



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

sid.inpe.br/mtc-m21b/2016/04.25.17.23 -TDI

## MÉTODOS CONVERGENTES DE OTIMIZAÇÃO GLOBAL BASEADOS NO VETOR Q-GRADIENTE

Érica Josiane Coelho Gouvêa

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Fernando Manuel Ramos, e Aline Cristina Soterroni, aprovada em 18 de maio de 2016.

URL do documento original:

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

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 E INOVAÇÃO  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**

sid.inpe.br/mtc-m21b/2016/04.25.17.23 -TDI

## MÉTODOS CONVERGENTES DE OTIMIZAÇÃO GLOBAL BASEADOS NO VETOR Q-GRADIENTE

Érica Josiane Coelho Gouvêa

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Fernando Manuel Ramos, e Aline Cristina Soterroni, aprovada em 18 de maio de 2016.

URL do documento original:

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

INPE  
São José dos Campos  
2016

Dados Internacionais de Catalogação na Publicação (CIP)

---

Gouvêa, Érica Josiane Coelho.

G475m Métodos convergentes de otimização global baseados no vetor q-Gradiente / Érica Josiane Coelho Gouvêa. – São José dos Campos : INPE, 2016.

xx + 91 p. ; (sid.inpe.br/mtc-m21b/2016/04.25.17.23 -TDI)

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

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

1. Otimização global. 2. q-cálculo. 3. Vetor q-gradiente. 4. Convergência. I. Título.

CDU 004.4.416:519.142

---



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): **Érica Josiane Coelho Gouvêa**

Título: " **MÉTODOS CONVERGENTES DE OTIMIZAÇÃO GLOBAL BASEADOS NO VETOR q-gradiente**".

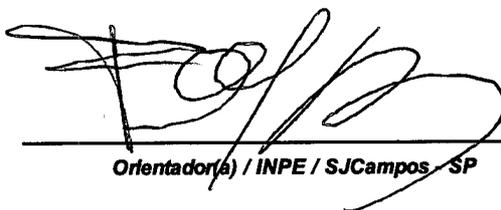
Aprovado (a) pela Banca Examinadora em cumprimento ao requisito exigido para obtenção do Título de **Doutor(a)** em **Computação Aplicada**

Dr. Solon Venâncio de Carvalho



Presidente / INPE / SJC Campos - SP

Dr. Fernando Manuel Ramos



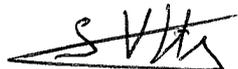
Orientador(a) / INPE / SJC Campos - SP

Dra. Aline Cristina Soterroni



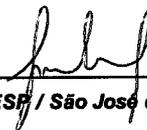
Orientador(a) / INPE / São José dos Campos - SP

Dr. Stephan Stephany



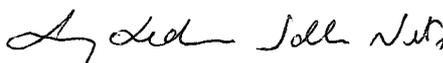
Membro da Banca / INPE / SJC Campos - SP

Dr. Erwin Doescher



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

Dr. Luiz Leduino de Salles Neto



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

Este trabalho foi aprovado por:

( ) maioria simples

unanimidade

São José dos Campos, 18 de Maio de 2016



*“If I have seen a little further it is by standing on the shoulders of  
Giants”.*

ISAAC NEWTON, 1676



*A minha família*



## AGRADECIMENTOS

Agradeço a Deus por estar sempre comigo, sendo o alicerce de minha vida.

Agradeço ao Dr. Fernando Manuel Ramos por toda serenidade, otimismo e competência na orientação desse trabalho. Muito obrigada por sua orientação, não só no trabalho científico, mas para a vida. Agradeço também a minha amiga desde a época do mestrado e que agora tive a honra de ser sua orientada, Dra. Aline Cristina Soterroni. Muito obrigada por toda paciência e dedicação em sua orientação.

Agradeço, de forma especial, ao Professor Dr. Regis G. Rommel pela oportunidade de estudo fora do país e principalmente por sua impecável orientação.

Agradeço ao Instituto Nacional de Pesquisas Espaciais (INPE) e a Saint Joseph's University (SJU) pela oportunidade de estudos e utilização de suas instalações.

Agradeço à CAPES por todo o suporte financeiro (nacional e internacional).

Agradeço aos professores do INPE pelo conhecimento compartilhado e à banca examinadora pelo aceite, pelos comentários e sugestões que em muito acrescentaram para o aprimoramento do trabalho.

Agradeço aos amigos do INPE por todo os momentos de descontração, pela força nas horas difíceis e pela amizade de todos vocês. E aos amigos com os quais dividi a mesma sala: Eduardo, Felipe, Heloisa e Marluce. Muito obrigada por cada momento em que passamos juntos, por toda a ajuda e por se mostrarem sempre presente em minha vida.

Agradeço aos amigos e colegas de profissão da Universidade de Taubaté e Fatec-Taubaté por me entenderem e me apoiarem em todos os momentos.

Agradeço aos meus queridos alunos, que me encorajaram e me fizeram sentir capaz de concluir o doutorado lecionando diversas disciplinas todos os dias da semana.

E por fim, agradeço a minha família, principalmente a meus pais Antonio e Vicentina por todo amor incondicional e por me mostrar que a educação é o caminho mais sublime dessa vida. Agradeço a minha irmã Carina, a meu cunhado João Evangelista e a meus sobrinhos Marina e João Guilherme, que sempre me deram forças e entenderam a minha ausência. E, de forma especial, agradeço a meu marido Felipe por toda sua dedicação e compreensão, dando força e amparo em todos os momentos. Muito obrigada por todo seu amor e por me fazer feliz.



## RESUMO

O vetor  $q$ -gradiente é um  $q$ -análogo do vetor gradiente clássico baseado na derivada de Jackson, com a propriedade de reduzir ao gradiente clássico quando o parâmetro  $q$  tende a 1. O primeiro método baseado nesses conceitos é o método  $q$ -G, uma generalização do método da máxima descida para problemas de otimização global contínuos, e que retorna a sua versão clássica quando  $q \rightarrow 1$ . A proposta do método  $q$ -G é definir a sua direção de busca a partir do vetor  $q$ -gradiente da função objetivo. Essa direção juntamente com estratégias apropriadas para a obtenção do parâmetro  $q$ , necessário para calcular o vetor  $q$ -gradiente, e o tamanho do passo fornecem ao método  $q$ -G mecanismos para escapar de mínimos locais por meio de uma transição suave entre busca global e busca local ao longo do procedimento iterativo. Este trabalho apresenta uma extensão desse estudo, com o desenvolvimento de novas  $q$ -versões, onde no limite,  $q \rightarrow 1$ , retomem suas versões clássicas. Foram desenvolvidas uma  $q$ -versão do método dos gradientes conjugados de Fletcher e Reeves, denominado método  $q$ -GC e duas  $q$ -versões dos métodos quase-Newton, método  $q$ -BFGS e método  $q$ -DFP, generalizações dos métodos de Broyden-Fletcher-Goldfarb-Shanno e Davidon-Fletcher-Powell, respectivamente. Assim como o método  $q$ -G, esses métodos são implementados de tal forma que o processo de busca muda gradualmente de busca global no início do procedimento iterativo, para busca local no final do procedimento iterativo. Além disso, perturbações gaussianas são usadas em algumas iterações para garantir a convergência desses métodos para o extremo global em um sentido probabilístico. As  $q$ -versões com prova de convergência foram comparadas com as suas versões clássicas e com outros métodos, incluindo uma estratégia evolutiva com matriz de covariância adaptada (CMA-ES), uma variação da busca aleatória controlada (CRS2-LM), um método de ponto interior que usa derivadas por diferenças finitas (ILOPT), um método de busca direta de Nelder-Mead e outra estratégia evolutiva (ISRES), totalizando 13 métodos diferentes. As comparações foram realizadas para 27 funções testes de 10 dimensões bem conhecidas na literatura. No geral, os resultados mostraram que os métodos baseados no vetor  $q$ -gradiente são competitivos e promissores, especialmente quando aplicados aos problemas de otimização multimodal. Além disso, os métodos também foram aplicados em dois problemas complexos de otimização e os resultados mostraram a viabilidade de seu uso em problemas de difícil solução.

Palavras-chave: otimização global.  $q$ -cálculo. vetor  $q$ -gradiente. convergência.



## ABSTRACT

The  $q$ -gradient vector is a  $q$ -analogue of the classical gradient vector based on the Jackson's derivative with the property of reducing the classical gradient when the parameter  $q$  tends to 1. The first method based on these concepts is the  $q$ -G method, a generalization of the steepest descent method to continuous global optimization problems, and it returns to its classical version when  $q \rightarrow 1$ . The proposal of the  $q$ -G method is to define the search direction from the  $q$ -gradient vector of the objective function. This direction together with appropriate strategies for obtaining the parameter  $q$  necessary for calculating the  $q$ -gradient vector, and the step length provide the  $q$ -G method mechanisms to escape local minima by a smooth transition between global search and local search during the iterative procedure. This work presents an extension of this study, with the development of the new  $q$ -versions where the limit  $q \rightarrow 1$ , returns its classical versions. We developed a  $q$ -version of the Fletcher-Reeves conjugate gradient method, called  $q$ -CG method and two  $q$ -versions of the quasi-Newton methods, called  $q$ -BFGS and  $q$ -DFP methods, generalizations of the methods of Broyden-Fletcher-Goldfarb-Shanno and Davidon-Fletcher-Powell, respectively. As the  $q$ -G method, the methods are implemented such that the search process gradually shifts from global search at the beginning of the iterative procedure to the local search at the end of the iterative procedure. Moreover, gaussian perturbations are used in some iteration to guarantee the convergence of the methods to the global minimum in a probabilistic sense. We compare the convergent  $q$ -versions with their classical versions and with other methods, including CMA-ES, a variant of Controlled Random Search, Controlled Random Search with Local Mutation (CRS2-LM), an interior point algorithm (IPOPT), another evolution strategy (ISRES), and the Nelder-Mead direct search method, amounting 13 different methods. The comparisons were performed to 27 well-known test problems in the literature. In general, the methods based on the  $q$ -gradient vector are competitive and promising, especially when applied to multimodal optimization problems. Moreover, the methods were applied to two complex optimization problems and the results showed the feasibility of their use in to solve hard problems.

Keywords: global optimization.  $q$ -calculus.  $q$ -gradient vector. convergence.



## LISTA DE FIGURAS

	<u>Pág.</u>
2.1 Exemplos de mínimo local e mínimo global . . . . .	6
2.2 Ilustração de conjuntos convexos . . . . .	7
2.3 Interpretação geométrica das condições de Kuhn-Tucker. . . . .	10
2.4 Método simplex de Nelder-Mead depois de uma etapa de reflexão e expansão	27
2.5 Método simplex de Nelder-Mead após uma contração externa, uma contração interna e uma redução . . . . .	27
3.1 Fluxograma do algoritmo do Método $q$ -G com perturbações gaussianas. . . .	40
3.2 Pontos gerados pelo método $q$ -G. Sem perturbação gaussiana <i>versus</i> com per- turbação gaussiana. . . . .	41
3.3 Fluxograma do algoritmo do Método $q$ -GC com perturbações gaussianas. . .	44
3.4 Fluxograma do algoritmo dos Métodos $q$ -BFGS e $q$ -DFP com perturbações gaussianas. . . . .	47
5.1 Perfis de dados dos algoritmos para 27 funções teste usando 30 diferentes pontos iniciais. . . . .	65
5.2 Perfis de dados dos algoritmos para 14 problemas multimodais e 13 proble- mas unimodais . . . . .	66
5.3 Perfis de dados das $q$ -versões e suas versões clássicas com vários valores de $\sigma^{(0)}$ para 16 funções teste usando 30 diferentes pontos iniciais . . . . .	68
5.4 Perfis de dados das $q$ -versões e suas versões clássicas com vários valores de $\sigma^{(0)}$ para 8 funções teste multimodais usando 30 diferentes pontos iniciais . .	70
5.5 Perfis de dados das $q$ -versões e suas versões clássicas com vários valores de $\sigma^{(0)}$ para 8 funções teste unimodais usando 30 diferentes pontos iniciais . . .	71
5.6 Perfis de dados dos algoritmos propostos para 16 funções teste usando 30 diferentes pontos iniciais. . . . .	72
5.7 Perfis de dados dos algoritmos propostos para 8 problemas multimodais e 8 problemas unimodais . . . . .	73



## LISTA DE TABELAS

	<u>Pág.</u>
5.1 Características das funções teste. . . . .	54
5.2 Estatística sobre o melhor valor da função objetivo sobre 30 execuções dos algoritmos para as funções teste multimodais . . . . .	57
5.3 Estatística sobre o melhor valor da função objetivo sobre 30 execuções dos algoritmos para as funções teste unimodais . . . . .	60
5.4 Tempo médio (em segundos) de 30 execuções dos algoritmos para as funções teste após 10.000 avaliações da função . . . . .	74
6.1 Resultados para o problema de minimização quadrática não-convexa . . . . .	80
6.2 Resultados para o problema de aniquilação da onda . . . . .	81



## SUMÁRIO

	<u>Pág.</u>
<b>1 INTRODUÇÃO</b> . . . . .	<b>1</b>
<b>2 CONCEITOS FUNDAMENTAIS DE OTIMIZAÇÃO</b> . . . . .	<b>5</b>
2.1 Problemas de Otimização . . . . .	5
2.2 Condições de Otimalidade . . . . .	8
2.3 Taxa de Convergência . . . . .	10
2.4 Critérios de Parada . . . . .	11
2.5 Métodos de Otimização Baseados em Gradientes . . . . .	12
2.5.1 Método da Máxima Descida . . . . .	14
2.5.2 Método de Newton . . . . .	15
2.5.3 Método dos Gradientes Conjugados . . . . .	16
2.5.4 Métodos quase-Newton . . . . .	19
2.6 Métodos de Otimização Baseados em Meta-Heurísticas . . . . .	21
2.6.1 Algoritmos Evolutivos . . . . .	21
2.6.2 Recozimento Simulado . . . . .	23
2.7 Métodos de Otimização Livres de Derivadas . . . . .	25
2.7.1 Método Simplex de Nelder-Mead . . . . .	25
<b>3 ALGORITMOS DE OTIMIZAÇÃO GLOBAL BASEADOS NO VETOR</b>	
<b><math>q</math>-GRADIENTE</b> . . . . .	<b>31</b>
3.1 Fundamentos de $q$ -cálculo . . . . .	31
3.1.1 $q$ -Cálculo . . . . .	31
3.1.2 $q$ -Derivada . . . . .	32
3.1.3 $q$ -Gradiente . . . . .	32
3.1.4 Método $q$ -G . . . . .	35
3.2 Método $q$ -G com Perturbações Gaussianas . . . . .	38
3.3 Método $q$ -GC com Perturbações Gaussianas . . . . .	41
3.4 Métodos $q$ -BFGS e $q$ -DFP com Perturbações Gaussianas . . . . .	45
<b>4 ANÁLISE DE CONVERGÊNCIA</b> . . . . .	<b>49</b>
4.1 Notações Preliminares . . . . .	49
4.2 Prova de Convergência . . . . .	52

<b>5</b>	<b>ANÁLISE DOS RESULTADOS</b>	<b>53</b>
5.1	Descrição dos Experimentos Computacionais	53
5.2	Estatísticas sobre a Comparação dos Algoritmos	56
5.3	Comparação dos Métodos Usando Perfis de Dados	63
5.4	Análise de Sensibilidade	67
5.5	Tempo Médio de Execução	70
<b>6</b>	<b>APLICAÇÃO A DOIS PROBLEMAS COMPLEXOS DE OTIMIZAÇÃO</b>	<b>77</b>
6.1	Minimização de Função Quadrática Não-Convexa	77
6.2	Problema de Aniquilação de Onda	79
6.3	Resultados Numéricos	80
<b>7</b>	<b>CONCLUSÕES</b>	<b>83</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>85</b>

## 1 INTRODUÇÃO

A essência da otimização está em melhorar algo em um conjunto de alternativas disponíveis. Trata-se de uma ferramenta de grande aplicabilidade que se constitui numa vasta e atraente área do conhecimento, sendo de grande importância para diversas áreas das ciências e engenharias. Um engenheiro, por exemplo, deseja projetar a melhor configuração de um equipamento com o menor custo possível. Indústrias procuram aumentar os lucros e em contrapartida diminuir os custos de produção (VANDERPLAATS, 1984). Até mesmo ao planejar férias, as pessoas tendem a maximizar o prazer, minimizando os custos. Com isso, o estudo da otimização se torna atraente tanto na área científica, quanto em aplicações práticas.

A otimização está diretamente ligada a matemática e a computação. Antes de otimizar um problema é necessário elaborar um modelo, expresso por meio de uma função objetivo sujeita a variáveis de decisão, sendo que este pode ser considerado o passo principal da otimização. O modelo não pode ser muito simples nem muito complexo, visto que, se muito simples poderá não representar o problema real, e se muito complexo dificultará a busca pela solução. Em muitos casos, a solução ideal não é encontrada de forma analítica. Assim, é necessário usar um algoritmo numérico, ou seja, procedimentos iterativos de cálculos que a cada passo procuram melhorar a solução atual, até que o extremo (mínimo ou máximo da função objetivo) seja encontrado, ou que algum critério de parada seja satisfeito. A construção de bons algoritmos, que resolvam problemas de otimização independentemente da dimensão e dos parâmetros envolvidos, também representa um passo importante nesse processo (GOLDBARG; LUNA, 2005).

Idealmente, um algoritmo de otimização não deve ser difícil de se utilizar, como por exemplo, um engenheiro estrutural com um conhecimento especializado em princípios mecânicos não tem que ser um especialista na área de otimização apenas para melhorar seus projetos. Além disso, o algoritmo deve ser bom o suficiente para convergir, de forma confiável, ao extremo da função objetivo com tempo de processamento viável. Assim, um método de otimização verdadeiramente útil, deve ser simples de implementar, fácil de usar, confiável e rápido.

Os métodos de otimização dependem, em grande parte, das características do problema em que se deseja otimizar. Os problemas reais modelados como problemas de otimização geram funções objetivos que variam consideravelmente. Identificar a característica dessas funções é uma fase importante no processo de otimização, uma vez que a escolha do algoritmo deve ser feita de acordo com as características do problema a ser otimizado (VANDERPLAATS, 1984).

Dentre as classes de problemas, destacam-se os problemas de otimização global. O adjetivo “global” indica que o problema de otimização pode ser de natureza muito geral, isto é, a função objetivo pode ser uma função não-linear, não-convexa, não-diferenciável e possivelmente descontínua ao longo de um domínio contínuo ou discreto. Além disso, problemas de otimização global podem conter diversos mínimos e/ou máximos locais e os métodos numéricos existentes não fornecem garantias de que o extremo encontrado seja de fato o extremo global, a não ser que exista uma estrutura clara indicando que a solução encontrada seja o extremo global. Apesar dos problemas de otimização global serem difíceis de se resolver, suas aplicações são predominantes em engenharia e sistemas do mundo real, dessa forma, o desenvolvimento de métodos eficazes capazes de resolver tais problemas com robustez e eficiência é importante (ZABINSKY, 2003).

Por um lado, muitos métodos de busca local, incluindo algoritmos de descida baseados em gradientes (como exemplo, método da máxima descida, método quase-Newton e método dos gradientes conjugados) e alguns métodos de busca direta, são geralmente rápidos e precisos. No entanto, a presença de múltiplos extremos locais torna a otimização global difícil para estes otimizadores locais, a menos que a busca seja iniciada suficientemente próximo do extremo global da função objetivo ou utilize vários pontos iniciais no procedimento iterativo. Por outro lado, meta-heurísticas, tais como algoritmos evolutivos e recozimento simulado, são mais adequadas para escapar dos múltiplos extremos locais, porém, em algumas vezes, podem exigir muitas avaliações da função objetivo para convergir ao extremo global.

O conflito entre robustez e eficiência computacional, busca local versus busca global, conduziu ao desenvolvimento de métodos híbridos, numa tentativa de combinar o melhor dos dois mundos. Por exemplo, Hedar and Fukushima (2006) e Chelouah and Siarry (2005) combinam a busca tabu com o método de busca direta de Nelder-Mead. Busca dispersa (HERRERA et al., 2006; MARTÍ et al., 2006) e busca em vizinhança variável (HANSEN; MLADENOVIC, 2001) são métodos conhecidos que efetivamente combinam pesquisa de busca global e local. Para uma revisão de algumas meta-herísticas híbridas veja Blum et al. (2008).

Neste sentido, Soterroni et al. (2011) desenvolveram um novo método que estende a estratégia de busca utilizado por algoritmos baseados em gradientes e o aplicou aos problemas de otimização global contínuos. Este método, denominado método do  $q$ -gradiente, ou simplesmente método  $q$ -G, é uma generalização do método da máxima descida que utiliza o conceito de  $q$ -derivada, proveniente da teoria do  $q$ -cálculo para balancear a busca local e global, sendo  $q$  um parâmetro usado para controlar a direção de busca. A proposta do

método  $q$ -G é utilizar um  $q$ -análogo ao gradiente da função objetivo que é então utilizado para determinar uma direção de busca apropriada.

A história do  $q$ -cálculo remonta ao início do século XVIII e XIX com os trabalhos pioneiros de Euler, Gauss e Heine. Mas, foi no início do século XX que Frank Hilton Jackson desenvolveu o  $q$ -cálculo de uma forma sistemática (ERNST, 2003). Seu trabalho deu origem a versões análogas, também chamadas de  $q$ -versões, de funções, séries e números especiais que, no limite  $q \rightarrow 1$ , retomam suas respectivas versões clássicas (ERNST, 2003; CHAUNDY, 1962). Mas, mais importante, ele reintroduziu o conceito de  $q$ -derivada (também conhecido como derivada de Jackson) (JACKSON, 1909), no qual Soterroni et al. (2011) usaram para desenvolver o vetor  $q$ -gradiente e aplicá-lo aos problemas de otimização global. O vetor  $q$ -gradiente é uma extensão do vetor gradiente clássico por meio de um parâmetro  $q$  e com a propriedade de retornar ao gradiente clássico quando no limite  $q$  tende a 1.

O método  $q$ -G foi extensivamente comparado com os Algoritmos Evolutivos (AEs) que são considerados eficientes na resolução de problemas de otimização global. Os resultados comprovaram a capacidade do método  $q$ -G de escapar de extremos locais e atingir a bacia de atração do extremo global em muitas das funções teste analisadas, sem a utilização de reinicializações. Vale resaltar que no limite, quando  $q \rightarrow 1$ , o método  $q$ -G tende ao método clássico da máxima descida. O método da máxima descida, por sua vez, é conhecido pela convergência lenta em funções mal escalonadas e que pode ser interpretada geometricamente pelo movimento em ziguezague na direção do extremo local (que porventura também pode ser global). Mesmo assim, o método  $q$ -G exibiu bons resultados sobretudo em funções multimodais que são caracterizadas pela existência de inúmeros extremos locais (SOTERRONI et al., 2011; SOTERRONI et al., 2012; SOTERRONI, 2012; SOTERRONI et al., 2013; SOTERRONI et al., 2015).

Dentro deste contexto, o objetivo desse trabalho é estender o estudo já realizado anteriormente por meio do desenvolvimento de novas  $q$ -versões, onde no limite,  $q \rightarrow 1$ , retomem suas versões clássicas. Foram desenvolvidas uma  $q$ -versão do método dos gradientes conjugados de Fletcher e Reeves, denominado método  $q$ -GC e duas  $q$ -versões dos métodos quase-Newton, método  $q$ -BFGS e método  $q$ -DFP, generalizações dos métodos de Broyden-Fletcher-Goldfarb-Shanno e Davidon-Fletcher-Powell, respectivamente. Em ambas generalizações, o vetor  $q$ -gradiente da função objetivo é utilizado em vez do gradiente clássico, quando a direção de busca é calculada. A vantagem de usar o vetor  $q$ -gradiente é que ele permite que a direção de busca seja realizada de uma forma mais diversificada, fazendo o possível para escapar dos extremos locais, já que as  $q$ -versões são

implementadas de tal forma que o processo de busca muda gradualmente para busca global no início do procedimento iterativo, e busca local no final do procedimento iterativo.

Além disso, esse trabalho apresenta prova de convergência para os métodos baseados no vetor  $q$ -gradiente. Dessa forma, perturbações gaussianas são utilizadas em algumas iterações nos métodos das  $q$ -versões, inclusive para o método  $q$ -G, para assegurar a convergência dos métodos para o extremo global em um sentido probabilístico. O desvio-padrão dessas perturbações gaussianas são reduzidos geometricamente, sempre que não houver progressos no sentido de facilitar a busca.

Para avaliar o desempenho dos métodos convergentes baseados no vetor  $q$ -gradiente, os mesmos foram comparados com suas versões clássicas e também com CMA-ES, uma variação da busca aleatória controlada, um método de ponto interior, o algoritmo simplex de Nelder-Mead e outro algoritmo evolutivo, para 27 funções teste de dez dimensões, bem conhecidas na literatura da otimização. Em geral, os métodos convergentes baseados no vetor  $q$ -gradiente são bastante promissores e competitivos, especialmente quando aplicados a problemas de otimização multimodais. Os métodos também foram aplicados em dois problemas complexos de otimização e os resultados mostraram a viabilidade de seu uso em problemas de difícil solução.

Este trabalho está organizado da seguinte forma. No Capítulo 2 são apresentados conceitos mais relevantes sobre os problemas de otimização e sobre os métodos de otimização local baseados no vetor gradiente. O Capítulo 3 apresenta os fundamentos do  $q$ -cálculo, trazendo conceitos sobre a  $q$ -derivada, vetor  $q$ -gradiente e o primeiro método de otimização baseado nesses conceitos. Os métodos convergentes baseados no vetor  $q$ -gradiente são também apresentados no Capítulo 3. Já no Capítulo 4, a prova de convergência desses métodos é realizada. A avaliação do desempenho computacional dos métodos desenvolvidos nesta tese é apresentada no Capítulo 5. O Capítulo 6 traz duas aplicações a problemas complexos de otimização e os resultados obtidos pelas  $q$ -versões desenvolvidas. Por fim, no Capítulo 7, são apresentadas as conclusões deste trabalho e sugestões para trabalhos futuros.

## 2 CONCEITOS FUNDAMENTAIS DE OTIMIZAÇÃO

Os conceitos mais relevantes a respeito dos problemas de otimização com apenas uma função objetivo são aqui apresentados. Primeiramente, uma breve explicação da formulação matemática sobre problemas de otimização é realizada. Em seguida, são abordados conceitos referentes às condições de otimalidade, taxas de convergência e possíveis critérios de parada. Por fim, são apresentados brevemente alguns dos principais algoritmos de otimização determinísticos e estocásticos.

### 2.1 Problemas de Otimização

Problemas de otimização tem por objetivo encontrar a melhor combinação dentre um conjunto de variáveis de decisão  $\mathbf{x}$  que minimize uma função objetivo  $f(\mathbf{x})$ . Há também, problemas de otimização que requerem maximizar a função objetivo. Estes problemas, no entanto, podem ser representados como problemas de minimização, uma vez que maximizar  $f(\mathbf{x})$  é equivalente a minimizar  $-f(\mathbf{x})$ . Dessa forma, por simplicidade e sem perda de generalidade, os problemas de otimização ao longo deste texto, serão tratados como problemas de minimização.

Um passo importante no processo da otimização é fazer a correta classificação dos problemas a serem otimizados, uma vez que os diversos algoritmos existentes são adaptados para cada tipo de problema.

Os problemas podem ser classificados com base na natureza das variáveis de decisão, função objetivo e restrições, podendo ser contínuos ou discretos, restritos ou irrestritos, lineares ou não-lineares, convexos ou não-convexos, unimodais ou multimodais, locais ou globais, monoobjetivo ou multiobjetivo, dentre outros.

