.91e+10	0.00e+00	2.84e-14	1.42e-13	-1.20e+04
	25° (Pior)	0.00e+00	5.68e-14	2.29e+10	0.00e+00	2.84e-14	1.42e-13	-1.17e+04
	Média	0.00e+00	5.46e-14	1.84e+10	0.00e+00	2.39e-14	1.42e-13	-1.20e+04
	Desvio Padrão	0.00e+00	1.14e-14	1.55e+09	0.00e+00	1.06e-14	0.00e+00	1.30e+02

As Figuras 5.2 e 5.3 exibem as curvas de convergência dos AEs participantes da competição do CEC'2008 e do método Fq-G para as 7 funções teste quando  $N = 1000$ . Para F1 a F6 tem-se gráficos do tipo semi-log, que representam o logaritmo dos valores medianos ( $f(\mathbf{x}) - f(\mathbf{x}^*)$ ) em 25 execuções versus o NAFO. No caso da F7, como  $f(\mathbf{x}^*)$  é desconhecido e pode assumir valores negativos, os gráficos representam os valores medianos de  $f(\mathbf{x}^*)$  em 25 execuções versus o NAFO. As curvas dos AEs foram extraídas dos artigos que descrevem cada um dos métodos, não foi possível retirar a curva de convergência da F7 para o método DMS-PSO, por esse motivo a Figura 5.3 possui apenas 7 curvas.

As curvas de convergência demonstram que o método Fq-G tem desempenho superior ao dos outros AEs para as funções F1, F2 e F4. Além disto, o método baseado em  $q$ -gradiente converge rapidamente quando aplicado na função Griewank Deslocada (F5), apresentando desempenho similar aos algoritmos evolutivos MTS e LSEDA-gl. Para as funções F6 e F7, sua convergência é mais lenta, utilizando o número máximo de avaliações da função objetivo disponível para atingir a bacia de atração do mínimo global. Para a Rosenbrock Deslocada, o algoritmo do Fq-G apresenta o pior desempenho dentre os oito algoritmos.

### 5.2.2 Segundo conjunto de funções teste

O conjunto de funções teste proposto no CEC'2010 (TANG et al., 2009), uma versão melhorada do conjunto do CEC'2008, foi desenvolvido para investigar o comportamento de algoritmos em problemas de grande porte em diferentes cenários. As 20 funções que compõe o CEC'2010, cujas expressões matemáticas são descritas a seguir, foram divididas em quatro grupos: funções separáveis; funções parcialmente separáveis, em que um número pequeno de variáveis são dependentes, enquanto todas as outras são independentes; funções parcialmente separáveis que consistem em vários subgrupos independentes, cada um dos quais são  $p$  não separáveis; e, as funções totalmente separáveis. Uma função não separável  $f(\mathbf{x})$  é chamada  $p$  não separável se no máximo  $p$  dos seus componentes  $x_i$  não são independentes (TANG et al., 2009).

- **Funções separáveis**

- **G1 - Função Elíptica Deslocada**

$$f(\mathbf{x}) = f_{eliptica}(\mathbf{z}) = \sum_{i=1}^N (10^6)^{\frac{i-1}{N-1}} z_i^2, \quad (5.2)$$

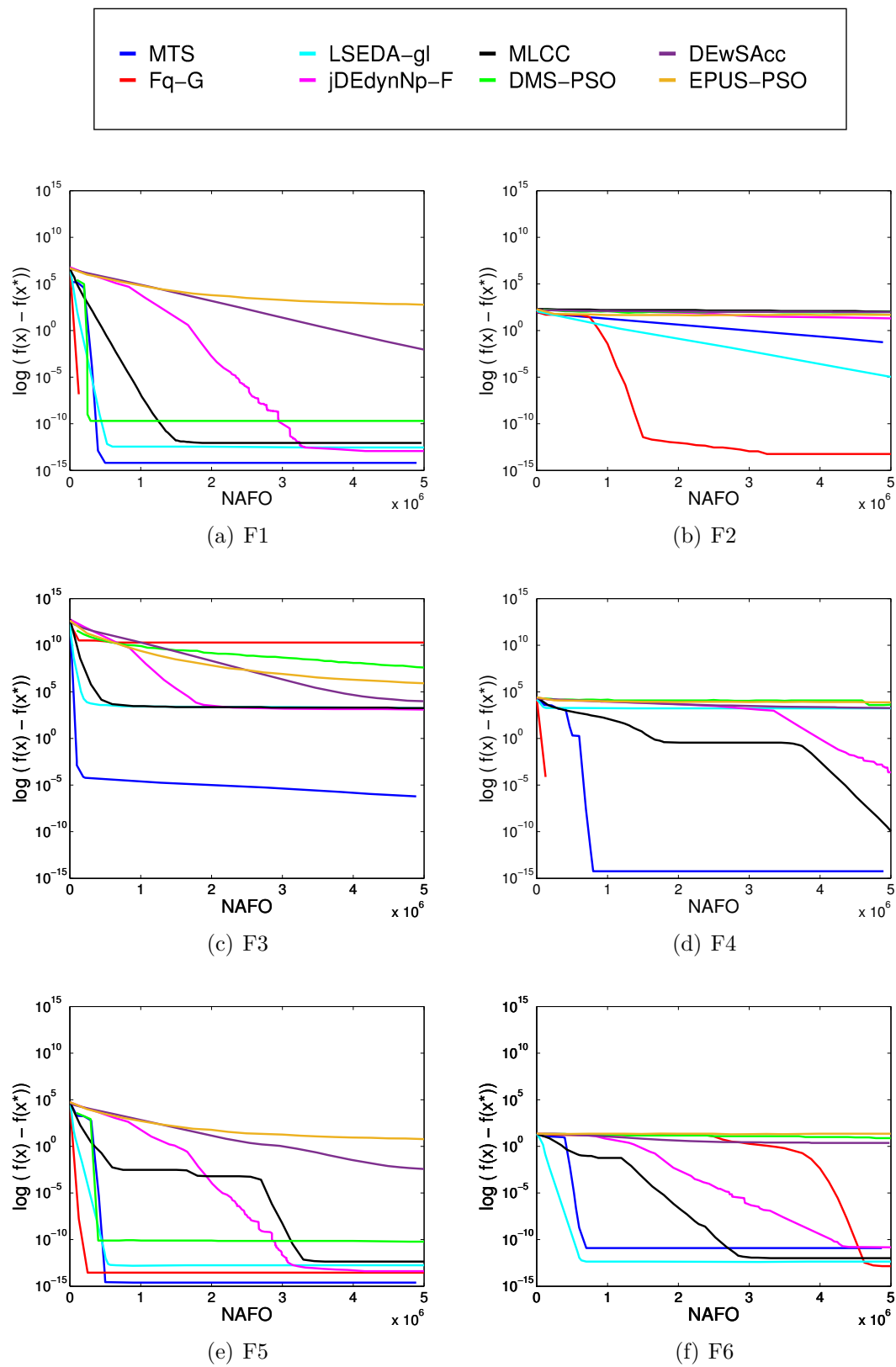
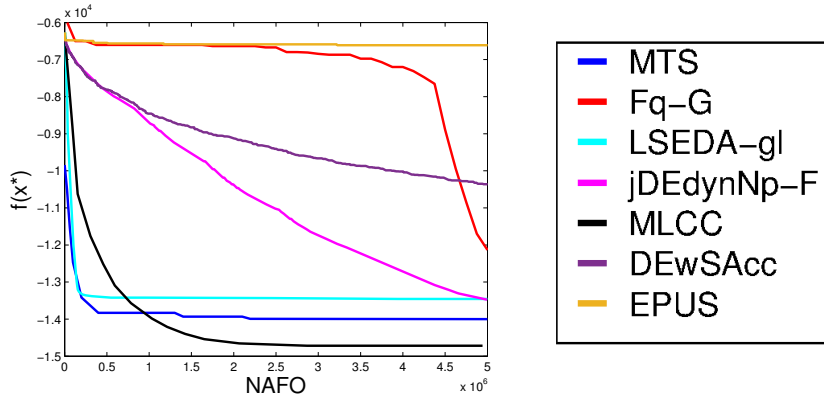


Figura 5.2 - Gráfico de convergência para as funções teste F1 a F6 do CEC'2008 e diferentes métodos.



(a) F7

Figura 5.3 - Gráfico de convergência para a função teste Rosenbrock Deslocada (F7) do CEC'2008 e diferentes métodos.

$$\mathbf{z} = \mathbf{x} - \mathbf{o}, \mathbf{x} \in [-100, 100]^N.$$

– **G2 - Função Rastrigin Deslocada**

$$f(\mathbf{x}) = f_{rastrigin}(\mathbf{z}) = \sum_{i=1}^N [z_i^2 - 10 \cos(2\pi z_i) + 10], \quad (5.3)$$

$$\mathbf{z} = \mathbf{x} - \mathbf{o}, \mathbf{x} \in [-5, 5]^N.$$

– **G3 - Função Ackley Deslocada**

$$f(\mathbf{x}) = f_{ackley}(\mathbf{z}) = -20 \exp \left( -0.2 \sqrt{\frac{1}{N} \sum_{i=1}^N z_i^2} \right) - \exp \left( \frac{1}{N} \sum_{i=1}^N \cos(2\pi z_i) \right) + 20 + e, \quad (5.4)$$

$$\mathbf{z} = \mathbf{x} - \mathbf{o}, \mathbf{x} \in [-32, 32]^N.$$

• **Funções  $p$  não separáveis em grupo único**

– **G4 - Elíptica Deslocada e  $p$ -rotacionada em grupo único**

$$f(\mathbf{x}) = f_{rot-eliptica}[\mathbf{z}(P_1 : P_p)] \cdot 10^6 + f_{eliptica}[\mathbf{z}(P_{p+1} : P_N)] \quad (5.5)$$

$\mathbf{z} = \mathbf{x} - \mathbf{o}$ ,  $P$  é uma permutação aleatória de  $\{1, 2, \dots, N\}$ ,  $f_{rot-eliptica}(\mathbf{x}) = f_{eliptica}(\mathbf{z}) = \mathbf{x} \cdot \mathbf{M}$ ,  $\mathbf{M}$  é uma matriz ortogonal  $N \times N$  e  $\mathbf{x} \in [-100, 100]^N$ .

- **G5 - Rastrigin Deslocada e  $p$ -rotacionada em grupo único**

$$f(\mathbf{x}) = f_{rot-rastrigin}[\mathbf{z}(P_1 : P_p)] \cdot 10^6 + f_{rastrigin}[\mathbf{z}(P_{p+1} : P_N)] \quad (5.6)$$

$\mathbf{z} = \mathbf{x} - \mathbf{o}$ ,  $P$  é uma permutação aleatória de  $\{1, 2, \dots, N\}$ ,  
 $f_{rot-rastrigin}(\mathbf{x}) = f_{rastrigin}(\mathbf{z}) = \mathbf{x} \cdot \mathbf{M}$ ,  $\mathbf{M}$  é uma matriz ortogonal  
 $N \times N$  e  $\mathbf{x} \in [-5, 5]^N$ .

- **G6 - Ackley Deslocada e  $p$ -rotacionada em grupo único**

$$f(\mathbf{x}) = f_{rot-ackley}[\mathbf{z}(P_1 : P_p)] \cdot 10^6 + f_{ackley}[\mathbf{z}(P_{p+1} : P_N)] \quad (5.7)$$

$\mathbf{z} = \mathbf{x} - \mathbf{o}$ ,  $P$  é uma permutação aleatória de  $\{1, 2, \dots, N\}$ ,  
 $f_{rot-ackley}(\mathbf{x}) = f_{ackley}(\mathbf{z}) = \mathbf{x} \cdot \mathbf{M}$ ,  $\mathbf{M}$  é uma matriz ortogonal  $N \times N$   
e  $\mathbf{x} \in [-32, 32]^N$ .

- **G7 - Prob. 1.2 Schwefel Deslocada e  $p$ -dimensional em grupo único**

$$f(\mathbf{x}) = f_{schwefel}[\mathbf{z}(P_1 : P_p)] \cdot 10^6 + f_{sphere}[\mathbf{z}(P_{p+1} : P_N)] \quad (5.8)$$

$\mathbf{z} = \mathbf{x} - \mathbf{o}$ ,  $P$  é uma permutação aleatória de  $\{1, 2, \dots, N\}$ ,  
 $f_{schwefel}(\mathbf{x}) = \sum_{i=1}^N \left( \sum_j^i x_j \right)^2$ ,  $f_{sphere}(\mathbf{x}) = \sum_{i=1}^N x_i^2$  e  $\mathbf{x} \in [-100, 100]^N$ .

- **G8 - Rosenbrock Deslocada e  $p$ -dimensional em grupo único**

$$f(\mathbf{x}) = f_{rosenbrock}[\mathbf{z}(P_1 : P_p)] \cdot 10^6 + f_{sphere}[\mathbf{z}(P_{p+1} : P_N)] \quad (5.9)$$

$\mathbf{z} = \mathbf{x} - \mathbf{o}$ ,  $P$  é uma permutação aleatória de  $\{1, 2, \dots, N\}$ ,  
 $f_{rosenbrock}(\mathbf{x}) = \sum_{i=1}^{N-1} [100(x_i^2 - x_{i+1}^2) + (x_i + 1)^2]$  e  $\mathbf{x} \in [-100, 100]^N$ .

- **Funções  $p$  não separáveis em  $N/2p$  grupos**

- **G9 - Elíptica Deslocada e  $p$ -rotacionada em  $N/2p$  grupos**

$$f(\mathbf{x}) = \sum_{k=1}^{\frac{N}{2p}} f_{rot-elliptica}[\mathbf{z}(P_{(k-1)p+1} : P_{kp})] + f_{elliptica}[\mathbf{z}(P_{\frac{N}{2}+1} : P_N)] \quad (5.10)$$

$\mathbf{z} = \mathbf{x} - \mathbf{o}$ ,  $P$  é uma permutação aleatória de  $\{1, 2, \dots, N\}$  e  
 $\mathbf{x} \in [-100, 100]^N$ .

- **G10 - Rastrigin Deslocada e p-rotacionada em N/2p grupos**

$$g(\mathbf{x}) = \sum_{k=1}^{\frac{N}{2p}} f_{rot-rastrigin}[\mathbf{z}(P_{(k-1)\cdot p+1} : P_{k\cdot p})] + f_{rastrigin}[\mathbf{z}(P_{\frac{N}{2}+1} : P_N)] \quad (5.11)$$

$\mathbf{z} = \mathbf{x} - \mathbf{o}$ , P é uma permutação aleatória de  $\{1, 2, \dots, N\}$  e  $\mathbf{x} \in [-5, 5]^N$ .

- **G11 - Ackley Deslocada e p-rotacionada em N/2p grupos**

$$f(\mathbf{x}) = \sum_{k=1}^{\frac{N}{2p}} f_{rot-ackley}[\mathbf{z}(P_{(k-1)\cdot p+1} : P_{k\cdot p})] + f_{ackley}[\mathbf{z}(P_{\frac{N}{2}+1} : P_N)] \quad (5.12)$$

$\mathbf{z} = \mathbf{x} - \mathbf{o}$ , P é uma permutação aleatória de  $\{1, 2, \dots, N\}$  e  $\mathbf{x} \in [-32, 32]^N$ .

- **G12 - Prob. 1.2 Schwefel Deslocada e p-dimensional em N/2p grupos**

$$f(\mathbf{x}) = \sum_{k=1}^{\frac{N}{2p}} f_{schwefel}[\mathbf{z}(P_{(k-1)\cdot p+1} : P_{k\cdot p})] + f_{sphere}[\mathbf{z}(P_{\frac{N}{2}+1} : P_N)] \quad (5.13)$$

$\mathbf{z} = \mathbf{x} - \mathbf{o}$ , P é uma permutação aleatória de  $\{1, 2, \dots, N\}$  e  $\mathbf{x} \in [-100, 100]^N$ .

- **G13 - Rosenbrock Deslocada e p-dimensional em N/2p grupos**

$$f(\mathbf{x}) = \sum_{k=1}^{\frac{N}{2p}} f_{rosenbrock}[\mathbf{z}(P_{(k-1)\cdot p+1} : P_{k\cdot p})] + f_{sphere}[\mathbf{z}(P_{\frac{N}{2}+1} : P_N)] \quad (5.14)$$

$\mathbf{z} = \mathbf{x} - \mathbf{o}$ , P é uma permutação aleatória de  $\{1, 2, \dots, N\}$  e  $\mathbf{x} \in [-100, 100]^N$ .

- **Funções p não separáveis em N/p grupos**

- **G14 - Elíptica Deslocada e p-rotacionada em N/p grupos**

$$f(\mathbf{x}) = \sum_{k=1}^{\frac{N}{p}} f_{rot-elliptica}[\mathbf{z}(P_{(k-1)\cdot p+1} : P_{k\cdot p})] \quad (5.15)$$

$\mathbf{z} = \mathbf{x} - \mathbf{o}$ , P é uma permutação aleatória de  $\{1, 2, \dots, N\}$  e  $\mathbf{x} \in [-100, 100]^N$ .

- **G15 - Rastrigin Deslocada e p-rotacionada em N/p grupos**

$$f(\mathbf{x}) = \sum_{k=1}^{\frac{N}{p}} f_{rot-rastrigin}[\mathbf{z}(P_{(k-1)\cdot p+1} : P_{k\cdot p})] \quad (5.16)$$

$\mathbf{z} = \mathbf{x} - \mathbf{o}$ , P é uma permutação aleatória de  $\{1, 2, \dots, N\}$  e  $\mathbf{x} \in [-5, 5]^N$ .

- **G16 - Ackley Deslocada e p-rotacionada em N/p grupos**

$$f(\mathbf{x}) = \sum_{k=1}^{\frac{N}{p}} f_{rot-ackley}[\mathbf{z}(P_{(k-1)\cdot p+1} : P_{k\cdot p})] \quad (5.17)$$

$\mathbf{z} = \mathbf{x} - \mathbf{o}$ , P é uma permutação aleatória de  $\{1, 2, \dots, N\}$  e  $\mathbf{x} \in [-32, 32]^N$ .

- **G17 - Prob. 1.2 Schwefel Deslocada e p-dimensional em N/p grupos**

$$f(\mathbf{x}) = \sum_{k=1}^{\frac{N}{p}} f_{schwefel}[\mathbf{z}(P_{(k-1)\cdot p+1} : P_{k\cdot p})] \quad (5.18)$$

$\mathbf{z} = \mathbf{x} - \mathbf{o}$ , P é uma permutação aleatória de  $\{1, 2, \dots, N\}$  e  $\mathbf{x} \in [-100, 100]^N$ .

- **G18 - Rosenbrock Deslocada e p-dimensional em N/2p grupos**

$$f(\mathbf{x}) = \sum_{k=1}^{\frac{N}{p}} f_{rosenbrock}[\mathbf{z}(P_{(k-1)\cdot p+1} : P_{k\cdot p})] \quad (5.19)$$

$\mathbf{z} = \mathbf{x} - \mathbf{o}$ , P é uma permutação aleatória de  $\{1, 2, \dots, N\}$  e  $\mathbf{x} \in [-100, 100]^N$ .

- **Funções não separáveis**

- **G19 - Prob. 1.2 Schwefel Deslocado**

$$f(\mathbf{x}) = f_{schwefel}(\mathbf{z}) = \sum_{i=1}^N \left( \sum_{j=1}^i x_j \right)^2 \quad (5.20)$$

$\mathbf{z} = \mathbf{x} - \mathbf{o}$ , P é uma permutação aleatória de  $\{1, 2, \dots, N\}$  e  $\mathbf{x} \in [-100, 100]^N$ .

– G20 - Rosenbrock Deslocada

$$f(\mathbf{x}) = f_{rosenbrock} = \sum_{i=1}^{N-1} [100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2] \quad (5.21)$$

$\mathbf{z} = \mathbf{x} - \mathbf{o}$ ,  $P$  é uma permutação aleatória de  $\{1, 2, \dots, N\}$  e  $\mathbf{x} \in [-100, 100]^N$ .

O mínimo global das funções é  $\mathbf{x}^* = \mathbf{o}$ , exceto para G8 que tem  $\mathbf{x}^*(P_1 : P_p) = \mathbf{o}(P_1 : P_p) + 1$ ,  $\mathbf{x}^*(P_{p+1} : P_N) = \mathbf{o}(P_{p+1} : P_N)$ , G13 com o mínimo global em  $\mathbf{x}^*(P_1 : P_{N/2}) = \mathbf{o}(P_1 : P_{N/2}) + 1$ ,  $\mathbf{x}^*(P_{N/2+1} : P_N) = \mathbf{o}(P_{N/2+1} : P_N)$ , e G18 e G20 com  $\mathbf{x}^* = \mathbf{o} + 1$ , com  $f(\mathbf{x}^*) = 0$  para todas as funções teste. Para a competição do CEC'2010, todas as funções foram resolvidas para 1000 dimensões e o parâmetro  $p$ , que define o grau de separabilidade, foi sugerido em (TANG et al., 2009) como 50. Os resultados obtidos pelo Fq-G neste conjunto de funções teste foram comparados com todos os AEs que participaram competição organizada pelo CEC'2010 e publicaram seus resultados: jDElsgo (BREST et al., 2010), DECC-DML (OMIDVAR et al., 2010), DMS-PSO-SHS (ZHAO et al., 2010), MA-SW-Chain (MOLINA et al., 2010), DASA (KOROŠEC et al., 2010), SDENS (WANG et al., 2010) e EOEa (WANG; LI, 2010).

Assim como os resultados do CEC'2008, a média de 25 execuções independentes, para todos os algoritmos e todas funções teste do CEC'2010 são apresentados na Tabela 5.7. Os melhores resultados estão em negrito.

Os organizadores da competição de otimizadores de problemas de grande porte do CEC'2010 solicitaram para que os participantes da competição exibissem as curvas de convergência de 8 das 20 funções teste: G2, G5, G8, G10, G13, G15, G18 e G20. As curvas apresentadas nas Figuras 5.4 e 5.5 foram retiradas dos artigos que descrevem os AEs participantes do CEC'2010, juntamente com as obtidas pelo Fq-G. Essas curvas representam os valores médios de  $f(\mathbf{x})$  em 25 execuções de cada algoritmo. Devido a falta de nitidez nas imagens disponibilizadas pelos autores do método DMS-PSO-SHO, não foi possível extrair as curvas de convergência de (ZHAO et al., 2010). Além disso, como os gráficos com as curvas de convergência das 8 funções obtidas pelo método DECC-DML e as curvas das funções G18 e G20 para o método MA-SW-Chains não consideraram o  $\text{NAFO}_{\max}$  igual a  $3.0e+06$ , decidiu-se rejeitar estes dados. Desta forma, as Figuras 5.4 e 5.5 apresentam 6 curvas de convergência, exceto pelos gráficos que ilustram as curvas para as funções G18 e G20, que possuem apenas 5.



Tabela 5.7 - CEC'2010:  $f(\mathbf{x}_{melhor})$ , média de 25 execuções independentes, para todos os algoritmos e todas as funções teste;  $f(\mathbf{x}^*) = 0$  é o mínimo global para G1 a G20; os melhores resultados estão em negrito.

	jDElsgo	DECC-DML	DMS-PSO-SHS	MA-SW-Chains	DASA	SDENS	EOEA	Fq-G
<b>G1</b>	8.86e-20	<b>1.93e-25</b>	5.51e-15	2.10e-14	1.52e-21	5.73e-06	2.20e-23	1.88e+01
<b>G2</b>	1.25e-01	2.17e+02	8.51e+01	8.10e+02	8.48e+00	2.21e+03	3.62e-01	<b>3.59e-13</b>
<b>G3</b>	3.81e-12	<b>1.18e-13</b>	5.52e-11	7.28e-13	7.20e-11	2.70e-05	1.67e-13	1.96e+00
<b>G4</b>	<b>8.06e+10</b>	3.58e+12	2.46e+11	3.53e+11	5.05e+11	5.11e+12	3.09e+12	7.93e+11
<b>G5</b>	9.72e+07	2.99e+08	8.36e+07	1.68e+08	6.20e+08	1.18e+08	2.24e+07	<b>9.92e+03</b>
<b>G6</b>	<b>1.70e-08</b>	7.93e+05	8.28e-02	8.14e+04	1.97e+07	2.02e-04	3.85e+06	3.17e+05
<b>G7</b>	<b>1.31e-02</b>	1.39e+08	1.95e+03	1.03e+02	7.78e+00	1.20e+08	1.24e+02	1.11e+04
<b>G8</b>	<b>3.15e+06</b>	3.46e+07	1.29e+07	1.41e+07	4.98e+07	5.12e+07	1.01e+07	5.09e+11
<b>G9</b>	3.11e+07	5.92e+07	8.72e+06	1.41e+07	3.60e+07	5.63e+08	4.63e+07	<b>4.09e+04</b>
<b>G10</b>	2.64e+03	1.25e+04	5.53e+03	2.07e+03	7.29e+03	6.87e+03	1.08e+03	<b>3.60e-13</b>
<b>G11</b>	2.20e+01	<b>1.80e-13</b>	3.25e+01	3.80e+01	1.98e+02	2.21e+02	3.86e+01	1.78e+02
<b>G12</b>	1.21e+04	3.80e+06	6.13e+02	3.62e-06	1.78e+03	4.13e+05	1.37e+04	<b>1.02e-06</b>
<b>G13</b>	<b>7.11e+02</b>	1.14e+03	1.12e+03	1.25e+03	1.21e+03	2.19e+03	1.24e+03	1.19e+10
<b>G14</b>	1.69e+08	1.89e+08	1.76e+07	3.11e+07	1.0e+08	1.88e+09	1.65e+08	<b>1.07e+05</b>
<b>G15</b>	5.84e+03	1.54e+04	4.08e+03	2.74e+03	1.45e+04	7.32e+03	2.14e+03	<b>3.67e-13</b>
<b>G16</b>	1.44e+02	<b>5.08e-02</b>	6.98e+01	9.98e+01	3.97e+02	4.08e+02	8.26e+01	3.27e+02
<b>G17</b>	1.02e+05	6.54e+06	3.83e+03	1.24e+00	1.03e+04	1.08e+06	7.93e+04	<b>7.91e-03</b>
<b>G18</b>	1.85e+03	2.47e+03	2.26e+03	<b>1.30e+03</b>	4.92e+03	3.08e+04	2.94e+03	1.78e+10
<b>G19</b>	2.74e+05	1.59e+07	1.17e+06	2.85e+05	8.34e+05	8.80e+05	1.84e+06	<b>7.78e+02</b>
<b>G20</b>	1.53e+03	9.91e+02	<b>3.52e+02</b>	1.07e+03	1.13e+03	9.90e+02	1.97e+03	1.81e+10

Considerando os valores médios de  $f(\mathbf{x})$ , o Fq-G supera o desempenho dos outros sete algoritmos em 9 das 20 funções teste. Contudo, sua performance deteriora-se significativamente para todas as funções Rosenbrock (G8, G13, G18, G20), como pode ser observado na Figura 5.5. Claramente o algoritmo baseado em Fq-gradiente consegue alcançar a bacia de atração do mínimo global para as funções Rastrigin (G2, G10, G15) e os Problemas Schwefel (G12 e G17). Na Figura 5.5 é possível observar que o método Fq-G converge rapidamente quando aplicados as funções Rastrigin. E apesar de não ter encontrado o mínimo nas funções G5, G9, G14 e G19, o resultado determinado pelo algoritmo Fq-G é o melhor entre os AEs participantes do CEC'2010.

De maneira semelhante ao CEC'2008, os algoritmos participantes das competições do CEC'2010 armazenaram, para cada uma das funções o 1º (melhor), o 13º (mediana) e o 25º (pior) valores da função, média e desvio padrão de 25 execuções, depois de  $1.2e+05$ ,  $6.0e+05$  e  $3.0e+06$  NAFO. Os resultados dos AEs estão reproduzidos no Anexo A e os do método Fq-G são apresentados nas Tabelas 5.8 e 5.9. Para as funções G2, G10 e G15, o método Fq-G atinge a bacia de atração do mínimo global com  $6.0e+05$  NAFO, e para G12 e G17 utiliza as  $3.0e+06$  avaliações da função

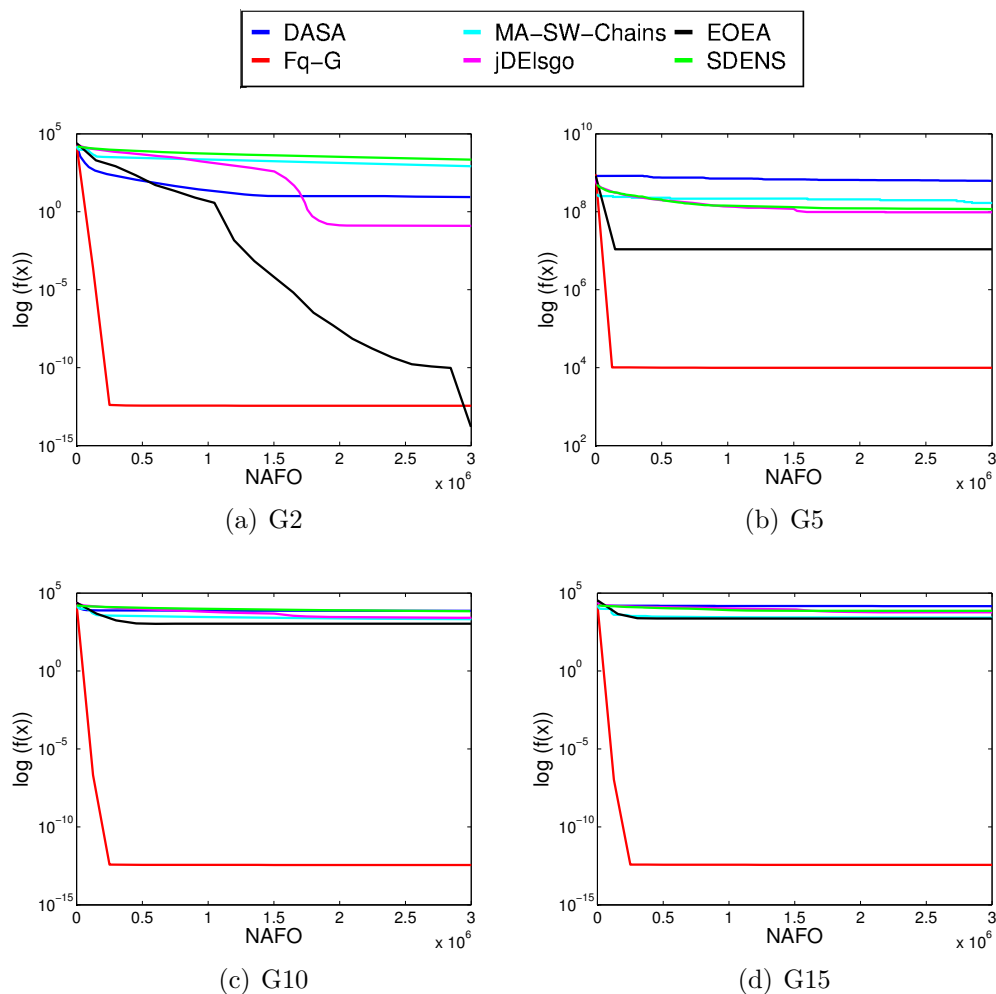


Figura 5.4 - Gráfico de convergência para as funções teste G2, G5, G10 e G15 do CEC'2010 e diferentes métodos.

objetivo disponíveis. Assim como no CEC'2008, fica evidente que não há diminuição significativa dos valores da função para as variações da função Rosenbrock (G8, G13, G18 e G20).

### 5.3 Classificação

Como os algoritmos participantes das competições do CEC'2008 e CEC'2010 armazenaram, para cada uma das funções o melhor, a mediana e o pior valor do erro  $f(\mathbf{x}_{melhor}) - f(\mathbf{x}^*)$ , média e desvio padrão de 25 execuções em três estágios de avaliação, uma pontuação foi dada para cada método seguindo o sistema de pontuação aplicado na competição do CEC'2010, que é igual ao utilizado pelos organizadores do Campeonato de Fórmula 1<sup>1</sup> no ano de 2010. Classificações dos métodos baseados

<sup>1</sup>No Grande Prêmio de Fórmula 1, os pontos são concedidos para os dez primeiros colocados de acordo com o seguinte sistema de pontuação: 1º - 25 pontos; 2º - 18 pontos; 3º - 15 pontos; 4º -

Tabela 5.8 - CEC'2010:  $f(\mathbf{x}_{melhor})$  para as funções G1 a G10 com  $N = 1000$ .

NAFO		G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
1.2e5	Melhor	1.13e+06	1.42e-04	2.00e+01	7.46e+11	9.81e+03	1.53e+07	2.09e+05	2.32e+15	1.95e+06	3.45e-07
	Mediana	1.43e+06	3.83e-04	2.00e+01	1.15e+12	1.02e+04	1.62e+07	2.39e+05	4.91e+15	2.27e+06	5.73e-07
	Pior	1.78e+06	2.67e-03	2.00e+01	1.59e+12	1.04e+04	1.67e+07	3.07e+05	7.38e+15	3.10e+06	7.94e-07
	Média	1.42e+06	6.54e-04	2.00e+01	1.11e+12	1.02e+04	1.61e+07	2.47e+05	4.68e+15	2.34e+06	5.60e-07
	Desvio Padrão	1.67e+05	6.62e-04	6.50e-05	1.96e+11	1.85e+02	3.64e+05	3.03e+04	1.25e+15	2.96e+05	1.22e-07
6.0e5	Melhor	2.87e+04	3.59e-13	1.60e+01	5.67e+11	9.58e+03	1.49e+07	1.80e+05	1.65e+15	2.13e+05	3.52e-13
	Mediana	4.66e+04	3.68e-13	1.64e+01	8.15e+11	1.00e+04	1.55e+07	2.03e+05	2.65e+15	2.61e+05	3.69e-13
	Pior	9.01e+04	3.77e-13	1.67e+01	1.12e+12	1.03e+04	1.59e+07	2.57e+05	3.84e+15	3.40e+05	3.77e-13
	Média	5.09e+04	3.67e-13	1.64e+01	8.38e+11	9.99e+03	1.55e+07	2.10e+05	2.68e+15	2.67e+05	3.68e-13
	Desvio Padrão	1.58e+04	4.73e-15	1.79e-01	1.42e+11	1.87e+02	2.89e+05	2.55e+04	5.16e+14	3.23e+04	6.31e-15
3.0e6	Melhor	4.14e+00	3.50e-13	1.90e+00	3.98e+11	9.56e+03	2.03e+05	7.08e+03	2.35e+11	3.00e+04	3.52e-13
	Mediana	1.22e+01	3.61e-13	1.96e+00	8.12e+11	9.90e+03	3.30e+05	1.10e+04	4.75e+11	4.10e+04	3.61e-13
	Pior	1.37e+02	3.68e-13	2.07e+00	1.12e+12	1.02e+04	4.08e+05	1.48e+04	9.45e+11	5.43e+04	3.69e-13
	Média	1.88e+01	3.59e-13	1.96e+00	7.93e+11	9.92e+03	3.17e+05	1.11e+04	5.09e+11	4.09e+04	3.60e-13
	Desvio Padrão	2.59e+01	4.63e-15	3.62e-02	1.61e+11	1.81e+02	5.79e+04	1.97e+03	2.09e+11	6.11e+03	5.25e-15

Tabela 5.9 - CEC'2010:  $f(\mathbf{x}_{melhor})$  para as funções G11 a G20 com  $N = 1000$ .