A formulação matemática para tais problemas pode ser dada por (LUENBERGER, 1984)

$$\min_{\mathbf{x} \in \Omega} f(\mathbf{x}), \quad (2.1)$$

sendo  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  a função objetivo que se deseja otimizar,  $\mathbf{x} = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$  as variáveis de decisão e  $\Omega \subseteq \mathbb{R}^n$  o espaço de busca viável.

Se  $\Omega = \mathbb{R}^n$ , o problema de otimização é dito sem restrições (ou irrestrito). Mas se  $\Omega$  é um subconjunto do  $\mathbb{R}^n$ , o problema é dito com restrições e sua formulação é dada por

$$\begin{aligned}
& \min_{\mathbf{x} \in \Omega} && f(\mathbf{x}) \\
& && g_j(\mathbf{x}) \leq 0 \quad j \in 1, 2, \dots, m \\
& && h_k(\mathbf{x}) = 0 \quad k \in 1, 2, \dots, l \\
& && \mathbf{x}^I \leq \mathbf{x} \leq \mathbf{x}^S,
\end{aligned} \tag{2.2}$$

em que  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$  é a restrição de desigualdade,  $h : \mathbb{R}^n \rightarrow \mathbb{R}^l$  é a restrição de igualdade e  $\mathbf{x}^I$  (limite inferior) e  $\mathbf{x}^S$  (limite superior) são as restrições laterais.

A solução procurada para o problema de otimização, seja ele restrito ou irrestrito, é classificada como global ou local, segundo a definição abaixo (SUN; YUAN, 2006).

**Definição 1.** Um ponto  $\mathbf{x}^*$  é mínimo local se existe um  $\varepsilon > 0$ , tal que,  $f(\mathbf{x}^*) \leq f(\mathbf{x})$  para todo  $\mathbf{x} \in \Omega$  satisfazendo  $\|\mathbf{x} - \mathbf{x}^*\| < \varepsilon$ . Um ponto  $\mathbf{x}^*$  é mínimo local estrito se existe um  $\varepsilon > 0$ , tal que,  $f(\mathbf{x}^*) < f(\mathbf{x})$  para todo  $\mathbf{x} \in \Omega$  com  $\mathbf{x} \neq \mathbf{x}^*$  e  $\|\mathbf{x} - \mathbf{x}^*\| < \varepsilon$ . Um ponto  $\mathbf{x}^*$  é mínimo global se  $f(\mathbf{x}^*) \leq f(\mathbf{x})$  para todo  $\mathbf{x} \in \Omega$ . Um ponto  $\mathbf{x}^*$  é mínimo global estrito se  $f(\mathbf{x}^*) < f(\mathbf{x})$  para todo  $\mathbf{x} \in \Omega$  com  $\mathbf{x} \neq \mathbf{x}^*$ .

A Figura 2.1 ilustra graficamente a Definição 1.

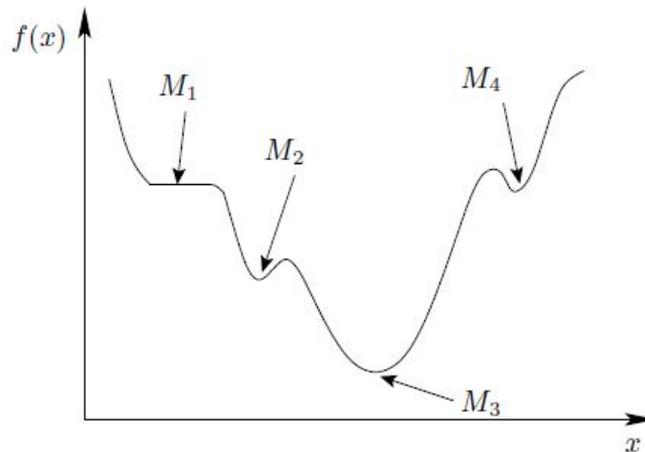


Figura 2.1 - Exemplos de mínimos:  $M_1$  é um mínimo local não estrito,  $M_2$  e  $M_4$  são mínimos locais estritos e  $M_3$  é o mínimo global.

Fonte: Collette and Siarry (2003).

Encontrar o mínimo global em um problema de otimização não é uma tarefa simples. Isso porque só se tem conhecimento dos valores para os pontos visitados durante a busca e os métodos disponíveis em geral, não fornecem garantias de que a solução encontrada seja de fato um mínimo global, mas apenas um mínimo local (NOCEDAL; WRIGHT, 2006).

Um caso especial e muito frequente em problemas de otimização é quando a função objetivo é convexa, dessa forma, o mínimo local é também o mínimo global da função. Um conjunto  $S \subset \mathbb{R}^n$  é dito convexo se o segmento de reta entre quaisquer dois pontos em  $S$  está contido em  $S$ , isto é, se para quaisquer  $\mathbf{x}, \mathbf{y} \in S$  e  $\theta \in [0, 1]$ , verifica-se que  $\theta\mathbf{x} + (1 - \theta)\mathbf{y} \in S$ , conforme apresenta a Figura 2.2 (SOUZA; DINIZ-EHRHARDT, 2011).

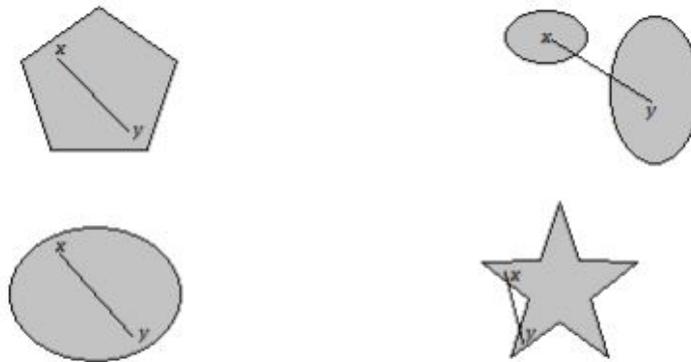


Figura 2.2 - Ilustração de definição de conjuntos convexos. À esquerda, exemplos de conjuntos convexos. À direita, exemplos de conjuntos não convexos.  
 Fonte: Souza and Diniz-Ehrhardt (2011).

Dessa forma, a noção de função convexa é definida como segue (SOUZA; DINIZ-EHRHARDT, 2011).

**Definição 2.** *Seja  $S$  um subconjunto convexo de  $\mathbb{R}^n$ . Uma função  $f : S \rightarrow \mathbb{R}$  é dita convexa se*

$$f(\theta\mathbf{x} + (1 - \theta)\mathbf{y}) \leq \theta f(\mathbf{x}) + (1 - \theta)f(\mathbf{y}), \quad \forall \mathbf{x}, \mathbf{y} \in S, \quad \forall \theta \in [0, 1].$$

A partir das definições acima, pode parecer que a única maneira de descobrir se um ponto  $\mathbf{x}^*$  é um mínimo local é examinar todos os pontos em sua vizinhança, e verificar se nenhum deles possui um valor de função objetivo menor. No entanto, existem maneiras mais práticas e eficientes para identificar os mínimos locais. As condições de otimalidade, vista na próxima seção, permitem determinar se um ponto  $\mathbf{x}^*$  é mínimo local do problema de otimização.

## 2.2 Condições de Otimalidade

Se  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  é uma função contínua, duas vezes diferenciável, então é possível determinar se um ponto  $\mathbf{x}^*$  é uma solução local do problema de otimização analisando apenas o gradiente da função objetivo  $\nabla f(\mathbf{x}^*)$ , e a matriz Hessiana  $\mathbf{H}(\mathbf{x}^*)$ , em que

$$\nabla f(\mathbf{x}^*)^T = \left[ \frac{\partial f(\mathbf{x}^*)}{\partial x_1} \quad \frac{\partial f(\mathbf{x}^*)}{\partial x_2} \quad \cdots \quad \frac{\partial f(\mathbf{x}^*)}{\partial x_n} \right], \quad (2.3)$$

e

$$\mathbf{H}(\mathbf{x}^*) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x}^*)}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{x}^*)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x}^*)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(\mathbf{x}^*)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\mathbf{x}^*)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(\mathbf{x}^*)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{x}^*)}{\partial x_n \partial x_1} & \frac{\partial^2 f(\mathbf{x}^*)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x}^*)}{\partial x_n^2} \end{bmatrix}. \quad (2.4)$$

A proposição seguinte apresenta a condição necessária de primeira ordem para  $\mathbf{x}^*$  ser um minimizador local (NOCEDAL; WRIGHT, 2006).

**Proposição 1.** *Se  $\mathbf{x}^*$  é um mínimo local e  $f$  é uma função contínua diferenciável em uma vizinhança de  $\mathbf{x}^*$ , então  $\nabla f(\mathbf{x}^*) = 0$ .*

Neste caso,  $\mathbf{x}^*$  também é conhecido como ponto estacionário do problema de otimização. As próximas proposições apresentam as condições necessárias e condições suficientes de segunda ordem, respectivamente. Essas condições requerer que a matriz Hessiana  $\mathbf{H}(\mathbf{x}^*)$  seja definida positiva, isto é,  $\mathbf{p}^T \mathbf{H}(\mathbf{x}^*) \mathbf{p} > 0$  para todo  $\mathbf{p} \neq 0$ , e semidefinida positiva se  $\mathbf{p}^T \mathbf{H}(\mathbf{x}^*) \mathbf{p} \geq 0$  para todo  $\mathbf{p} \in \mathbb{R}^n$  (NOCEDAL; WRIGHT, 2006).

**Proposição 2.** *Se  $\mathbf{x}^*$  é um mínimo local de  $f$  e  $\mathbf{H}(\mathbf{x}^*)$  é contínua em uma vizinhança de  $\mathbf{x}^*$ , então  $\nabla f(\mathbf{x}^*) = 0$  e  $\mathbf{H}(\mathbf{x}^*)$  é positiva semidefinida.*

**Proposição 3.** *Ao supor que  $\mathbf{H}(\mathbf{x}^*)$  seja contínua em uma vizinhança de  $\mathbf{x}^*$  e que  $\nabla f(\mathbf{x}^*) = 0$  e  $\mathbf{H}(\mathbf{x}^*)$  é positiva definida, então  $\mathbf{x}^*$  é um mínimo local estrito de  $f$ .*

Quando a função objetiva é convexa, os minimizadores globais e locais são simples de se caracterizar, conforme apresenta a definição seguinte (NOCEDAL; WRIGHT, 2006).

**Proposição 4.** Quando  $f$  é convexo, qualquer mínimo local  $\mathbf{x}^*$  é também mínimo global de  $f$ . Além disso, se  $f$  é diferenciável, então qualquer ponto estacionário  $\mathbf{x}^*$  é um mínimo global de  $f$ .

Já para o problema com restrições, da forma 2.2, um ponto de mínimo deverá satisfazer as condições de Kuhn-Tucker (VANDERPLAATS, 1984).

**Proposição 5.** Seja  $\mathbf{x}^*$  um mínimo local do problema de otimização com restrições, para  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$  e  $h : \mathbb{R}^n \rightarrow \mathbb{R}^l$  diferenciáveis, então as seguintes condições de Kuhn-Tucker devem ser satisfeitas:

**Condição 1.**  $\mathbf{x}^*$  é viável, ou seja, não viola nenhuma restrição.

**Condição 2.**  $\lambda_j g_j(\mathbf{x}^*) = 0$ ,  $\lambda_j \geq 0$ ,  $j = 1, \dots, m$ .

**Condição 3.**

$$\nabla f(\mathbf{x}^*) + \sum_{j=1}^m \lambda_j \nabla g_j(\mathbf{x}^*) + \sum_{k=1}^l \lambda_{m+k} \nabla h_k(\mathbf{x}^*) = \mathbf{0}, \quad \lambda_j \geq 0, \quad \lambda_{m+k} \text{ com qualquer sinal.}$$

Essas condições são definidas como os pontos estacionários do Lagrangeano:

$$\mathbf{L}(\mathbf{x}; \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{j=1}^m \lambda_j g_j(\mathbf{x}) + \sum_{k=1}^l \lambda_{k+m} h_k(\mathbf{x}), \quad (2.5)$$

onde  $\lambda$  é o multiplicador de Lagrange.

A primeira condição é uma exigência óbvia de que a variável de decisão deve satisfazer todas as restrições. A segunda condição impõe a exigência de que, se a restrição  $g_j(\mathbf{x})$  não for satisfeita (isto é,  $g_j(\mathbf{x}) < 0$ ) então o multiplicador de Lagrange correspondente deve ser zero. E a terceira condição define o gradiente nulo.

A interpretação geométrica das condições de Kuhn-Tucker pode ser compreendida através da Figura 2.3, que mostra um problema de minimização de duas variáveis com três condições de desigualdade. Note que  $\mathbf{x}^*$  é viável, logo a primeira condição é satisfeita. A restrição  $g_3(\mathbf{x}^*)$  não é satisfeita e, portanto,  $\lambda_3 = 0$ . Uma vez que  $g_1(\mathbf{x}^*) = 0$  e  $g_2(\mathbf{x}^*) = 0$ , a segunda condição de Kuhn-Tucker é satisfeita em relação a essas restrições. A terceira condição exige que, ao multiplicar o gradiente de cada restrição estacionária, no caso  $g_1(\mathbf{x}^*)$  e  $g_2(\mathbf{x}^*)$ , por seu multiplicador de Lagrange correspondente, a soma vetorial do

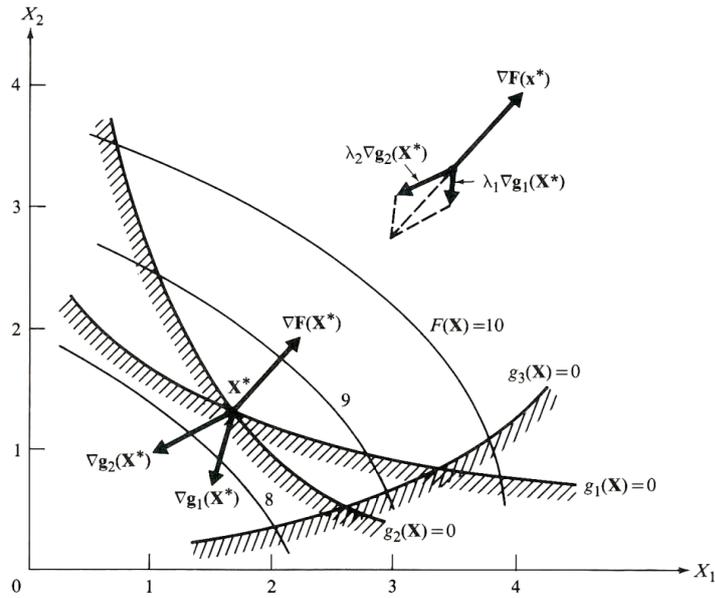


Figura 2.3 - Interpretação geométrica das condições de Kuhn-Tucker.  
 Fonte: Vanderplaats (1984).

resultado deve ser igual ao negativo do vetor gradiente da função objetivo. Assim,

$$\nabla f(\mathbf{x}^*) + \lambda_1 \nabla g_1(\mathbf{x}^*) + \lambda_2 \nabla g_2(\mathbf{x}^*) = 0, \quad (2.6)$$

com  $\lambda_1 \geq 0$  e  $\lambda_2 \geq 0$ . Portanto, cada uma das condições necessárias de Kuhn-Tucker é satisfeita.

### 2.3 Taxa de Convergência

Um comportamento típico de um algoritmo de otimização considerado aceitável é quando uma determinada iteração  $\mathbf{x}^{(k)}$  se move no sentido de um minimizador local  $\mathbf{x}^*$ , e rapidamente converge para o ponto  $\mathbf{x}^*$ .

A taxa de convergência é uma maneira de medir a eficácia de um método de otimização. Seja  $\mathbf{x}^{(k)}, k = 0, 1, \dots, n$  uma sequência de pontos obtidos por um algoritmo de otimização. Então, o algoritmo converge para  $\mathbf{x}^*$ , se (SUN; YUAN, 2006)

$$\lim_{k \rightarrow \infty} \|\mathbf{x}^{(k)} - \mathbf{x}^*\| = 0. \quad (2.7)$$

A noção mais empregada de taxa de convergência é a que considera o quociente entre duas sucessivas iterações, isto é,  $\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\| / \|\mathbf{x}^{(k)} - \mathbf{x}^*\|$ , supondo  $\mathbf{x}^{(k)} \neq \mathbf{x}^*$  para

todo  $k$ . Assim, se existe um número real  $p$  e uma constante positiva  $\beta$ , o qual independe da iteração  $k$ , a taxa de convergência é dada por

$$\lim_{k \rightarrow \infty} \frac{\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\|}{\|\mathbf{x}^{(k)} - \mathbf{x}^*\|^p} = \beta. \quad (2.8)$$

Em particular,

- 1) Quando  $p = 1$  e  $\beta \in (0, 1)$ , a sequência  $\mathbf{x}^{(k)}$  converge linearmente para  $\mathbf{x}^*$ .
- 2) Quando  $p = 1$  e  $\beta = 0$ , ou  $1 < p < 2$  e  $\beta > 0$ , a sequência  $\mathbf{x}^{(k)}$  possui convergência superlinear.
- 3) Quando  $p = 2$ , a convergência é dita quadrática.

Embora estimativas de taxa de convergência serem úteis na comparação qualitativa entre métodos de otimização, é importante frisar que a taxa de convergência não é a única característica relevante nesse sentido. É indispensável, por exemplo, levar em conta o custo computacional de uma iteração, pois um algoritmo com uma taxa de convergência rápida pode ser computacionalmente lento (IZMAILOV; SOLODOV, 2007).

## 2.4 Critérios de Parada

Quando a regra de convergência é satisfeita, a iteração deverá ser terminada. Em geral, o critério de parada mais natural é dado por

$$\|\nabla f(\mathbf{x}^{(k)})\| \leq \varepsilon, \quad (2.9)$$

onde  $\varepsilon$  é uma tolerância estabelecida. Se a condição 2.9 é satisfeita, implica que o vetor gradiente  $\nabla f(\mathbf{x}^{(k)})$  tende a zero e a sequência  $\mathbf{x}^{(k)}$  converge para o ponto estacionário do problema.

Outros critérios de parada que garantem a convergência dos algoritmos requerem que

$$|f(\mathbf{x}^{(k)}) - f(\mathbf{x}^*)| \leq \varepsilon, \quad (2.10)$$

ou

$$\| \mathbf{x}^{(k)} - \mathbf{x}^* \| \leq \varepsilon, \quad (2.11)$$

onde o parâmetro  $\varepsilon$  é definido pelo usuário. Porém, a menos que se conheça o mínimo global a priori, esses critérios não são utilizados na prática, uma vez que precisam da informação da solução  $\mathbf{x}^*$  do problema.

Também existem os critérios de parada que se baseiam na parte mais recente da sequência  $\{\mathbf{x}^{(k)}\}$  ou da sequência  $f(\mathbf{x}^{(k)})$ , isto é,

$$\| \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \| \leq \varepsilon, \quad (2.12)$$

ou

$$|f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)})| \leq \varepsilon, \quad (2.13)$$

para  $\varepsilon > 0$  e pequeno.

No entanto, regras de parada descritas nas Equações 2.12 e 2.13 não são teoricamente confiáveis, pois em geral, essas regras não garantem a proximidade do iterando  $\mathbf{x}^{(k+1)}$  a uma solução do problema em nenhum sentido. Mesmo assim, elas são muito utilizadas em métodos computacionais, principalmente por serem facilmente implementadas (IZMAILOV; SOLODOV, 2007).

Além disso, um critério de parada que pode estar implícito no algoritmo é o número máximo de avaliações da função objetivo. Na prática, para problemas de difícil resolução em que o procedimento iterativo é lento, o método é parado por exaustão do tempo disponível. Naturalmente, nesses casos a aproximação obtida pode não ser a solução do problema, mas é a melhor encontrada diante das ferramentas disponíveis (VANDERPLAATS, 1984) (IZMAILOV; SOLODOV, 2007). É importante nestes casos, utilizar alguma combinação de várias regras de parada, ordenadas numa hierarquia, sendo definida diante do problema em questão e da experiência computacional do programador.

## 2.5 Métodos de Otimização Baseados em Gradientes

A maioria dos métodos de otimização baseados em gradientes utilizam um procedimento iterativo para encontrar o menor valor da função objetivo. A ideia básica é que, dado um ponto inicial  $\mathbf{x}^{(0)} \in \mathbb{R}^n$ , uma sequência  $\mathbf{x}^{(k)}$  é gerada por algum método iterativo, de modo que, quando  $\mathbf{x}^{(k)}$  é uma sequência finita, o último ponto obtido é a solução ideal do problema de otimização.

Um procedimento iterativo muito utilizado por esses algoritmos consiste em iniciar a busca a partir de um ponto  $\mathbf{x}^{(0)}$ , escolhido pelo usuário ou definido de maneira arbitrária, e gerar uma sequência de possíveis soluções por meio da expressão (NOCEDAL; WRIGHT, 2006)

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)}, \quad (2.14)$$

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  $\mathbf{x}^{(k)} \in \mathbb{R}^n$  é o vetor de variáveis de decisão do problema na iteração  $k$ .

A maioria dos métodos de otimização que utilizam o procedimento iterativo (2.14) determinam o tamanho do passo  $\alpha^{(k)}$  na direção  $\mathbf{d}^{(k)}$  por meio de uma técnica de busca linear (PILLO; PALAGI, 2002). Essa técnica exige que  $\mathbf{d}^{(k)}$  seja uma direção de descida, ou seja,

$$\nabla f(\mathbf{x}^{(k)}) \mathbf{d}^{(k)} < 0. \quad (2.15)$$

Se  $\mathbf{d}^{(k)}$  é uma direção de descida, então existirá um  $\alpha^{(k)} > 0$  tal que

$$f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)}). \quad (2.16)$$

Um algoritmo base dos métodos de otimização baseados em gradientes é dado a seguir.

---

**Algoritmo 1:** Esquema básico dos métodos de otimização baseados em gradientes

---

**Entradas**

- (1) Função  $f(\mathbf{x})$  contínua e diferenciável com  $\mathbf{x} \in \mathbb{R}^n$
- (2) Ponto inicial  $\mathbf{x}^{(0)}$
- (3) Tolerância  $\varepsilon > 0$

Passo 1: **Faça**  $k = 0$

Passo 2: **Enquanto**  $|\nabla f(\mathbf{x}^{(k)})| \leq \varepsilon$ , **faça**

- (2a) Determinar a direção de busca  $\mathbf{d}^{(k)}$  por algum método de otimização
- (2b) Obter o tamanho do passo  $\alpha^{(k)}$  tal que  $f(\mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)}) < f(\mathbf{x}^{(k)})$
- (2c)  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)}$
- (2d)  $k = k + 1$

Passo 3: **Retorna**  $\mathbf{x}^k$

---

Os métodos de otimização baseados em gradientes diferem na forma em que a direção e o tamanho do passo são calculados no procedimento iterativo 2.14. As direções podem ser obtidas simplesmente pela direção contrária à direção do vetor gradiente da função obje-

tivo, outras, mais custosas computacionalmente, são obtidas por meio da matriz Hessiana da função, e ainda, algumas direções utilizam combinações que levam em consideração essas duas últimas a fim de direcionar a busca para o mínimo da função. As próximas subseções apresentam os principais métodos que utilizam informações dessas direções.

### 2.5.1 Método da Máxima Descida

O método da máxima descida (*steepest descent method*) é um dos mais antigos e simples métodos de otimização. Consiste em uma estratégia de busca que se move na direção contrária à direção do gradiente da função objetivo no ponto atual da busca e com um tamanho de passo apropriado a cada iteração (NOCEDAL; WRIGHT, 2006). Ou seja, a direção de busca é dada por

$$\mathbf{d}^{(k)} = -\nabla f(\mathbf{x}^{(k)}), \quad (2.17)$$

e a cada iteração o método faz uma busca unidimensional para encontrar um passo  $\alpha^{(k)}$  tal que

$$f(\mathbf{x}^{(k)} - \alpha^{(k)}\nabla f(\mathbf{x}^{(k)})) < f(\mathbf{x}^{(k)}). \quad (2.18)$$

As principais etapas do algoritmo para esse método, adaptado de Vanderplaats (1984), é apresentado a seguir.

---

#### Algoritmo 2: Método da Máxima Descida

---

##### Entradas

- (1) Função  $f(\mathbf{x})$  contínua e diferenciável com  $\mathbf{x} \in \mathbb{R}^n$
- (2) Ponto inicial  $\mathbf{x}^{(0)}$

Passo 1: **Faça**  $k = 0$

Passo 2: **Enquanto**  $\nabla f(\mathbf{x}^{(k)}) \neq 0$ , **faça**

(2a)  $\mathbf{d}^{(k)} = -\nabla f(\mathbf{x}^{(k)})$

(2b) Obtenha  $\alpha^{(k)} > 0$  tal que  $f(\mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{d}^{(k)}) < f(\mathbf{x}^{(k)})$

(2c)  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{d}^{(k)}$

(2d)  $k = k + 1$

Passo 3: **Retorna**  $\mathbf{x}^k$

---

Apesar do método da máxima descida possuir convergência lenta em muitos casos, ele é muito utilizado na prática. Isso porque o método é teoricamente simples e requer apenas

o cálculo da primeira derivada da função objetivo em cada ponto  $\mathbf{x}^{(k)}$  da busca. Dessa forma, o método da máxima descida é um padrão de referência para outros métodos mais avançados (LUENBERGER, 1984).

### 2.5.2 Método de Newton

A ideia principal do método de Newton é usar de forma iterativa a aproximação quadrática  $g^{(k)}$  da função objetivo  $f$  e minimizar a aproximação  $g^{(k)}$ .

Seja  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  uma função contínua, duas vezes diferenciável,  $\mathbf{x}^{(k)} \in \mathbb{R}^n$  e a matriz Hessiana  $\mathbf{H}(\mathbf{x}^k)$  definida positiva. O modelo da função  $f$  pela aproximação quadrática  $g^{(k)}$  é dada por

$$f(\mathbf{x}^{(k)} + \mathbf{s}) \approx g^{(k)}(\mathbf{s}) = f(\mathbf{x}^{(k)}) + \nabla f(\mathbf{x}^{(k)})^\top \mathbf{s} + \frac{1}{2} \mathbf{s}^\top \mathbf{H}(\mathbf{x}^k) \mathbf{s}, \quad (2.19)$$

onde  $\mathbf{s} = \mathbf{x} - \mathbf{x}^{(k)}$ . A minimização de  $g^{(k)}(\mathbf{s})$  é obtida por

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - [\mathbf{H}(\mathbf{x}^k)]^{-1} \nabla f(\mathbf{x}^{(k)}), \quad (2.20)$$

e este esquema iterativo é conhecido como método de Newton puro (SUN; YUAN, 2006).

Claramente, a direção de busca do método de Newton é uma direção de descida, pois satisfaz a propriedade 2.15, isto é

$$-\nabla f(\mathbf{x}^{(k)})^\top \mathbf{H}(\mathbf{x}^k)^{-1} \nabla f(\mathbf{x}^{(k)}) < 0. \quad (2.21)$$

Para uma função quadrática, o método de Newton pode convergir para o mínimo local  $\mathbf{x}^*$  com apenas uma iteração. Porém, para uma função geral, não quadrática, não existem garantias de que o método de Newton convirja para o mínimo com iterações finitas. No entanto, uma vez que a função objetivo é aproximada por uma função quadrática, e se o ponto inicial  $\mathbf{x}^{(0)}$  estiver próximo do mínimo local  $\mathbf{x}^*$ , então o método de Newton irá convergir rapidamente. Caso contrário, se  $\mathbf{x}^{(0)}$  está longe de  $\mathbf{x}^*$ , o método de Newton pode não convergir, pois não há garantias da matriz Hessiana ser definida positiva e a direção  $\mathbf{d}^{(k)}$  ser uma direção de descida. Nestes casos, o método de Newton requer modificações (SUN; YUAN, 2006).

Uma modificação que garanta convergência é empregar ao método de Newton o uso de um

tamanho de passo  $\alpha^{(k)}$  a cada iteração, além de testar a positividade da matriz Hessiana  $\mathbf{H}(\mathbf{x}^k)$  de acordo com o Método de Goldfeld. Mais precisamente, quando  $\mathbf{H}(\mathbf{x}^k)$  não é definida positiva, a matriz Hessiana é substituída por  $[\epsilon^{(k)}\mathbf{I} + \mathbf{H}(\mathbf{x}^k)]^{-1}$ , onde  $\epsilon^{(k)} > 0$  arbitrário (SUN; YUAN, 2006). Dessa forma, o procedimento iterativo da Eq. 2.14 é dado por

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha^{(k)}[\epsilon^{(k)}\mathbf{I} + \mathbf{H}(\mathbf{x}^k)]^{-1}\nabla f(\mathbf{x}^{(k)}), \quad (2.22)$$

onde  $\alpha^{(k)}$  é um parâmetro de busca positivo. Assim, se  $\mathbf{H}(\mathbf{x}^k)$  é definida positiva,  $\epsilon^{(k)} = 0$ , caso contrário  $\epsilon^{(k)} > 0$ . Dessa forma, a direção de busca do método de Newton no procedimento iterativo da Eq. 2.14 é dada por

$$\mathbf{d}^{(k)} = -[\epsilon^{(k)}\mathbf{I} + \mathbf{H}(\mathbf{x}^k)]^{-1}\nabla f(\mathbf{x}^{(k)}). \quad (2.23)$$

As principais etapas do algoritmo para esse método, adaptado de Sun and Yuan (2006), é apresentada a seguir.

---

**Algoritmo 3:** Método de Newton

---

**Entradas**

- (1) Função  $f(\mathbf{x})$  contínua e duas vezes diferenciável com  $\mathbf{x} \in \mathbb{R}^n$
- (2) Ponto inicial  $\mathbf{x}^{(0)}$
- (3)  $\epsilon^{(k)} > 0$

Passo 1: **Faça**  $k = 0$

Passo 2: **Enquanto**  $\nabla f(\mathbf{x}^{(k)}) \neq 0$ , **faça**

(2a) Obtenha  $\mathbf{d}^{(k)}$  pela Eq 2.23

(2b) Obtenha  $\alpha^{(k)} > 0$  tal que  $f(\mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{d}^{(k)}) < f(\mathbf{x}^{(k)})$

(2c)  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{d}^{(k)}$

(2d)  $k = k + 1$

Passo 3: **Retorna**  $\mathbf{x}^k$

---

Algoritmos que usam a direção do método de Newton possuem uma taxa de convergência rápida, tipicamente quadrática, isso depois que a vizinhança da solução é alcançada por alguma iteração. Porém, a necessidade de calcular a inversa da matriz Hessiana de  $f$  a cada iteração, torna o método custoso do ponto de vista computacional.

### 2.5.3 Método dos Gradientes Conjugados

O método dos gradientes conjugados está entre o método da máxima descida e o método de Newton. Este método é motivado pelo desejo de acelerar a convergência tipicamente

lenta do método da máxima descida, e ao mesmo tempo, evitar as informações associadas as derivadas de segunda ordem, as quais são necessárias pelo método de Newton. Dessa forma, uma vez que não precisa calcular a matriz Hessiana ou a sua aproximação, o método dos gradientes conjugados é amplamente utilizado na resolução de problemas de otimização de grande escala (LUENBERGER, 1984).

Inicialmente, o método dos gradientes conjugados foi desenvolvido para a minimização de uma função quadrática da forma (HESTENES, 1980)

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x}, \quad (2.24)$$

em que  $\mathbf{A} \in \mathbb{R}^{n \times n}$  é a matriz Hessiana definida positiva, isto é,  $\mathbf{A}^T = \mathbf{A}$  e  $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0, \forall \mathbf{x} \neq \mathbf{0}$  e  $\mathbf{b} \in \mathbb{R}^n$ .

Uma importante propriedade do método dos gradientes conjugados é a sua capacidade de gerar um conjunto de vetores conjugados  $\{\mathbf{d}^{(0)}, \mathbf{d}^{(1)}, \dots, \mathbf{d}^{(k)}\} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$  com respeito a matriz Hessiana  $\mathbf{A}$  tais que (HESTENES, 1980)

$$\langle \mathbf{A} \mathbf{d}^i, \mathbf{d}^j \rangle = 0 \quad \forall i, j = 0, 1, \dots, k, \quad i \neq j. \quad (2.25)$$

Qualquer conjunto de vetores que satisfaçam essa propriedade são linearmente independentes. Assim, para um passo  $\alpha^{(k)}$  obtido por uma busca linear exata, o método dos gradientes conjugados convergem em, no máximo,  $n$  iterações quando aplicado em funções quadráticas da forma 2.24, com matriz Hessiana definida positiva (IZMAILOV; SOLODOV, 2007).

O método dos gradientes conjugados utiliza o procedimento iterativo 2.14 para uma dada direção de busca  $\mathbf{d}^{(k)}$  e um dado tamanho do passo  $\alpha^{(k)}$ . Para funções quadráticas o tamanho do passo é obtido por busca linear exata e calculado explicitamente por (NOCEDAL; WRIGHT, 2006)

$$\alpha^{(k)} = - \frac{\mathbf{d}^{(k)T} \mathbf{d}^{(k)}}{\mathbf{d}^{(k)T} \mathbf{A} \mathbf{d}^{(k)}}. \quad (2.26)$$

A direção inicial  $\mathbf{d}^{(0)}$  é dada pela direção de máxima descida no ponto inicial  $\mathbf{x}^{(0)}$  ( $\mathbf{d}^{(0)} = -\nabla f(\mathbf{x}^{(0)})$ ). As demais direções  $\mathbf{d}^{(k)}$  são combinações lineares de  $-\nabla f(\mathbf{x}^{(k)})$  e da direção anterior  $\mathbf{d}^{(k-1)}$ , ou seja,

$$\mathbf{d}^{(k)} = -\nabla f(\mathbf{x}^{(k)}) + \delta^{(k)} \mathbf{d}^{(k-1)}, k \geq 1, \quad (2.27)$$

em que  $\delta^{(k)}$  é um escalar escolhido de modo que  $\mathbf{d}^{(k-1)}$  e  $\mathbf{d}^{(k)}$  sejam vetores conjuga-

dos em relação à matriz  $\mathbf{A}$ . Multiplicando a equação anterior por  $\mathbf{d}^{(k-1)T} \mathbf{A}$  e impondo a condição  $\mathbf{d}^{(k-1)T} \mathbf{A} \mathbf{d}^{(k)} = 0$ , tem-se

$$\delta^{(k)} = \frac{\nabla f(\mathbf{x}^{(k)})^T \mathbf{A} \mathbf{d}^{(k-1)}}{\mathbf{d}^{(k-1)T} \mathbf{A} \mathbf{d}^{(k-1)}}. \quad (2.28)$$

Note que o método dos gradientes conjugados gera um conjunto de vetores conjugados de forma que o vetor  $\mathbf{d}^{(k)}$  é obtido com base apenas no vetor anterior  $\mathbf{d}^{(k-1)}$ . Com isso, não é preciso armazenar na iteração  $k$  todos os vetores conjugados  $\mathbf{d}^{(0)}, \mathbf{d}^{(1)}, \dots, \mathbf{d}^{(k)}$  e, dessa forma, o método requer pouco armazenamento e o custo computacional é reduzido (LUENBERGER, 1984).

Um dos primeiros métodos dos gradientes conjugados para funções não-lineares foi introduzido por Fletcher e Reeves na década de 60 e é uma das primeiras técnicas utilizadas na resolução de problemas de otimização não-lineares de grande escala (FLETCHER; REEVES, 1964). A principal diferença entre os métodos dos gradientes conjugados para funções lineares e para funções não lineares está no cálculo do escalar  $\delta^{(k)}$ . Dentre outras formas para o cálculo desse escalar, uma expressão muito utilizada é a definida por Fletcher and Reeves (1964) e dada por

$$\delta^{(k)} = \frac{\|\nabla f(\mathbf{x}^{(k)})\|^2}{\|\nabla f(\mathbf{x}^{(k-1)})\|^2}. \quad (2.29)$$

As principais etapas do algoritmo para esse método, adaptado de Fletcher and Reeves (1964), é apresentada a seguir.

---

**Algoritmo 4:** Método dos Gradientes Conjugados de Fletcher e Reeves

---

**Entradas**

- (1) Função  $f(\mathbf{x})$  contínua e diferenciável com  $\mathbf{x} \in \mathbb{R}^n$
- (2) Ponto inicial  $\mathbf{x}^{(0)}$

Passo 1: **Faça**  $k = 0$

Passo 2: **Enquanto**  $\nabla f(\mathbf{x}^{(k)}) \neq 0$ , **faça**

(2a) **Se**  $k = 0$  **então**  $\mathbf{d}^{(k)} = -\nabla f(\mathbf{x}^{(k)})$  e vá para o Passo (2d)

(2b) Calcule  $\delta^{(k)}$  pela Equação 2.29

(2c)  $\mathbf{d}^{(k)} = -\nabla f(\mathbf{x}^{(k)}) + \delta^{(k)} \mathbf{d}^{(k-1)}$

(2d) Obtenha  $\alpha^{(k)}$  tal que  $f(\mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)}) < f(\mathbf{x}^{(k)})$

(2e)  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)}$

(2f)  $k = k + 1$

Passo 3: **Retorna**  $\mathbf{x}^k$

---

A convergência para o método de Fletcher e Reeves em funções quadráticas de  $n$  variáveis ocorre no máximo em  $n$  passos. Para funções não quadráticas a convergência não é garantida da mesma forma, mas mesmo assim é rápida pois toda função duas vezes diferenciável pode ser aproximada localmente por uma função quadrática (IZMAILOV; SOLODOV, 2007). Além disso, para funções objetivos mais gerais com termos não quadráticos, o método dos gradientes conjugados pode apresentar perda de conjugação ao longo do procedimento iterativo e gerar direções ineficientes. Uma estratégia para contornar esse problema e assegurar as propriedades de convergência do método é reinicializar a direção de busca como a direção de máxima descida do último ponto visitado sempre que houver perda de conjugação (PILLO; PALAGI, 2002).

#### 2.5.4 Métodos quase-Newton

É sabido que o Método de Newton é um poderoso método, devido em sua direção de busca utilizar informações da matriz Hessiana, a qual fornece informações úteis sobre a curvatura de uma função objetivo. No entanto, em várias aplicações práticas, a avaliação computacional da matriz Hessiana é dispendiosa, ou ainda, a mesma não está disponível em sua forma analítica. Dessa forma, o método quase-Newton surgiu com a proposta de não calcular a matriz Hessiana, mas sim, realizar uma aproximação iterativa de sua inversa, reduzindo assim o custo computacional do método de Newton e ao mesmo tempo mantendo uma rápida taxa de convergência (SUN; YUAN, 2006).

Os métodos quase-Newton utilizam o procedimento iterativo 2.14 com direção de busca dada por (LUENBERGER, 1984)

$$\mathbf{d}^{(k)} = -\mathbf{B}_k \nabla f(\mathbf{x}^{(k)}), \quad (2.30)$$

em que  $\mathbf{B}_k$  é uma matriz simétrica definida positiva. Note que, se  $\mathbf{B}_k = I^{(n)}$  ( $I^{(n)}$  é a matriz identidade de ordem  $n$  por  $n$ ), a direção de busca  $\mathbf{d}^{(k)}$  do método quase-Newton se transforma na direção do método da máxima descida e, se  $\mathbf{B}_k = [\mathbf{H}(\mathbf{x}^k)]^{-1}$ , a direção de busca se transforma na direção do método de Newton.

A matriz  $\mathbf{B}_k$  é atualizada a cada iteração  $k$ , de forma que aproxime da inversa da matriz Hessiana de  $f$  e satisfaça a relação

$$\mathbf{B}_{k+1} \mathbf{s}^{(k)} = \mathbf{r}^{(k)}, \quad (2.31)$$

a qual é conhecida como equação de quase-Newton (ou equação da secante), onde

$$\mathbf{r}^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}, \quad (2.32)$$

e

$$\mathbf{s}^{(k)} = \nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)}). \quad (2.33)$$

Assim, a matriz  $\mathbf{B}_k$  é atualizada pela expressão

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\mathbf{r}^{(k)}(\mathbf{r}^{(k)})^\top}{(\mathbf{r}^{(k)})^\top \mathbf{s}^{(k)}} - \frac{\mathbf{B}_k \mathbf{s}^{(k)}(\mathbf{B}_k \mathbf{s}^{(k)})^\top}{(\mathbf{s}^{(k)})^\top \mathbf{B}_k \mathbf{s}^{(k)}} + c(\mathbf{s}^{(k)})^\top \mathbf{B}_k \mathbf{s}^{(k)} \mathbf{v}^{(k)}(\mathbf{v}^{(k)})^\top, \quad (2.34)$$

com

$$\mathbf{v}^{(k)} = \frac{\mathbf{r}^{(k)}}{(\mathbf{r}^{(k)})^\top \mathbf{s}^{(k)}} - \frac{\mathbf{B}_k \mathbf{s}^{(k)}}{(\mathbf{s}^{(k)})^\top \mathbf{B}_k \mathbf{s}^{(k)}}. \quad (2.35)$$

Os métodos quase-Newton diferem entre si na forma como as atualizações da matriz  $\mathbf{B}_k$  são realizadas. Para  $c = 0$  na expressão 2.34, tem-se a fórmula de Davidon-Fletcher-Powell (DFP). Já para  $c = 1$ , tem-se a fórmula de Broyden-Fletcher-Goldfarb-Shanno (BFGS) (PILLO; PALAGI, 2002; IZMAILOV; SOLODOV, 2007).

Atualmente, o método quase-Newton BFGS é considerado o mais eficiente, com boas propriedades de convergência e com um desempenho numérico superior ao método quase-Newton DFP (SUN; YUAN, 2006).

As principais etapas do algoritmo para esse método, adaptado de Luenberger (1984), é apresentada a seguir

---

#### Algoritmo 5: Métodos quase-Newton

---

##### Entradas

(1) Função  $f(\mathbf{x})$  contínua e diferenciável com  $\mathbf{x} \in \mathbb{R}^n$

(2) Ponto inicial  $\mathbf{x}^{(0)}$

Passo 1: **Faça**  $k = 0$

Passo 2: **Enquanto**  $\nabla f(\mathbf{x}^{(k)}) \neq 0$ , **faça**

(2a) **Se**  $k = 0$  **então**  $B_k = I^n$ ,  $\mathbf{d}^{(k)} = -B_k \nabla f(\mathbf{x}^{(k)})$  e vá para o passo (2e)

(2b) Calcule  $r^{(k)}$  e  $s^{(k)}$  pelas Equações 2.32 e 2.33, respectivamente

(2c) Calcule  $B_k$  pela Equação 2.34

(2d)  $\mathbf{d}^{(k)} = -B_k \nabla f(\mathbf{x}^{(k)})$

(2e) Obtenha  $\alpha^{(k)}$  tal que  $f(\mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)}) < f(\mathbf{x}^{(k)})$

(2f)  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)}$

(2g)  $k = k + 1$

Passo 3: **Retorna**  $\mathbf{x}^k$

---

Os métodos quase-Newton geram aproximações  $\mathbf{x}^{(k)}$  para o mínimo da função com boas propriedades de convergência e sem a necessidade de avaliar a matriz Hessiana a cada iteração, como é feito no método de Newton, porém para problemas de grande escala o armazenamento da matriz  $\mathbf{B}_k$  pode inviabilizar o uso desses métodos (PILLO; PALAGI, 2002).

## 2.6 Métodos de Otimização Baseados em Meta-Heurísticas

Os métodos de otimização baseados em meta-heurísticas realizam a busca pelo novo ponto  $\mathbf{x}^{(k)}$  utilizando heurísticas tradicionais a fim de encontrar o mínimo global  $\mathbf{x}^*$  da função objetivo. Além disso, esses métodos possuem mecanismos que permitem a busca escapar de mínimos locais (GENDREAU; POTVIN, 2010). Exemplos de meta-heurísticas são os algoritmos evolutivos e recozimento simulado, apresentados a seguir.

### 2.6.1 Algoritmos Evolutivos

Os algoritmos evolutivos (AEs) são algoritmos estocásticos, cuja a busca pelo melhor valor da função objetivo é baseada em fenômenos naturais propostos por Darwin (1859). A ideia básica é começar com um conjunto de soluções, geralmente criados aleatoriamente, denominado população. Em cada iteração, novas populações são geradas a partir do conjunto existente, por meio da aplicação de operadores genéticos como reprodução, mutação e seleção. O melhor conjunto de população é mantido para a iteração seguinte, em que um novo ciclo de operações genéticas são realizadas. Assim, o algoritmo procede de uma maneira evolutiva, onde os indivíduos mais aptos sobrevivem. Embora simplista do ponto de vista de um biólogo, os algoritmos evolutivos são suficientemente complexos, no sentido de proporcionar mecanismos de busca de uma forma adaptativa e robusta (SPEARS, 2000).

A ideia de aplicar os princípios da evolução natural como técnica de busca remonta a trabalhos na década de 1950 (SPALL, 2003). Mais tarde, três abordagens gerais foram desenvolvidas de forma independente: Programação Evolutiva (PE) por Fogel et al. (1966) com o objetivo de utilizar os conceitos de evolução no desenvolvimento da inteligência artificial, Estratégias Evolutivas (EEs) por Rechenberg (1973) com o objetivo de solucionar problemas de otimização de parâmetros e Algoritmos Genéticos (AGs) por Goldberg (1989) e Holland (1992) com o objetivo de obter soluções computacionais eficientes para problemas de otimização. No entanto, a estrutura de qualquer método evolutivo é sempre a mesma. As principais etapas de um algoritmo evolutivo, adaptado de Back et al. (1997), é apresentada a seguir.

---

**Algoritmo 6:** Algoritmo Evolutivo Padrão

---

**Entradas**

(1) População inicial de indivíduos  $\mathbf{P}^{(k)}$

Passo 1: **Faça**  $k = 0$

Passo 2: Avalie os indivíduos de  $\mathbf{P}^{(k)}$  e obtenha a aptidão para cada um

Passo 3: **Enquanto** algum critério de parada não for atingido, **faça**

(3a) Selecione os indivíduos de  $\mathbf{P}^{(k)}$  com melhor aptidão

(3b) Recombine e gere novos indivíduos para  $\mathbf{P}^{(k)}$

(3c) Modifique os novos indivíduos de  $\mathbf{P}^{(k)}$

(3d) Avalie os novos indivíduos de  $\mathbf{P}^{(k)}$  e obtenha a aptidão para cada um

(3e) Selecione uma nova população  $\mathbf{P}^{(k+1)}$  a partir de  $\mathbf{P}^{(k)}$

(3f)  $k = k + 1$

Passo 4: **Retorna** indivíduo com melhor aptidão

---

Geralmente, um algoritmo evolutivo inicializa a sua população aleatoriamente, embora o conhecimento específico do domínio também possa ser utilizado. A avaliação mede a aptidão de cada indivíduo, de acordo com o seu valor no ambiente em que se encontra, podendo ser simples como calcular uma função matemática, ou complexa, como executar uma simulação elaborada. A seleção dos indivíduos no Passo (3a) decide quem serão os pais, isto é, os indivíduos com melhor aptidão são mais propensos a serem pais e terem mais filhos (indivíduos). Os filhos são criados através de recombinação e mutação. Na recombinação, os filhos trocam informações entre os pais, transmitindo para as novas gerações as boas propriedades das gerações anteriores. Na mutação, os filhos são perturbados ainda mais, com o intuito de diversificar a população e assim terem mais chances de escapar dos mínimos locais. Os filhos são então avaliados e finalmente uma nova população é gerada (Passo (3e)). O processo se repete até que um critério de parada seja satisfeito, podendo esse ser um número máximo de gerações ou qualquer outro critério que esteja intrínscio ao problema.

Os algoritmos evolutivos representam uma grande classe de algoritmos baseados na evolução natural. Atualmente existem diversas extensões e aprimoramentos desses algoritmos para as mais diversas aplicações. A principal diferença entre os algoritmos evolutivos está justamente na definição dos indivíduos ou variáveis de decisão, podendo ser com codificação binária, inteira ou real. Além disso, diferentes estratégias para os operadores de cruzamento e mutação definem as particularidades de cada algoritmo evolutivo. Por exemplo, a estratégia evolutiva com matriz de covariância adaptada (CMA-ES) é um dos mais poderosos algoritmos evolutivos para a otimização de codificação real. Esse algoritmo emprega uma análise dos componentes principais dos passos de mutação selecionados na geração anterior para calcular a distribuição a ser empregada pela mutação na

próxima geração (HANSEN; MLADENOVIĆ, 1997). Já o algoritmo ISRES, uma estratégia evolutiva estocástica melhorada, se baseia em uma combinação de regras de mutações e atualizações, semelhantes ao método de Nelder-Mead livres de derivadas, com o intuito de encontrar o mínimo global da função objetivo (RUNARSSON; YAO, 2005).

Existem algumas características nos algoritmos evolutivos no qual seu sucesso é atribuído. Primeiro, os AEs trabalham com uma população em vez de um único ponto. Assim, a probabilidade de ficar preso em um mínimo local é reduzida em comparação com os métodos que procedem a busca ponto a ponto. Em segundo, os AES são de fácil implementação, o que os torna muito atrativos no ponto de vista computacional. Além disso, são adaptáveis a problemas com diversos tipos de variáveis e função objetivo (RANGAIAH, 2010).

Porém, como desvantagem, os AEs podem sofrer convergência prematura em algumas aplicações, além da existência de vários parâmetros de ajuste. Contudo, estas desvantagens podem ser atenuadas adicionando procedimentos específicos para manter a diversidade da população e utilizando a lista de referência dos parâmetros de ajuste disponível na literatura para serem empregados em problemas gerais (LACERDA; CARVALHO, 1999; RANGAIAH, 2010).

### 2.6.2 Recozimento Simulado

O Recozimento Simulado (RS), do inglês *simulated annealing*, teve seu início com o trabalho de Kirkpatrick et al. (1983) para o problema de otimização combinatorial. A ideia principal do RS é reproduzir certos princípios termodinâmicos da produção de um cristal ideal. O método faz uso do algoritmo de Metropolis et al. (1953), originalmente proposto para determinar estados de equilíbrio de uma coleção de átomos a uma determinada temperatura  $T$ . Esta temperatura é o parâmetro crucial do RS que influencia tanto a confiabilidade quanto a eficiência da otimização.

Para produzir um cristal, a temperatura  $T$  precisa ser diminuída. Uma diminuição rápida de  $T$  resulta em irregularidades na estrutura cristalina, o que é semelhante à insuficiência para localizar o mínimo global, mas que é aceita probabilisticamente. Já uma redução extremamente lenta de  $T$  resultaria em um custo computacional proibitivo. Assim, a escolha do processo de resfriamento de  $T$  implica significativamente no desempenho do algoritmo. Em geral, a temperatura  $T$  decresce seguindo a regra  $T^{k+1} = \tau T^k$ , onde  $\tau$  é o fator de decréscimo ou resfriamento (GOLDEN; WASIL, 2002).

Dessa forma, a ideia básica do algoritmo do recozimento simulado é usar busca aleatória que aceita não só mudanças que melhoram a função objetivo, mas também, mudanças que

não são ideais. Em um problema de minimização, por exemplo, quaisquer movimentos ou alterações que diminuam o valor da função objetivo serão aceitos; entretanto, algumas mudanças que aumentem a função objetivo também serão aceitos probabilisticamente. As principais etapas do algoritmo do recozimento simulado, adaptado de Yang (2010), é apresentada a seguir.

---

**Algoritmo 7:** Recozimento Simulado

---

**Entradas**

- (1) Ponto inicial  $\mathbf{x}^{(0)} \in \mathbb{R}^n$
- (2) Temperatura inicial  $T^{(0)}$
- (3) Fator de resfriamento  $0 < \tau < 1$

Passo 1: **Faça**  $k = 0$

Passo 2: **Faça**  $\mathbf{x}_{melhor} = \mathbf{x}^{(0)}$

Passo 3: **Enquanto** algum critério de parada não for atingido, **faça**

- (3a) Gere uma solução  $\mathbf{y}^{(k)}$
- (3b) Gere  $r \in [0; 1]$  com distribuição uniforme
- (3c) Calcule  $\Delta f = f(\mathbf{y}^{(k)}) - f(\mathbf{x}^{(k)})$
- (3d) **Se**  $\Delta f < 0$  ou  $\exp[-\Delta f/T^{(k)}] > r$  **então**  
 $\mathbf{x}^{(k+1)} = \mathbf{y}^{(k)}$
- (3e) **Caso contrário**  
 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$
- (3f) **Se**  $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}_{melhor})$  **então**  $\mathbf{x}_{melhor} = \mathbf{x}^{(k+1)}$
- (3g)  $T^{(k+1)} = \tau \cdot T^{(k)}$
- (3h)  $k = k + 1$

Passo 4: **Retorna**  $\mathbf{x}_{melhor}$

---

A desvantagem do método é justamente nos ajustes da temperatura inicial  $T^0$  e no fator de resfriamento  $\tau$ . Se o fator de resfriamento for muito pequeno, a temperatura irá cair rapidamente, a busca se tornará local em poucas iterações do algoritmo e consequentemente com chances de ficar preso em um mínimo local, já que a probabilidade de aceitar soluções piores é muito pequena. Já se o fator de resfriamento for grande, a temperatura cairá lentamente, a busca permanecerá global por várias iterações do algoritmo, pois a probabilidade de aceitar soluções piores será grande, porém poderá resultar em uma não convergência do método. Por outro lado, o método do recozimento simulado tem a vantagem de ser flexível e de fácil implementação, podendo ser aplicado em problemas de variáveis discretas, contínuas ou mistas (DRÉO A. PÉROWSKI; TAILLARD, 2006).

## 2.7 Métodos de Otimização Livres de Derivadas

Os métodos de otimização para funções livres de derivadas tem se tornado uma importante e desafiadora área. A necessidade cada vez maior em resolver problemas de otimização definidos por funções nas quais as derivadas estão indisponíveis ou disponíveis a um custo proibitivo e a crescente complexidade na modelagem matemática são algumas das razões pelas quais a otimização livre de derivadas é atualmente uma área de grande demanda (CONN et al., 1996).

O desenvolvimento de algoritmos livre de derivadas remonta aos trabalhos de Spendley et al. (1962) e Nelder and Mead (1965) com seus algoritmos baseados em simplex. Trabalhos seguintes tiveram um progresso significativo, fornecendo provas de convergência (CONN et al., 1996; LUCIDI; SCIANDRONE, 2002) e incorporando o uso de modelos alternativos (BOOKER et al., 1998). Dessa forma, pode-se afirmar que existem três características presentes em todos os algoritmos livres de derivadas que são globalmente convergentes: (1) os métodos incorporam mecanismos para impor que a direção de busca seja sempre de descida, assim como nos métodos baseados em gradientes. Além disso, combinando esses mecanismos com uma possível redução do parâmetro do tamanho do passo, a direção de descida é garantida longe do ponto estacionário do problema; (2) os métodos devem garantir alguma forma de controle da geometria do conjunto de pontos onde a função é avaliada. Essencialmente, essas operações asseguram que qualquer indício de estacionariedade é de fato verdadeira. Não impor uma boa geometria implica em falta de convergência do método; (3) os métodos devem conduzir a sequência do parâmetro de tamanho do passo para zero. Neste caso, o tamanho do passo possui um duplo propósito, além de controlar o tamanho da área de onde a função é amostrada em torno da iteração atual, o tamanho do passo está intimamente ligado com o critério de parada do algoritmo. Embora nos métodos baseados em gradientes esse tipo de terminação do algoritmo seja prematuro e uma indicação de falha do método, talvez porque as derivadas ou não são precisas o suficiente ou mal codificadas, nos métodos livres de derivadas esse tipo de terminação é uma consequência implícita dos mecanismos de algoritmos eficazes. Desse modo, um critério de parada com base no tamanho do passo é natural (CONN et al., 1996).