NAFO		G11	G12	G13	G14	G15	G16	G17	G18	G19	G20
1.2e5	Melhor	1.83e+02	4.32e+02	2.26e+10	3.21e+06	1.97e-07	3.29e+02	2.43e+03	2.30e+10	2.33e+04	2.78e+10
	Mediana	1.88e+02	4.89e+02	2.70e+10	3.54e+06	2.95e-07	3.38e+02	2.57e+03	3.62e+10	2.57e+04	3.70e+10
	Pior	1.90e+02	5.18e+02	3.09e+10	4.19e+06	4.60e-07	3.43e+02	2.77e+03	4.42e+10	2.90e+04	4.55e+10
	Média	1.88e+02	4.86e+02	2.71e+10	3.55e+06	2.93e-07	3.38e+02	2.58e+03	3.55e+10	2.61e+04	3.73e+10
	Desvio Padrão	1.56e+00	2.39e+01	2.50e+09	2.23e+05	6.55e-08	3.99e+00	1.03e+02	4.70e+09	1.52e+03	4.70e+09
6.0e5	Melhor	1.77e+02	3.29e+00	1.35e+10	5.08e+05	3.43e-13	3.22e+02	9.61e+01	1.70e+10	4.71e+03	1.79e+10
	Mediana	1.80e+02	4.18e+00	1.72e+10	5.76e+05	3.73e-13	3.28e+02	1.09e+02	2.19e+10	5.32e+03	2.14e+10
	Pior	1.83e+02	5.00e+00	2.08e+10	6.59e+05	3.82e-13	3.32e+02	1.26e+02	2.61e+10	5.90e+03	2.77e+10
	Média	1.80e+02	4.15e+00	1.76e+10	5.81e+05	3.72e-13	3.28e+02	1.08e+02	2.17e+10	5.34e+03	2.16e+10
	Desvio Padrão	1.90e+00	4.05e-01	2.07e+09	4.48e+04	8.62e-15	2.45e+00	7.19e+00	2.57e+09	2.73e+02	2.18e+09
3.0e6	Melhor	1.73e+02	5.34e-07	9.16e+09	9.01e+04	3.43e-13	3.22e+02	6.02e-03	1.38e+10	6.71e+02	1.38e+10
	Mediana	1.78e+02	9.69e-07	1.20e+10	1.06e+05	3.69e-13	3.28e+02	7.86e-03	1.76e+10	7.60e+02	1.87e+10
	Pior	1.82e+02	1.54e-06	1.41e+10	1.26e+05	3.77e-13	3.32e+02	1.09e-02	2.16e+10	9.04e+02	2.21e+10
	Média	1.78e+02	1.02e-06	1.19e+10	1.07e+05	3.67e-13	3.27e+02	7.91e-03	1.78e+10	7.78e+02	1.81e+10
	Desvio Padrão	2.18e+00	2.64e-07	1.34e+09	9.70e+03	7.23e-15	2.60e+00	1.34e-03	1.79e+09	5.65e+01	2.37e+09



Tabela 5.10 - CEC'2008: classificação baseada em 15 estatísticas.

Posição	Algoritmo	Pontos
1	MTS	2002
2	<b>Fq-G</b>	1871
3	LSEDA-gl	1680
4	MLCC	1202
5	DMS-PSO	1144
6	jDEdynNP-F	1015
9	EPUS-PSO	859
7	DEwSAcc	790

Tabela 5.11 - CEC'2010: classificação baseada em 15 estatísticas.

Posição	Algoritmo	Pontos
1	<b>Fq-G</b>	4842
2	MA-SSW-Chains	4666
3	EOEA	3919
4	DASA	3864
5	DMS-PSO-SHS	3815
6	DECC-DML	3022
7	jDElsgo	3016
8	SDENS	2278

Tabela 5.12 - CEC'2008: classificação baseada nos resultados da Tabela 5.3 .

Posição	Algoritmo	Pontos
1	MTS	145
2	<b>Fq-G</b>	129
3	LSEDA-gl	93
4	jDEdynNP-F	91
5	MLCC	87
6	DMS-PSO	82
9	DEwSAcc	52
7	EPUS-PSO	38

Tabela 5.13 - CEC'2010: classificação baseada nos resultados da Tabela 5.7.

Posição	Algoritmo	Pontos
1	jDElsgo	311
2	<b>Fq-G</b>	293
3	DMS-PSO-SHS	290
4	MA-SSW-Chains	271
5	EOEA	242
6	DECC-DML	216
7	DASA	193
8	SDENS	144

## 6 Conclusões

Problemas reais que utilizam um grande número de variáveis de decisão têm surgido com muita frequência nos últimos anos e por isto o interesse em resolvê-los tem aumentado significativamente. Neste trabalho, apresentou-se o método  $Fq$ -G, desenvolvido especialmente para resolver problemas de otimização de grande porte. O método  $Fq$ -G é uma versão rápida do método  $q$ -G proposto por (SOTERRONI et al., 2011; SOTERRONI, 2012; SOTERRONI et al., 2012; SOTERRONI et al., 2013; SOTERRONI et al., 2015). É rápida no sentido que se baseia em perturbações simultâneas para calcular uma aproximação do vetor  $q$ -gradiente, vetor  $Fq$ -gradiente, que necessita de apenas duas avaliações da função objetivo por iteração, independentemente da dimensionalidade do problema a ser resolvido. Além de definir o vetor  $Fq$ -gradiente, desenvolveu-se uma nova estratégia de passo baseada na combinação da interpolação parabólica com o valor do parâmetro  $q$  na iteração  $k$ . Como esperado, a comparação entre os dois métodos mostrou o ganho do método  $Fq$ -G em relação ao método  $q$ -G. Além disso o novo método, segue a estrutura de GARS, desenvolvido por (REGIS, 2010), e portanto converge no sentido probabilístico para o ótimo global da função a ser otimizada.

O desempenho do algoritmo  $Fq$ -G foi avaliado em 27 funções teste de  $N$  dimensões, propostas na competição do IEEE *Congress on Evolutionary Computation* de 2008 e 2010. Os valores dos parâmetros de ajuste foram fixados segundo a heurística desenvolvida para o método  $Fq$ -G, assim para cada dimensão apenas um conjunto de parâmetros é utilizado na resolução de todas as funções. Os resultados obtidos pelo método  $Fq$ -G foram comparados com os obtidos por 14 AEs participantes das duas competições. Quando os algoritmos são ordenados segundo os resultados alcançados em determinadas etapas do processo de busca, o método  $Fq$ -G fica em segundo lugar no CEC'2008 e em primeiro lugar no CEC'2010. Como desejado, os resultados mostraram que o  $Fq$ -G é promissor e competitivo para solucionar problemas de grande porte.

Por outro lado, o método  $Fq$ -G apresentou um fraco desempenho na resolução da função Rosenbrock e suas variações. A topologia da função de Rosenbrock, com o mínimo global localizado dentro de um vale estreito em forma parabólica, geralmente requer muitas iterações de algoritmos de primeira ordem que dependem apenas de informações de descida do ponto de iteração atual. A solução usual para esta dificuldade consiste em introduzir informações obtidas a partir da iteração anterior ou da curvatura local (métodos de segunda ordem) no processo de busca.

Desta forma, como trabalho futuro está o desenvolvimento de uma versão rápida do  $q$ -Gradientes Conjugados ( $q$ -GC) e das  $q$ -versões dos métodos quase-Newton ( $q$ -BFGS e  $q$ -DFP) recentemente propostas (GOUVÊA et al., 2016; GOUVÊA, 2016). Sugere-se, igualmente, o desenvolvimento de uma versão do método F $q$ -G que resolva problemas que além de restrições laterais possuam restrições de igualdade e de desigualdade, inclusive não lineares. Por fim, a elaboração de uma estratégia de passo que consiga adaptar-se dependendo de características da função objetivo e estágio do processo de busca seria desejável.



## REFERÊNCIAS BIBLIOGRÁFICAS

- ABDULSADDA, A. T.; IQBAL, K. An improved spsa algorithm for system identification using fuzzy rules for training neural networks. **International Journal of Automation and Computing**, Springer, v. 8, n. 3, p. 333–339, 2011. 27
- AHMAD, B.; NTOUYAS, S.; TARIBOON, J. **Quantum calculus: new concepts, impulsive IVPs and BVPs, inequalities**. [S.l.]: World Scientific, 2016. (Trends in Abstract and Applied Analysis, v. 4). 17, 18
- ALESSANDRI, A.; PARISINI, T. Nonlinear modeling of complex large-scale plants using neural networks and stochastic approximation. **IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans**, IEEE, v. 27, n. 6, p. 750–757, 1997. 28
- ANDREANI, R.; BIRGIN, E. G.; MARTÍNEZ, J. M.; SCHUVERDT, M. L. On augmented lagrangian methods with general lower-level constraints. **SIAM Journal on Optimization**, SIAM, v. 18, n. 4, p. 1286–1309, 2007. 11
- ANDREI, N. Another conjugate gradient algorithm with guaranteed descent and conjugacy conditions for large-scale unconstrained optimization. **Journal of Optimization Theory and Applications**, v. 159, n. 1, p. 159–182, 2013. 10
- BARZILAI, J.; BORWEIN, J. M. Two-point step size gradient methods. **IMA Journal of Numerical Analysis**, Oxford University Press, v. 8, n. 1, p. 141–148, 1988. 11
- BAZARAA, M. S.; SHERALI, H. D.; SHETTY, C. M. **Nonlinear programming: theory and algorithms**. [S.l.]: John Wiley & Sons, 2013. 6, 8, 9, 10
- BENSON, H. Y.; SHANNO, D. F.; VANDERBEI, R. J. A comparative study of large-scale nonlinear optimization algorithms. In: PILLO, G. D.; MURLI, A. (Ed.). **High performance algorithms and software for nonlinear optimization**. Boston, MA: Springer US, 2003. p. 95–127. 9
- BERGH, F. van den; ENGELBRECHT, A. P. A cooperative approach to particle swarm optimization. **IEEE Transactions on Evolutionary Computation**, v. 8, n. 3, p. 225–239, June 2004. ISSN 1089-778X. 12

BHATNAGAR, S. Adaptive multivariate three-timescale stochastic approximation algorithms for simulation based optimization. **ACM Transactions on Modeling and Computer Simulation (TOMACS)**, ACM, v. 15, n. 1, p. 74–107, 2005. 27

BIRGIN, E. G.; MARTÍNEZ, J. M.; RAYDAN, M. Nonmonotone spectral projected gradient methods on convex sets. **SIAM Journal on Optimization**, SIAM, v. 10, n. 4, p. 1196–1211, 2000. 11

BOOLE, G. **A treatise on the calculus of finite differences**. 3. ed. [S.l.]: Macmillan and Company, 1880. 1

\_\_\_\_\_. **Calculus of finite differences**. 4. ed. [S.l.]: Chelsea Publishing Company, 1958. 1

BOUSSAÏD, I.; LEPAGNOT, J.; SIARRY, P. A survey on optimization metaheuristics. **Information Sciences**, Elsevier, v. 237, p. 82–117, 2013. 11

BREST, J.; ZAMUDA, A.; BOŠKOVIĆ, B.; MAUČEC, M. S.; ŽUMER, V. High-dimensional real-parameter optimization using self-adaptive differential evolution algorithm with population size reduction. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE), 2008, Hong Kong. **Proceedings...** [S.l.]: IEEE, 2008. p. 2032–2039. xvi, 14, 53, 89, 96

BREST, J.; ZAMUDA, A.; FISTER, I.; MAUČEC, M. S. Large scale global optimization using self-adaptive differential evolution algorithm. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION, 2010, Barcelona. **Proceedings...** [S.l.]: IEEE, 2010. p. 1–8. xvi, 14, 66, 90, 100, 101

BYRD, R. H.; NOCEDAL, J.; WALTZ, R. A. Knitro: An integrated package for nonlinear optimization. In: **Large-scale nonlinear optimization**. [S.l.]: Springer, 2006. p. 35–59. 11

CHEN, H.; ZHU, Y.; HU, K.; HE, X.; ; NIU, B. Cooperative approaches to bacterial foraging optimization. In: ADVANCED INTELLIGENT COMPUTING THEORIES AND APPLICATIONS WITH ASPECTS OF ARTIFICIAL INTELLIGENCE: 4TH INTERNATIONAL CONFERENCE ON INTELLIGENT COMPUTING, 2008, Shanghai. **Proceedings...** Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 541–548. ISBN 978-3-540-85984-0. 12

CHEN, W.; WEISE, T.; YANG, Z.; TANG, K. Large-scale global optimization using cooperative coevolution with variable interaction learning. In: 11TH

INTERNATIONAL CONFERENCE PARALLEL PROBLEM SOLVING FROM NATURE, 2010, Kraków. **Proceedings...** Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 300–309. ISBN 978-3-642-15871-1. Disponível em: <[http://dx.doi.org/10.1007/978-3-642-15871-1\\_31](http://dx.doi.org/10.1007/978-3-642-15871-1_31)>. 14

CHENG, S.; TING, T. O.; YANG, X.-S. Large-scale global optimization via swarm intelligence. In: KOZIEL, S.; LEIFSSON, L.; ; YANG, X.-S. (Ed.). **Solving computationally expensive engineering problems: methods and applications**. Cham: Springer International Publishing, 2014. p. 241–253. 9

COLLETTE, Y.; SIARRY, P. **Multiobjective optimization: principles and case studies**. [S.l.]: Springer Science & Business Media, 2003. 6

CONN, A. R.; GOULD, G.; TOINT, P. L. **LANCELOT**:: a fortran package for large-scale nonlinear optimization (release a). [S.l.]: Springer Verlag, 1992. 11

DAI, Y.-h.; NI, Q. Testing different conjugate gradient methods for large-scale unconstrained optimization. **Journal of Computational Mathematics**, JSTOR, p. 311–320, 2003. 10

DENG, S.; WAN, Z. A three-term conjugate gradient algorithm for large-scale unconstrained optimization problems. **Applied Numerical Mathematics**, Elsevier, v. 92, p. 70–81, 2015. 10

DONG, N.; CHEN, Z. A novel data based control method based upon neural network and simultaneous perturbation stochastic approximation. **Nonlinear Dynamics**, Springer, v. 67, n. 2, p. 957–963, 2012. 27

EL-ABD, M. A cooperative approach to the artificial bee colony algorithm. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION. **Proceedings...** [S.l.]: IEEE, 2010. p. 1–5. ISSN 1089-778X. 12

ERNST, T. **A comprehensive treatment of q-calculus**. [S.l.]: Birkhäuser Basel, 2012. 1, 17

FELLER, W. **An introduction to probability theory and its applications**: volume i. [S.l.]: John Wiley & Sons London-New York-Sydney-Toronto, 1968. 87

FLETCHER, R.; REEVES, C. M. Function minimization by conjugate gradients. **The computer journal**, Br Computer Soc, v. 7, n. 2, p. 149–154, 1964. 10, 23

GILL, P. E.; MURRAY, W.; SAUNDERS, M. A. **User's guide for SNOPT 5.3: A Fortran package for large-scale nonlinear programming.** Citeseer, 1999. 11

GILL, P. E.; MURRAY, W.; WRIGHT, M. H. **Practical optimization.** Academic press, 1981. 1

GOUVÊA, E. J. C. **Métodos convergentes de otimização global baseados no vetor  $q$ -gradiente.** Tese (Doutorado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais (INPE), 2016. 2, 23, 25, 74

GOUVÊA, É. J. C.; REGIS, R. G.; SOTERRONI, A. C.; SCARABELLO, M. C.; RAMOS, F. M. Global optimization using  $q$ -gradients. **European Journal of Operational Research**, Elsevier, v. 251, n. 3, p. 727–738, 2016. 2, 15, 17, 22, 23, 24, 32, 42, 46, 74