A seguir, um dos mais famosos métodos para otimização livres de derivadas denominado Método Simplex de Nelder-Mead é apresentado.

### 2.7.1 Método Simplex de Nelder-Mead

Desde sua publicação em 1965, o método simplex de Nelder Mead (NELDER; MEAD, 1965) tornou-se um dos métodos mais utilizados na área da otimização não-linear, principal-

mente na área de química e medicina (WRIGHT, 1995). O objetivo do método é minimizar uma função não-linear utilizando apenas os valores da função objetivo, sem qualquer informação de sua derivada de forma explícita ou implícita.

O método de Nelder-Mead, assim como grande parte dos métodos de busca direta, utilizam os conceitos de um simplex no  $\mathbb{R}^n$ , ou seja, uma figura geométrica em  $n$  dimensões que é fecho convexo com  $n + 1$  vértices. Para duas variáveis, por exemplo, um simplex é um triângulo, e o método compara os valores da função nos três vértices do triângulo. O pior vértice, onde o valor da função objetivo é maior, é rejeitado e substituído por um novo vértice. Um novo triângulo é formado e a pesquisa se continua. Esse processo gera uma sequência de triângulos (podendo ter diferentes formas), para o qual os valores da função nos vértices ficam cada vez menores. O tamanho dos triângulos é reduzido e as coordenadas do ponto de mínimo da função são encontradas (LAGARIAS et al., 1998).

Sejam  $x_1^{(k)}, x_2^{(k)}, \dots, x_{n+1}^{(k)}$  os vértices do simplex na iteração  $k$ , o método de Nelder-Mead gera um sequência iterativa de simplexes para aproximar do valor ótimo da função. A cada iteração, os vértices são ordenados de acordo com os valores da função objetivo (LAGARIAS et al., 1998)

$$f(x_1^{(k)}) \leq f(x_2^{(k)}) \leq \dots \leq f(x_{n+1}^{(k)}), \quad (2.36)$$

onde  $x^{(1)}$  é o melhor vértice e  $x^{(n+1)}$  o pior, isto é, o vértice com o maior valor da função objetivo.

O algoritmo usa quatro operações possíveis: reflexão ( $\alpha$ ), expansão ( $\beta$ ), contração externa e interna ( $\gamma$ ) e redução ( $\delta$ ). De acordo com Nelder and Mead (1965) esses parâmetros devem satisfazer  $\alpha > 0$ ,  $\beta > 1$ ,  $0 < \gamma < 1$  e  $0 < \delta < 1$ . Uma escolha, praticamente universal, usado pelo algoritmo de Nelder-Mead padrão é definir  $\alpha = 1$ ,  $\beta = 2$ ,  $\gamma = 1/2$  e  $\delta = 1/2$  (LAGARIAS et al., 1998).

Cada uma das quatro operações visa obter um novo vértice, o qual substituirá o pior vértice. Para definir as operações do método simplex de Nelder-Mead é preciso fazer uso do centroide dos  $n$  melhores vértices ( $\bar{\mathbf{x}}$ ), isto é

$$\bar{\mathbf{x}} = \frac{1}{2} \sum_{i=1}^n \mathbf{x}_i. \quad (2.37)$$

Assim, as principais etapas do algoritmo para o método simplex de Nelder-Mead, adap-

tado de Lagarias et al. (1998) é apresentada a seguir.

Observe que se o ponto refletido é melhor que o vértice, então o simplex foi refletido numa direção que minimiza a função objetivo. Neste caso é importante expandir o simplex nessa direção. Caso contrário, se o ponto refletido não possuir um valor melhor da função objetivo, então é provável que o simplex esteja próximo do ponto de mínimo da função, por isso são testados os pontos de contração externa e interna (Passo (3e) e (3f)). Se nenhuma das operações resultou em um novo vértice com valor da função objetivo pelo menos melhor do que aquele correspondente ao vértice a ser rejeitado, então deve-se reduzir o simplex, pois o ponto de mínimo está em seu interior. A operação de redução do simplex é feita preservando o vértice  $x_1$  e aproximando os demais vértices na direção de  $x_1$ . As Figuras 2.4 e 2.5 ilustram esses conceitos.

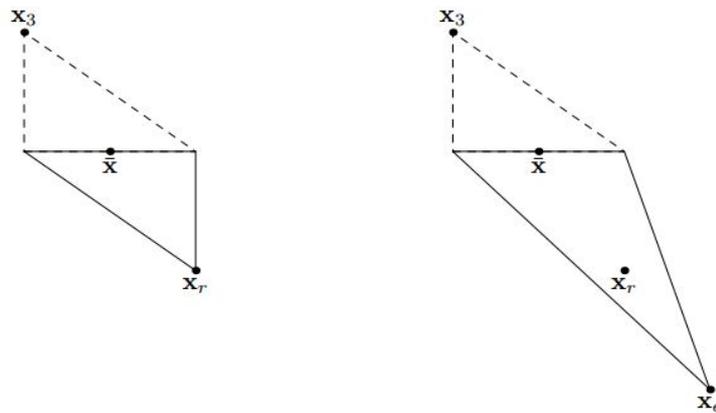


Figura 2.4 - Método simplex de Nelder-Mead depois de uma etapa de reflexão e expansão. O simplex original é mostrado com uma linha tracejada.

Fonte: (LAGARIAS et al., 1998).

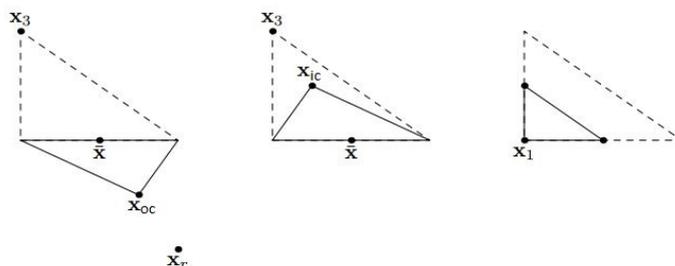


Figura 2.5 - Método simplex de Nelder-Mead após uma contração externa, uma contração interna e uma redução. O simplex original é mostrado com uma linha tracejada.

Fonte: (LAGARIAS et al., 1998).

---

**Algoritmo 8: Método Simplex de Nelder-Mead**

---

**Entradas**

- (1) Função  $f(\mathbf{x})$  com  $\mathbf{x} \in \mathbb{R}^n$
- (2) Conjunto de vértices iniciais:  $\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}, \dots, \mathbf{x}_{n+1}^{(k)} \in \mathbb{R}^n$
- (3) Parâmetro de reflexão  $\alpha > 0$
- (4) Parâmetro de expansão  $\beta > 1$
- (5) Parâmetro de contração  $0 < \gamma < 1$
- (6) Parâmetro de redução  $0 < \delta < 1$

Passo 1: **Faça**  $k = 0$

Passo 2: Calcule o valor da função objetivo para os  $n + 1$  vértices e ordene-os de forma que atenda a condição da Eq 2.36

Passo 3: **Enquanto** algum critério de parada não for atingido, **faça**

(3a) Calcule o ponto de reflexão  $\mathbf{x}_r^{(k)} = \bar{\mathbf{x}}^{(k)} + \alpha(\bar{\mathbf{x}}^{(k)} - \mathbf{x}_{n+1}^{(k)})$

(3b) Calcule  $f(\mathbf{x}_r^{(k)})$

(3c) **Se**  $f(\mathbf{x}_1^{(k)}) \leq f(\mathbf{x}_r^{(k)}) < f(\mathbf{x}_n^{(k)})$  **então**

$$\mathbf{x}_{n+1}^{(k)} = \mathbf{x}_r^{(k)}$$

(3d) **Se**  $f(\mathbf{x}_r^{(k)}) < f(\mathbf{x}_1^{(k)})$  **então**

$$\text{Calcule o ponto de expansão: } \mathbf{x}_e^{(k)} = \bar{\mathbf{x}}^{(k)} + \beta(\mathbf{x}_r^{(k)} - \bar{\mathbf{x}}^{(k)})$$

$$\text{Calcule } f(\mathbf{x}_e^{(k)})$$

**Se**  $f(\mathbf{x}_e^{(k)}) \leq f(\mathbf{x}_r^{(k)})$  **então**  $\mathbf{x}_{n+1}^{(k)} = \mathbf{x}_e^{(k)}$

**Senão**  $\mathbf{x}_{n+1}^{(k)} = \mathbf{x}_r^{(k)}$

(3e) **Se**  $f(\mathbf{x}_n^{(k)}) \leq f(\mathbf{x}_r^{(k)}) < f(\mathbf{x}_{n+1}^{(k)})$  **então**

$$\text{Calcule o ponto de contração externa: } \mathbf{x}_{oc}^{(k)} = \bar{\mathbf{x}}^{(k)} + \gamma(\mathbf{x}_r^{(k)} - \bar{\mathbf{x}}^{(k)})$$

$$\text{Calcule } f(\mathbf{x}_{oc}^{(k)})$$

**Se**  $f(\mathbf{x}_{oc}^{(k)}) \leq f(\mathbf{x}_r^{(k)})$  **então**  $\mathbf{x}_{n+1}^{(k)} = \mathbf{x}_{oc}^{(k)}$

**Senão** vá para o Passo (3g)

(3f) **Se**  $f(\mathbf{x}_r^{(k)}) \geq f(\mathbf{x}_{n+1}^{(k)})$  **então**

$$\text{Calcule o ponto de contração interna: } \mathbf{x}_{ic}^{(k)} = \bar{\mathbf{x}}^{(k)} - \gamma(\mathbf{x}_r^{(k)} - \bar{\mathbf{x}}^{(k)})$$

$$\text{Calcule } f(\mathbf{x}_{ic}^{(k)})$$

**Se**  $f(\mathbf{x}_{ic}^{(k)}) \leq f(\mathbf{x}_{n+1}^{(k)})$  **então**  $\mathbf{x}_{n+1}^{(k)} = \mathbf{x}_{ic}^{(k)}$

**Senão** vá para o Passo (3g)

(3g) Redução: para  $2 \leq i \leq n + 1$  faça  $\mathbf{x}_i^{(k)} = \mathbf{x}_1^{(k)} + \delta(\mathbf{x}_i^{(k)} - \mathbf{x}_1^{(k)})$

(3h)  $k = k + 1$  e volte ao Passo 2

Passo 4: **Retorna**  $\mathbf{x}$

---

O critério de parada, em geral, se baseia no volume do simplex. Por exemplo, caso o tamanho das arestas  $\| \mathbf{x}_i - \mathbf{x}_1 \|$  estejam abaixo de uma tolerância, considera-se que o método convergiu. Apesar da falta de uma teoria satisfatória de convergência, o método simplex de Nelder-Mead geralmente funciona bem para problemas de baixa dimensionalidade. Porém, para problemas com grandes dimensões, o método pode tornar-se ineficiente (GAO; HAN, 2012).



### 3 ALGORITMOS DE OTIMIZAÇÃO GLOBAL BASEADOS NO VETOR $q$ -GRADIENTE

Este capítulo apresenta os conceitos de  $q$ -derivada e  $q$ -gradiente, além de  $q$ -versões convergentes de três dos principais métodos clássicos de otimização local baseados no vetor gradiente. Uma versão convergente do método  $q$ -G é também apresentada.

#### 3.1 Fundamentos de $q$ -cálculo

##### 3.1.1 $q$ -Cálculo

O  $q$ -cálculo surgiu da generalização de expressões matemáticas através de um parâmetro multiplicativo  $q$ . Esse parâmetro  $q$  é muitas vezes visto como um parâmetro complexo, devido a ampla gama de diferentes generalizações que o envolvem (ERNST, 2012). É o caso dos  $q$ -números,  $q$ -derivadas,  $q$ -integrais, funções  $q$ -hipergeométricas, dentre outros, que no limite  $q \rightarrow 1$  retomam suas versões clássicas.

A história do  $q$ -cálculo remonta ao século XVIII e XIX, quando Leonhard Euler (1707-1783) introduziu o primeiro  $q$  nas séries infinitas de Newton. Em 1748, Euler descobriu as primeiras duas funções  $q$ -exponenciais, um prelúdio para o teorema  $q$ -binomial, e ao mesmo tempo, introduziu um operador que cem anos mais tarde conduziria ao operador  $q$ -derivada. O progresso do  $q$ -cálculo continuou nos termos de Johann Carl Friedrich Gauss (1777-1855), onde em 1812 criou a série hipergeométrica e suas relações de contiguidade. Além disso, baseado nos trabalhos de Euler, Gauss criou os coeficientes  $q$ -binomiais, provando identidades, e fazendo a base para o  $q$ -cálculo. Em 1846, Heinrich Eduard Heine (1821-1881) também fez sua colaboração, introduzindo a função  $q$ -hipergeométrica (ERNST, 2012). Mas, foi no início do século XX que o  $q$ -cálculo recebeu importantes contribuições, dentre elas a do reverendo inglês Frank Hilton Jackson (1870-1960). A partir de 1904, Jackson publicou uma série de artigos dedicados inteiramente ao  $q$ -cálculo, que duraria até 1951. Ele estudou funções elípticas, funções especiais e a obra de Heine, além de encontrar  $q$ -análogos de funções trigonométricas, funções de Bessel, polinômios de Legendre e da função gama (ERNST, 2000). Considerado como um dos maiores heróis do  $q$ -cálculo, Frank Hilton Jackson reintroduziu o operador  $q$ -derivada, que ficou amplamente conhecido como derivada de Jackson, e criou o conceito de  $q$ -integral definida (JACKSON, 1904), (JACKSON, 1909), (JACKSON, 1910b), (JACKSON, 1910a).

Nos últimos anos, o interesse quanto ao  $q$ -cálculo aumentou significativamente. Isto é, naturalmente, devido ao fato do  $q$ -cálculo provar ser extremamente frutífero em vários

campos, com aplicações de grande alcance em áreas tão vitais como ciência da computação e física de partículas, além de atuar também como uma ferramenta importante para pesquisadores que trabalham com a teoria analítica dos números ou física teórica (ERNST, 2012).

### 3.1.2 $q$ -Derivada

A derivada clássica avalia o quanto uma dada função  $f(x)$  é sensível a pequenas *translações* ( $h$ ) em sua variável independente ( $x$ ) por meio da equação

$$\frac{df(x)}{dx} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}. \quad (3.1)$$

A generalização do conceito de derivada desenvolvida por Jackson é baseada em *deformações* na variável independente, ou seja, em vez da variável independente  $x$  ser transladada por uma quantidade  $h$ , ela é dilatada ou contraída por uma quantidade  $q$  (SOTERRONI et al., 2015). Então, dada uma função diferenciável de uma única variável  $f(x)$  e um parâmetro  $q \in \mathbb{R}$ , a  $q$ -derivada (ou *derivada de Jackson*) de  $f(x)$  é definida como (JACKSON, 1909)

$$D_q f(x) = \frac{f(qx) - f(x)}{qx - x}, \quad (3.2)$$

para  $q \neq 1$  e  $x \neq 0$ . No limite,  $q \rightarrow 1$ , a  $q$ -derivada retorna à derivada clássica, ou seja,

$$\frac{df(x)}{dx} = \lim_{q \rightarrow 1} \frac{f(qx) - f(x)}{qx - x}. \quad (3.3)$$

### 3.1.3 $q$ -Gradiente

Analogamente, para funções  $f(\mathbf{x})$  diferenciáveis de  $n$  variáveis, Soterroni (2012) definiu a  $q$ -derivada parcial de primeira ordem com respeito à variável  $x_i$  por meio da expressão

$$D_{q_i, x_i} f(\mathbf{x}) = \begin{cases} \frac{f(x_1, \dots, q_i x_i, \dots, x_n) - f(x_1, \dots, x_i, \dots, x_n)}{q_i x_i - x_i} & \text{se } q_i \neq 1 \text{ e } x_i \neq 0 \\ \frac{\partial f(\mathbf{x})}{\partial x_i} & \text{outros casos} \end{cases} \quad (3.4)$$

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

onde  $e^{(i)}$  é a  $i$ -ésima coluna da matriz identidade  $I_n$ .

Note que um parâmetro  $q_i$  é necessário para cada variável  $x_i$ , pois o parâmetro  $q$  é, na verdade, um vetor de  $n$  variáveis, isto é  $\mathbf{q} = (q_1, \dots, q_i, \dots, q_n)$ .

Dessa forma, dado um vetor  $\mathbf{q} = (q_1, \dots, q_n) \in \mathbb{R}^n$ , o  $q$ -gradiente é definido como sendo um vetor das  $n$   $q_i$ -derivadas parciais de primeira ordem de  $f$  (SOTERRONI, 2012)

$$\nabla_{\mathbf{q}} f(\mathbf{x}) = [D_{q_1, x_1} f(\mathbf{x}) \dots D_{q_i, x_i} f(\mathbf{x}) \dots D_{q_n, x_n} f(\mathbf{x})]^T. \quad (3.5)$$

E no limite, quando  $q_i \rightarrow 1$  ( $\forall i = 1, \dots, n$ ), o vetor  $q$ -gradiente retorna ao vetor gradiente clássico.

Embora Soterroni et al. (2015) tenha definido o vetor  $q$ -gradiente para funções diferenciáveis, é possível defini-lo da mesma maneira para funções não diferenciáveis ou ainda para funções descontínuas desde que  $q_i \neq 1$  e  $x_i \neq 0$  para todo  $i$ .

O vetor  $q$ -gradiente é um caso especial do gradiente simplex (CONN et al., 2009; REGIS, 2015), que é amplamente utilizado em métodos de otimização sem derivadas. Dado um conjunto de  $n + 1$  pontos afim independentes  $\mathcal{X} = \{x^{(0)}, x^{(1)}, \dots, x^{(n)}\} \subset \mathbb{R}^n$  onde  $f$  tenha sido avaliada, o gradiente simplex de  $f$  com respeito a  $\mathcal{X}$  é o gradiente do modelo linear que interpola os pontos  $(x^{(0)}, f(x^{(0)}))$ ,  $(x^{(1)}, f(x^{(1)}))$ ,  $\dots$ ,  $(x^{(n)}, f(x^{(n)}))$ . Ou seja, seja  $S(\mathcal{X}) = [x^{(1)} - x^{(0)} \dots x^{(k)} - x^{(0)}] \in \mathbb{R}^{n \times k}$  e  $\delta_f(\mathcal{X}) = [f(x^{(1)}) - f(x^{(0)}) \dots f(x^{(k)}) - f(x^{(0)})]^T \in \mathbb{R}^k$ , o gradiente simplex de  $f$  em relação a  $\mathcal{X}$ , denotado por  $\nabla_S f(\mathcal{X})$  é dado por (CONN et al., 2009)

$$\nabla_S f(\mathcal{X}) = S(\mathcal{X})^{-T} \delta_f(\mathcal{X}). \quad (3.6)$$

Dessa forma, seja  $\mathbf{q}, \mathbf{x} \in \mathbb{R}^n$ , onde  $q_i \neq 1$  e  $x_i \neq 0$  para todo  $i$ , os pontos  $\mathbf{x}, \mathbf{x} + (q_1 - 1)x_1 e^{(1)}, \dots, \mathbf{x} + (q_n - 1)x_n e^{(n)}$  são afim independentes e  $\nabla_{\mathbf{q}} f(\mathbf{x})$  é o gradiente simplex de  $f$  com os respectivos pontos.

A seguinte proposição diz que quando  $f(x)$  é uma função linear, o vetor  $q$ -gradiente é idêntico ao gradiente simplex e ao vetor gradiente clássico.

**Proposição 6.** Se  $f(\mathbf{x}) = c_0 + c^T \mathbf{x}$ , onde  $c_0 \in \mathbb{R}$  e  $c \in \mathbb{R}^n$ , então para qualquer  $\mathbf{x}, \mathbf{q} \in \mathbb{R}^n$ ,  $\nabla_{\mathbf{q}} f(\mathbf{x}) = \nabla_{\mathbf{s}} f(\mathbf{x}) = \nabla f(\mathbf{x}) = c$ .

Regis (2015) provou regras de cálculo para os gradientes simplex que são semelhantes às aquelas satisfeitas pelos gradientes clássicos. Dessa forma, regras similares foram aplicadas ao vetor  $q$ -gradiente, incluindo regras do produto e regras do quociente. Nas proposições abaixo,  $f$  e  $g$  são funções de  $\mathbb{R}^n$  para  $\mathbb{R}$  e  $\text{diag}(a_1, \dots, a_n)$  representa uma matriz diagonal cujas entradas diagonais são  $a_1, \dots, a_n$ . A prova quando  $x_i \neq 0$  e  $q_i \neq 1$  para todo  $i$  podem ser encontradas em Regis (2015).

**Proposição 7.** Para qualquer  $\mathbf{x}, \mathbf{q} \in \mathbb{R}^n$  e para qualquer constante  $c$ ,

$$(a) \nabla_{\mathbf{q}}(f + g)(x) = \nabla_{\mathbf{q}}f(x) + \nabla_{\mathbf{q}}g(x), e$$

$$(b) \nabla_{\mathbf{q}}(cf)(x) = c\nabla_{\mathbf{q}}f(x).$$

**Proposição 8.** Para qualquer  $\mathbf{x}, \mathbf{q} \in \mathbb{R}^n$ ,

$$\nabla_{\mathbf{q}}(fg)(x) = f(x)\nabla_{\mathbf{q}}g(x) + \text{diag}(g(x + (q_1 - 1)x_1e^{(1)}), \dots, g(x + (q_n - 1)x_n e^{(n)}))\nabla_{\mathbf{q}}f(x).$$

**Proposição 9.** Para qualquer  $\mathbf{x}, \mathbf{q} \in \mathbb{R}^n$  para qual  $g(x), g(x + (q_1 - 1)x_1e^{(1)}), \dots, g(x + (q_n - 1)x_n e^{(n)})$  são todos diferentes de zero,

$$\nabla_{\mathbf{q}}\left(\frac{f}{g}\right)(x) = \text{diag}\left(\frac{1}{g(x + (q_1 - 1)x_1e^{(1)})}, \dots, \frac{1}{g(x + (q_n - 1)x_n e^{(n)})}\right) \left[\frac{g(x)\nabla_{\mathbf{q}}f(x) - f(x)\nabla_{\mathbf{q}}g(x)}{g(x)}\right].$$

Uma questão que surge quando se trabalha com interpolação é a forma de medir a precisão de  $\nabla_{\mathbf{q}}f(\mathbf{x})$  como uma aproximação de  $\nabla f(\mathbf{x})$ . Neste caso, o vetor  $q$ -gradiente satisfaz um *error bound* semelhante ao dos gradientes simplex. A próxima proposição segue imediatamente do Teorema 2.11 de Conn et al. (2009).

**Proposição 10.** Seja  $x^{(0)} \in \mathbb{R}^n$  e seja  $q \in \mathbb{R}^n$  tal que  $x_i^{(0)} \neq 0$  e  $q_i \neq 1$  para todo  $i$ . Supõe que  $f$  seja continuamente diferenciável em um domínio aberto  $\Omega$  que contem a bola fechada  $B(x^{(0)}, \Delta) = \{x \in \mathbb{R}^n : \|x - x^{(0)}\| \leq \Delta\}$ , onde  $\Delta = \max_{1 \leq i \leq n} |(q_i - 1)x_i^{(0)}|$ . Além disso, supõe que  $\nabla f$  seja Lipschitz contínua<sup>1</sup> em  $\Omega$  com constante de Lipschitz

<sup>1</sup>Um função  $f : S \rightarrow \mathbb{R}^n$  é dita Lipschitz contínua em  $\mathbf{x} \in S$ , se existir uma constante  $C > 0$ , tal que

$$\|f(\mathbf{y}) - f(\mathbf{x})\| \leq C \|\mathbf{y} - \mathbf{x}\|,$$

para todo  $\mathbf{y} \in S$  suficientemente próximo de  $\mathbf{x}$ .

$v > 0$ . Então para todo  $x \in B(x^{(0)}, \Delta)$

$$\|\nabla f(x) - \nabla_q f(x)\| \leq v \left(1 + n^{1/2} \|\widehat{L}^{-1}\|/2\right) \Delta,$$

onde  $\widehat{L} = \frac{1}{\Delta} \text{diag}((q_1 - 1)x_1^{(0)}, \dots, (q_n - 1)x_n^{(0)})$ .

### 3.1.4 Método $q$ -G

O primeiro método de otimização baseado no vetor  $q$ -gradiente foi desenvolvido recentemente (SOTERRONI et al., 2011; SOTERRONI et al., 2012; SOTERRONI, 2012; SOTERRONI et al., 2013; SOTERRONI et al., 2015). Inicialmente denominado por método do  $q$ -gradiente e, posteriormente, por método  $q$ -G, o método é uma generalização do método da máxima descida para problemas de otimização global contínuos. Neste método a direção de busca é definida a partir do vetor  $q$ -gradiente da função objetivo. O uso dessa direção com estratégias apropriadas para a obtenção tanto do parâmetro  $q$  quanto do tamanho do passo fornecem ao método  $q$ -G mecanismos para escapar de mínimos locais, e realizar uma transição suave entre busca global e busca local ao longo do procedimento iterativo. O método  $q$ -G introduziu o conceito de vetor  $q$ -gradiente na área de otimização global contínua.

O método da máxima descida, por sua vez, assim como outros métodos de otimização que utilizam informações do gradiente da função objetivo, se caracterizam em como a direção de busca e o tamanho do passo na Equação 2.14 são calculados. Assim, a principal diferença entre o método  $q$ -G e o método da máxima descida, está na substituição do vetor gradiente clássico  $\nabla f(\mathbf{x})$  pelo vetor  $q$ -gradiente  $\nabla_q f(\mathbf{x})$  (Eq. 3.5) no cálculo da direção de busca, além da forma como o tamanho do passo é calculado (SOTERRONI, 2012). Isto é, dada uma função objetivo  $f$  diferenciável com  $n$  variáveis, o método  $q$ -G utiliza o procedimento iterativo 2.14 a partir de um ponto inicial  $\mathbf{x}^{(0)} \in \mathbb{R}^n$  e ao invés de utilizar a Equação 2.17 como direção de busca, usa-se a expressão abaixo

$$\mathbf{d}^{(k)} = -\nabla_q f(\mathbf{x}^{(k)}), \quad (3.7)$$

onde  $\mathbf{d}^{(k)} \in \mathbb{R}^n$  na iteração  $k$ .

Conforme observado, o método  $q$ -G faz uso de um parâmetro  $q$ , e a correta especificação desse parâmetro é crítica para o bom desempenho do método. Assim, Soterroni et al. (2015) elaborou a estratégia de geração do fator  $q^{(k)} \mathbf{x}^{(k)}$  segundo uma distribuição gaussiana centrada no ponto atual da busca  $\mathbf{x}^{(k)}$  e com desvio-padrão  $\sigma^{(k)}$ . Dessa forma, dada uma função  $f(\mathbf{x})$  de  $n$  variáveis, um conjunto de  $n$  diferentes fatores  $q_i x_i \neq 1$  ( $i = 1, \dots, n$ )

são necessários para calcular o vetor  $q$ -gradiente de  $f$ . Logo, cada coordenada  $q_i$  é obtida a partir do fator  $q_i x_i$  gerado segundo uma distribuição gaussiana com função densidade de probabilidade dada por

$$f(q_i x_i; \sigma, \mu) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left[ -\frac{(q_i x_i - \mu)^2}{2\sigma^2} \right], \quad (3.8)$$

com desvio-padrão  $\sigma = \sigma^{(k)}$  e média  $\mu = x_i^{(k)}$ , em que  $x_i^{(k)}$  é a  $i$ -ésima coordenada do ponto atual da busca  $\mathbf{x}^{(k)}$  na iteração  $k$ . O desvio-padrão inicial é diferente de zero ( $\sigma^{(0)} > 0$ ) e tende a zero ao longo do procedimento iterativo por meio da expressão

$$\sigma^{(k+1)} = \beta \cdot \sigma^{(k)}, \quad (3.9)$$

em que  $\beta \in (0, 1)$  é o fator de redução.