GRIPPO, L.; LAMPARIELLO, F.; LUCIDI, S. A nonmonotone line search technique for newton's method. **SIAM Journal on Numerical Analysis**, SIAM, v. 23, n. 4, p. 707–716, 1986. 11

HAGER, W. W.; HEARN, D. W.; PARDALOS, P. M. **Large scale optimization: state of the art.** [S.l.]: Springer US, 1994. 3

HANSEN, N.; MÜLLER, S. D.; KOUMOUTSAKOS, P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). **Evolutionary computation**, MIT Press, v. 11, n. 1, p. 1–18, 2003. 14

HESTENES, M. R.; STIEFEL, E. Methods of conjugate gradients for solving linear systems. **Journal of Research of the National Bureau of Standards**, v. 49, p. 409–436, 1952. 10

HILL, S. D.; FU, M. C. Transfer optimization via simultaneous perturbation stochastic approximation. In: SIMULATION CONFERENCE, 1995, Arlington. **Proceedings...** [S.l.]: IEEE, 1995. p. 242–249. 27

HSIEH, S.-T.; SUN, T.-Y.; LIU, C.-C.; TSAI, S.-J. Solving large scale global optimization using improved particle swarm optimizer. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE), 2008, Hong Kong. **Proceedings...** [S.l.]: IEEE, 2008. p. 1777–1784. xvi, 14, 53, 89, 99

- IZMAILOV, A.; SOLODOV, M. **Otimização**: - volume 2 métodos computacionais. [S.l.]: IMPA, 2007. 34
- JACKSON, F. H. Xi.on  $q$ -functions and a certain difference operator. **Transactions of the Royal Society of Edinburgh**, Cambridge Univ Press, v. 46, n. 02, p. 253–281, 1909. 2, 18
- \_\_\_\_\_. On  $q$ -definite integrals. **Quart. J. Pure Appl. Math.**, v. 41, p. 193–203, 1910. 2, 18
- \_\_\_\_\_.  $q$ -difference equations. **American Journal of Mathematics**, v. 32, p. 307–314, 1910. 2, 18
- KAC, V.; CHEUNG, P. **Quantum calculus**. [S.l.]: Springer-Verlag New York, 2002. (Universitext). 1, 17
- KARUSH, W. **Minima of Functions of Several Variables with Inequalities as Side Conditions**. Dissertação (Mestrado) — University of Chicago, Department of Mathematics, 1939. 8
- KIEFER, J.; WOLFOWITZ, J. et al. Stochastic estimation of the maximum of a regression function. **The Annals of Mathematical Statistics**, Institute of Mathematical Statistics, v. 23, n. 3, p. 462–466, 1952. 27
- KIRKPATRICK, S.; JR., C. D. G.; VECCHI, M. P. Optimizing by simulated annealing. **Science**, v. 220, n. 4598, p. 671–680, 1983. 21
- KOEKOEK, J.; KOEKOEK, R. A note on the  $q$ -derivative operator. **Journal of Mathematical Analysis and Applications**, v. 176, n. 2, p. 627–634, 1993. 18
- KOROŠEC, P.; TASHKOVA, K.; ŠILC, J. The differential ant-stigmergy algorithm for large-scale global optimization. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION, 2010, Barcelona. **Proceedings...** [S.l.]: IEEE, 2010. p. 1–8. xvi, 66, 91, 108, 109
- KUHN, H. W.; TUCKER, A. W. Nonlinear programming. In: **Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability**. Berkeley, Calif.: University of California Press, 1951. p. 481–492. Disponível em: <<http://projecteuclid.org/euclid.bsmsp/1200500249>>. 8
- LIU, J.; TANG, K. Scaling up covariance matrix adaptation evolution strategy using cooperative coevolution. In: YIN, H.; TANG, K.; GAO, Y.; KLAWONN, F.; LEE, M.; WEISE, T.; LI, B.; YAO, X. (Ed.). **Proceedings...** [S.l.], 2013. 14

LOSHCHILOV, I. A computationally efficient limited memory cma-es for large scale optimization. In: ACM. **Proceedings of the 2014 conference on Genetic and evolutionary computation**. [S.l.], 2014. p. 397–404. 11

LUENBERGER, D. G.; YE, Y. **Linear and nonlinear programming**. 3. ed. Stanford University, USA: Springer, 2008. 8

MACNISH, C.; YAO, X. Direction matters in high-dimensional optimisation. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE), 2008, Hong Kong. **Proceedings...** [S.l.]: IEEE, 2008. p. 2372–2379. ISSN 1089-778X. 53

MAHDAVI, S.; SHIRI, M. E.; RAHNAMAYAN, S. Metaheuristics in large-scale global continues optimization: A survey. **Information Sciences**, Elsevier, v. 295, p. 407–428, 2015. 3, 11, 12, 13, 14

MARYAK, J. L.; CHIN, D. C. Global random optimization by simultaneous perturbation stochastic approximation. **Johns Hopkins APL technical digest**, Citeseer, v. 25, n. 2, p. 91–100, 2004. 27

MILNE-THOMSON, L. M. **The calculus of finite differences**. [S.l.]: American Mathematical Soc., 1951. 1

MISHRA, V.; BHATNAGAR, S.; HEMACHANDRA, N. Discrete parameter simulation optimization algorithms with applications to admission control with dependent service times. In: 46TH IEEE CONFERENCE ON DECISION AND CONTROL, 2007, New Orleans. **Proceedings...** [S.l.]: IEEE, 2007. p. 2986–2991. 27

MOLINA, D.; LOZANO, M.; HERRERA, F. Ma-sw-chains: Memetic algorithm based on local search chains for large scale continuous global optimization. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION, 2010, Barcelona. **IEEE Congress on Evolutionary Computation**. [S.l.]: IEEE, 2010. p. 1–8. xvi, 15, 66, 91, 106, 107

NOCEDAL, J. Updating quasi-newton matrices with limited storage. **Mathematics of computation**, v. 35, n. 151, p. 773–782, 1980. 10, 11

NOCEDAL, J.; WRIGHT, S. **Numerical optimization**. [S.l.]: Springer Science & Business Media, 2006. 1

- OMIDVAR, M. N.; LI, X.; TANG, K. Designing benchmark problems for large-scale continuous optimization. **Information Sciences**, Elsevier, v. 316, p. 419–436, 2015. 3
- OMIDVAR, M. N.; LI, X.; YAO, X. Cooperative co-evolution with delta grouping for large scale non-separable function optimization. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION, 2010, Barcelona. **Proceedings...** [S.l.]: IEEE, 2010. p. 1–8. xvi, 66, 90, 102, 103
- POTTER, M. A.; JONG, K. A. D. A cooperative coevolutionary approach to function optimization. In: DAVIDOR, Y.; SCHWEFEL, H.-P.; ; MÄNNER, R. (Ed.). **Proceedings...** [S.l.]: Springer Berlin Heidelberg, 1994. p. 249–257. 12
- RAY, T.; YAO, X. A cooperative coevolutionary algorithm with correlation based adaptive variable partitioning. In: 2009 IEEE CONGRESS ON EVOLUTIONARY COMPUTATION, 2009, Trondheim. **Proceedings...** [S.l.]: IEEE, 2009. p. 983–989. ISSN 1089-778X. 13
- RAYDAN, M. The barzilai and borwein gradient method for the large scale unconstrained minimization problem. **SIAM Journal on Optimization**, SIAM, v. 7, n. 1, p. 26–33, 1997. 11
- REGIS, R. G. Convergence guarantees for generalized adaptive stochastic search methods for continuous global optimization. **European Journal of Operational Research**, Elsevier, v. 207, n. 3, p. 1187–1202, 2010. 23, 42, 43, 44, 45, 46, 48, 73
- RESNICK, S. I. **A probability path**. [S.l.]: Birkauer Boston, 1999. 43, 45, 87
- RIBEIRO, A. A.; KARAS, E. W. **Otimização contínua: aspectos teóricos e computacionais**. [S.l.]: Cengage Learning, 2013. 33
- ROMA, M. Large scale unconstrained optimization. In: FLOUDAS, C. A.; PARDALOS, P. M. (Ed.). **Encyclopedia of Optimization**. Boston, MA: Springer US, 2001. p. 1233–1240. ISBN 978-0-306-48332-5. Disponível em: <[http://dx.doi.org/10.1007/0-306-48332-7\\_250](http://dx.doi.org/10.1007/0-306-48332-7_250)>. 9
- ROYDEN, H. **Real analysis: 3d ed.** [S.l.]: Macmillan, 1988. 87
- SHANNO, D. F. Conditioning of quasi-newton methods for function minimization. **Mathematics of computation**, v. 24, n. 111, p. 647–656, 1970. 10
- SHI, Y.-j.; TENG, H.-f.; LI, Z.-q. Cooperative co-evolutionary differential evolution for function optimization. In: WANG, L.; CHEN, K.; ONG, Y. S. (Ed.).

- Proceedings...** Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 1080–1088. ISBN 978-3-540-31858-3. Disponível em: [http://dx.doi.org/10.1007/11539117\\_147](http://dx.doi.org/10.1007/11539117_147). 12
- SOLIS, F. J.; WETS, R. J.-B. Minimization by random search techniques. **Mathematical Operations Research**, INFORMS, v. 6, n. 1, p. 19–30, 1981. 15, 91
- SOTERRONI, A. C. **O método do  $q$ -gradiente para otimização global**. Tese (Doutorado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais (INPE), 2012. 19, 21, 73
- SOTERRONI, A. C.; GALSKI, R. L.; RAMOS, F. M. The  $q$ -gradient vector for unconstrained continuous optimization problems. In: **Operations Research Proceedings 2010**. [S.l.]: Springer, 2011. p. 365–370. 15, 17, 18, 21, 22, 73
- \_\_\_\_\_. The  $q$ -gradient method for global optimization. **arXiv preprint arXiv:1209.2084**, 2012. 1, 2, 20, 22, 23, 73
- \_\_\_\_\_. The  $q$ -gradient method for continuous global optimization. **AIP Conference Proceedings**, v. 1558, n. 1, p. 2389–2393, 2013. 22, 73
- SOTERRONI, A. C.; GALSKI, R. L.; SCARABELLO, M. C.; RAMOS, F. M. The  $q$ -g method. **SpringerPlus**, Springer, v. 4, n. 1, p. 1–15, 2015. 20, 22, 32, 40, 73
- SPALL, J. C. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. **Automatic Control, IEEE Transactions on**, v. 37, n. 3, p. 332–341, Mar 1992. 27, 28, 29, 31
- \_\_\_\_\_. Implementation of the simultaneous perturbation algorithm for stochastic optimization. **Aerospace and Electronic Systems, IEEE Transactions on**, v. 34, n. 3, p. 817–823, Jul 1998. 29
- \_\_\_\_\_. An overview of the simultaneous perturbation method for efficient optimization. **Johns Hopkins APL Technical Digest**, v. 19, n. 4, p. 482 – 492, 1998. 28
- \_\_\_\_\_. **Introduction to stochastic search and optimization: estimation, simulation, and control**. [S.l.]: John Wiley & Sons, 2005. 29, 30, 31
- \_\_\_\_\_. Stochastic optimization. In: GENTLE, J. E.; HÄRDLE, W. K.; MORI, Y. (Ed.). **Handbook of Computational Statistics**. [S.l.]: Springer Berlin



Heidelberg, 2012, (Springer Handbooks of Computational Statistics). p. 173–201. ISBN 978-3-642-21550-6. 3

TANG, K.; LI, X.; SUGANTHAN, P. N.; YANG, Z.; WEISE, T. **Benchmark Functions for the CEC'2010 Special Session and Competition on Large Scale Global Optimization**. Nature Inspired Computation and Applications Laboratory, USTC, China: [s.n.], 2009. Disponível em: <<http://nical.ustc.edu.cn/cec10ss.php>>. 12, 51, 60, 66

TANG, K.; YÁO, X.; SUGANTHAN, P. N.; MACNISH, C.; CHEN, Y.-P.; CHEN, C.-M.; YANG, Z. **Benchmark functions for the CEC'2008 special session and competition on large scale global optimization**. Nature Inspired Computation and Applications Laboratory, USTC, China: [s.n.], November 2007. 153–177 p. Disponível em: <<http://nical.ustc.edu.cn/cec08ss.php>>. 3, 40, 51, 52, 54

TSENG, L.-Y.; CHEN, C. Multiple trajectory search for large scale global optimization. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE), 2008, Hong Kong. **Proceedings...** [S.l.]: IEEE, 2008. p. 3052–3059. xvi, 15, 53, 89, 98

VANDERBEI, R. J.; SHANNO, D. F. An interior-point algorithm for nonconvex nonlinear programming. **Computational Optimization and Applications**, Springer, v. 13, n. 1-3, p. 231–252, 1999. 11

VITSENTIY, V. Improvement of human-machine interaction with applications to information retrieval system. In: 2002 FIRST INTERNATIONAL IEEE SYMPOSIUM INTELLIGENT SYSTEMS, 2002, Varna. **Proceedings...** [S.l.]: IEEE, 2002. v. 1, p. 363–368. 28

WÄCHTER, A.; BIEGLER, T. L. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. **Mathematical Programming**, v. 106, n. 1, p. 25–57, 2006. ISSN 1436-4646. Disponível em: <<http://dx.doi.org/10.1007/s10107-004-0559-y>>. 11

WANG, H.; WU, Z.; RAHNAMAYAN, S.; JIANG, D. Sequential de enhanced by neighborhood search for large scale global optimization. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION, 2010, Barcelona. **IEEE Congress on Evolutionary Computation**. [S.l.]: IEEE, 2010. p. 1–7. xvi, 14, 66, 91, 110, 111

WANG, Y.; LI, B. A restart univariate estimation of distribution algorithm: sampling under mixed gaussian and lévy probability distribution. In: IEEE

CONGRESS ON EVOLUTIONARY COMPUTATION (IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE), 2008, Hong Kong. **Proceedings...** [S.l.]: IEEE, 2008. p. 3917–3924. [xvi](#), [53](#), [90](#), [97](#)

\_\_\_\_\_. Two-stage based ensemble optimization for large-scale global optimization. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION, 2010, Barcelona. **Proceedings...** [S.l.]: IEEE, 2010. p. 1–8. [xvi](#), [15](#), [66](#), [91](#), [112](#), [113](#)

WOLPERT, D. H.; MACREADY, W. G. No free lunch theorems for optimization. **IEEE transactions on evolutionary computation**, v. 1, n. 1, p. 67–82, 1997. [1](#)

XING, X.; DAMODARAN, M. Assessment of simultaneous perturbation stochastic approximation method for wing design optimization. **Journal of Aircraft**, v. 39, n. 2, p. 379–381, 2002. [27](#)

\_\_\_\_\_. Inverse design of transonic airfoils using parallel simultaneous perturbation stochastic approximation. **Journal of aircraft**, v. 42, n. 2, p. 568–570, 2005. [27](#)

XU, Y.; SONG, J.; NECHYBA, M. C.; YAM, Y. Performance evaluation and optimization of human control strategy. **Robotics and Autonomous Systems**, Elsevier, v. 39, n. 1, p. 19–36, 2002. [28](#)

YANG, Z.; TANG, K.; YAO, X. Large scale evolutionary optimization using cooperative coevolution. **Information Sciences**, v. 178, n. 15, p. 2985 – 2999, 2008. ISSN 0020-0255. Nature Inspired Problem-Solving. [12](#)

\_\_\_\_\_. Multilevel cooperative coevolution for large scale optimization. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE), 2008, Hong Kong. **Proceedings...** [S.l.]: IEEE, 2008. p. 1663–1670. [xvi](#), [13](#), [53](#), [89](#), [95](#)

\_\_\_\_\_. Self-adaptive differential evolution with neighborhood search. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE), 2008, Hong Kong. **Proceedings...** [S.l.]: IEEE, 2008. p. 1110–1116. ISSN 1089-778X. [13](#)

YU, G.; ZHAO, Y.; WEI, Z. A descent nonlinear conjugate gradient method for large-scale unconstrained optimization. **Applied Mathematics and Computation**, v. 187, n. 2, p. 636 – 643, 2007. ISSN 0096-3003. [10](#)

ZAMUDA, A.; BREST, J.; BOŠKOVIĆ, B. et al. Large scale global optimization using differential evolution with self-adaptation and cooperative co-evolution. In:

IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE), 2008, Hong Kong. **Proceedings...** [S.l.]: IEEE, 2008. p. 3718–3725. xv, 53, 89, 93

ZHANG, J.; SANDERSON, A. C. Jade: Adaptive differential evolution with optional external archive. **IEEE Transactions on Evolutionary Computation**, v. 13, n. 5, p. 945–958, Oct 2009. ISSN 1089-778X. 14

ZHAO, S.-Z.; LIANG, J. J.; SUGANTHAN, P. N.; TASGETIREN, M. F. Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE), 2008, Hong Kong. **Proceedings...** [S.l.]: IEEE, 2008. p. 3845–3852. xv, 14, 53, 90, 94

ZHAO, S. Z.; SUGANTHAN, P. N.; DAS, S. Dynamic multi-swarm particle swarm optimizer with sub-regional harmony search. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION, 2010, Barcelona. **Proceedings...** [S.l.]: IEEE, 2010. p. 1–8. xvi, 66, 90, 104, 105



## APÊNDICE A - Definições

Neste apêndice seção são apresentadas algumas definições utilizadas na seção 4.5.

**Definição A.1.** (ROYDEN, 1988) O conjunto  $E$  é mensurável se para cada  $A$  tem-se  $\mu^*A = \mu^*(A \cap E) + \mu^*(A \cap E^C)$ , onde  $\mu^*$  é uma medida exterior.

**Definição A.2.** (ROYDEN, 1988) Uma função  $f : E \rightarrow \bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$  é mensurável se  $E$  é mensurável e  $f$  satisfaz uma das condições abaixo:

- i.  $\forall \gamma \in \mathbb{R}, \{x \in E : f(x) > \gamma\}$  é mensurável;
- ii.  $\forall \gamma \in \mathbb{R}, \{x \in E : f(x) \geq \gamma\}$  é mensurável;
- iii.  $\forall \gamma \in \mathbb{R}, \{x \in E : f(x) < \gamma\}$  é mensurável;
- iv.  $\forall \gamma \in \mathbb{R}, \{x \in E : f(x) \leq \gamma\}$  é mensurável.

**Definição A.3.** (RESNICK, 1999) Uma  $\sigma$ -álgebra  $\mathcal{B}$  sobre  $\Omega$  é uma coleção não vazia de subconjuntos de  $\Omega$  que é fechado sobre operações contáveis de união, interseção e complemento de conjuntos. O conjunto mínimo de postulados para  $\mathcal{B}$  ser uma  $\sigma$ -álgebra é:

- i.  $\Omega \in \mathcal{B}$ .
  - ii.  $B \in \mathcal{B}$ , implica que  $B^C \in \mathcal{B}$ .
  - iii.  $B_i \in \mathcal{B}, i \geq 1$ , implica que  $\cup_{i=1}^{\infty} B_i \in \mathcal{B}$ .
- Além disto, se  $B_i \in \mathcal{B}$ , para  $i \geq 1$ , então  $\cap_{i=1}^{\infty} B_i \in \mathcal{B}$ .

Na Teoria da probabilidade, o espaço amostral é uma  $\sigma$ -álgebra. Isto permite flexibilidade suficiente para construir novos eventos a partir de velhos eventos (fechados) (RESNICK, 1999).

**Definição A.4.** (FELLER, 1968) Dada uma família  $\mathcal{F}$  de conjuntos de  $\Omega$ , a menor  $\sigma$ -álgebra contendo todos os conjuntos de  $\mathcal{F}$  é chamada de  $\sigma$ -álgebra gerada por  $\mathcal{F}$ . Em particular, os conjuntos gerados por intervalos do  $\mathbb{R}^N$  são chamados conjuntos de Borel do  $\mathbb{R}^N$ .



## APÊNDICE B - Breve descrição dos Algoritmos Evolutivos que participaram do CEC'2008 e do CEC'2010

Neste apêndice é apresentado uma breve descrição dos algoritmos evolutivos que participaram das competições do CEC de 2008 e 2010, e que foram comparados com o Fq-G.

### B.1 CEC'2008

- **MLCC** *Multilevel Cooperative Coevolution* (YANG et al., 2008b)  
Algoritmo baseado em coevolução cooperativa, o MLCC inicialmente cria um conjunto com diversos tamanhos possíveis de grupos. O algoritmo se autoadapta para selecionar os tamanhos dos subproblemas mais apropriados com base nos registros do desempenho.
- **EPUS-PSO** *Efficient Population Utilization Strategy for Particle Swarm Optimizer* (HSIEH et al., 2008)  
Baseado em Otimização por Enxame de Partícula, o EPUS-PSO utiliza um gerenciador de população para aumentar a capacidade de busca e direcionar as partículas no enxame de forma mais eficiente. Além disso, estratégias de compartilhamento são utilizadas para impedir que as soluções fiquem presas no mínimo local e para que o método consiga encontrar a solução global de maneira mais fácil.
- **jDEdynNP-F** (BREST et al., 2008)  
É um algoritmo de Evolução Diferencial (ED) que possui mecanismos de autoajuste dos parâmetros de controle F (controle da mutação) e CR (controle do crossover), e de mudança do sinal do parâmetro F. O jDEdynNP-F também utiliza uma estratégia de redução do tamanho da população durante o processo de otimização.
- **MTS** *Multiple Trajectory Search* (TSENG; CHEN, 2008)  
MTS utiliza múltiplos agentes para encontrar as melhores soluções dentro do espaço de busca. Cada agente é aplicado a um dos três métodos de busca local disponíveis. O algoritmo também utiliza o conceito de soluções de “primeiro plano”, que são soluções de alta qualidade encontradas nas iterações anteriores e usadas como base para encontrar novas e melhores soluções nas novas iterações do algoritmo.
- **DEwSAcc** *Differential Evolution with Self-Adaptive cooperative coevolution* (ZAMUDA et al., 2008)

Baseado em ED, o DEwSAcc inclui uma autoadaptação dos dois principais parâmetros de controle F e CR, e a utilização da coevolução cooperativa para decompor o problema.

- **DMS-PSO** *Dynamic Multi-Swarm Particle Swarm Optimizer* (ZHAO et al., 2008)

No DMS-PSO a população é dividida em um grande número de subenxames; esses subenxames são frequentemente reagrupados e informações são trocadas entre as partículas em todo o enxame. O método quase-Newton é combinado ao DMS-PSO para melhorar a habilidade de busca local do algoritmo.

- **LSEDA-gl** (WANG; LI, 2008)

Este método é uma versão do Algoritmo de Estimação de Distribuição (AED) desenvolvida para problemas de grande porte. LSEDA-gl utiliza três estratégias: amostragem sob distribuição de probabilidade de Gauss e Lévy, estratégia de controle do desvio padrão e de reinicialização.

## B.2 CEC'2010

- **jDElsgo** (BREST et al., 2010)

jDElsgo, é uma extensão do algoritmo jDEdynNP-F desenvolvido pelos mesmos autores. Esse algoritmo usa um valor pequeno para o parâmetro de controle F, com o objetivo de aprimorar o melhor indivíduo conhecido depois que metade do número máximo de avaliações da função objetivo é atingido.

- **DECC-DML** (OMIDVAR et al., 2010)

Um algoritmo de evolução diferencial que adota a coevolução cooperativa. DECC-DML utiliza um método para identificar a interação entre as variáveis de decisão e também uma decomposição semelhante à usada no MLCC.

- **DMS-PSO-SHS** *Dynamic Multi-Swarm Particle Swarm Optimizer with Sub-regional Harmony Search* (ZHAO et al., 2010)

O DMS-PSO-SHS é a hibridização do DMS-PSO com o SHS (*Sub-regional Harmony Search*). O SHS é aplicado aos subenxames e gera novas harmonias de acordo com o melhor indivíduo conhecido, a harmonia com melhor aptidão substituirá o melhor indivíduo dentre os mais próximos.



- **MA-SW-Chain** *Memetic Algorithm Based on Local Search Chains* (MOLINA et al., 2010)  
O MA-SW-Chain (MOLINA et al., 2010) é um algoritmo memético que utiliza o método de busca local *Solis Wets* (SOLIS; WETS, 1981). O método foi projetado para adaptar a intensidade do algoritmo de busca local, com o propósito de explorar mais a região que tem os indivíduos mais promissores.
- **DASA** *Differential Ant-Stigmergy Algorithm* (KOROŠEC et al., 2010)  
DASA é uma adaptação do algoritmo da otimização por colônia de formigas para problemas de otimização contínua. Esse algoritmo usa uma nova maneira de transformar um problema de otimização contínuo em um problema de busca em grafo. Os parâmetros atribuídos aos vértices grafo são usados como um meio para mover-se através do espaço de busca.
- **SDENS** *Sequential Differential Evolution Enhanced by Neighborhood Search* (WANG et al., 2010)  
O SDENS é um algoritmo sequencial DE reforçado com busca na vizinhança que consiste basicamente de dois passos: primeiro, para cada indivíduo, é criado dois indivíduos experimentais através de estratégias de busca em vizinhança local e global; segundo, seleciona o indivíduo mais apto entre o indivíduo atual e os dois indivíduos experimentais para fazer parte da nova geração.
- **EOEA** *Two-stage based ensemble optimization evolutionary algorithm*(WANG; LI, 2010)  
No EOEA o procedimento de pesquisa é dividido em duas etapas. A fase de encolhimento global procura encolher o espaço de busca a procura de regiões promissoras o mais rápido possível. A etapa de exploração local tem como objetivo explorar intensamente a área que foi limitada na primeira fase para encontrar a melhor solução possível.



## ANEXO A - Resultados obtidos pelos Algoritmos Evolutivos que participaram do CEC'2008 e do CEC'2010

Neste anexo são reproduzidos os resultados obtidos AEs participantes do CEC'2008 e do CEC'2010 e utilizados na classificação realizada no Capítulo 5.

Tabela A.1 - DEwSacc (ZAMUDA et al., 2008); CEC'2008;  $f(\mathbf{x}) - f(\mathbf{x}^*)$  para F1 a F6;  $f(\mathbf{x})$  para F7;  $N = 1000$ .

NAFO		F1	F2	F3	F4	F5	F6	F7
5e4	1° (Melhor)	3.15E+06	1.53E+02	1.68E+12	1.76e+04	2.83e+04	2.11e+01	-6.89e+03
	13° (Mediana)	3.35e+06	1.62e+02	1.91e+12	1.80e+04	2.97e+04	2.12e+01	-6.79e+03
	25° (Pior)	3.54e+06	1.66e+02	2.28e+12	1.86e+04	3.14e+04	2.12e+01	-6.71e+03
	Média	3.34e+06	1.61e+02	1.91e+12	1.81e+04	2.97e+04	2.12e+01	-6.79e+03
	Desvio-padrão	8.77e+04	3.2054	1.56e+11	2.44e+02	8.38e+02	2.86e-02	3.91e+01
5e5	1° (Melhor)	4.96e+05	1.22e+02	1.40e+11	1.07e+04	4.23e+03	1.73e+01	-7.94e+03
	13° (Mediana)	5.22e+05	1.29e+02	1.53e+11	1.10e+04	4.72e+03	1.78e+01	-7.85e+03
	25° (Pior)	6.22e+05	1.36e+02	1.87e+11	1.15e+04	5.64e+03	1.85e+01	-7.76e+03
	Média	5.36e+05	1.28e+02	1.55e+11	1.10e+04	4.76e+03	1.78e+01	-7.85e+03
	Desvio Padrão	3.52e+04	3.4718	1.10e+10	1.94e+02	3.31e+02	2.69e-01	5.09e+01
5e6	1° (Melhor)	3.17e-03	9.25e+01	6.93e+03	1.58e+03	2.32e-04	1.5605	-1.15e+04
	13° (Mediana)	8.06e-03	9.64e+01	9.01e+03	1.82e+03	6.79e-04	2.3699	-1.04e+04
	25° (Pior)	2.65e-02	9.92e+01	1.16e+04	2.10e+03	2.03e-02	2.745	-1.03e+04
	Média	8.79e-03	9.61e+01	9.15e+03	1.82e+03	3.58e-03	2.2956	-1.06e+04
	Desvio Padrão	5.27e-03	1.8194	1.26e+03	1.38e+02	5.74e-03	2.98e-01	4.18e+02

Tabela A.2 - DMS-PSO (ZHAO et al., 2008); CEC'2008;  $f(\mathbf{x}) - f(\mathbf{x}^*)$  para F1 a F6;  $f(\mathbf{x})$  para F7;  $N = 1000$ .

NAFO		F1	F2	F3	F4	F5	F6	F7
5e4	1° (Melhor)	4.24e+05	1.76e+02	3.81e+12	2.13e+04	5.98e+03	2.14e+01	-6.55e+03
	13° (Mediana)	4.76e+05	1.81e+02	4.20e+12	2.18e+04	6.95e+03	2.14e+01	-6.45e+03
	25° (Pior)	4.90e+05	1.83e+02	4.66e+12	2.23e+04	8.39e+03	2.14e+01	-6.39e+03
	Média	4.65e+05	1.81e+02	3.31e+12	2.18e+04	7.11e+03	2.14e+01	-6.44e+03
	Desvio-padrão	2.81e+04	1.74e+00	1.58e+12	2.49e+02	7.81e+02	1.16e-02	4.83e+01
5e5	1° (Melhor)	0.00e+00	1.14e+02	3.69e+11	1.42e+04	0.00e+00	1.91e+01	-6.74e+03
	13° (Mediana)	0.00e+00	1.17e+02	3.91e+11	1.44e+04	0.00e+00	1.92e+01	-6.71e+03
	25° (Pior)	0.00e+00	1.19e+02	4.26e+11	1.45e+04	0.00e+00	1.92e+01	-6.66e+03
	Média	0.00e+00	1.17e+02	2.94e+11	1.43e+04	0.00e+00	1.92e+01	-6.71e+03
	Desvio Padrão	0.00e+00	1.07e+00	1.69e+11	7.36e+01	0.00e+00	4.06e-02	2.07e+01
5e6	1° (Melhor)	0.00e+00	9.00e+01	8.28e+09	3.55e+03	0.00e+00	7.58e+00	-7.54e+03
	13° (Mediana)	0.00e+00	9.17e+01	9.05e+09	3.82e+03	0.00e+00	7.76e+00	-7.51e+03
	25° (Pior)	0.00e+00	9.28e+01	9.45e+09	4.11e+03	0.00e+00	7.90e+00	-7.49e+03
	Média	0.00e+00	9.15e+01	8.98e+09	3.84e+03	0.00e+00	7.76e+00	-7.51e+03
	Desvio Padrão	0.00e+00	7.14e-01	4.39e+08	1.71e+02	0.00e+00	8.92e-02	1.64e+01

Tabela A.3 - MLCC (YANG et al., 2008b); CEC'2008;  $f(\mathbf{x}) - f(\mathbf{x}^*)$  para F1 a F6;  $f(\mathbf{x})$  para F7;  $N = 1000$ .

NAFO		F1	F2	F3	F4	F5	F6	F7
5e4	1° (Melhor)	2.55e+05	1.77e+02	1.61e+10	5.57e+03	2.38e+03	1.83e+01	-1.08e+04
	13° (Mediana)	4.71e+05	1.85e+02	6.35e+10	9.93e+03	4.16e+03	2.00e+01	-8.74e+03
	25° (Pior)	5.79e+05	1.89e+02	1.00e+11	1.05e+04	5.08e+03	2.05e+01	-7.75e+03
	Média	4.23e+05	1.85e+02	6.34e+10	8.80e+03	3.83e+03	1.98e+01	-8.90e+03
	Desvio-padrão	9.95e+04	2.78e+00	2.32e+10	1.85e+03	8.68e+02	5.93e-01	1.06e+03
5e5	1° (Melhor)	1.10e-01	1.59e+02	5.85e+03	5.90e+02	3.90e-03	2.53e-02	-1.30e+04
	13° (Mediana)	3.59e-01	1.73e+02	8.03e+03	7.42e+02	2.56e-02	1.03e-01	-1.28e+04
	25° (Pior)	2.47e+00	1.79e+02	1.04e+04	1.22e+03	8.75e-02	1.37e+00	-1.23e+04
	Média	6.15e-01	1.72e+02	7.91e+03	7.83e+02	2.98e-02	2.02e-01	-1.27e+04
	Desvio Padrão	6.58e-01	4.00e+00	1.12e+03	1.47e+02	2.26e-02	3.14e-01	1.94e+02
5e6	1° (Melhor)	7.39e-13	9.97e+01	1.46e+03	2.39e-12	3.69e-13	9.10e-13	-1.47e+04
	13° (Mediana)	8.53e-13	1.08e+02	1.83e+03	2.84e-12	4.26e-13	1.02e-12	-1.47e+04
	25° (Pior)	9.10e-13	1.19e+02	2.05e+03	1.14e-09	4.55e-13	1.19e-12	-1.47e+04
	Média	8.46e-13	1.09e+02	1.80e+03	1.37e-10	4.18e-13	1.06e-12	-1.47e+04
	Desvio Padrão	5.01e-14	4.75e+00	1.58e+02	3.37e-10	2.78e-14	7.68e-14	1.52e+01

Tabela A.4 - jDEdynNP-F (BREST et al., 2008); CEC'2008;  $f(\mathbf{x}) - f(\mathbf{x}^*)$  para F1 a F6;  $f(\mathbf{x})$  para F7;  $N = 1000$ .