A estratégia de redução do desvio-padrão faz com que o algoritmo mude de uma amostragem aleatória global durante as iterações iniciais para uma busca local, quase determinística, perto do final do processo de otimização. Como  $\sigma^{(k)}$  se aproxima de zero, os valores de  $q_i$  tendem a 1, e o método  $q$ -G tende a se comportar como o método da máxima descida. É importante enfatizar que se  $\sigma^{(k)}$  diminuir muito rapidamente, o algoritmo pode ficar preso em um mínimo local.

Além disso, o cálculo do comprimento do passo  $\alpha$  é um compromisso entre a obtenção de uma redução sensível no valor da função objetivo e o número de avaliações da função objetivo necessários para obtê-la (NOCEDAL; WRIGHT, 2006). Geralmente, os métodos baseados em gradientes realizam busca linear a cada iteração para determinar o comprimento do passo a ser dado em uma direção que é de descida (PILLO; PALAGI, 2002). Uma vez que as direções de busca geradas pelos algoritmos baseados em  $q$ -gradiente não são necessariamente direções de descida, uma estratégia que vem sendo adotada é a do comprimento do passo na iteração  $k$  calculado simplesmente através de uma recursão geométrica dada por

$$\alpha^{(k+1)} = \beta \cdot \alpha^{(k)}, \quad (3.10)$$

onde  $0 < \beta < 1$ . Por simplicidade, o fator de redução  $\beta$  é o mesmo utilizado no decréscimo do desvio-padrão  $\sigma^{(k)}$  (Equação 3.9), responsável por gerar o vetor de parâmetros  $\mathbf{q}$ . Como o tamanho do passo decresce (e os valores de  $q_i^{(k)}$  tendem a 1), uma transição

suave para um processo de busca cada vez mais local ocorre. As principais etapas do algoritmo para o método  $q$ -G são dadas abaixo. Mais detalhes sobre esse método podem ser encontrados em Soterroni (2012).

---

**Algoritmo 9:** Método  $q$ -G

---

**Entradas**

- (1) Função  $f(\mathbf{x})$  contínua e diferenciável com  $\mathbf{x} \in \mathbb{R}^n$
- (2) Ponto inicial  $\mathbf{x}^{(0)}$
- (3)  $\sigma^{(0)} > 0$
- (4)  $\alpha^{(0)} > 0$
- (5)  $0 < \beta < 1$

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

Passo 2: **Enquanto** não atingir um critério de parada, **faça**

- (2a) Obtenha  $q^{(k)}$  segundo uma distribuição gaussiana com  $\mu = \mathbf{x}^{(k)}$  e  $\sigma^{(k)}$
- (2b)  $\mathbf{d}^{(k)} = -\nabla_{\mathbf{q}} f(\mathbf{x}^{(k)})$
- (2c)  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)}$
- (2d) **Se**  $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}_{melhor})$  **então**  $\mathbf{x}_{melhor} = \mathbf{x}^{(k+1)}$
- (2e)  $\sigma^{(k+1)} = \beta \cdot \sigma^{(k)}$
- (2f)  $\alpha^{(k+1)} = \beta \cdot \alpha^{(k)}$
- (2g)  $k = k + 1$

Passo 3: **Retorna**  $\mathbf{x}_{melhor}$

---

O critério de parada do algoritmo pode ser um número máximo de avaliações da função objetivo ou uma precisão desejada, conforme descritos na Seção 2.4. O desvio padrão inicial  $\sigma^{(0)}$  determina quão global a busca está. Para funções multimodais, o desvio padrão deve ser suficientemente grande no início para permitir que o método faça uma amostragem adequada do espaço de busca. O fator de redução  $\beta$  controla a velocidade de transição de busca global para busca local. Um  $\beta$  próximo de 1 reduz o risco do algoritmo ficar preso em um mínimo local.

O método  $q$ -G foi comparado com os Algoritmos Evolutivos (AEs), que são considerados eficientes na resolução de problemas de otimização global, e os resultados comprovaram a capacidade do método  $q$ -G de escapar de extremos locais e atingir a bacia de atração do extremo global em muitas das funções teste analisadas (SOTERRONI et al., 2015).

A próxima seção apresenta uma modificação do método  $q$ -G necessária para garantir a sua convergência no sentido probabilístico, bem como  $q$ -versões convergentes do método dos gradientes conjugados e dos métodos quase-Newton Broyden-Fletcher-Goldfarb-Shanno (BFGS) e Davidon-Fletcher-Powell (DFP).

### 3.2 Método $q$ -G com Perturbações Gaussianas

Conforme já visto, o método  $q$ -G é uma generalização do método da máxima descida em que a direção de busca é dada pela direção contrária à direção do vetor  $q$ -gradiente da função objetivo. O método foi testado em várias funções teste unimodais e multimodais da literatura de otimização contínua e exibiu bons resultados, sobretudo para as funções teste multimodais (SOTERRONI et al., 2015).

Embora o método  $q$ -G original não tenha prova de convergência, os resultados numéricos mostram que o método possui mecanismos para escapar de mínimos locais. Dessa forma, esta seção apresenta modificações no algoritmo do método  $q$ -G, a fim de provar a sua convergência. A primeira modificação está na introdução de restrições laterais na variável independente com o objetivo de limitar o espaço de busca. A segunda, e última, modificação é a utilização de perturbações gaussianas em algumas iterações. A importância de se aplicar essas perturbações gaussianas é explicitado mais abaixo.

As principais etapas do algoritmo para o método  $q$ -G com perturbações gaussianas para resolver um problema de otimização com restrições laterais são dadas abaixo (Algoritmo 10).

No Passo 1, o melhor ponto é definido como o ponto inicial. Em seguida, no Passo 2, podemos distinguir entre dois tipos de iterações: a iteração regular que usa o vetor  $q$ -gradiente e uma iteração que usa perturbações gaussianas. Se o algoritmo está realizando a iteração do vetor  $q$ -gradiente, então o parâmetro  $q^{(k)}$  é selecionado no Passo 2 de tal forma que  $q_i^{(k)} x_i^{(k)}$  é obtido por uma distribuição normal com média  $x_i^{(k)}$  e desvio padrão  $\sigma^{(k)}$ . Caso contrário, é dado um salto para o Passo 9 para executar a iteração gaussiana. No Passo 3, o vetor  $q$ -gradiente é calculado no ponto atual da busca  $\mathbf{x}^{(k)}$ . Em seguida, no Passo 4, a direção da busca  $\mathbf{d}^{(k)}$  é calculada de acordo com a Equação 3.7. No Passo 5, é calculado o próximo ponto  $\mathbf{x}^{(k+1)}$ . Logo após, no Passo 6, é utilizado uma transformação de absorção  $\rho_{\mathcal{D}}$  com o intuito de garantir que o ponto atual esteja dentro do espaço de busca, e assim avaliar a função objetivo. Neste trabalho, a transformação de absorção utilizada se baseia no critério da restrição por reflexão, isto é, se o ponto atual estiver fora do espaço de busca ele é refletido para dentro do espaço de busca nas mesmas proporções. Nos Passos 7 e 8, o melhor ponto da busca e os parâmetros  $\sigma^{(k)}$  e  $\alpha^{(k)}$  são atualizados. Por fim, o algoritmo salta para o Passo 11 para verificar se algum critério de parada é satisfeito.

Se o algoritmo está executando a iteração gaussiana, então uma série de perturbações gaussianas  $r$  são geradas no Passo 9. Assim, no Passo 10, o novo ponto  $\mathbf{x}^{(k+1)}$  é definido

---

**Algoritmo 10:** Método  $q$ -G com perturbações gaussianas para otimização global e com restrições laterais

---

**Entradas**

- (1)  $f : \mathcal{D} \rightarrow \mathbb{R}$ , onde  $\mathcal{D} = [\ell, u] \subseteq \mathbb{R}^n$  (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  $k = 0$ ,  $\mathbf{x}_{best} = \mathbf{x}^{(0)}$  e  $\theta^{(m)} = \theta^{(0)}$

**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  $q_i^{(k)} x_i^{(k)} \notin [\ell_i, u_i]$ , então faça  $q_i^{(k)} x_i^{(k)}$  como sendo o ponto mais próximo dentro do espaço de busca e atualize o valor de  $q_i^{(k)}$

Fim do Passo 2

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

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

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

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

Fim do Passo 3

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

**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  $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}_{best})$  faça  $\mathbf{x}_{best} = \mathbf{x}^{(k+1)}$

**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  $f(\mathbf{y}^{(k,i)}) < f(\mathbf{x}_{best})$ , atualize  $\mathbf{x}_{best} = \mathbf{y}^{(k,i)}$

Fim do Passo 9

**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  $\mathbf{x}_{best}$  e  $f(\mathbf{x}_{best})$ . Caso contrário, incremente  $k = k + 1$  e vá para o Passo 2

---

como sendo o melhor entre os pontos anteriores  $\mathbf{x}^{(k)}$  e o resultado das perturbações gaussianas  $\mathbf{y}^{(k,1)}, \dots, \mathbf{y}^{(k,r)}$ . Além disso, o desvio-padrão das perturbações gaussianas é reduzido caso a iteração gaussiana não melhore o ponto atual  $\mathbf{x}^{(k)}$ . O fluxograma apresentado na Figura 3.1 ilustra os passos do algoritmo.

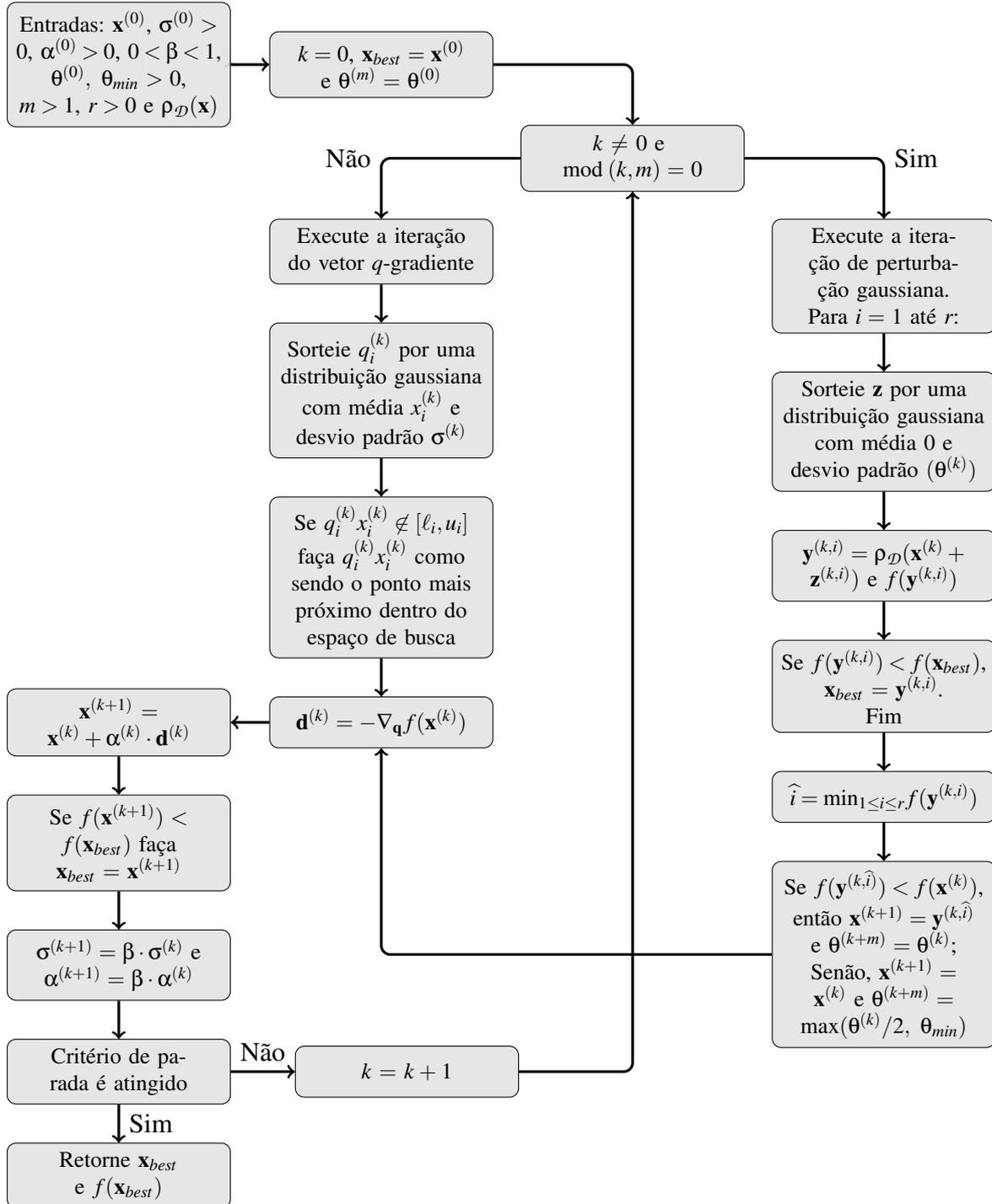


Figura 3.1 - Fluxograma do algoritmo do Método  $q$ -G com perturbações gaussianas.

Assim como descrito no algoritmo do método  $q$ -G original apresentado no Algoritmo 9, o critério de parada pode ser um número máximo de avaliações da função objetivo ou uma precisão desejada.

Além disso, perturbações gaussianas são usadas em algumas iterações (Passo 9) para garantir a convergência dos métodos para o mínimo global em um sentido probabilístico. A importância das perturbações gaussianas é bem ilustrada na Figura 3.2, a qual mostra as linhas de contorno da função  $f(x_1, x_2) = 2 + (x_1 - 2)^2 + (x_2 - 2)^2$  se  $(x_1 - 2)^2 + (x_2 - 2)^2 \leq 1$ ; senão  $f(x_1, x_2) = 3$  com ponto inicial em  $(x_1, x_2) = (0.5, 0.5)$  e os pontos amostrados pelo algoritmo do método  $q$ -G.

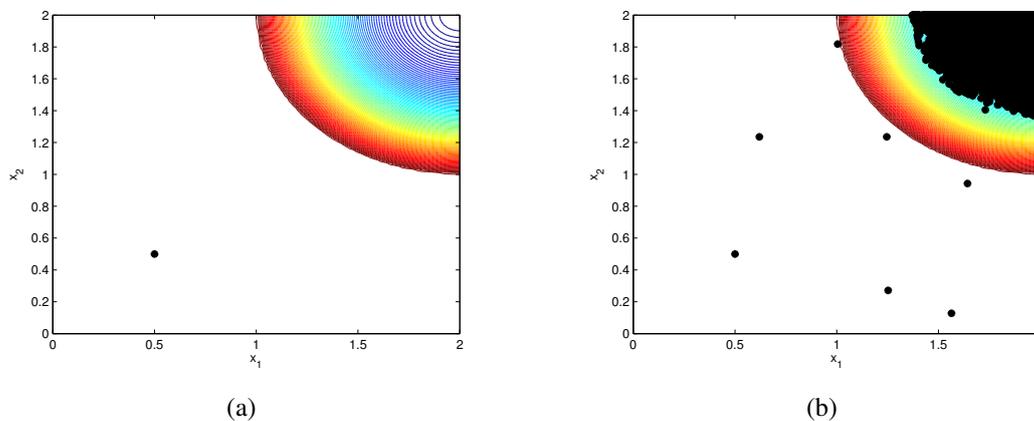


Figura 3.2 - Pontos gerados pelo método  $q$ -G. Sem perturbação gaussiana *versus* com perturbação gaussiana.

Note que a função  $f$  possui um mínimo global em  $(x_1, x_2) = (2, 2)$ . Para a Figura 3.2(a), onde o método  $q$ -G não usa perturbações gaussianas, os pontos estão amostrados apenas no ponto inicial para este tipo de função, uma vez que o vetor  $q$ -gradiente neste ponto é sempre um vetor nulo, devido a função neste ponto ser um platô. Quando as perturbações gaussianas no método  $q$ -G são usadas, os pontos podem potencialmente alcançar qualquer região do espaço de busca, sendo possível encontrar o mínimo global (veja Figura 3.2(b)).

### 3.3 Método $q$ -GC com Perturbações Gaussianas

Inspirado no método  $q$ -G, foi criada uma generalização do método dos gradientes conjugados para funções não-lineares, denominado método  $q$ -GC. O método dos gradientes conjugados, conforme apresentado na seção 2.5.3, utiliza o procedimento iterativo dado

na equação 2.14, onde a primeira direção é dada pelo método da máxima descida e as outras direções são combinações lineares da direção atual com as direções anteriores associadas com um escalar  $\delta$ , este último definido pela expressão de Fletcher and Reeves (1964).

Dessa forma, a primeira direção de busca do método  $q$ -GC é dada pela direção do método  $q$ -G ( $\mathbf{d}^{(1)} = -\nabla_{\mathbf{q}}f(\mathbf{x}^{(0)})$ ) e as outras direções são obtidas por meio da expressão

$$\mathbf{d}^{(k)} = -\nabla_{\mathbf{q}}f(\mathbf{x}^{(k)}) + \delta^{(k)}\mathbf{d}^{(k-1)}, \quad (3.11)$$

onde  $\delta^{(k)}$  é definido através da generalização da expressão de Fletcher and Reeves (1964) na Equação 2.29

$$\delta^{(k)} = \frac{\nabla_{\mathbf{q}}f(\mathbf{x}^{(k)})^T \nabla_{\mathbf{q}}f(\mathbf{x}^{(k)})}{\nabla_{\mathbf{q}}f(\mathbf{x}^{(k-1)})^T \nabla_{\mathbf{q}}f(\mathbf{x}^{(k-1)})}. \quad (3.12)$$

As principais etapas do algoritmo para o método  $q$ -GC com perturbações gaussianas para resolver um problema de otimização com restrições laterais são dadas abaixo (Algoritmo 11). As explicações de cada passo do algoritmo, bem como os critérios de parada, foram definidas na seção anterior. Note que a única alteração do algoritmo do método  $q$ -G com perturbações gaussianas para o algoritmo do método  $q$ -GC com perturbações gaussianas ocorre no Passo 4. O fluxograma apresentado na Figura 3.3 ilustra os passos do algoritmo.

---

**Algoritmo 11:** Método  $q$ -GC com perturbações gaussianas para otimização global e com restrições laterais

---

**Entradas**

- (1)  $f : \mathcal{D} \rightarrow \mathbb{R}$ , onde  $\mathcal{D} = [\ell, u] \subseteq \mathbb{R}^n$  (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  $k = 0$ ,  $\mathbf{x}_{best} = \mathbf{x}^{(0)}$  e  $\theta^{(m)} = \theta^{(0)}$

**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  $q_i^{(k)} x_i^{(k)} \notin [\ell_i, u_i]$ , então faça  $q_i^{(k)} x_i^{(k)}$  como sendo o ponto mais próximo dentro do espaço de busca e atualize o valor de  $q_i^{(k)}$

Fim do Passo 2

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

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

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

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

Fim do Passo 3

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:** Se  $k > 0$ , calcule  $\delta^{(k)}$  por meio da Equação 3.12; senão, faça  $\delta^{(k)} = 0$ . Calcule  $\mathbf{d}^{(k)} = -\nabla_{\mathbf{q}} f(\mathbf{x}^{(k)}) + \delta^{(k)} \mathbf{d}^{(k-1)}$

**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  $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}_{best})$  faça  $\mathbf{x}_{best} = \mathbf{x}^{(k+1)}$

**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  $f(\mathbf{y}^{(k,i)}) < f(\mathbf{x}_{best})$ , atualize  $\mathbf{x}_{best} = \mathbf{y}^{(k,i)}$

Fim do Passo 9

**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  $\mathbf{x}_{best}$  e  $f(\mathbf{x}_{best})$ . Caso contrário, incremente  $k = k + 1$  e vá para o Passo 2

---

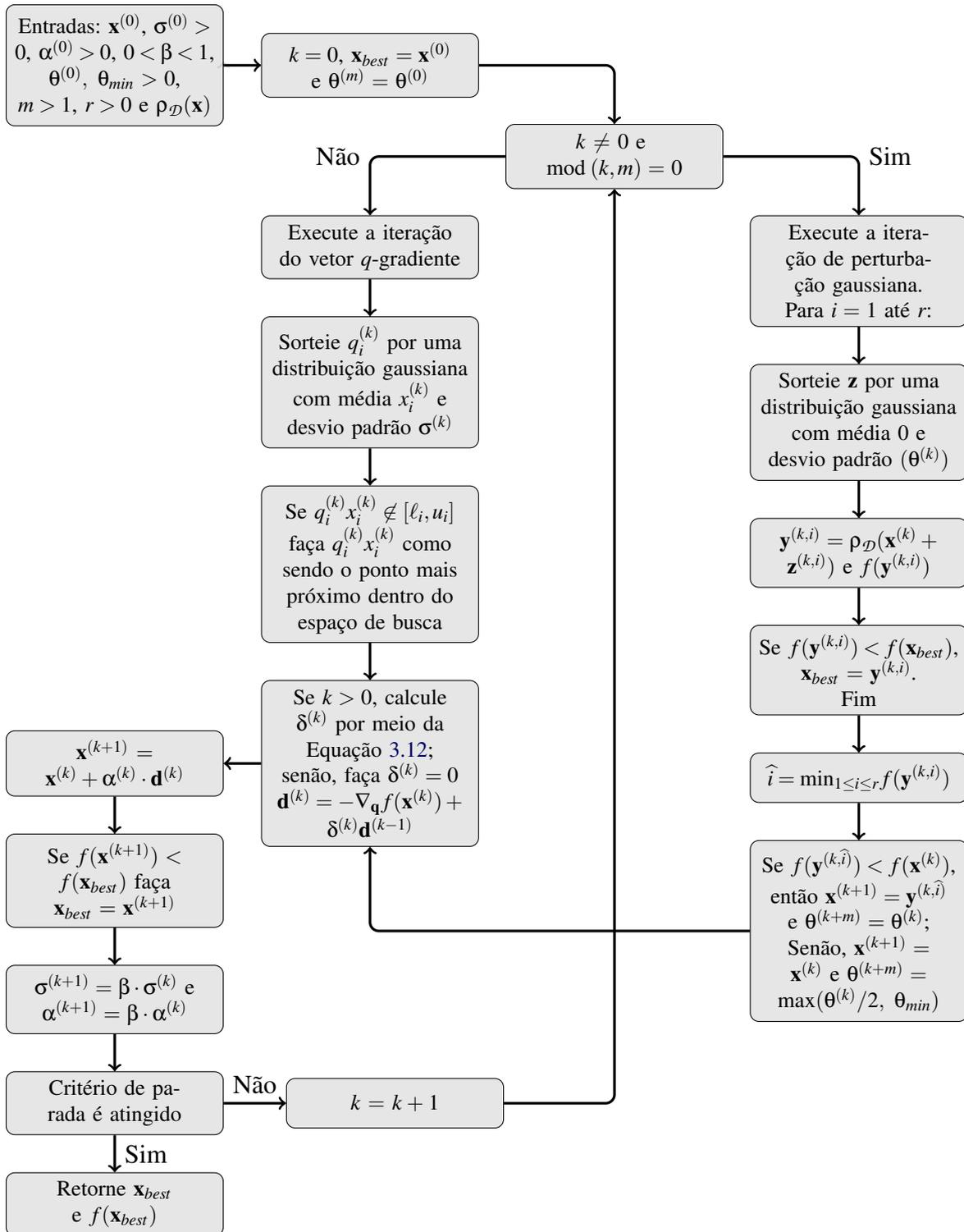


Figura 3.3 - Fluxograma do algoritmo do Método  $q$ -GC com perturbações gaussianas.

### 3.4 Métodos $q$ -BFGS e $q$ -DFP com Perturbações Gaussianas

Os métodos quase-Newton surgem naturalmente no estudo de métodos baseados em gradientes. Aqui, os métodos  $q$ -BFGS e  $q$ -DFP são generalizações dos métodos quase-Newton utilizando a expressão desenvolvida por Broyden-Fletcher-Goldfarb-Shanno (BFGS) e Davidon-Fletcher-Powell (DFP), respectivamente, conforme apresentado na seção 2.5.4.

Os métodos quase-Newton utilizam o procedimento iterativo dado pela equação 2.14, sendo que a direção de busca é dada por uma matriz simétrica definida positiva associada com a direção contrária à direção do vetor gradiente da função objetivo, isto é,  $\mathbf{d}^{(k)} = -\mathbf{B}_k \nabla f(\mathbf{x}^{(k)})$ , onde a matriz  $\mathbf{B}_k$  é atualizada a cada iteração através da Equação 2.34.

Assim, dada uma matriz  $\mathbf{B}_k$  inicial (normalmente  $\mathbf{B}_0 = I_n$ , sendo  $I_n$  a matriz identidade de ordem  $n$  por  $n$ ), a direção de busca dos métodos  $q$ -BFGS e  $q$ -DFP é obtida por meio da substituição do vetor gradiente clássico pelo vetor  $q$ -gradiente, ou seja

$$\mathbf{d}^{(k)} = -\mathbf{B}_k \nabla_{\mathbf{q}} f(\mathbf{x}^{(k)}), \quad (3.13)$$

sendo  $\mathbf{B}_k$  atualizada a cada iteração  $k$  por meio da expressão

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\mathbf{r}^{(k)}(\mathbf{r}^{(k)})^\top}{(\mathbf{r}^{(k)})^\top \mathbf{s}^{(k)}} - \frac{\mathbf{B}_k \mathbf{s}^{(k)}(\mathbf{B}_k \mathbf{s}^{(k)})^\top}{(\mathbf{s}^{(k)})^\top \mathbf{B}_k \mathbf{s}^{(k)}} + c(\mathbf{s}^{(k)})^\top \mathbf{B}_k \mathbf{s}^{(k)} \mathbf{v}^{(k)}(\mathbf{v}^{(k)})^\top,$$

onde  $\mathbf{r}^{(k)}$  e  $\mathbf{v}^{(k)}$  são obtidos através das Equação 2.32 e 2.35, respectivamente, e  $\mathbf{s}^{(k)}$  é obtido por

$$\mathbf{s}^{(k)} = \nabla_{\mathbf{q}} f(\mathbf{x}^{(k+1)}) - \nabla_{\mathbf{q}} f(\mathbf{x}^{(k)}). \quad (3.14)$$

Se  $c = 0$  na Equação 3.4, tem-se a expressão para o método  $q$ -DFP. Já para  $c = 1$ , tem-se a expressão para o método  $q$ -BFGS.

O algoritmo para os métodos  $q$ -BFGS e  $q$ -DFP com perturbações gaussianas para resolver um problema de otimização com restrições laterais é dado no Algoritmo 12. As explicações de cada passo do algoritmo, bem como os critérios de parada, foram definidas na seção 3.2. Note que a única alteração do algoritmo do método  $q$ -G com perturbações gaussianas para o algoritmo dos métodos  $q$ -BFGS e  $q$ -DFP com perturbações gaussianas ocorre no Passo 4. O fluxograma apresentado na Figura 3.4 ilustra os passos do algoritmo.

---

**Algoritmo 12:** Métodos  $q$ -BFGS e  $q$ -DFP com perturbações gaussianas para otimização global e com restrições laterais

---

**Entradas**

- (1)  $f : \mathcal{D} \rightarrow \mathbb{R}$ , onde  $\mathcal{D} = [\ell, u] \subseteq \mathbb{R}^n$  (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  $k = 0$ ,  $\mathbf{x}_{best} = \mathbf{x}^{(0)}$  e  $\theta^{(m)} = \theta^{(0)}$

**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  $q_i^{(k)} x_i^{(k)} \notin [\ell_i, u_i]$ , então faça  $q_i^{(k)} x_i^{(k)}$  como sendo o ponto mais próximo dentro do espaço de busca e atualize o valor de  $q_i^{(k)}$

Fim do Passo 2

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

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

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

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

Fim do Passo 3

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:** Se  $k > 0$ , faça  $\mathbf{B}_k$  por meio da Eq. 3.4; senão, faça  $\mathbf{B}_k = I_n$  sendo  $I_n$  a matriz identidade de ordem  $n$  por  $n$ . Para  $c = 0$ , tem-se o método  $q$ -DFP e para  $c = 1$ , tem-se o método  $q$ -BFGS.