NAFO		F1	F2	F3	F4	F5	F6	F7
5e4	1° (Melhor)	3.64e+06	1.73e+02	2.97e+12	1.94e+04	3.33e+04	2.14e+01	-6.93e+03
	13° (Mediana)	4.22e+06	1.77e+02	3.55e+12	2.05e+04	3.74e+04	2.14e+01	-6.69e+03
	25° (Pior)	4.39e+06	1.78e+02	4.23e+12	2.09e+04	3.99e+04	2.14e+01	-6.62e+03
	Média	4.14e+06	1.77e+02	3.57e+12	2.04e+04	3.74e+04	2.14e+01	-6.71e+03
	Desvio-padrão	2.20e+05	1.1312	3.31e+11	3.58e+02	1.52e+03	1.47e-02	6.79e+01
5e5	1° (Melhor)	2.69e+05	1.52e+02	5.80e+10	1.09e+04	2.15e+03	1.75e+01	-8.07e+03
	13° (Mediana)	2.94e+05	1.54e+02	7.24e+10	1.12e+04	2.58e+03	1.81e+01	-7.86e+03
	25° (Pior)	3.17e+05	1.57e+02	8.11e+10	1.13e+04	2.78e+03	1.84e+01	-7.75e+03
	Média	2.92e+05	1.54e+02	7.10e+10	1.12e+04	2.53e+03	1.81e+01	-7.86e+03
	Desvio Padrão	1.22e+04	9.80e-01	5.99e+09	1.14e+02	1.66e+02	1.81e-01	6.86e+01
5e6	1° (Melhor)	1.14e-13	1.60e+01	1.14e+03	9.27e-06	2.84e-14	1.31e-12	-1.36e+04
	13° (Mediana)	1.14e-13	1.93e+01	1.29e+03	4.22e-05	2.84e-14	6.85e-12	-1.35e+04
	25° (Pior)	1.14e-13	2.32e+01	1.57e+03	1.48e-03	5.68e-14	1.20e-10	-1.34e+04
	Média	1.14e-13	1.95e+01	1.31e+03	2.17e-04	3.98e-14	1.47e-11	-1.35e+04
	Desvio Padrão	0	2.2525	1.36e+02	4.06e-04	1.42e-14	2.43e-11	4.60e+01

Tabela A.5 - LSEDA-lg (WANG; LI, 2008); CEC'2008;  $f(\mathbf{x}) - f(\mathbf{x}^*)$  para F1 a F6;  $f(\mathbf{x})$  para F7;  $N = 1000$ .

NAFO		F1	F2	F3	F4	F5	F6	F7
5e4	1° (Melhor)	7.98e+03	9.05e+01	7.38e+08	1.34e+04	7.83e+01	5.09e+00	-9.86e+03
	13° (Mediana)	9.75e+03	9.32e+01	8.97e+08	1.36e+04	8.59e+01	5.55e+00	-9.64e+03
	25° (Pior)	1.16e+04	9.41e+01	1.06e+09	1.37e+04	9.89e+01	5.68e+00	-9.42e+03
	Média	9.95e+03	9.28e+01	9.05e+08	1.36e+04	8.83e+01	5.47e+00	-9.66e+03
	Desvio-padrão	1.08e+03	1.10e+00	9.81e+07	9.35e+01	7.97e+00	2.11e-01	1.43e+02
5e5	1° (Melhor)	9.09e-13	1.48e+01	3.25e+03	4.17e+03	2.64e-12	2.27e-10	-1.35e+04
	13° (Mediana)	2.05e-12	1.56e+01	3.62e+03	4.47e+03	3.38e-12	3.19e-10	-1.34e+04
	25° (Pior)	1.89e-08	1.75e+01	3.97e+03	4.75e+03	3.44e-02	1.33e-04	-1.33e+04
	Média	2.47e-09	1.59e+01	3.59e+03	4.49e+03	1.57e-03	2.29e-05	-1.34e+04
	Desvio Padrão	6.26e-09	1.01e+00	2.33e+02	1.36e+02	1.54e-02	4.77e-05	6.17e+01
5e6	1° (Melhor)	2.84e-13	9.79e-06	1.57e+03	5.05e+02	1.71e-13	4.26e-13	-1.36e+04
	13° (Mediana)	3.41e-13	1.04e-05	1.68e+03	5.45e+02	1.71e-13	4.26e-13	-1.35e+04
	25° (Pior)	3.41e-13	1.12e-05	2.01e+03	5.81e+02	1.71e-13	4.26e-13	-1.34e+04
	Média	3.22e-13	1.04e-05	1.73e+03	5.45e+02	1.71e-13	4.26e-13	-1.35e+04
	Desvio Padrão	2.84e-14	5.11e-07	1.40e+02	1.80e+01	0.00e+00	0.00e+00	5.72e+01

Tabela A.6 - MTS (TSENG; CHEN, 2008); CEC'2008;  $f(\mathbf{x}) - f(\mathbf{x}^*)$  para F1 a F6;  $f(\mathbf{x})$  para F7;  $N = 1000$ .

NAFO		F1	F2	F3	F4	F5	F6	F7
5e4	1° (Melhor)	1.82e+05	9.48e+01	5.09e-01	1.83e+03	1.37e+01	1.35e+01	-1.19e+04
	13° (Mediana)	2.01e+05	9.49e+01	1.45e+00	4.73e+03	1.96e+03	1.37e+01	-1.18e+04
	25° (Pior)	2.51e+05	1.20e+02	5.50e+01	4.92e+03	2.20e+03	1.43e+01	-1.18e+04
	Média	2.10e+05	9.60e+01	3.64e+00	4.51e+03	1.91e+03	1.37e+01	-1.18e+04
	Desvio-padrão	2.13e+04	5.00e+00	1.07e+01	7.90e+02	4.11e+02	1.90e-01	1.51e+01
5e5	1° (Melhor)	0.00e+00	4.14e+01	2.70e-04	0.00e+00	0.00e+00	2.69e-11	-1.40e+04
	13° (Mediana)	0.00e+00	4.68e+01	2.31e-02	9.92e+00	5.68e-14	1.92e-09	-1.39e+04
	25° (Pior)	3.21e-02	4.73e+01	2.32e-02	1.49e+03	1.21e-06	1.30e-02	-1.39e+04
	Média	1.36e-03	4.65e+01	1.98e-02	2.83e+02	9.48e-08	7.20e-04	-1.39e+04
	Desvio Padrão	6.42e-03	1.16e+00	7.82e-03	4.75e+02	2.61e-07	2.67e-03	3.45e+01
5e6	1° (Melhor)	0.00e+00	3.50e-02	2.70e-06	0.00e+00	0.00e+00	1.16e-11	-1.41e+04
	13° (Mediana)	0.00e+00	4.75e-02	3.99e-04	0.00e+00	0.00e+00	1.24e-11	-1.40e+04
	25° (Pior)	0.00e+00	7.64e-02	4.00e-04	0.00e+00	0.00e+00	1.32e-11	-1.39e+04
	Média	0.00e+00	4.72e-02	3.41e-04	0.00e+00	0.00e+00	1.24e-11	-1.40e+04
	Desvio Padrão	0.00e+00	8.58e-03	1.38e-04	0.00e+00	0.00e+00	4.48e-13	2.94e+01



Tabela A.7 - EPUS-PSO (HSIEH et al., 2008); CEC'2008;  $f(\mathbf{x}) - f(\mathbf{x}^*)$  para F1 a F6;  $f(\mathbf{x})$  para F7;  $N = 1000$ .

NAFO		F1	F2	F3	F4	F5	F6	F7
5e4	1° (Melhor)	2.56e+06	5.01e+01	8.37e+11	1.52e+04	2.19e+04	2.10e+01	-6.61e+03
	13° (Mediana)	2.62e+06	6.10e+01	8.83e+11	1.62e+04	2.34e+04	2.10e+01	-6.48e+03
	25° (Pior)	2.74e+06	9.66e+01	9.26e+11	1.72e+04	2.43e+04	2.10e+01	-6.44e+03
	Média	2.63e+06	6.79e+01	8.88e+11	1.61e+04	2.34e+04	2.10e+01	-6.49e+03
	Desvio-padrão	4.35e+04	1.65e+01	2.30e+10	6.11e+02	5.01e+02	1.33e-02	3.42e+01
5e5	1° (Melhor)	3.21e+05	4.75e+01	3.13e+10	1.03e+04	2.82e+03	2.09e+01	-6.62e+03
	13° (Mediana)	3.44e+05	4.81e+01	3.73e+10	1.06e+04	3.10e+03	2.10e+01	-6.55e+03
	25° (Pior)	3.74e+05	5.77e+01	4.63e+10	1.08e+04	3.32e+03	2.10e+01	-6.51e+03
	Média	3.46e+05	4.85e+01	3.77e+10	1.06e+04	3.09e+03	2.10e+01	-6.55e+03
	Desvio Padrão	1.48e+04	1.90e+00	3.30e+09	1.42e+02	1.30e+02	1.62e-02	2.59e+01
5e6	1° (Melhor)	5.07e+02	4.59e+01	5.35e+05	7.27e+03	5.15e+00	1.33e+01	-6.72e+03
	13° (Mediana)	5.54e+02	4.66e+01	8.22e+05	7.60e+03	5.93e+00	2.03e+01	-6.62e+03
	25° (Pior)	6.14e+02	4.73e+01	1.30e+06	8.01e+03	6.81e+00	2.09e+01	-6.58e+03
	Média	5.53e+02	4.66e+01	8.37e+05	7.58e+03	5.89e+00	1.89e+01	-6.62e+03
	Desvio Padrão	2.86e+01	4.00e-01	1.52e+05	1.51e+02	3.91e-01	2.49e+00	3.18e+01

Tabela A.8 - jDElsgo (BREST et al., 2010); CEC'2010;  $f(\mathbf{x})$  para G1 a G10.

NAFO		G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
1.2e5	Melhor	2.78e+09	1.06e+04	1.81e+01	8.06e+13	2.98e+08	3.36e+06	2.89e+10	1.04e+09	1.43e+10	1.31e+04
	Mediana	3.72e+09	1.09e+04	1.88e+01	1.43e+14	3.38e+08	4.24e+06	5.40e+10	2.29e+09	1.59e+10	1.43e+04
	Pior	4.89e+09	1.13e+04	1.97e+01	2.30e+14	3.75e+08	4.84e+06	7.23e+10	5.42e+09	2.07e+10	1.51e+04
	Média	3.70e+09	1.09e+04	1.87e+01	1.40e+14	3.39e+08	4.26e+06	5.39e+10	2.39e+09	1.64e+10	1.43e+04
	Desvio Padrão	5.11e+08	1.75e+02	4.46e-01	3.69e+13	1.82e+07	3.81e+05	1.07e+10	9.13e+08	1.73e+09	4.38e+02
6.0e5	Melhor	7.04e+04	3.67e+03	9.70e-01	7.89e+12	1.42e+08	2.20e+01	3.36e+09	3.57e+07	1.45e+09	7.66e+03
	Mediana	8.71e+04	3.93e+03	1.18e+00	1.29e+13	1.87e+08	3.24e+01	6.41e+09	4.65e+07	1.64e+09	8.69e+03
	Pior	1.23e+05	4.20e+03	1.58e+00	2.67e+13	2.30e+08	3.14e+02	1.10e+10	1.39e+08	1.82e+09	9.49e+03
	Média	8.99e+04	3.95e+03	1.22e+00	1.39e+13	1.88e+08	5.07e+01	6.43e+09	6.82e+07	1.66e+09	8.67e+03
	Desvio Padrão	1.39e+04	1.32e+02	1.38e-01	4.60e+12	2.31e+07	5.81e+01	2.12e+09	3.53e+07	8.29e+07	3.99e+02
3.0e6	Melhor	4.78e-20	1.09e-11	1.63e-12	3.09e+10	7.42e+07	7.14e-09	2.69e-05	3.40e-03	2.36e+07	2.10e+03
	Mediana	6.63e-20	4.69e-11	2.35e-12	8.28e+10	9.82e+07	7.22e-09	1.04e-04	1.25e+06	3.04e+07	2.66e+03
	Pior	2.24e-19	1.12e+00	2.24e-11	1.34e+11	1.24e+08	2.10e-07	3.19e-01	8.08e+06	4.22e+07	3.30e+03
	Média	8.86e-20	1.25e-01	3.81e-12	8.06e+10	9.72e+07	1.70e-08	1.31e-02	3.15e+06	3.11e+07	2.64e+03
	Desvio Padrão	4.51e-20	3.45e-01	5.02e-12	3.08e+10	1.44e+07	4.03e-08	6.38e-02	3.27e+06	5.00e+06	3.19e+02

Tabela A.9 - jDElsgo (BREST et al., 2010); CEC'2010;  $f(\mathbf{x})$  para G11 a G20.

NAFO		G11	G12	G13	G14	G15	G16	G17	G18	G19	G20
1.2e5	Melhor	2.02e+02	2.76e+06	2.23e+09	1.95e+10	1.44e+04	4.09e+02	4.28e+06	5.25e+10	2.07e+07	5.80e+10
	Mediana	2.20e+02	3.18e+06	3.70e+09	2.32e+10	1.55e+04	4.17e+02	4.79e+06	6.37e+10	2.88e+07	8.14e+10
	Pior	2.26e+02	3.65e+06	5.44e+09	2.76e+10	1.59e+04	4.24e+02	5.71e+06	8.76e+10	3.56e+07	1.11e+11
	Média	2.19e+02	3.15e+06	3.76e+09	2.32e+10	1.54e+04	4.17e+02	4.85e+06	6.60e+10	2.85e+07	7.99e+10
	Desvio Padrão	5.92e+00	2.19e+05	1.04e+09	2.03e+09	3.33e+02	3.28e+00	3.53e+05	9.47e+09	3.38e+06	1.25e+10
6.0e5	Melhor	7.73e+01	8.61e+05	2.90e+04	3.61e+09	1.10e+04	2.67e+02	1.81e+06	6.95e+05	4.16e+06	5.98e+05
	Mediana	1.14e+02	9.35e+05	4.95e+04	4.11e+09	1.22e+04	2.97e+02	1.97e+06	1.04e+06	5.48e+06	1.05e+06
	Pior	1.49e+02	9.88e+05	9.02e+04	4.72e+09	1.26e+04	3.53e+02	2.10e+06	1.39e+06	1.22e+07	1.66e+06
	Média	1.17e+02	9.39e+05	5.32e+04	4.10e+09	1.20e+04	2.99e+02	1.95e+06	1.03e+06	6.09e+06	1.01e+06
	Desvio Padrão	1.87e+01	2.96e+04	1.70e+04	2.89e+08	5.30e+02	1.91e+01	6.54e+04	2.08e+05	1.65e+06	2.48e+05
3.0e6	Melhor	1.27e+00	8.32e+03	4.79e+02	1.28e+08	5.20e+03	7.30e+01	7.75e+04	1.31e+03	2.39e+05	1.24e+03
	Mediana	1.95e+01	1.18e+04	6.91e+02	1.72e+08	5.78e+03	1.46e+02	1.00e+05	1.88e+03	2.77e+05	1.55e+03
	Pior	5.81e+01	1.71e+04	1.02e+03	2.02e+08	6.84e+03	2.00e+02	1.28e+05	2.57e+03	3.21e+05	1.83e+03
	Média	2.20e+01	1.21e+04	7.11e+02	1.69e+08	5.84e+03	1.44e+02	1.02e+05	1.85e+03	2.74e+05	1.53e+03
	Desvio Padrão	1.53e+01	2.04e+03	1.37e+02	2.08e+07	4.48e+02	3.43e+01	1.26e+04	3.18e+02	2.12e+04	1.32e+02

Tabela A.10 - DECC-DML (OMIDVAR et al., 2010); CEC'2010;  $f(\mathbf{x})$  para G1 a G10.

NAFO		G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
1.2e5	Melhor	2.28e+08	5.51e+03	8.22e+00	3.80e+13	1.43e+08	1.25e+06	2.65e+09	2.23e+09	4.09e+09	1.32e+04
	Mediana	2.85e+08	5.76e+03	9.71e+00	6.40e+13	2.85e+08	1.96e+06	5.50e+09	4.92e+09	4.91e+09	1.39e+04
	Pior	7.02e+08	5.96e+03	1.01e+01	1.20e+14	5.21e+08	2.00e+07	1.17e+10	1.35e+10	5.54e+09	1.45e+04
	Média	4.09e+08	5.75e+03	9.51e+00	6.76e+13	3.00e+08	2.70e+06	5.97e+09	5.57e+09	4.89e+09	1.38e+04
	Desvio Padrão	1.75e+08	1.35e+02	5.55e-01	2.02e+13	9.31e+07	3.62e+06	2.49e+09	2.56e+09	3.77e+08	3.24e+02
6.0e5	Melhor	6.95e+01	2.51e+03	1.06e-02	7.92e+12	1.42e+08	4.59e+01	3.14e+08	4.19e+07	2.82e+08	1.25e+04
	Mediana	4.63e+02	2.64e+03	1.83e-02	1.51e+13	2.85e+08	1.09e+02	5.42e+08	1.15e+08	3.85e+08	1.30e+04
	Pior	1.22e+03	2.78e+03	2.20e-02	3.29e+13	5.20e+08	1.98e+07	9.17e+08	2.38e+08	4.21e+08	1.36e+04
	Média	6.02e+02	2.64e+03	1.81e-02	1.61e+13	2.99e+08	7.94e+05	5.84e+08	1.24e+08	3.73e+08	1.30e+04
	Desvio Padrão	4.11e+02	5.88e+01	3.08e-03	6.19e+12	9.31e+07	3.97e+06	1.68e+08	5.40e+07	3.13e+07	2.93e+02
3.0e6	Melhor	9.05e-27	1.62e+02	1.10e-13	1.38e+12	1.42e+08	3.55e-09	7.09e+07	7.34e+05	4.51e+07	1.21e+04
	Mediana	1.22e-25	2.12e+02	1.14e-13	3.32e+12	2.85e+08	7.11e-09	1.23e+08	1.57e+07	5.97e+07	1.24e+04
	Pior	7.12e-25	2.94e+02	1.35e-13	6.89e+12	5.20e+08	1.98e+07	4.82e+08	1.21e+08	7.09e+07	1.30e+04
	Média	1.93e-25	2.17e+02	1.18e-13	3.58e+12	2.99e+08	7.93e+05	1.39e+08	3.46e+07	5.92e+07	1.25e+04
	Desvio Padrão	1.86e-25	2.98e+01	8.22e-15	1.54e+12	9.31e+07	3.97e+06	7.72e+07	3.56e+07	4.71e+06	2.66e+02

Tabela A.11 - DECC-DML (OMIDVAR et al., 2010); CEC'2010;  $f(\mathbf{x})$  para G11 a G20.