Calcule  $\mathbf{d}^{(k)} = -\mathbf{B}_k \nabla_{\mathbf{q}} f(\mathbf{x}^{(k)})$

**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  $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}_{best})$  faça  $\mathbf{x}_{best} = \mathbf{x}^{(k+1)}$

**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  $f(\mathbf{y}^{(k,i)}) < f(\mathbf{x}_{best})$ , atualize  $\mathbf{x}_{best} = \mathbf{y}^{(k,i)}$

Fim do Passo 9

**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  $\mathbf{x}_{best}$  e  $f(\mathbf{x}_{best})$ . Caso contrário, incremente  $k = k + 1$  e vá para o Passo 2

---

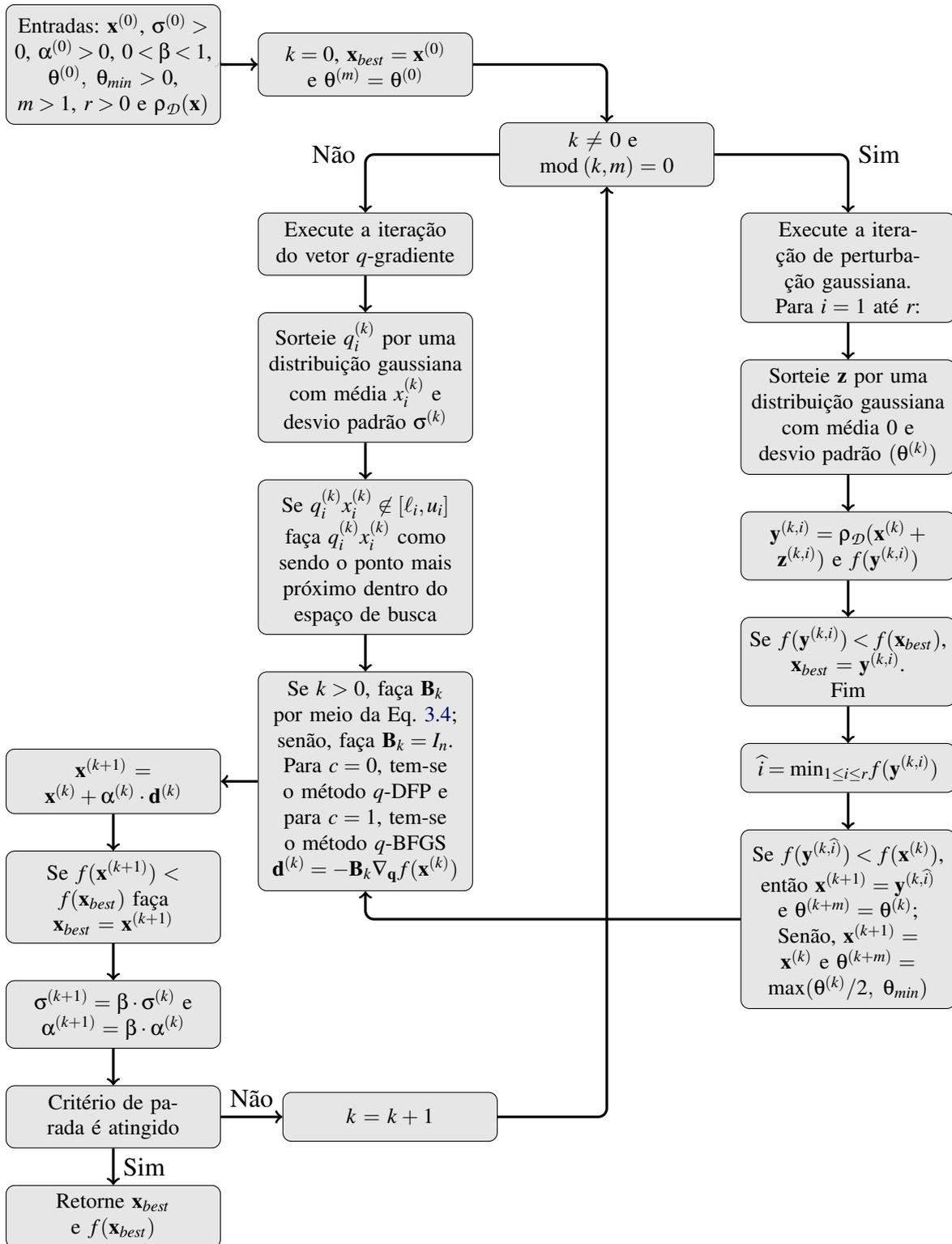


Figura 3.4 - Fluxograma do algoritmo dos Métodos  $q$ -BFGS e  $q$ -DFP com perturbações gaussianas.

É importante ressaltar que ambas  $q$ -versões retomam as suas respectivas versões clássicas

quando os parâmetros  $q_i$ 's ( $i = 1, \dots, n$ ) tendem a 1. Em outras palavras, quando  $\mathbf{q} \rightarrow \mathbf{1}$ , o método  $q$ -G com perturbações gaussianas retorna ao método da máxima descida, o método  $q$ -GC com perturbações gaussianas retorna ao método dos gradientes conjugados de Fletcher e Reeves e os métodos  $q$ -BFGS e  $q$ -DFP com perturbações gaussianas retornam aos métodos quase-Newton BFGS e DFP, respectivamente.

O próximo capítulo apresenta a prova de convergência dos métodos baseados no vetor  $q$ -gradiente que usam perturbações gaussianas.

## 4 ANÁLISE DE CONVERGÊNCIA

Neste capítulo são apresentadas as provas de convergência para os métodos de otimização global baseados no vetor  $q$ -gradiente apresentados anteriormente.

### 4.1 Notações Preliminares

Seja  $f$  uma função objetivo determinística definida em um conjunto  $\mathcal{D} \subseteq \mathbb{R}^n$ . O intuito é encontrar o mínimo global de  $f$  sobre  $\mathcal{D}$ , caso ele exista. Para isso, vamos nos concentrar no caso onde  $\mathcal{D} = [\ell, u] \subseteq \mathbb{R}^n$  é uma região com restrições laterais. É sabido que se  $f$  é contínua sobre  $\mathcal{D}$ , então  $f$  possui um ponto de mínimo global em  $\mathcal{D}$ .

Uma vez que os métodos de otimização baseados em  $q$ -gradientes são estocásticos e a função objetivo  $f$  é determinística, a convergência desses métodos será estabelecida com base na estrutura de Busca Aleatória Adaptativa Generalizada, do inglês, *Generalized Adaptive Random Search* (GARS), desenvolvida por Regis (2010), a qual é descrita abaixo. As iterações dos métodos baseados em  $q$ -gradiente serão tratadas como vetores  $n$ -dimensionais cujos pontos estão em  $\mathcal{D} \subseteq \mathbb{R}^n$ .

Considere um algoritmo estocástico no qual as iterações são obtidas pela sequência de vetores aleatórios  $\{Y^{(k)}\}_{k \geq 1}$  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 de subconjuntos de  $\Omega$ , e  $\mathcal{B}(\mathcal{D})$  são os conjuntos de Borel em  $\mathcal{D}$ <sup>1</sup>. Este algoritmo é dito seguir o framework GARS se obedecer a seguinte estrutura:

ALGORITMO (Framework GARS (REGIS, 2010))

#### Entradas

- (1) A 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,  $\rho_{\mathcal{D}}(x) = x$

---

<sup>1</sup>A dupla  $(\Omega, \mathbb{F})$  se chama espaço mensurável se  $\Omega \neq \emptyset$  e  $\mathbb{F}$  uma  $\sigma$ -álgebra (em  $\Omega$ ), isto é, uma coleção de subconjuntos de  $\Omega$  que satisfazem às propriedades seguintes:

- a)  $\Omega \in \mathbb{F}$
- b) Se  $A \in \mathbb{F}$ , então  $A^c := \Omega \setminus A \in \mathbb{F}$
- c) Se  $A_i \in \mathbb{F}, i = 1, \dots, \infty$ , então  $\bigcup_{i=1}^{\infty} A_i \in \mathbb{F}$

Os elementos de  $\mathbb{F}$  são chamados de conjuntos mensuráveis. A menor  $\sigma$ -álgebra sobre  $\mathbb{R}$  que contém todos os intervalos ou conjuntos gerados a partir de uniões ou complementos constituem a  $\sigma$ -álgebra de Borel ( $\mathcal{B}$ ). Os elementos de  $\mathcal{B}$  são os conjuntos de Borel (SOLANO, 2014).

para todo  $x \in \mathcal{D}$ .

- (3) Uma coleção de elementos aleatórios intermediários  $\{\Lambda_{i,j} : (\Omega, \mathcal{B}) \rightarrow (\Omega_{i,j}, \mathcal{B}_{i,j}) : i \geq 0 \text{ e } j = 0, 1, \dots, r_i\}$  que são usados para determinar o experimento na iteração dos vetores aleatórios. 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 os vetores aleatórios  $Y^{(k)} : (\Omega, \mathcal{B}) \rightarrow (\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$  como segue:

**Passo 1.1** Para cada  $j = 0, \dots, r_k$ , gere os elementos aleatórios intermediários  $\Lambda_{k,j} : (\Omega, \mathcal{B}) \rightarrow (\Omega_{k,j}, \mathcal{B}_{k,j})$  de acordo com alguma distribuição de probabilidade.

**Passo 1.2** 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 o conjunto 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.

Regis (2010) provou diversos teoremas que fornecem condições necessárias para garantir a convergência de um algoritmo que segue o framework GARS para o mínimo global em um sentido probabilístico. Dessa forma, é utilizado um desses teoremas para provar a convergência dos métodos baseados no vetor  $q$ -gradiente que usam perturbações gaussianas. O teorema aplicado aos algoritmos GARS, que utilizamos neste trabalho, usa uma distribuição elíptica, que nada mais é que uma generalização da distribuição gaussiana.

Seja  $Z : (\Omega, \mathcal{B}) \rightarrow (\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$  um vetor aleatório com uma distribuição elíptica. Se  $Z$  tem uma densidade, então ela tem a forma (FANG; ZHANG, 1990)

$$g(z) = \gamma [\det(V)]^{-1/2} \Psi((z - u)^T V^{-1}(z - u)), \quad z \in \mathbb{R}^n \quad (4.1)$$

onde  $u \in \mathbb{R}^n$ ,  $V$  é uma matriz definida positiva e simétrica,  $\Psi$  é uma função não negativa sobre os reais positivos, tais que  $\int_0^\infty z^{(n/2)-1} \Psi(z) dz < \infty$ , e  $\gamma$  é uma constante de normalização dada por

$$\gamma = \frac{1}{2} \pi^{-n/2} \Gamma(n/2) \left( \int_0^\infty z^{n-1} \Psi(z^2) dz \right)^{-1}. \quad (4.2)$$

As distribuições elípticas incluem algumas das mais importantes distribuições usadas em algoritmos estocásticos. Por exemplo, se  $\Psi(y) = e^{-y/2}$  na definição acima, tem-se a distribuição gaussiana multivariada. Já se  $\Psi(y) = (1+y)^{-\binom{d+1}{2}}$ , tem-se a distribuição de Cauchy multivariada (REGIS, 2010). A seguinte proposição mostra que os algoritmos que seguem o framework GARS e usam distribuições elípticas, sendo  $\Psi$  monótona não-crescente e os autovalores de  $V$  delimitados longe do 0, convergem para o mínimo global da função quase certamente (q.c.)<sup>2</sup>

**Proposição 11.** (REGIS, 2010) *Seja  $\mathcal{D}$  um subconjunto limitado de  $\mathbb{R}^n$  tal que  $\Psi_{\mathcal{D}}(\delta) := \inf_{w \in \mathcal{D}} \mu(B(w, \delta) \cap \mathcal{D}) > 0$  para todo  $\delta > 0$ , onde  $B(w, \delta)$  é a bola aberta centrada em  $w$  com raio  $\delta$  e  $\mu$  é a medida de Lebesgue<sup>3</sup> em  $\mathbb{R}^n$ . Seja  $f$  uma função real definida em  $\mathcal{D}$  tal que  $f^* := \inf_{x \in \mathcal{D}} f(x) > -\infty$  e assumamos que  $f$  é contínua em um minimizador global  $x^*$  de  $f$  sobre  $\mathcal{D}$ . Considere um algoritmo GARS cujas iterações são  $\{Y^{(k)} : k \geq 0\}$  e cuja sequência de melhores iterações são  $\{Y^{(k)*} : k \geq 0\}$ . Suponha que haja uma subsequência  $\{k_t\}_{t \geq 1}$  tal que para cada  $t \geq 1$ , temos  $Y^{(k_t)} = U^{(t)} + Z^{(t)}$ , 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 dada  $\sigma(\mathcal{E}_{(k_t)-1})$  é uma distribuição elíptica com densidade condicional dada por*

$$g_t(z | \sigma(\mathcal{E}_{(k_t)-1})) = \gamma [\det(V_t)]^{-1/2} \Psi(z^T V_t^{-1} z), \quad z \in \mathbb{R}^n, \quad (4.3)$$

onde  $\gamma$  é definido em (4.2). Para cada  $t \geq 1$ , seja  $\lambda_t$  o menor autovalor de  $V_t$ . Além disso, suponha que as seguintes propriedades são válidas:

[P1]  $\Psi$  é monótona não-crescente; e

[P2]  $\inf_{t \geq 1} \lambda_t > 0$ .

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

<sup>2</sup>Sejam  $\{X^{(k)}\}_{k \geq 1}$  uma sequência de variáveis aleatórias e  $X^{(k)*}$  uma variável aleatória definida no mesmo espaço de probabilidade. Diz-se que  $X^{(k)}$  converge quase certamente para  $X^{(k)*}$ , isto é  $X^{(k)} \rightarrow X^{(k)*}$ , se (SOONG, 1973)

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

<sup>3</sup>Um conjunto  $E \subset \mathbb{R}^n$  é dito ser Lebesgue mensurável se, para cada  $\varepsilon > 0$ , existir um conjunto aberto  $U \subset \mathbb{R}^n$  contendo  $E$  tal que  $m^*(U \setminus E) \leq \varepsilon$ . Se  $E$  é Lebesgue mensurável, então refere-se a  $m(E) := m^*(E)$  como a medida de Lebesgue de  $E$  (TAO, 2011).

## 4.2 Prova de Convergência

A proposição seguinte apresenta a prova de convergência dos métodos baseados no vetor  $q$ -gradiente que usam perturbações gaussianas.

**Proposição 12.** *Suponha que os métodos  $q$ -G,  $q$ -GC,  $q$ -BFGS ou  $q$ -DFP com perturbações gaussianas seja aplicado a uma função de valor real  $f$  em  $\mathcal{D} = [\ell, u] \subseteq \mathbb{R}^n$  tal que  $f^* = \inf_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x}) > -\infty$ . Além disso, suponha que  $f$  é contínua no minimizador global  $\mathbf{x}^*$  de  $f$  sobre  $\mathcal{D}$ . Então  $f(\mathbf{x}_{best}) \rightarrow f^*$  quase certamente (q.c.).*

*Proof.* Usamos a mesma notação nas descrições dos algoritmos  $q$ -G,  $q$ -GC,  $q$ -BFGS e  $q$ -DFP com perturbações gaussianas exceto para alguns vetores, no qual utilizamos em letras maiúscula para enfatizar que eles são vetores aleatórios (por exemplo,  $Y^{(k,i)}$  ao invés de  $\mathbf{y}^{(k,i)}$ ). Por conveniência, definimos  $Y^{(k,0)} = X^{(k)}$  para cada  $k \geq 0$ . Ainda, seja  $r_k = n$  para  $k = 0$  e todo  $k$  tal que  $\text{mod}(k, m) \neq 0$  (iterações que usam  $q$ -gradiente); senão, seja  $r_k = r$  (iterações gaussianas). Note que  $\{Y^{(k,i)} : k \geq 0, i = 0, 1, \dots, r_k\}$  é o conjunto de todos os pontos onde a função objetivo é avaliada se os algoritmos são executados infinitamente. Aqui,  $k$  é o número de iteração e  $i$  representa o índice de um ponto da avaliação da função dentro de cada iteração. Além disso, seja  $\mathcal{E}_{k,i}$  o conjunto de todos os vetores aleatórios que foram gerados até  $Y^{(k,i)}$  (isto inclui os vetores aleatórios  $\mathbf{q}^{(j)}$ ,  $j \leq k$  e os vetores aleatórios gaussianos anteriores). Seja  $\{k_t\}_{t \geq 1}$  a subsequência das iterações gaussianas, isto é,  $k_t = tm$  para todo  $t \geq 1$ . A partir da descrição do algoritmo,

$$Y^{(k_t,i)} = Y^{(k_t,0)} + Z^{(k_t,i)}, \quad i = 1, \dots, r, \quad (4.4)$$

onde  $Z^{(k_t,i)}$  é o vetor aleatório cuja distribuição condicional dada  $\sigma(\mathcal{E}_{k_t,i-1})$  ( $\sigma$ -álgebra gerado pelos vetores aleatórios em  $\mathcal{E}_{k_t,i-1}$ ) é uma distribuição gaussiana com média no vetor  $0_{n \times 1}$  e matriz de covariância  $V_t = \theta_{k_t}^2 I_n$ . Consequentemente,  $Z^{(k_t,i)}$  tem uma densidade condicional

$$g_t(z | \sigma(\mathcal{E}_{k_t,i-1})) = (2\pi)^{-n/2} [\det(V_t)]^{-1/2} \exp(-z^T V_t^{-1} z / 2), \quad \forall t \geq 1, i = 1, \dots, r.$$

Na Equação 4.4, note que  $Y^{(k_t,0)} = \Phi_t(\mathcal{E}_{k_t,i-1})$  para alguma função determinística  $\Phi_t$ . Além disso, note que  $Z^{(k_t,i)}$  tem uma distribuição elíptica onde  $\Psi(w) = e^{-w/2}$  em 4.1 é uma função monótona não-crescente. Por fim, note que  $\theta_{k_t}^2$  é o único autovalor de  $V_t$  (e possui multiplicidade  $n$ ). Desde que  $\inf_{t \geq 1} \theta_{k_t} \geq \theta_{min} > 0$ , segue da Proposição 6 (que representa o Teorema 6 em Regis (2010)) que  $f(\mathbf{x}_{best}) \rightarrow f^*$  q.c.

## 5 ANÁLISE DOS RESULTADOS

Os principais resultados e análises dos experimentos numéricos dos métodos baseados no vetor  $q$ -gradiente aqui propostos são apresentados a seguir. O desempenho dos métodos é avaliado em funções teste com 10 dimensões, escolhidas do repertório de problemas de otimização contínua, não linear e com restrições disponível na literatura especializada. São consideradas funções teste unimodais e multimodais com o intuito de comprovar a aplicabilidade e eficiência dos métodos quando aplicados a funções com diferentes aspectos. As comparações de desempenho são realizadas com outros nove algoritmos de otimização. Além de suas versões clássicas, os métodos baseados no vetor  $q$ -gradiente são também comparados com métodos de otimização determinísticos, métodos estocásticos e métodos livres de derivadas.

Os algoritmos para os métodos baseados no vetor  $q$ -gradiente, assim como suas versões clássicas, foram implementados em Fortran 90 para o compilador IFORT (Intel Fortran Compiler Professional Edition) versão 10.1.018 e executados em um laptop com processador Intel(R) Core(TM) i5-2410M 2.30GHz com 6GB RAM e sistema operacional Ubuntu Linux 12.04.4 64-bit. Os outros métodos foram implementados em Matlab R2013b e executados em um desktop Intel(R) Core(TM) i7-4770 CPU 3.4GHz 3.00GHz com 16GB RAM e sistema operacional Windows XP 64-bit.

### 5.1 Descrição dos Experimentos Computacionais

O desempenho dos métodos baseados no vetor  $q$ -gradiente é avaliado sobre 27 funções teste com 10 dimensões, sendo 14 funções do *Special Session on Real-Parameter Optimization do IEEE Congress on Evolutionary Computation 2005* (CEC-2005) definidas em Suganthan et al. (2005), 5 funções multimodais que são bem conhecidas na literatura e 8 funções unimodais definidas em Moré et al. (1981). No geral, foram utilizadas 14 funções multimodais e 13 funções unimodais. A Tabela 5.1 resume as características dessas funções teste.

Tabela 5.1 - Características das funções teste.

Função teste	Domínio	Valor do mínimo global
Ackley	$[-30, 30]^{10}$	0
Griewank	$[-600, 600]^{10}$	0
Levy	$[-5, 5]^{10}$	$< -11$
Michalewicz	$[0, \pi]^{10}$	-9.66
Rastrigin	$[-5, 5]^{10}$	0
F6: Rosenbrock Deslocada	$[-100, 100]^{10}$	390
F7: Griewank Rotacionada e Deslocada	$[0, 600]^{10}$	-180
F8: Ackley Rotacionada e Deslocada com Ótimo Global na Fronteira	$[-32, 32]^{10}$	-140
F9: Rastrigin Deslocada	$[-5, 5]^{10}$	-330
F10: Rastrigin Rotacionada e Deslocada	$[-5, 5]^{10}$	-330
F11: Weierstrass Rotacionada e Deslocada	$[-0.5, 0.5]^{10}$	90
F12: Schwefel Problema 2.13	$[-\pi, \pi]^{10}$	-460
F13: Griewank + Rosenbrock Deslocada e Expandida	$[-3, 1]^{10}$	-130
F14: Scaffer Rotacionada, Deslocada e Expandida	$[-100, 100]^{10}$	-300
Brown	$[-2, 2]^{10}$	0
Broyden Banded	$[-1, 1]^{10}$	0
Broyden Tridiagonal	$[-1, 1]^{10}$	0
Discrete Boundary	$[-3, 3]^{10}$	0
Discrete Integral	$[-1, 3]^{10}$	0
Penalty I	$[-1, 3]^{10}$	0
Rosenbrock	$[-2, 048, 2, 048]^{10}$	0
Trigonometric	$[-1, 3]^{10}$	0
F1: Sphere Deslocada	$[-100, 100]^{10}$	-450
F2: Schwefel Deslocada Problema 1.2	$[-100, 100]^{10}$	-450
F3: Elliptic Deslocada, Rotacionada Alta Condicionada	$[-100, 100]^{10}$	-450
F4: Schwefel Problema 1.2 Deslocada com Ruído	$[-100, 100]^{10}$	-450
F5: Schwefel Problem 2.6 com Ótimo Global na Fronteira	$[-100, 100]^{10}$	-310

Os métodos  $q$ -G,  $q$ -GC,  $q$ -BFGS e  $q$ -DFP com prova de convergência e perturbações gaussianas são comparados com nove algoritmos de otimização: máxima descida (MD), gradientes conjugados não linear (GC) (FLETCHER; REEVES, 1964), métodos quase-Newton BFGS e DFP, um algoritmo de ponto interior (IPOPT) (WÄCHTER; BIEGLER, 2006), uma estratégia evolutiva com matriz de covariância adaptada (CMA-ES) (HANSEN; MLADENOVIC, 1997), um algoritmo de busca aleatória controlada com mutação local (CRS2-LM) (KAELO, 2006), outra estratégia evolutiva (ISRES) (RUNARSSON; YAO, 2005) e o método de busca direta de Nelder-Mead (NELDER; MEAD, 1965). Os algoritmos IPOPT, ISRES, CMA-ES, CRS2-LM, e Nelder-Mead foram executados através do *toolbox* OPTI (CURRIE; WILSON, 2012) com os parâmetros de ajuste desse pacote e o software NLOpt (JOHNSON, 2008).

São realizadas 30 execuções independentes de todos os algoritmos para todas as funções teste e o critério de parada é um número de avaliações da função objetivo fixado em 10.000. Para garantir uma comparação equitativa, os diferentes algoritmos utilizam o mesmo conjunto de pontos iniciais. Além disso, para que os métodos de busca local (MD, GC, BFGS, DFP, IPOPT e Nelder-Mead) tenham o mesmo potencial de encontrar o mínimo global da função, eles são reinicializados a partir de um ponto aleatório com distribuição uniforme dentro do espaço de busca cada vez que convergem, desde que o tempo computacional não tenha sido esgotado.

Para fins de comparação, as definições dos parâmetros devem ser fixos com valores no mínimo razoáveis para todas as funções teste, como recomendado por Barr et al. (1995). Os métodos propostos nesta tese possuem 4 parâmetros de ajuste: desvio-padrão inicial ( $\sigma^{(0)}$ ) que determina o quanto a busca é global ou local, tamanho do passo inicial ( $\alpha^{(0)}$ ), fator de redução ( $\beta$ ) que controla a velocidade de transição entre busca global e busca local, sendo utilizado tanto para reduzir o parâmetro  $\sigma$  quanto para  $\alpha$  (SOTERRONI et al., 2015) e desvio-padrão da perturbação gaussiana ( $\theta$ ), utilizado para garantir a convergência em probabilidade dos métodos. Após rodadas preliminares, esses parâmetros foram ajustados da seguinte maneira:  $\sigma^{(0)} = 0,2 \times L$  (para os métodos  $q$ -G e  $q$ -GC) e  $\sigma^{(0)} = 0,05 \times L$  (para os métodos  $q$ -BFGS e  $q$ -DFP),  $\alpha^{(0)} = 0,1 \times L$ ,  $\beta = 0,999$ ,  $\theta = 0,2 \times L$  utilizado a cada  $n$  avaliações da função objetivo, onde  $n$  é dimensão do problema (no caso  $n = 10$ ) e  $L$  a maior distância dentro do espaço de busca definido por  $L = \sqrt{\sum_{i=1}^n (\mathbf{x}_{sup_i} - \mathbf{x}_{inf_i})^2}$ . Na Seção 5.4, é realizada uma análise de sensibilidade do desempenho dos métodos baseados no vetor  $q$ -gradiente no que diz respeito a  $\sigma^{(0)}$ . Para os outros nove algoritmos utilizados para fins de comparação são utilizados os valores de parâmetros definidos na literatura.

Os algoritmos são comparados de duas maneiras. Em primeiro lugar, os algoritmos são

comparados em termos de melhores valores da função objetivo obtidos após 10.000 avaliações da função. Em seguida, os algoritmos são comparados usando perfis de dados (MORÉ; WILD, 2009), que são particularmente adequados quando as avaliações das funções são computacionalmente caras. Essas comparações são discutidas nas próximas duas seções.

## 5.2 Estatísticas sobre a Comparação dos Algoritmos

As Tabelas 5.2 e 5.3 apresentam o mínimo (melhor), o máximo (pior), a mediana, a média e o erro padrão dos valores ótimos obtidos após 10.000 avaliações das funções teste, considerando 30 execuções independentes tanto para as funções multimodais quanto para as funções unimodais. Os melhores valores de cada estatística estão realçados. As estatísticas foram calculadas para 11 algoritmos de otimização, excluindo os métodos ISRES e Nelder-Mead.

A Tabela 5.2 mostra que os métodos baseados em  $q$ -gradiente são melhores do que suas versões clássicas para as funções multimodais. Em particular, os métodos  $q$ -G e  $q$ -GC são melhores ou iguais à suas respectivas versões clássicas, MD e GC em 12 das 14 funções multimodais (com exceção das funções: Griewank e F07). Similarmente, o método  $q$ -BFGS é melhor ou igual ao método BFGS em 11 dessas funções (com exceção das funções: Griewank, F06, F07), enquanto o método  $q$ -DFP é melhor quando comparado ao método DFP em 10 das funções multimodais utilizadas neste trabalho (com exceção das funções: Griewank, F06, F07, F12). Além disso, em geral, os métodos baseados no vetor  $q$ -gradiente são melhores do que o método IPOPT e são competitivos com os métodos CMA-ES e CRS2-LM para os problemas multimodais.