NAFO		G11	G12	G13	G14	G15	G16	G17	G18	G19	G20
1.2e5	Melhor	1.02e+02	4.07e+06	1.09e+08	1.26e+10	1.58e+04	3.22e+02	7.48e+06	1.56e+09	1.77e+07	2.04e+09
	Mediana	1.22e+02	4.68e+06	1.82e+08	1.37e+10	1.65e+04	3.73e+02	8.77e+06	3.30e+09	2.23e+07	3.93e+09
	Pior	1.70e+02	5.35e+06	3.72e+08	1.51e+10	1.73e+04	4.28e+02	1.01e+07	4.06e+09	2.72e+07	5.09e+09
	Média	1.24e+02	4.70e+06	2.11e+08	1.37e+10	1.65e+04	3.75e+02	8.81e+06	3.08e+09	2.20e+07	3.84e+09
	Desvio Padrão	1.38e+01	2.99e+05	9.68e+07	6.86e+08	3.61e+02	3.60e+01	6.86e+05	7.84e+08	2.36e+06	7.72e+08
6.0e5	Melhor	4.00e-01	3.64e+06	8.29e+02	9.82e+08	1.53e+04	3.46e+00	6.50e+06	5.64e+03	1.54e+07	1.43e+03
	Mediana	7.09e-01	4.22e+06	1.71e+03	1.18e+09	1.59e+04	8.65e+00	7.29e+06	1.48e+04	1.84e+07	1.67e+03
	Pior	1.72e+00	4.65e+06	1.47e+04	1.29e+09	1.67e+04	4.28e+02	7.99e+06	3.96e+04	2.46e+07	2.02e+03
	Média	7.66e-01	4.19e+06	3.15e+03	1.17e+09	1.59e+04	4.47e+01	7.27e+06	1.74e+04	1.87e+07	1.69e+03
	Desvio Padrão	2.81e-01	2.18e+05	3.09e+03	8.20e+07	3.63e+02	1.16e+02	3.77e+05	8.26e+03	1.99e+06	1.58e+02
3.0e6	Melhor	1.63e-13	3.46e+06	6.19e+02	1.54e+08	1.48e+04	2.74e-13	5.65e+06	1.64e+03	1.30e+07	9.69e+02
	Mediana	1.78e-13	3.81e+06	1.06e+03	1.89e+08	1.53e+04	3.20e-13	6.55e+06	2.21e+03	1.59e+07	9.75e+02
	Pior	2.03e-13	4.11e+06	2.09e+03	2.22e+08	1.62e+04	1.27e+00	7.63e+06	7.52e+03	2.16e+07	1.10e+03
	Média	1.80e-13	3.80e+06	1.14e+03	1.89e+08	1.54e+04	5.08e-02	6.54e+06	2.47e+03	1.59e+07	9.91e+02
	Desvio Padrão	9.88e-15	1.50e+05	4.31e+02	1.49e+07	3.59e+02	2.54e-01	4.63e+05	1.18e+03	1.72e+06	3.51e+01

Tabela A.12 - DMS-PSO-SHS (ZHAO et al., 2010); CEC'2010;  $f(\mathbf{x})$  para G1 a G10.

NAFO		G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
1.2e5	Melhor	8.36e+08	7.30e+03	1.89e+01	8.96e+12	2.03e+08	5.44e+01	1.69e+09	8.20e+08	4.26e+08	8.33e+03
	Mediana	5.18e+09	8.16e+03	1.89e+01	1.16e+13	2.18e+08	5.68e+01	2.15e+09	9.43e+08	6.72e+08	1.10e+04
	Pior	5.91e+09	9.46e+03	1.93e+01	1.81e+13	2.39e+08	6.97e+01	2.50e+09	1.09e+09	8.36e+08	1.52e+04
	Média	5.48e+09	8.94e+03	1.90e+01	1.25e+13	2.87e+08	5.78e+01	2.25e+09	9.60e+08	6.95e+08	1.21e+04
	Desvio Padrão	1.25e+09	1.00e+03	1.52e+00	2.64e+12	2.16e+07	2.16e+00	3.49e+08	8.17e+07	5.15e+07	4.22e+03
6.0e5	Melhor	1.18e-08	8.45e+02	2.54e-02	1.89e+12	7.15e+07	1.58e+01	1.03e+08	3.74e+07	1.11e+08	5.10e+03
	Mediana	1.39e-05	9.85e+02	2.95e-01	4.01e+12	7.42e+07	1.74e+01	1.17e+08	4.01e+07	1.52e+08	5.51e+03
	Pior	1.64e-04	1.64e+03	5.91e-01	7.22e+12	8.02e+08	1.88e+01	2.27e+08	1.35e+08	2.05e+08	6.29e+03
	Média	1.00e-04	1.28e+03	3.72e-01	4.82e+12	7.66e+07	1.76e+01	1.21e+08	6.05e+07	1.69e+08	5.64e+03
	Desvio Padrão	1.16e-04	3.95e+02	3.57e-02	4.36e+11	1.49e+07	1.81e+00	2.42e+07	4.22e+06	1.15e+07	7.15e+02
3.0e6	Melhor	5.20e-25	1.39e+01	4.69e-14	1.25e+11	2.02e+07	5.83e-12	9.29e+02	2.74e+03	3.58e+06	4.95e+03
	Mediana	2.61e-19	7.16e+01	1.28e-12	2.04e+11	6.10e+07	5.84e-05	1.34e+03	1.03e+07	7.34e+06	5.26e+03
	Pior	1.01e-14	1.05e+02	5.10e-10	6.64e+11	1.12e+08	1.51e+01	3.91e+03	1.15e+08	1.04e+07	6.01e+03
	Média	5.51e-15	8.51e+01	5.52e-11	2.46e+11	8.36e+07	8.28e-02	1.95e+03	1.29e+07	8.72e+06	5.53e+03
	Desvio Padrão	4.00e-14	2.06e+01	3.25e-10	3.31e+10	6.10e+06	9.96e-01	1.56e+02	1.91e+06	6.51e+05	5.18e+02

Tabela A.13 - DMS-PSO-SHS (ZHAO et al., 2010); CEC'2010;  $f(x)$  para G11 a G20.

NAFO		G11	G12	G13	G14	G15	G16	G17	G18	G19	G20
1.2e5	Melhor	1.05e+02	3.55e+06	4.16e+06	6.12e+09	6.04e+03	6.26e+02	3.18e+06	8.98e+06	3.28e+07	9.13e+06
	Mediana	1.16e+02	3.73e+06	7.23e+06	1.01e+10	8.23e+03	7.42e+02	4.20e+06	1.18e+07	3.40e+07	1.24e+07
	Pior	1.47e+02	5.44e+06	8.67e+06	1.42e+10	1.04e+04	7.94e+02	7.39e+06	1.76e+07	4.08e+07	2.05e+07
	Média	1.20e+02	4.09e+06	7.83e+06	1.29e+10	8.74e+03	7.49e+02	5.02e+06	1.53e+07	3.81e+07	1.56e+07
	Desvio Padrão	2.21e+01	4.26e+05	5.16e+05	2.91e+09	2.05e+02	8.16e+01	4.15e+05	2.15e+06	2.15e+06	5.33e+06
6.0e5	Melhor	7.01e+01	1.56e+05	1.10e+03	3.18e+08	9.56e+02	8.87e+01	2.34e+05	9.52e+03	6.73e+06	8.97e+02
	Mediana	8.00e+01	1.87e+05	5.42e+03	5.23e+08	5.11e+03	9.06e+01	4.44e+05	5.75e+04	7.83e+06	2.88e+03
	Pior	1.08e+02	3.82e+05	8.69e+03	7.14e+08	8.53e+03	2.68e+02	5.76e+05	8.06e+04	8.43e+06	4.65e+03
	Média	8.19e+01	2.13e+05	5.81e+03	6.08e+08	5.82e+03	9.94e+01	4.79e+05	6.79e+04	7.91e+06	3.05e+03
	Desvio Padrão	1.06e+01	2.13e+04	7.98e+02	2.45e+07	8.46e+02	6.01e+00	3.25e+04	8.15e+03	8.15e+03	5.15e+05
3.0e6	Melhor	1.76e+01	3.50e+02	5.95e+02	8.97e+06	9.25e+02	3.67e+01	6.58e+02	1.92e+03	1.00e+06	9.26e+01
	Mediana	3.48e+01	6.02e+02	1.01e+03	1.67e+07	4.01e+03	6.49e+01	1.14e+03	2.04e+03	1.10e+06	2.84e+02
	Pior	7.02e+01	7.28e+02	2.56e+03	1.93e+07	5.17e+03	8.31e+01	1.25e+04	4.11e+03	1.51e+06	6.21e+02
	Média	3.25e+01	6.13e+02	1.12e+03	1.76e+07	4.08e+03	6.98e+01	3.83e+03	2.26e+03	1.17e+06	3.52e+02
	Desvio Padrão	3.00e+00	6.00e+01	1.05e+02	1.55e+06	2.17e+02	4.24e+00	4.15e+02	1.16e+02	1.16e+02	1.06e+05

Tabela A.14 - MA-SW-Chain (MOLINA et al., 2010); CEC'2010;  $f(\mathbf{x})$  para G1 a G10.

NAFO		G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
1.2e5	Melhor	2.15e+07	3.32e+03	1.13e+01	1.22e+12	9.35e+07	2.02e+01	4.54e+06	3.30e+07	4.48e+08	3.62e+03
	Mediana	2.76e+07	3.75e+03	1.15e+01	2.04e+12	2.64e+08	2.08e+01	4.91e+06	4.17e+07	5.60e+08	4.15e+03
	Pior	3.51e+07	1.00e+04	1.22e+01	3.35e+12	3.42e+08	1.16e+06	5.71e+06	8.55e+08	6.45e+08	1.00e+04
	Média	2.83e+07	5.09e+03	1.16e+01	2.12e+12	2.52e+08	8.14e+04	4.90e+06	1.21e+08	5.54e+08	5.12e+03
	Desvio Padrão	3.06e+06	2.38e+03	2.68e-01	6.21e+11	6.49e+07	2.84e+05	2.59e+05	2.11e+08	5.20e+07	2.20e+03
6.0e5	Melhor	8.52e+02	2.36e+03	3.44e+00	4.29e+11	3.68e+07	3.61e+00	6.33e+04	3.42e+06	6.93e+07	2.79e+03
	Mediana	1.55e+03	2.68e+03	3.83e+00	5.75e+11	2.59e+08	1.78e+01	7.78e+05	1.90e+07	8.08e+07	3.25e+03
	Pior	7.28e+03	2.97e+03	4.60e+00	7.42e+11	3.24e+08	1.16e+06	4.61e+06	6.11e+08	1.00e+08	3.54e+03
	Média	2.24e+03	2.67e+03	3.84e+00	5.79e+11	2.17e+08	8.14e+04	8.35e+05	6.13e+07	8.18e+07	3.22e+03
	Desvio Padrão	1.71e+03	1.63e+02	2.13e-01	6.46e+10	8.56e+07	2.84e+05	9.08e+05	1.27e+08	8.36e+06	1.85e+02
3.0e6	Melhor	3.18e-15	7.04e+02	3.34e-13	3.04e+11	2.89e+07	8.13e-07	3.35e-03	1.54e+06	1.19e+07	1.81e+03
	Mediana	1.50e-14	7.90e+02	6.11e-13	3.54e+11	2.31e+08	1.60e+00	9.04e+01	3.43e+06	1.40e+07	2.07e+03
	Pior	8.15e-14	9.37e+02	1.58e-12	3.97e+11	2.90e+08	1.16e+06	2.68e+02	1.80e+08	1.62e+07	2.28e+03
	Média	2.10e-14	8.10e+02	7.28e-13	3.53e+11	1.68e+08	8.14e+04	1.03e+02	1.41e+07	1.41e+07	2.07e+03
	Desvio Padrão	1.99e-14	5.88e+01	3.40e-13	3.12e+10	1.04e+08	2.84e+05	8.70e+01	3.68e+07	1.15e+06	1.44e+02



Tabela A.15 - MA-SW-Chain (MOLINA et al., 2010); CEC'2010;  $f(\mathbf{x})$  para G11 a G20.

NAFO		G11	G12	G13	G14	G15	G16	G17	G18	G19	G20
1.2e5	Melhor	5.01e+01	2.20e+05	7.64e+05	8.16e+08	3.94e+03	2.01e+02	5.80e+05	2.22e+04	3.23e+06	2.01e+03
	Mediana	6.41e+01	2.40e+05	9.04e+05	8.81e+08	4.29e+03	2.12e+02	6.78e+05	5.18e+04	3.63e+06	2.22e+03
	Pior	7.13e+01	2.62e+05	1.11e+06	1.04e+09	9.63e+03	2.31e+02	7.50e+05	8.33e+04	4.05e+06	4.69e+03
	Média	6.31e+01	2.40e+05	9.13e+05	8.95e+08	4.83e+03	2.13e+02	6.78e+05	5.14e+04	3.63e+06	2.43e+03
	Desvio Padrão	5.53e+00	1.26e+04	8.09e+04	6.60e+07	1.51e+03	9.19e+00	3.52e+04	1.64e+04	1.94e+05	5.43e+02
6.0e5	Melhor	2.77e+01	1.39e+03	1.08e+03	1.51e+08	2.95e+03	8.51e+01	3.59e+04	1.80e+03	1.29e+06	1.02e+03
	Mediana	3.79e+01	1.64e+03	3.06e+03	1.70e+08	3.19e+03	9.71e+01	4.29e+04	3.89e+03	1.42e+06	1.18e+03
	Pior	5.15e+01	1.91e+03	1.07e+04	1.95e+08	3.45e+03	1.26e+02	5.01e+04	1.61e+04	1.58e+06	1.65e+03
	Média	3.83e+01	1.63e+03	4.34e+03	1.69e+08	3.19e+03	1.02e+02	4.31e+04	5.53e+03	1.41e+06	1.21e+03
	Desvio Padrão	7.23e+00	1.53e+02	3.21e+03	1.17e+07	1.46e+02	1.42e+01	3.42e+03	3.94e+03	7.44e+04	1.42e+02
3.0e6	Melhor	2.74e+01	2.65e-06	3.86e+02	2.79e+07	2.56e+03	8.51e+01	1.04e+00	7.83e+02	2.49e+05	9.25e+02
	Mediana	3.75e+01	3.50e-06	1.07e+03	3.09e+07	2.72e+03	9.44e+01	1.26e+00	1.19e+03	2.85e+05	1.06e+03
	Pior	5.11e+01	4.98e-06	2.92e+03	3.67e+07	2.96e+03	1.24e+02	1.63e+00	2.55e+03	3.32e+05	1.21e+03
	Média	3.80e+01	3.62e-06	1.25e+03	3.11e+07	2.74e+03	9.98e+01	1.24e+00	1.30e+03	2.85e+05	1.07e+03
	Desvio Padrão	7.35e+00	5.92e-07	5.72e+02	1.93e+06	1.22e+02	1.40e+01	1.25e-01	4.36e+02	1.78e+04	7.29e+01

Tabela A.16 - DASA (KOROŠEC et al., 2010); CEC'2010;  $f(\mathbf{x})$  para G1 a G10.