Para as funções unimodais, a Tabela 5.3 mostra que o método IPOPT possui o melhor desempenho; isto não é surpresa, uma vez que este é um método baseado em gradientes especialmente desenvolvido para otimização local. Como esperado, os métodos baseados no vetor  $q$ -gradiente não possuem um desempenho tão bom quando comparado com suas versões clássicas em muitas das funções unimodais, uma vez que são destinados principalmente a problemas de otimização global. Já para as funções unimodais do CEC-2005, os métodos  $q$ -G e  $q$ -GC são, em geral, melhores do que os métodos  $q$ -BFGS,  $q$ -DFP, IPOPT e suas versões clássicas. Além disso, os métodos  $q$ -G e  $q$ -GC são competitivos com os métodos CMA-ES e CRS2-LM, com exceção da F03.

Tabela 5.2 - Estatística sobre o melhor valor da função objetivo sobre 30 execuções dos algoritmos para as funções teste multimodais após 10.000 avaliações da função.

Função	MD	GC	BFGS	DPP	q-G	q-GC	q-BFGS	q-DPP	IPOPT	CMA-ES	CRS2-LM	
Ackley	mínimo	1,84E+01	1,84E+01	1,84E+01	1,03E-04	3,59E-05	3,86E-07	2,19E-01	1,17E-04	1,53E-10	2,02E-06	
	máximo	1,98E+01	1,98E+01	1,98E+01	5,35E-04	3,59E-04	1,64E+00	3,47E+00	5,39E-04	1,99E+01	7,80E-04	
	mediano	1,94E+01	1,94E+01	1,94E+01	3,20E-04	2,34E-04	1,19E-04	2,64E+00	1,92E+01	2,42E-10	7,93E-06	
	média	1,93E+01	1,94E+01	1,94E+01	3,19E-04	2,35E-04	5,50E-02	2,36E+00	1,90E+01	1,33E+00	4,52E-05	
	erro padrão	6,33E-02	6,33E-02	6,32E-02	6,31E-02	2,65E-05	1,35E-05	5,49E-02	1,68E-01	9,50E-02	9,22E-01	2,60E-05
Griewank	mínimo	-2,22E-16	-2,22E-16	6,66E-15	9,86E-03	2,58E-02	1,08E-06	1,01E-02	4,08E-12	3,33E-16	3,75E-05	
	máximo	7,08E+00	4,96E+00	4,96E+00	2,73E-01	8,87E-01	7,37E-02	1,19E+00	3,20E-02	4,18E-02	6,63E-01	
	mediano	1,11E-02	3,45E-02	1,23E-02	2,24E-01	1,14E-01	2,82E-01	4,64E-01	9,86E-03	7,39E-03	8,39E-02	
	média	6,51E-01	4,39E-01	5,32E-01	6,92E-01	1,07E-01	3,19E-01	2,57E-02	5,44E-01	9,52E-03	2,03E-01	
	erro padrão	2,89E-01	1,85E-01	1,96E-01	2,12E-01	1,02E-02	2,90E-02	3,47E-03	5,51E-02	1,87E-03	3,86E-02	
Levy	mínimo	-2,12E+01	-3,30E+01	-2,06E+01	-1,71E+01	-1,15E+01	-1,15E+01	-1,15E+01	-1,15E+01	-1,15E+01	-1,15E+01	
	máximo	8,22E+01	1,29E+02	9,03E+01	1,35E+02	7,60E+01	-1,14E+01	-1,09E+01	1,35E+01	-1,02E+01	-1,15E+01	
	mediano	2,21E+00	-1,82E+00	-2,31E+00	4,47E+00	-1,15E+01	-1,15E+01	-1,13E+01	-5,40E+00	-1,15E+01	-1,15E+01	
	média	1,38E+01	2,72E+00	5,44E+00	1,76E+01	-1,14E+01	-1,15E+01	-1,13E+01	-4,98E+00	-1,13E+01	-1,15E+01	
	erro padrão	6,00E+00	6,13E+00	3,99E+00	5,84E+00	1,33E-02	8,00E-03	9,96E-03	2,50E-02	1,01E+00	6,09E-02	2,35E-05
Mich	mínimo	-6,43E+00	-6,47E+00	-5,82E+00	-5,75E+00	-9,42E+00	-9,27E+00	-8,72E+00	-9,23E+00	-8,95E+00	-9,55E+00	
	máximo	-1,35E+00	-8,97E-01	-1,72E+00	-1,71E+00	-6,49E+00	-7,14E+00	-6,14E+00	-6,44E+00	-6,44E+00	-5,61E+00	-5,72E+00
	mediano	-4,14E+00	-4,41E+00	-4,09E+00	-3,95E+00	-8,06E+00	-8,43E+00	-7,69E+00	-7,07E+00	-7,77E+00	-7,40E+00	-8,72E+00
	média	-4,19E+00	-4,18E+00	-4,12E+00	-3,75E+00	-8,13E+00	-8,28E+00	-7,62E+00	-7,01E+00	-7,80E+00	-7,49E+00	-8,53E+00
	erro padrão	2,56E-01	2,51E-01	1,96E-01	2,12E-01	1,09E-01	9,59E-02	1,34E-01	1,38E-01	1,33E-01	1,51E-01	1,63E-01
Rastrigin	mínimo	4,48E+01	4,48E+01	4,48E+01	4,48E+01	3,97E+00	7,95E+00	6,82E+00	2,49E+01	5,97E+00	9,95E-01	
	máximo	1,26E+02	1,27E+02	1,59E+02	1,26E+02	2,08E+01	4,09E+01	5,51E+01	5,97E+01	4,73E+01	3,09E+01	
	mediano	8,61E+01	8,61E+01	8,61E+01	8,61E+01	7,46E+00	2,19E+01	3,29E+01	4,43E+01	1,49E+01	7,25E+00	
	média	8,50E+01	8,41E+01	8,61E+01	8,45E+01	8,85E+00	2,35E+01	3,27E+01	4,30E+01	1,77E+01	9,41E+00	
	erro padrão	4,57E+00	4,36E+00	5,03E+00	4,41E+00	7,75E-01	8,60E-01	1,62E+00	1,94E+00	1,74E+00	1,37E+00	
F06	mínimo	3,90E+02	1,31E+03	3,90E+02	3,90E+02	3,90E+02	4,59E+02	4,06E+03	3,90E+02	3,90E+02	3,90E+02	
	máximo	7,73E+03	1,76E+05	3,94E+03	6,15E+02	1,06E+03	6,49E+04	1,17E+05	3,95E+02	4,03E+02	3,94E+02	
	mediano	3,90E+02	2,67E+04	3,90E+02	3,90E+02	3,90E+02	1,43E+03	2,86E+04	3,90E+02	3,90E+02	3,90E+02	
	média	9,49E+02	5,14E+04	3,90E+02	3,99E+02	4,29E+02	5,25E+03	3,29E+04	3,91E+02	3,91E+02	3,91E+02	

Função	MD	GC	BFGS	DFP	q-G	q-GC	q-BFGS	q-DFP	IPOPT	CMA-ES	CRS2-LM
	erro padrão	2,80E+02	1,07E+04	2,76E+01	7,47E+00	2,54E+01	7,31E+01	2,22E+03	2,92E-01	5,33E-01	2,52E-01
F07	mínimo	-8,19E+01	-5,53E+01	-5,53E+01	1,08E+03	1,08E+03	1,08E+03	1,09E+03	1,09E+03	1,09E+03	1,09E+03
	máximo	-2,65E+01	2,44E+02	2,44E+02	1,08E+03	1,08E+03	1,09E+03	1,09E+03	1,09E+03	1,09E+03	1,09E+03
	mediano	-4,35E+01	6,65E+01	5,94E+01	1,08E+03	1,08E+03	1,09E+03	1,09E+03	1,09E+03	1,09E+03	1,09E+03
	média	-4,70E+01	8,51E+01	8,39E+01	1,08E+03	1,08E+03	1,09E+03	1,09E+03	1,09E+03	1,09E+03	1,09E+03
	erro padrão	2,70E+00	1,44E+01	1,45E+01	2,06E-03	7,04E-05	2,90E-02	3,29E-02	1,26E-13	8,29E-03	2,48E-02
F08	mínimo	-1,20E+02	-1,19E+02								
	máximo	-1,20E+02	-1,20E+02	-1,20E+02	-1,20E+02	-1,20E+02	-1,20E+02	-1,19E+02	-1,20E+02	-1,19E+02	-1,19E+02
	mediano	-1,20E+02	-1,19E+02	-1,19E+02							
	média	-1,20E+02	-1,19E+02	-1,19E+02							
	erro padrão	1,49E-05	2,95E-03	0,00E+00	1,65E-02	1,55E-02	1,43E-02	2,02E-02	0,00E+00	1,93E-02	1,63E-02
F09	mínimo	-2,90E+02	-2,91E+02	-2,90E+02	-2,90E+02	-3,27E+02	-3,21E+02	-3,25E+02	-3,15E+02	-3,26E+02	-3,26E+02
	máximo	-1,24E+02	-1,23E+02	-1,31E+02	-1,24E+02	-3,11E+02	-2,87E+02	-2,80E+02	-2,56E+02	-3,05E+02	-2,91E+02
	mediano	-1,83E+02	-1,88E+02	-1,88E+02	-1,91E+02	-3,22E+02	-3,13E+02	-3,05E+02	-2,91E+02	-3,21E+02	-3,19E+02
	média	-1,91E+02	-1,95E+02	-1,94E+02	-1,92E+02	-3,22E+02	-3,10E+02	-3,02E+02	-2,90E+02	-3,20E+02	-3,15E+02
	erro padrão	8,19E+00	7,60E+00	7,89E+00	8,19E+00	7,00E-01	1,80E+00	2,14E+00	2,33E+00	9,50E-01	1,93E+00
F10	mínimo	-2,42E+02	-2,69E+02	-2,42E+02	-2,42E+02	-3,25E+02	-3,21E+02	-3,16E+02	-3,08E+02	-3,27E+02	-3,25E+02
	máximo	1,48E+02	7,09E+01	1,01E+02	1,47E+02	-2,92E+02	-2,78E+02	-2,68E+02	-2,17E+02	-3,13E+02	-2,87E+02
	mediano	-7,23E+01	-1,12E+02	-7,88E+01	-6,19E+01	-3,13E+02	-3,15E+02	-2,98E+02	-2,71E+02	-3,21E+02	-2,96E+02
	média	-6,47E+01	-9,36E+01	-7,99E+01	-5,78E+01	-3,12E+02	-3,14E+02	-2,97E+02	-2,66E+02	-3,21E+02	-2,99E+02
	erro padrão	1,87E+01	1,62E+01	1,74E+01	1,74E+01	1,27E+00	1,29E+00	2,08E+00	4,49E+00	5,99E-01	1,95E+00
F11	mínimo	9,89E+01	9,85E+01	9,99E+01	9,99E+01	9,00E+01	9,06E+01	9,34E+01	9,98E+01	9,10E+01	9,00E+01
	máximo	1,05E+02	1,04E+02	1,06E+02	1,05E+02	9,33E+01	9,77E+01	9,73E+01	1,04E+02	9,59E+01	9,80E+01
	mediano	1,01E+02	1,01E+02	1,02E+02	1,03E+02	9,10E+01	9,36E+01	9,55E+01	1,02E+02	9,30E+01	9,21E+01
	média	1,01E+02	1,01E+02	1,02E+02	1,02E+02	9,12E+01	9,35E+01	9,54E+01	1,02E+02	9,31E+01	9,28E+01
	erro padrão	2,93E-01	2,43E-01	2,59E-01	2,68E-01	1,73E-01	2,71E-01	1,85E-01	1,87E-01	2,19E-01	4,17E-01
F12	mínimo	-4,59E+02	-4,59E+02	-4,60E+02	-4,60E+02	-4,60E+02	-4,60E+02	-4,12E+02	-3,90E+02	-4,60E+02	-4,60E+02
	máximo	3,24E+04	1,13E+03	2,31E+03	1,09E+03	1,65E+04	1,17E+05	5,71E+03	5,64E+03	1,69E+04	1,01E+04
	mediano	2,09E+03	-4,37E+02	-4,41E+02	-4,41E+02	-4,49E+02	-4,50E+02	4,09E+01	-4,50E+02	-1,97E+02	-2,93E+02
	média	5,14E+03	-2,49E+02	-1,27E+02	-1,71E+02	1,03E+03	1,88E+03	7,37E+03	5,20E+02	8,10E+01	2,59E+03
	erro padrão	1,35E+03	7,13E+01	1,25E+02	9,85E+01	6,22E+02	7,72E+02	4,37E+03	2,53E+02	2,57E+02	8,73E+02
mínimo	-1,29E+02	-1,29E+02	-1,29E+02	-1,29E+02	-1,30E+02	-1,30E+02	-1,29E+02	-1,29E+02	-1,29E+02	-1,29E+02	-1,29E+02

Função	MD	GC	BFGS	DFP	q-G	q-GC	q-BFGS	q-DFP	IPOPT	CMA-ES	CRS2-LM
F13	máximo	-1,00E+02	-1,20E+02	-1,12E+02	-1,28E+02	-1,28E+02	-1,28E+02	-1,27E+02	-9,51E+00	-1,29E+02	-1,26E+02
	mediano	-1,28E+02	-1,27E+02	-1,25E+02	-1,29E+02	-1,29E+02	-1,29E+02	-1,28E+02	-1,25E+02	-1,29E+02	-1,27E+02
	média	-1,27E+02	-1,26E+02	-1,24E+02	-1,29E+02	-1,29E+02	-1,29E+02	-1,28E+02	-1,15E+02	-1,29E+02	-1,27E+02
	erro padrão	9,56E-01	3,34E-01	8,36E-01	7,75E-01	9,54E-02	7,37E-02	1,04E-01	4,97E+00	4,73E-02	1,29E-01
F14	mínimo	-2,95E+02	-2,95E+02	-2,95E+02	-2,97E+02	-2,97E+02	-2,97E+02	-2,97E+02	-2,96E+02	-2,96E+02	-2,97E+02
	máximo	-2,95E+02	-2,95E+02	-2,95E+02	-2,96E+02	-2,96E+02	-2,96E+02	-2,96E+02	-2,95E+02	-2,96E+02	-2,96E+02
	mediano	-2,95E+02	-2,95E+02	-2,95E+02	-2,96E+02	-2,96E+02	-2,96E+02	-2,96E+02	-2,95E+02	-2,96E+02	-2,96E+02
	média	-2,95E+02	-2,95E+02	-2,95E+02	-2,96E+02	-2,96E+02	-2,96E+02	-2,96E+02	-2,95E+02	-2,96E+02	-2,96E+02
erro padrão	2,28E-02	2,24E-02	2,28E-02	4,84E-02	6,45E-02	6,16E-02	5,78E-02	4,45E-02	4,48E-02	5,67E-02	

Tabela 5.3 - Estatística sobre o melhor valor da função objetivo sobre 30 execuções dos algoritmos para as funções teste unimodais após 10.000 avaliações da função.

Função	MD	GC	BFGS	DFP	q-G	q-GC	q-BFGS	q-DFP	IPOPT	CMA-ES	CRS2-LM
Brown	mínimo	7,35E-23	6,31E-07	9,42E-27	1,34E-07	2,86E-06	3,04E-06	4,51E-03	7,78E-18	3,16E-16	1,31E-07
	máximo	1,84E+00	1,84E+00	1,84E+00	4,17E-05	4,65E-05	1,91E+00	1,75E+00	2,35E-14	5,50E-15	7,17E-05
Brown	mediano	1,42E-07	4,96E-06	1,04E-20	1,40E-05	1,21E-05	4,64E-04	3,12E-02	1,39E-15	1,16E-15	3,68E-06
	média	3,07E-01	2,46E-01	1,84E-01	1,49E-05	1,47E-05	6,54E-02	1,08E-01	3,36E-15	1,46E-15	9,69E-06
Brown	erro padrão	1,27E-01	1,16E-01	1,02E-01	2,15E-06	2,04E-06	6,37E-02	6,09E-02	1,01E-15	2,09E-16	2,93E-06
	mínimo	2,90E-15	2,58E-15	2,81E-15	7,59E-11	2,93E-11	1,52E-08	5,28E-07	1,17E-19	7,11E-17	7,05E-08
Broyden	máximo	6,88E-15	6,89E-15	8,65E-12	5,86E+00	6,19E-09	3,99E-07	3,64E+00	1,74E-17	7,47E-15	9,82E-06
	mediano	6,61E-15	6,24E-15	2,81E-14	2,16E-10	9,68E-10	8,10E-08	1,13E-03	1,90E-18	1,06E-15	2,02E-06
Broyden	média	6,15E-15	5,49E-15	5,54E-13	1,14E+00	1,40E-09	1,11E-07	1,25E-01	3,69E-18	1,61E-15	3,14E-06
	erro padrão	2,07E-16	2,88E-16	3,33E-13	4,25E-01	2,55E-10	1,83E-08	1,21E-01	7,90E-19	3,16E-16	5,59E-07
Broyden	mínimo	2,16E-01	2,16E-01	8,66E-15	3,67E-01	3,67E-01	3,28E-01	3,70E-01	1,15E-17	1,02E+00	1,02E+00
	máximo	2,76E+00	2,76E+00	3,82E-01	3,67E-01	3,67E-01	3,82E-01	2,52E+00	1,02E+00	3,03E+00	1,20E+00
Tridiagonal	mediano	3,66E-01	3,67E-01	3,63E-01	3,67E-01	3,67E-01	3,67E-01	3,82E-01	1,02E+00	1,21E+00	1,03E+00
	média	7,86E-01	8,14E-01	2,95E-01	3,67E-01	3,67E-01	3,67E-01	4,52E-01	7,16E-01	1,34E+00	1,10E+00
Discrete	erro padrão	1,56E-01	1,66E-01	2,18E-02	1,34E-08	1,74E-08	1,44E-03	7,13E-02	8,22E-02	8,92E-02	1,66E-02
	mínimo	7,71E-19	7,71E-19	3,17E-19	1,40E-03	7,52E-06	8,30E-07	2,30E-03	5,29E-17	4,07E-16	6,70E-07
Boundary	máximo	2,63E-11	2,80E-11	1,94E-05	1,97E-02	2,40E-05	1,83E-03	2,57E-01	6,77E-15	5,46E-15	6,04E-05
	mediano	1,02E-12	2,09E-12	4,19E-17	8,07E-03	1,37E-05	4,18E-05	1,19E-01	6,88E-03	1,03E-15	1,09E-05
Boundary	média	3,10E-12	3,85E-12	6,47E-07	8,60E-03	1,49E-05	1,48E-04	1,33E-01	1,40E-15	1,42E-15	1,77E-05
	erro padrão	1,03E-12	1,08E-12	6,47E-18	9,50E-04	9,81E-07	6,26E-05	9,74E-03	2,70E-16	1,99E-16	3,21E-06
Discrete	mínimo	8,36E-21	8,76E-21	4,07E-17	7,66E-10	1,08E-06	7,17E-05	5,48E-03	1,57E-19	2,79E-16	1,84E-07
	máximo	1,08E-18	1,12E-18	1,20E-11	8,87E-03	2,13E-04	4,35E-04	4,42E-02	1,92E-17	4,21E-15	2,20E-05
Integral	mediano	1,90E-19	3,25E-19	4,55E-13	2,12E-04	2,13E-04	2,42E-04	2,01E-02	1,14E-18	1,28E-15	9,38E-07
	média	3,34E-19	4,25E-19	1,35E-12	1,27E-03	1,71E-04	2,43E-04	2,23E-02	2,27E-18	1,55E-15	2,76E-06
Penalty I	erro padrão	6,33E-20	7,08E-20	4,34E-13	3,83E-04	1,57E-05	1,78E-05	1,87E-03	6,64E-19	2,03E-16	8,40E-07
	mínimo	7,09E-05	7,09E-05	7,45E-05	7,25E-05	7,38E-05	7,24E-05	7,32E-05	7,09E-05	7,09E-05	7,37E-05
Penalty I	máximo	8,16E-05	7,64E-05	9,25E-05	7,90E-05	8,18E-05	7,87E-05	7,89E-05	7,09E-05	7,35E-05	8,45E-05
	mediano	7,09E-05	7,09E-05	8,18E-05	7,48E-05	7,78E-05	7,40E-05	7,62E-05	7,09E-05	7,10E-05	7,88E-05

Função	MD	GC	BFGS	DFP	q-G	q-GC	q-BFGS	q-DFP	IPOPT	CMA-ES	CRS2-LM
Penalty I	média	7,12E-05	7,09E-05	8,20E-05	7,52E-05	7,76E-05	7,46E-05	7,63E-05	7,09E-05	7,13E-05	7,91E-05
	erro padrão	4,45E-07	8,33E-13	8,28E-07	2,71E-07	3,77E-07	3,18E-07	2,66E-07	1,46E-13	1,23E-07	5,02E-07
Rosenbrock	mínimo	7,86E-10	1,54E-12	1,54E-12	1,22E+00	3,06E+00	2,17E-04	7,75E+00	2,64E-16	3,87E-16	6,52E-04
	máximo	3,98E+00	8,93E+00	8,93E+00	7,03E+00	8,49E+00	2,95E-02	9,19E+01	4,35E-14	3,98E+00	8,13E-01
	mediano	2,24E-06	1,02E-11	1,02E-11	5,48E+00	5,98E+00	1,78E-03	8,25E+00	6,44E-15	1,60E-15	2,68E-03
	média	7,97E-01	9,62E-01	9,62E-01	4,87E+00	5,81E+00	3,22E-03	8,26E+00	1,08E-14	2,66E-01	3,24E-02
Trigonometric	erro padrão	2,96E-01	3,89E-01	3,89E-01	3,33E-01	2,72E-01	9,70E-04	6,31E-02	2,07E-15	1,85E-01	2,69E-02
	mínimo	8,47E-16	7,38E-16	6,69E-12	4,49E-07	9,41E-13	4,52E-20	1,33E-20	3,33E-10	3,89E-16	3,88E-07
	máximo	4,22E-05	8,79E-04	8,80E-04	1,42E-03	4,72E-04	8,79E-04	3,91E-03	2,80E-05	8,79E-04	4,32E-05
	mediano	2,79E-05	2,24E-07	5,82E-05	4,22E-05	4,62E-05	2,79E-05	9,79E-04	1,42E-05	2,79E-05	3,77E-06
F01	média	1,92E-05	9,78E-05	2,05E-04	2,93E-04	6,64E-05	5,84E-05	1,35E-03	1,40E-05	7,78E-05	1,40E-05
	erro padrão	3,04E-06	4,84E-05	5,58E-05	8,76E-05	1,93E-05	2,84E-05	2,00E-04	2,58E-06	3,98E-05	2,75E-06
	mínimo	-4,50E+02									
	máximo	-4,50E+02	-4,50E+02	-4,50E+02	-4,50E+02	-4,50E+02	-4,50E+02	-4,49E+02	-4,50E+02	-4,50E+02	-4,50E+02
F02	mediano	-4,50E+02									
	média	-4,50E+02									
	erro padrão	0,00E+00	0,00E+00	0,00E+00	0,00E+00	7,35E-07	0,00E+00	4,06E-02	0,00E+00	0,00E+00	1,21E-04
	mínimo	-4,50E+02	-3,74E+02	-4,50E+02	-4,50E+02	-4,50E+02	-4,50E+02	-4,37E+02	-4,50E+02	-4,50E+02	-4,50E+02
F03	máximo	-4,50E+02	3,88E+04	-4,50E+02	-4,50E+02	-4,50E+02	-4,50E+02	-3,82E+02	-4,50E+02	-4,50E+02	-4,50E+02
	mediano	-4,50E+02	5,81E+03	-4,50E+02	-4,50E+02	-4,50E+02	-4,50E+02	-4,15E+02	-4,50E+02	-4,50E+02	-4,50E+02
	média	-4,50E+02	8,72E+03	-4,50E+02	-4,50E+02	-4,50E+02	-4,50E+02	-4,11E+02	-4,50E+02	-4,50E+02	-4,50E+02
	erro padrão	0,00E+00	1,79E+03	0,00E+00	1,55E-06	1,06E-05	5,94E-05	2,68E+00	0,00E+00	0,00E+00	3,24E-04
F04	mínimo	2,05E+03	-4,48E+02	-4,50E+02	9,55E+03	2,13E+04	8,39E+03	2,34E+05	-4,49E+02	-4,50E+02	-4,15E+02
	máximo	1,32E+05	1,00E+05	4,46E+06	4,67E+05	5,31E+05	1,90E+06	2,11E+06	1,82E+02	-4,50E+02	3,72E+02
	mediano	4,50E+04	-3,92E+02	-4,50E+02	1,75E+05	1,29E+05	2,74E+05	6,57E+05	-4,44E+02	-4,50E+02	-2,77E+02
	média	4,54E+04	4,00E+03	2,88E+04	1,92E+05	1,86E+05	4,04E+05	8,70E+05	1,97E+05	-4,50E+02	-2,18E+02
F04	erro padrão	5,58E+03	3,40E+03	2,48E+04	2,41E+04	2,58E+04	7,25E+04	9,80E+04	2,20E+04	0,00E+00	3,15E+01
	mínimo	-4,50E+02	-4,50E+02	7,20E+03	-4,50E+02	-4,50E+02	-4,32E+02	-4,14E+02	1,79E+04	-4,50E+02	-4,50E+02
	máximo	3,73E+04	3,08E+04	5,46E+04	-4,49E+02	-4,50E+02	-3,14E+02	-3,30E+02	8,49E+05	-4,50E+02	-4,50E+02
	mediano	4,79E+03	7,82E+03	2,03E+04	-4,50E+02	-4,50E+02	-3,70E+02	-3,67E+02	5,53E+04	-4,50E+02	-4,50E+02
média	7,05E+03	7,00E+03	2,28E+04	-4,50E+02	-4,50E+02	-3,71E+02	-3,70E+02	1,07E+05	-4,50E+02	-4,50E+02	

Função	MD	GC	BFGS	DFP	q-G	q-GC	q-BFGS	q-DFP	IPOPT	CMA-ES	CRS2-LM
F04	1,55E+03	1,24E+03	1,98E+03	2,14E+03	-2,80E-02	2,29E-05	5,09E+00	4,19E+00	3,19E+04	0,00E+00	2,66E-03
	-3,10E+02	-3,10E+02	3,99E+03	7,68E+03	-3,10E+02	-3,10E+02	-2,86E+02	-1,80E+01	-3,09E+02	-3,10E+02	-3,10E+02
	6,69E+03	5,77E+03	3,62E+04	3,62E+04	-3,10E+02	-3,10E+02	5,89E+02	5,25E+02	-3,02E+02	-3,10E+02	-3,10E+02
F05	-2,89E+02	-3,08E+02	1,66E+04	1,73E+04	-3,10E+02	-3,10E+02	-2,56E+01	2,96E+02	-3,09E+02	-3,10E+02	-3,10E+02
	2,83E+02	2,51E+02	1,66E+04	1,84E+04	-3,10E+02	-3,10E+02	3,66E+00	2,69E+02	-3,08E+02	-3,10E+02	-3,10E+02
	2,59E+02	2,79E+02	1,46E+03	1,24E+03	3,03E-03	3,65E-03	3,83E+01	2,69E+01	3,48E-01	7,84E-05	1,09E-04

### 5.3 Comparação dos Métodos Usando Perfis de Dados

O desempenho dos métodos baseados no vetor  $q$ -gradiente foi também avaliado com o uso de perfis de dados (MORÉ; WILD, 2009). Embora as funções teste utilizadas neste trabalho não sejam realmente computacionalmente dispendiosas, esta abordagem é particularmente adequada quando o custo computacional para avaliar a função-objetivo é dominante na execução do algoritmo, o que ocorre com frequência em aplicações de interesse prático.