NAFO		G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
1.2e5	Melhor	1.15e+05	4.65e+02	2.10e+00	2.05e+12	5.51e+08	1.98e+07	1.78e+07	1.00e+07	6.52e+08	7.02e+03
	Mediana	1.50e+05	5.24e+02	2.34e+00	6.61e+12	8.06e+08	1.99e+07	4.14e+07	1.05e+08	7.58e+08	7.91e+03
	Pior	1.94e+05	5.95e+02	2.68e+00	1.04e+13	1.05e+09	2.00e+07	1.02e+08	1.17e+09	9.34e+08	8.63e+03
	Média	1.50e+05	5.18e+02	2.35e+00	7.00e+12	8.32e+08	1.99e+07	4.53e+07	2.36e+08	7.67e+08	7.87e+03
	Desvio Padrão	1.76e+04	2.77e+01	1.29e-01	2.21e+12	1.30e+08	5.50e+04	1.91e+07	3.27e+08	7.30e+07	4.55e+02
6.0e5	Melhor	9.30e-08	5.37e+01	2.82e-07	8.01e+11	5.03e+08	1.97e+07	1.23e+05	1.49e+05	1.34e+08	6.81e+03
	Mediana	1.73e-07	6.77e+01	4.41e-07	2.05e+12	7.45e+08	1.98e+07	1.73e+05	3.54e+07	1.62e+08	7.71e+03
	Pior	3.27e-07	9.75e+01	5.77e-07	4.29e+12	9.72e+08	1.99e+07	3.00e+05	3.77e+08	2.17e+08	8.43e+03
	Média	1.86e-07	6.87e+01	4.34e-07	2.04e+12	7.52e+08	1.98e+07	1.81e+05	7.69e+07	1.62e+08	7.66e+03
	Desvio Padrão	6.15e-08	1.05e+01	8.39e-08	8.79e+11	1.22e+08	5.02e+04	4.46e+04	9.40e+07	2.06e+07	4.56e+02
3.0e6	Melhor	5.59e-23	3.98e+00	5.54e-11	2.26e+11	4.41e+08	1.96e+07	2.57e+00	2.84e+03	2.83e+07	6.78e+03
	Mediana	5.42e-22	7.96e+00	7.37e-11	4.94e+11	6.36e+08	1.97e+07	7.18e+00	1.21e+06	3.58e+07	7.33e+03
	Pior	8.44e-21	1.29e+01	8.56e-11	1.10e+12	7.45e+08	1.98e+07	1.49e+01	3.48e+08	5.18e+07	7.75e+03
	Média	1.52e-21	8.48e+00	7.20e-11	5.05e+11	6.20e+08	1.97e+07	7.78e+00	4.98e+07	3.60e+07	7.29e+03
	Desvio Padrão	2.33e-21	2.52e+00	8.27e-12	2.22e+11	7.87e+07	4.45e+04	3.10e+00	8.95e+07	4.78e+06	2.69e+02

Tabela A.17 - DASA (KOROŠEC et al., 2010); CEC'2010;  $f(\mathbf{x})$  para G11 a G20.

NAFO		G11	G12	G13	G14	G15	G16	G17	G18	G19	G20
1.2e5	Melhor	2.02e+02	6.40e+05	1.14e+04	1.64e+09	1.43e+04	3.97e+02	1.55e+06	2.16e+05	5.01e+06	1.93e+05
	Mediana	2.03e+02	6.82e+05	3.04e+04	2.00e+09	1.56e+04	3.97e+02	1.67e+06	2.80e+05	5.78e+06	2.49e+05
	Pior	2.03e+02	7.62e+05	4.87e+04	2.19e+09	1.63e+04	3.98e+02	1.79e+06	4.41e+05	6.58e+06	3.67e+05
	Média	2.03e+02	6.92e+05	3.11e+04	1.97e+09	1.54e+04	3.97e+02	1.67e+06	2.90e+05	5.91e+06	2.57e+05
	Desvio Padrão	3.69e-01	3.93e+04	8.29e+03	1.39e+08	5.30e+02	2.96e-01	6.06e+04	4.05e+05	4.80e+04	4.80e+04
6.0e5	Melhor	1.98e+02	1.06e+05	1.58e+03	4.23e+08	1.40e+04	3.97e+02	3.49e+05	1.93e+04	2.28e+06	1.60e+03
	Mediana	1.99e+02	1.27e+05	1.35e+04	4.89e+08	1.51e+04	3.97e+02	3.77e+05	4.20e+04	2.64e+06	2.11e+03
	Pior	1.99e+02	1.51e+05	2.50e+04	5.78e+08	1.60e+04	3.98e+02	4.11e+05	6.53e+04	2.90e+06	4.02e+03
	Média	1.99e+02	1.28e+05	1.39e+04	4.96e+08	1.51e+04	3.97e+02	3.82e+05	4.30e+04	2.63e+06	2.17e+03
	Desvio Padrão	2.27e-01	1.03e+04	5.79e+03	3.62e+07	5.08e+02	2.57e-01	1.99e+04	1.35e+04	1.27e+05	5.68e+02
3.0e6	Melhor	1.98e+02	1.40e+03	5.16e+02	8.35e+07	1.38e+04	3.96e+02	8.52e+03	2.37e+03	7.18e+05	7.75e+02
	Mediana	1.98e+02	1.75e+03	9.72e+02	1.00e+08	1.45e+04	3.97e+02	1.03e+04	3.95e+03	8.28e+05	1.19e+03
	Pior	1.99e+02	2.22e+03	3.65e+03	1.14e+08	1.51e+04	3.97e+02	1.23e+04	1.09e+04	9.43e+05	1.49e+03
	Média	1.98e+02	1.78e+03	1.21e+03	1.00e+08	1.45e+04	3.97e+02	1.03e+04	4.92e+03	8.34e+05	1.13e+03
	Desvio Padrão	1.57e-01	2.25e+02	7.39e+02	7.88e+06	3.69e+02	2.18e-01	8.99e+02	2.28e+03	5.26e+04	1.79e+02

Tabela A.18 - SDENS (WANG et al., 2010); CEC'2010;  $f(\mathbf{x})$  para G1 a G10.

NAFO		G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
1.2e5	Melhor	3.93e+09	1.16e+04	1.99e+01	3.90e+13	3.14e+08	9.88e+05	3.07e+10	6.05e+08	1.13e+10	1.37e+04
	Mediana	4.74e+09	1.19e+04	2.01e+01	4.60e+13	3.32e+08	2.03e+06	3.57e+10	6.23e+08	1.52e+10	1.38e+04
	Pior	6.19e+09	1.20e+04	2.02e+01	7.90e+13	3.41e+08	2.39e+06	4.70e+10	1.20e+09	1.89e+10	1.42e+04
	Média	5.01e+09	1.19e+04	2.01e+01	5.10e+13	3.29e+08	1.84e+06	3.75e+10	7.71e+08	1.56e+10	1.39e+04
	Desvio Padrão	9.18e+08	9.89e+01	1.17e-01	1.46e+13	1.04e+07	4.77e+05	5.46e+09	2.27e+08	2.77e+09	2.51e+02
6.0e5	Melhor	3.82e+06	7.00e+03	5.13e+00	8.47e+12	1.51e+08	1.38e+01	5.73e+09	4.64e+07	1.78e+09	1.02e+04
	Mediana	4.59e+06	7.12e+03	6.27e+00	1.53e+13	1.83e+08	1.53e+01	7.73e+09	6.40e+07	2.13e+09	1.09e+04
	Pior	1.95e+07	7.17e+03	6.76e+00	2.85e+13	2.12e+08	1.74e+01	1.36e+10	1.09e+08	2.88e+09	1.15e+04
	Média	7.87e+06	7.09e+03	6.12e+00	1.72e+13	1.81e+08	1.53e+01	9.28e+09	7.41e+07	2.23e+09	1.10e+04
	Desvio Padrão	5.94e+06	6.76e+01	6.30e-01	6.68e+12	2.29e+07	1.18e+00	3.44e+09	2.73e+07	3.70e+08	4.59e+02
3.0e6	Melhor	1.75e-06	2.14e+03	1.23e-05	3.26e+12	7.66e+07	1.53e-04	6.36e+07	3.96e+07	4.77e+08	5.78e+03
	Mediana	2.54e-06	2.17e+03	2.35e-05	3.72e+12	1.17e+08	1.76e-04	8.57e+07	4.09e+07	5.75e+08	7.03e+03
	Pior	1.16e-05	2.39e+03	5.50e-05	8.99e+12	1.52e+08	2.57e-04	2.39e+08	9.35e+07	6.38e+08	7.37e+03
	Média	5.73e-06	2.21e+03	2.70e-05	5.11e+12	1.18e+08	2.02e-04	1.20e+08	5.12e+07	5.63e+08	6.87e+03
	Desvio Padrão	4.46e-06	8.95e+01	1.54e-05	2.16e+12	2.88e+07	4.29e-05	6.56e+07	2.12e+07	5.78e+07	5.60e+02

Tabela A.19 - SDENS (WANG et al., 2010); CEC'2010;  $f(\mathbf{x})$  para G11 a G20.

NAFO		G11	G12	G13	G14	G15	G16	G17	G18	G19	G20
1.2e5	Melhor	2.27e+02	2.71e+06	1.70e+10	1.42e+10	1.36e+04	4.15e+02	3.84e+06	2.00e+11	1.19e+07	2.39e+11
	Mediana	2.27e+02	2.83e+06	1.91e+10	1.73e+10	1.45e+04	4.15e+02	4.25e+06	2.09e+11	1.57e+07	2.62e+11
	Pior	2.28e+02	3.29e+06	2.01e+10	2.31e+10	1.45e+04	4.15e+02	4.98e+06	2.35e+11	2.31e+07	2.82e+11
	Média	2.27e+02	2.95e+06	1.88e+10	1.84e+10	1.43e+04	4.15e+02	4.31e+06	2.11e+11	1.67e+07	2.61e+11
	Desvio Padrão	3.49e-01	2.37e+05	1.07e+09	3.56e+09	3.72e+02	1.08e-01	4.04e+05	1.27e+10	1.27e+10	3.71e+06
6.0e5	Melhor	2.25e+02	1.25e+06	4.37e+05	3.91e+09	7.32e+03	4.13e+02	1.96e+06	1.65e+08	4.92e+06	1.36e+08
	Mediana	2.26e+02	1.30e+06	6.67e+05	5.02e+09	1.18e+04	4.13e+02	2.02e+06	1.86e+08	5.39e+06	2.78e+08
	Pior	2.26e+02	1.42e+06	7.64e+05	6.93e+09	1.26e+04	4.14e+02	2.29e+06	3.00e+08	6.18e+06	3.52e+08
	Média	2.26e+02	1.32e+06	6.43e+05	5.14e+09	1.03e+04	4.13e+02	2.07e+06	2.02e+08	5.41e+06	2.69e+08
	Desvio Padrão	3.83e-01	5.98e+04	1.10e+05	9.89e+08	2.29e+03	3.49e-01	1.17e+05	5.02e+07	5.02e+07	4.31e+05
3.0e6	Melhor	2.20e+02	3.80e+05	1.16e+03	1.61e+09	7.14e+03	4.03e+02	8.78e+05	1.16e+04	7.57e+05	9.81e+02
	Mediana	2.21e+02	3.95e+05	1.80e+03	1.86e+09	7.32e+03	4.09e+02	1.14e+06	3.32e+04	8.02e+05	9.83e+02
	Pior	2.22e+02	4.97e+05	4.13e+03	2.30e+09	7.44e+03	4.10e+02	1.18e+06	4.51e+04	1.19e+06	1.02e+03
	Média	2.21e+02	4.13e+05	2.19e+03	1.88e+09	7.32e+03	4.08e+02	1.08e+06	3.08e+04	8.80e+05	9.90e+02
	Desvio Padrão	5.09e-01	4.28e+04	1.03e+03	2.33e+08	9.63e+01	2.53e+00	1.11e+05	1.22e+04	1.22e+04	1.59e+05

Tabela A.20 - EOEa (WANG; LI, 2010); CEC'2010;  $f(\mathbf{x})$  para G1 a G10.

NAFO		G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
1.2e5	Melhor	1.02e+08	3.70e+03	3.74e+00	7.36e+12	1.79e+07	3.46e+06	2.13e+09	4.76e+09	5.75e+08	6.02e+03
	Mediana	1.44e+08	4.11e+03	4.24e+00	1.18e+13	2.79e+07	4.11e+06	3.23e+09	1.21e+10	6.62e+08	6.35e+03
	Pior	2.08e+08	4.30e+03	4.58e+00	2.43e+13	7.79e+07	5.06e+06	4.04e+09	1.95e+10	7.94e+08	7.11e+03
	Média	1.52e+08	4.06e+03	4.22e+00	1.41e+13	3.63e+07	4.27e+06	3.02e+09	1.23e+10	6.76e+08	6.46e+03
	Desvio Padrão	3.11e+07	2.25e+02	2.20e-01	5.00e+12	1.77e+07	4.50e+05	6.63e+08	5.18e+09	6.94e+07	3.43e+02
6.0e5	Melhor	7.56e+01	5.05e+01	3.93e-03	4.06e+12	1.29e+07	2.99e+06	4.48e+06	3.24e+05	1.65e+08	1.02e+03
	Mediana	1.09e+02	8.51e+01	5.93e-03	7.78e+12	2.39e+07	3.97e+06	2.79e+07	1.34e+07	2.37e+08	1.12e+03
	Pior	9.49e+02	1.76e+02	1.45e-02	1.25e+13	3.08e+07	4.44e+06	1.54e+08	4.53e+07	2.54e+08	1.25e+03
	Média	2.28e+02	9.10e+01	7.74e-03	7.75e+12	2.24e+07	3.85e+06	5.41e+07	1.45e+07	2.24e+08	1.13e+03
	Desvio Padrão	2.77e+02	3.67e+01	3.74e-03	2.60e+12	5.91e+06	4.97e+05	4.97e+07	1.39e+07	3.01e+07	6.54e+01
3.0e6	Melhor	0.00e+00	3.55e-15	1.53e-13	1.25e+12	1.29e+07	2.99e+06	3.99e+00	9.50e+01	3.75e+07	9.50e+02
	Mediana	1.35e-24	1.78e-14	1.71e-13	2.57e+12	2.39e+07	3.97e+06	5.49e+01	5.51e+06	4.74e+07	1.07e+03
	Pior	7.95e-23	1.99e+00	1.81e-13	5.96e+12	3.08e+07	4.44e+06	5.27e+02	4.07e+07	5.31e+07	1.18e+03
	Média	2.20e-23	3.62e-01	1.67e-13	3.09e+12	2.24e+07	3.85e+06	1.24e+02	1.01e+07	4.63e+07	1.08e+03
	Desvio Padrão	2.87e-23	6.71e-01	1.13e-14	1.61e+12	5.91e+06	4.97e+05	1.55e+02	1.28e+07	4.78e+06	6.91e+01

Tabela A.21 - EOEa (WANG; LI, 2010); CEC'2010;  $f(\mathbf{x})$  para G11 a G20.

NAFO		G11	G12	G13	G14	G15	G16	G17	G18	G19	G20
1.2e5	Melhor	6.01e+01	7.77e+05	1.63e+06	1.90e+09	6.70e+03	1.08e+02	1.59e+06	3.51e+07	9.35e+06	6.32e+07
	Mediana	6.29e+01	8.08e+05	2.04e+06	2.14e+09	7.35e+03	1.22e+02	1.73e+06	6.73e+07	1.13e+07	9.81e+07
	Pior	1.12e+02	9.34e+05	3.36e+06	2.23e+09	7.77e+03	1.53e+02	1.79e+06	9.02e+07	1.25e+07	1.17e+08
	Média	7.26e+01	8.35e+05	2.26e+06	2.13e+09	7.31e+03	1.27e+02	1.71e+06	6.42e+07	1.12e+07	9.12e+07
	Desvio Padrão	1.71e+01	5.21e+04	5.77e+05	9.67e+07	2.82e+02	1.55e+01	6.35e+04	1.91e+07	1.05e+06	1.77e+07
6.0e5	Melhor	2.92e+01	1.89e+05	1.64e+03	6.37e+08	1.97e+03	6.58e+01	5.57e+05	1.80e+04	5.04e+06	3.85e+03
	Mediana	3.19e+01	2.55e+05	2.58e+03	7.53e+08	2.13e+03	7.85e+01	7.14e+05	5.00e+04	5.50e+06	4.48e+03
	Pior	8.14e+01	2.99e+05	5.31e+03	9.64e+08	2.33e+03	1.19e+02	8.35e+05	6.55e+04	6.42e+06	5.16e+03
	Média	4.03e+01	2.48e+05	2.83e+03	7.79e+08	2.14e+03	8.33e+01	7.08e+05	4.80e+04	5.56e+06	4.42e+03
	Desvio Padrão	1.68e+01	3.24e+04	1.16e+03	8.32e+07	1.22e+02	1.71e+01	7.69e+04	1.35e+04	3.98e+05	4.09e+02
3.0e6	Melhor	2.68e+01	9.57e+03	7.98e+02	1.55e+08	1.97e+03	6.57e+01	6.29e+04	1.99e+03	1.69e+06	1.65e+03
	Mediana	3.03e+01	1.35e+04	1.12e+03	1.63e+08	2.13e+03	7.85e+01	8.12e+04	2.89e+03	1.82e+06	1.96e+03
	Pior	7.87e+01	1.99e+04	2.27e+03	1.85e+08	2.33e+03	1.17e+02	9.61e+04	4.10e+03	2.02e+06	2.34e+03
	Média	3.86e+01	1.37e+04	1.24e+03	1.65e+08	2.14e+03	8.26e+01	7.93e+04	2.94e+03	1.84e+06	1.97e+03
	Desvio Padrão	1.65e+01	2.90e+03	4.59e+02	8.95e+06	1.22e+02	1.68e+01	8.80e+03	6.92e+02	9.97e+04	2.35e+02





## **PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE**

### **Teses e Dissertações (TDI)**

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

### **Manuais Técnicos (MAN)**

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

### **Notas Técnico-Científicas (NTC)**

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programas de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

### **Relatórios de Pesquisa (RPQ)**

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

### **Propostas e Relatórios de Projetos (PRP)**

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

### **Publicações Didáticas (PUD)**

Incluem apostilas, notas de aula e manuais didáticos.

### **Publicações Seriadas**

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Constam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

### **Programas de Computador (PDC)**

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. Aceitam-se tanto programas fonte quanto os executáveis.

### **Pré-publicações (PRE)**

Todos os artigos publicados em periódicos, anais e como capítulos de livros.