Seja  $\mathcal{S}$  o conjunto de 13 algoritmos de otimização:  $q$ -G,  $q$ -GC,  $q$ -BFGS,  $q$ -DFP, MD, GC, BFGS, DFP, IPOPT, ISRES, NelderMead, CMA-ES e CRS2-LM. Seja também  $\mathcal{P}$  o conjunto de problemas onde cada problema  $p$  corresponde a um par constituído de uma função (problema) teste particular (por exemplo, Rastrigin) e uma execução particular. Uma vez que o número de funções teste é igual a 27 funções e são realizadas 30 execuções para cada função, então existe um total de  $27 \times 30 = 810$  problemas para geração do perfil de dados de um dado algoritmo de otimização. Assim, conforme Moré and Wild (2009), define-se o perfil de dados de um dado algoritmo  $s \in \mathcal{S}$  em relação a  $\gamma$  como sendo

$$d_s(\gamma) = \frac{1}{|\mathcal{P}|} \left| \{p \in \mathcal{P} : t_{p,s} \leq \gamma(n_p + 1)\} \right|, \quad (5.1)$$

onde  $\gamma$  é o número de avaliações da função objetivo,  $n_p$  é o número de variáveis do problema  $p$  e  $|\mathcal{P}|$  denota a cardinalidade de  $\mathcal{P}$ . Para um dado solucionador  $s$  e qualquer  $\gamma > 0$ ,  $d_s(\gamma)$  é a fração dos problemas resolvidos segundo um critério de convergência por  $s$  dentro de  $\gamma(n_p + 1)$  avaliações da função objetivo (equivalente ao  $\gamma$  estimado pelo gradiente simplex (MORÉ; WILD, 2009)). No presente caso, o perfil de dados de um dado algoritmo fornece a fração dos 810 problemas que foram resolvidos segundo um dado critério de convergência, após  $\gamma$  avaliações das 27 funções teste.

Como dito acima, a geração de perfis de dados requer a definição de um critério de convergência. Este pode ser a precisão da iteração atual com relação a solução exata do problema, ou pode estar relacionado a algum parâmetro no algoritmo. Usuários que trabalham com funções teste computacionalmente caras podem optar por um critério de convergência baseado na taxa de diminuição do valor da função objetivo (MORÉ; WILD, 2009). O critério de convergência aqui utilizado baseia-se em Moré and Wild (2009), onde dada uma tolerância  $\tau > 0$ , um ponto  $x$  obtido por algum algoritmo é dito satisfazer

o critério de convergência se

$$f(x^{(0)}) - f(x) \geq (1 - \tau)(f(x^{(0)}) - f_L), \quad (5.2)$$

onde  $x^{(0)}$  é o ponto inicial do problema em questão e  $f_L$  é calculado para cada problema  $p \in \mathcal{P}$  como o menor valor de  $f$  obtido por qualquer solucionador com um dado número limite de avaliações da função objetivo. Aqui, é necessário que  $f(x^{(0)}) - f(x)$  alcance no mínimo a redução de  $1 - \tau$  vezes a melhor redução possível ( $f(x^{(0)}) - f_L$ ).

O critério de convergência definido em 5.2 foi usado por diversos autores. [Elster and Neumaier \(1995\)](#) definiram  $f_L$  como sendo uma estimativa precisa de  $f$  em um minimizador global. Já [Marazzi and Nocedal \(2002\)](#), definiu  $f_L$  como uma estimativa precisa de  $f$  em um minimizador local obtido por um algoritmo que utiliza derivadas. Porém, segundo [Moré and Wild \(2009\)](#), definir  $f_L$  como uma estimativa precisa de  $f$  em um minimizador não é apropriado nos casos em que a avaliação de  $f$  é cara computacionalmente, pois o solucionador pode não ser capaz de satisfazer o critério de convergência em 5.2 dentro dos limites computacionais do usuário. Mesmo para problemas com  $f$  simples e barata, um algoritmo que não utiliza derivadas pode não atingir a precisão quando comparada a um otimizador com derivadas. Por outro lado, se  $f_L$  é definido como sendo o menor valor de  $f$ , então pelo menos um algoritmo irá satisfazer 5.2 para qualquer  $\tau > 0$ .

A tolerância  $\tau > 0$  em 5.2 representa a redução percentual a partir do valor inicial  $f(x^{(0)})$ . Um valor de  $\tau = 0,1$  pode representar um pequeno decréscimo, isto é, uma redução de 90% do total possível, enquanto que menores valores de  $\tau$  correspondem a decréscimos maiores. A medida que  $\tau$  decresce, a precisão de  $f(x)$  como uma aproximação de  $f_L$  aumenta. Para os nossos experimentos numéricos definimos  $\tau = 0,05$ , assim, um algoritmo satisfaz o critério de convergência para o problema  $p \in \mathcal{P}$  se atingir 95% da melhor redução possível por qualquer um dos solucionadores do problema  $p$ .

A Figura 5.1 apresenta a curva dos perfis de dados dos vários algoritmos para as funções teste em estudo. Os perfis de dados foram calculados até 9.900 avaliações da função objetivo, equivalentes a 900 gradientes simplex estimados para problemas de 10 dimensões (ou seja,  $(10 + 1) \times 900 = 9.900$ ). Além disso, a Figura 5.2 apresenta a curva dos perfis de dados separados para problemas multimodais e problemas unimodais.

Os perfis de dados da Figura 5.1 mostram que os métodos  $q$ -G,  $q$ -GC e  $q$ -BFGS são muito melhores quando comparados as suas versões clássicas (MD, GC, BFGS) e aos algoritmos IPOPT, ISRES e Nelder-Mead. Já o método  $q$ -DFP não teve um desempenho tão bom

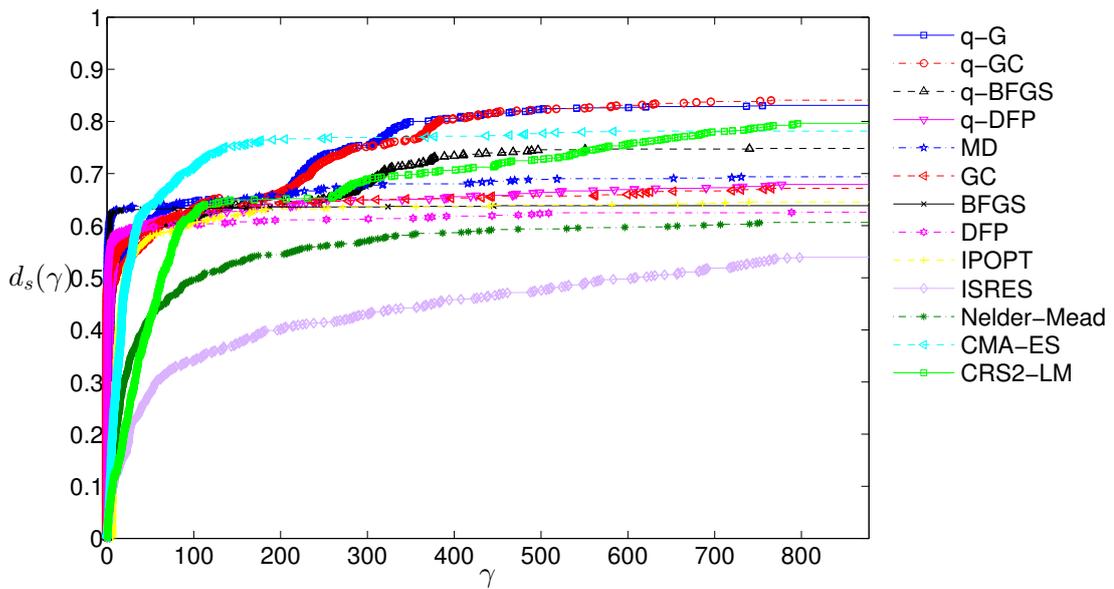
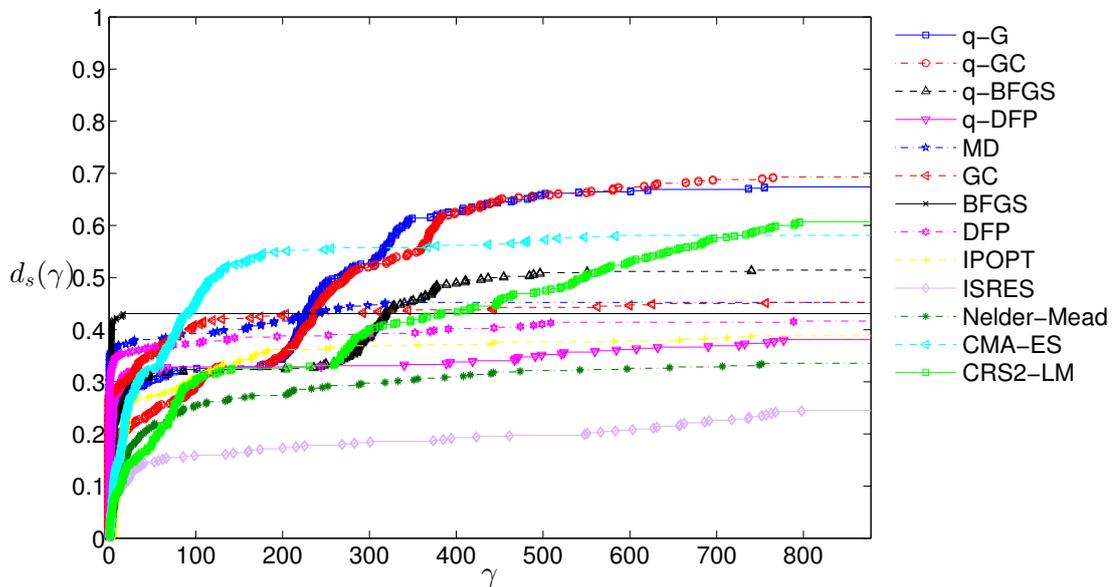


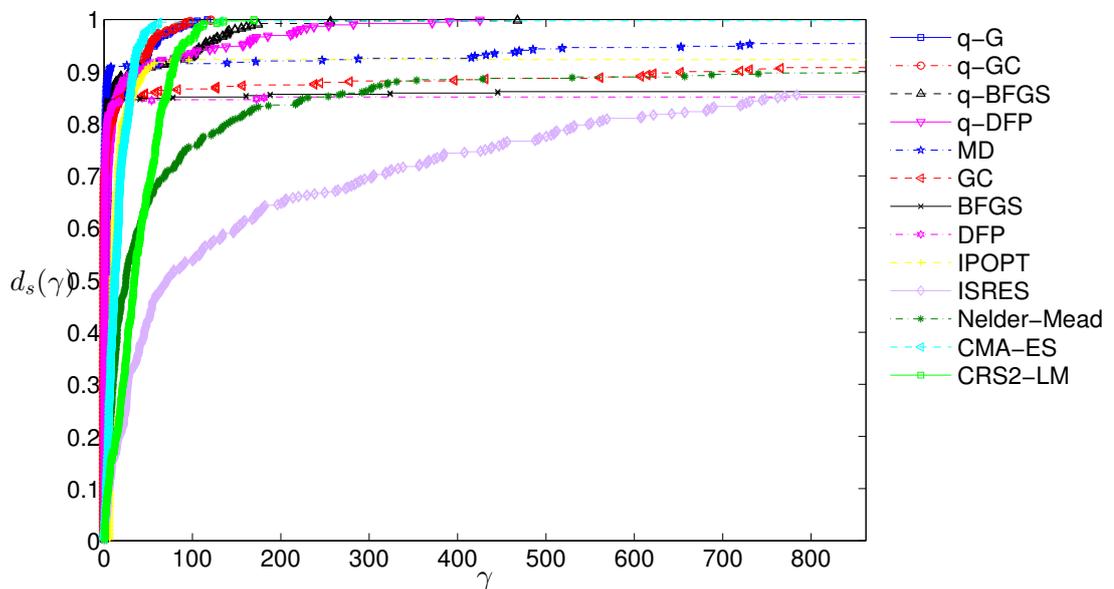
Figura 5.1 - Perfis de dados dos algoritmos para 27 funções teste usando 30 diferentes pontos iniciais.

quanto o método  $q$ -BFGS. Porém, isso já era esperado, uma vez que, o método clássico BFGS é considerado o mais eficiente dos métodos quase-Newton, atualmente. Mesmo assim, o método  $q$ -DFP é melhor que a sua versão clássica, IPOPT, ISRES e Nelder-Mead. Além disso, os perfis de dados mostram que os métodos  $q$ -G,  $q$ -GC e  $q$ -BFGS são competitivos com CMA-ES e CRS2-LM para as funções teste desse estudo. Em particular, ambos,  $q$ -G e  $q$ -GC satisfazem o critério de convergência em aproximadamente 85% dos problemas após 500 gradientes simplex estimados ou 5.500 avaliações da função objetivo. Enquanto CMA-ES e CRS2-LM satisfazem o teste de convergência em aproximadamente 80% dos problemas, e o método  $q$ -BFGS satisfaz em aproximadamente 75% dos problemas. Em contrapartida, as versões clássicas, MD, GC, BFGS e DFP juntamente com os métodos  $q$ -DFP, IPOPT e Nelder-Mead satisfazem o critério de convergência em aproximadamente 65% das funções com o mesmo critério computacional. Por fim, o desempenho do método ISRES é o pior de todos, satisfazendo o critério de convergência em aproximadamente 55% das funções teste.

Na Figura 5.2, os perfis de dados são apresentados separadamente para as funções multimodais (5.2(a)) e funções unimodais (5.2(b)). Percebe-se que para as funções multimodais, os métodos  $q$ -G,  $q$ -GC e  $q$ -BFGS são competitivos com os métodos CMA-ES e CRS2-LM e são muito melhores quando comparados com as suas versões clássicas e com os outros métodos, especialmente nas iterações finais (depois de 300 gradientes



(a)



(b)

Figura 5.2 - Perfis de dados dos algoritmos para 14 problemas multimodais (a) e 13 problemas unimodais (b) usando 30 diferentes pontos iniciais.

simplex estimados). Em especial, os métodos  $q$ -G e  $q$ -GC satisfazem o critério de convergência em aproximadamente 70% das funções teste, enquanto os métodos CMA-ES e CRS2-LM satisfazem o critério de convergência em aproximadamente 60%. Já o método  $q$ -BFGS satisfaz o mesmo critério de convergência em 50%, e o método  $q$ -DFP, com um pior desempenho, satisfaz o critério de convergência em apenas 40% das funções teste aproximadamente.

De forma similar, para as funções unimodais, os métodos  $q$ -G e  $q$ -GC são competitivos com os métodos CMA-ES e CRS2-LM e são melhores do que os outros métodos, satisfazendo o critério de convergência em 100% após 100 gradientes simplex estimados (ou 1.100 avaliações da função objetivo). Os métodos  $q$ -BFGS e  $q$ -DFP também satisfazem em 100% o critério de convergência, porém, utilizando um pouco mais avaliações da função objetivo (4.400 aproximadamente). Mesmo não sendo a aplicação ideal para os métodos baseados no vetor  $q$ -gradiente, devido serem métodos de otimização global, os mesmos se mostraram superiores, no sentido do critério de convergência, quando comparados as suas respectivas versões clássicas e aos métodos IPOPT, Nelder-Mead e ISRES, que apresentaram um desempenho inferior para esse conjunto de funções teste.

O fato dos métodos baseados no vetor  $q$ -gradiente, principalmente os métodos  $q$ -G e  $q$ -GC neste caso, serem competitivos com o método CMA-ES em uma ampla gama de funções teste é um indicativo da eficiência desses métodos, uma vez que o método CMA-ES é considerado um importante algoritmo na área da computação evolutiva e na comunidade de otimização numérica. Variantes do CMA-ES têm consistentemente conseguido alcançar o topo em várias competições do CEC.

Além disso, é importante enfatizar que o mesmo conjunto de parâmetros ( $\sigma^{(0)}$ ,  $\alpha^{(0)}$ ,  $\beta$  e  $\theta$ ) está sendo utilizado em todas as  $q$ -versões e para todas as funções. Um resultado importante que mostra o quão robusto os métodos baseados no vetor  $q$ -gradiente são. Na seção seguinte é apresentada uma análise sensitiva a respeito do parâmetro  $\sigma$ , parâmetro fundamental que difere a direção de busca dos métodos.

#### 5.4 Análise de Sensibilidade

A robustez dos métodos  $q$ -G,  $q$ -GC,  $q$ -BFGS e  $q$ -DFP é verificada através de uma análise de sensibilidade quando há variação do parâmetro  $\sigma^{(0)}$  (o desvio padrão inicial da distribuição gaussiana usada para calcular o vetor  $q$ -gradiente). Para essa análise, os valores de  $\sigma^{(0)}$  foram verificados para diversos valores, porém, os melhores resultados para o conjunto de funções teste aqui analisadas foram obtidos para  $\sigma^{(0)} = 0,01 \times L; 0,05 \times L; 0,1 \times L; 0,2 \times L; 0,5 \times L$ , onde  $L$ , conforme já visto, é a maior dis-

tância dentro do espaço de busca. Após rodadas preliminares, os outros parâmetros foram fixados como:  $\alpha^{(0)} = 0,1 \times L$ ,  $\beta = 0,999$  e  $\theta = 0,2 \times L$ . Como antes, perturbações gaussianas são utilizadas a cada 10 avaliações da função objetivo. Como uma proposta de análise, são consideradas apenas 16 funções da Tabela 5.1. Por apresentarem um menor custo computacional, as funções multimodais escolhidas são: Ackley, Griewank, Levy, Michalewicz, Rastrigin, F7, F8 e F9. E as funções unimodais são: Brown, Broyden Banded, Broyden Tridiagonal, Discrete Boundary, Discrete Integral, Penalty I, Rosenbrock e Trigonometric.

A Figura 5.3 apresenta os perfis de dados considerando diferentes valores de  $\sigma^{(0)}$ , para os métodos  $q$ -G,  $q$ -GC,  $q$ -BFGS e  $q$ -DFP com suas respectivas versões clássicas: método da máxima descida (MD), métodos dos gradientes conjugados (GC), método quase-Newton Broyden-Fletcher-Goldfarb-Shanno (BFGS) e método quase-Newton Davidon-Fletcher-Powell (DFP) para as 16 funções teste selecionadas.

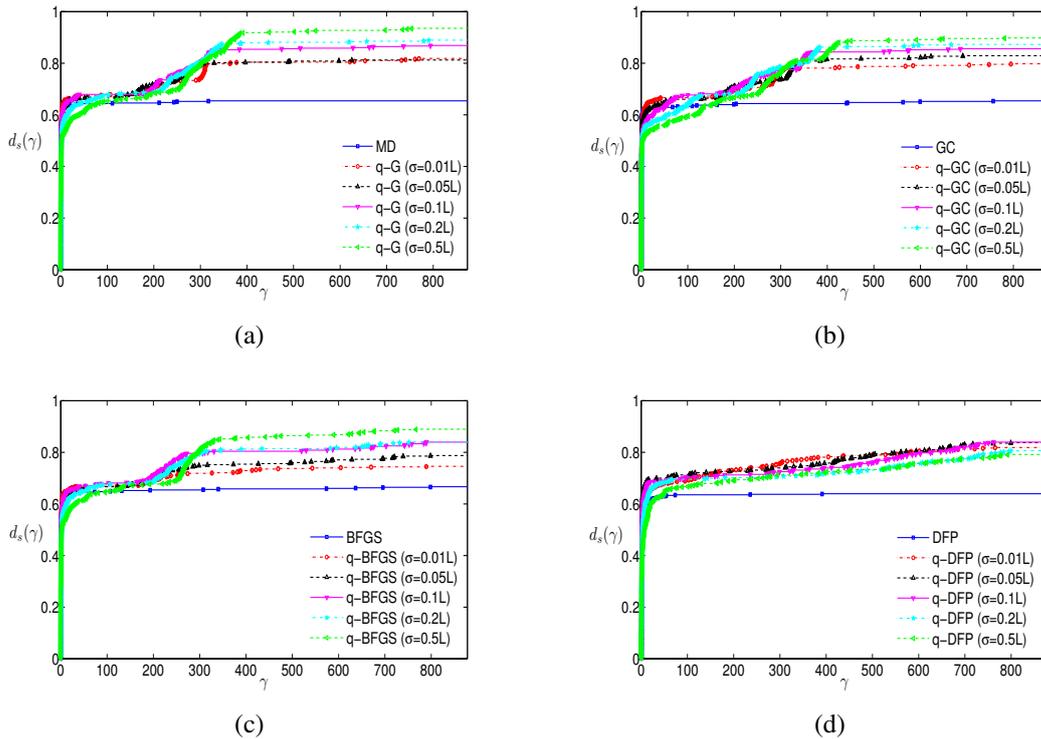


Figura 5.3 - Perfis de dados das  $q$ -versões e suas versões clássicas com vários valores de  $\sigma^{(0)}$  para 16 funções teste usando 30 diferentes pontos iniciais - (a) método da máxima descida versus método  $q$ -G, (b) método dos gradientes conjugados versus método  $q$ -GC, (c) método BFGS versus método  $q$ -BFGS e (d) método DFP versus método  $q$ -DFP.

Os perfis de dados da Figura 5.3 mostram que os métodos baseados no vetor  $q$ -gradiente possuem um desempenho similar e são melhores quando comparados com suas versões clássicas. Enquanto os métodos  $q$ -G,  $q$ -GC e  $q$ -BFGS satisfazem o critério de convergência em aproximadamente 90% das funções teste depois de aproximadamente 4.400 avaliações da função objetivo ou 400 gradientes simplex estimados, as versões clássicas (MD, GC e BFGS) satisfazem o critério de convergência em apenas 65% das funções com o mesmo critério computacional. De forma similar, porém um pouco pior, o método  $q$ -DFP satisfaz o critério de convergência em aproximadamente 80% das funções teste depois de aproximadamente 6.600 avaliações da função objetivo ou 600 gradientes simplex estimados, enquanto sua versão clássica (DFP) satisfaz o critério de convergência em aproximadamente 65% das funções com o mesmo critério computacional.

As Figuras 5.4 e 5.5 mostram os perfis de dados dos métodos separados para 8 funções multimodais e 8 funções unimodais, respectivamente. Note que os métodos baseados no vetor  $q$ -gradiente são claramente melhores do que suas versões clássicas para as funções multimodais. Já para as funções unimodais, as  $q$ -versões são similares e comportam-se à suas versões clássicas para pequenos valores de  $\sigma$ .

Embora esses resultados mostrem que os métodos baseados no vetor  $q$ -gradiente sejam relativamente robustos com relação a  $\sigma^{(0)}$ , é ainda importante selecionar este e os outros parâmetros livres adequadamente. O valor de  $\sigma$  não deve ser muito pequeno para que o algoritmo não se comporte rapidamente como sua versão clássica. Mas também não poderá ser tão grande de tal forma que as iterações escapem do espaço de busca. Assim, fixamos  $\sigma^{(0)} = 0,2 \times L$  para todas as funções teste para os algoritmos  $q$ -G e  $q$ -GC e fixamos  $\sigma^{(0)} = 0,05 \times L$  para todas as funções teste para os algoritmos  $q$ -BFGS e  $q$ -DFP, e os perfis de dados mostraram que esta é uma configuração razoável para as funções aqui analisadas. É importante salientar que pode não ser o melhor ajuste de parâmetros para cada função teste, mas é bom o suficiente para uma grande variedade de funções.

A Figura 5.6 apresenta os perfis de dados dos algoritmos propostos neste trabalho com suas respectivas versões clássicas para as 16 funções teste. Nota-se que as  $q$ -versões aqui implementadas possuem um melhor desempenho quando comparadas a suas versões clássicas, para esse conjunto de funções. Em especial, os métodos  $q$ -G e  $q$ -GC satisfazem o critério de convergência em aproximadamente 80% das funções, e os métodos  $q$ -BFGS e  $q$ -DFP satisfazem o critério de convergência em aproximadamente 70% das funções teste analisadas.

Na Figura 5.7, os perfis de dados são apresentados separadamente para as funções multimodais (5.7(a)) e funções unimodais (5.7(b)). Percebe-se que para as funções multimo-

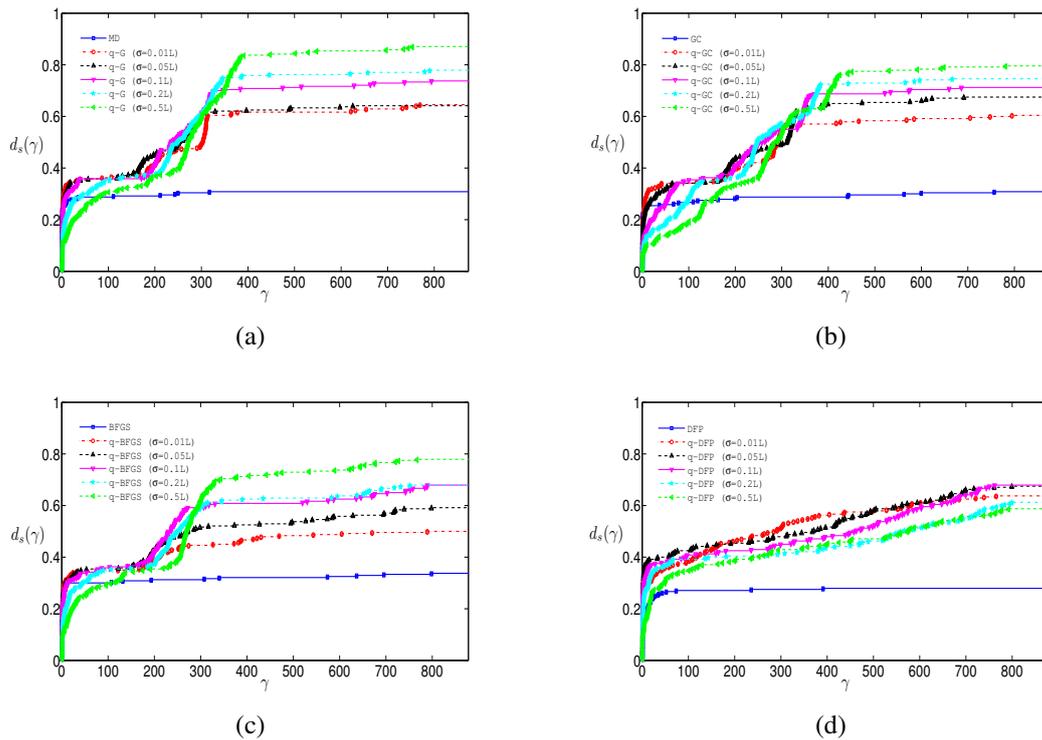


Figura 5.4 - Perfis de dados das  $q$ -versões e suas versões clássicas com vários valores de  $\sigma^{(0)}$  para 8 funções teste multimodais usando 30 diferentes pontos iniciais - (a) método da máxima descida versus método  $q$ -G, (b) método dos gradientes conjugados versus método  $q$ -GC, (c) método BFGS versus método  $q$ -BFGS e (d) método DFP versus método  $q$ -DFP.

dais, os métodos  $q$ -G e  $q$ -GC são muito melhores quando comparados com as suas versões clássicas e com as outras  $q$ -versões ( $q$ -BFGS e  $q$ -DFP), especialmente após 3850 avaliações da função objetivo, satisfazendo o critério de convergência em aproximadamente 65% das funções teste. Já o método  $q$ -BFGS satisfaz o mesmo critério de convergência em aproximadamente 45%, e o método  $q$ -DFP, com um pior desempenho, satisfaz o critério de convergência em apenas 35% das funções teste aproximadamente. Já para as funções unimodais, as  $q$ -versões se mostraram competitivas com suas respectivas versões clássicas, no sentido do critério de convergência, para esse conjunto de funções teste.

## 5.5 Tempo Médio de Execução

As médias dos tempos para as 30 execuções de todos os algoritmos em todas as funções teste, após 10.000 avaliações da função, são apresentadas na Tabela 5.4. Nota-se que o tempo médio de execução para os métodos baseados no vetor  $q$ -gradiente são consistentemente menores quando comparados a suas versões clássicas, quando executados na

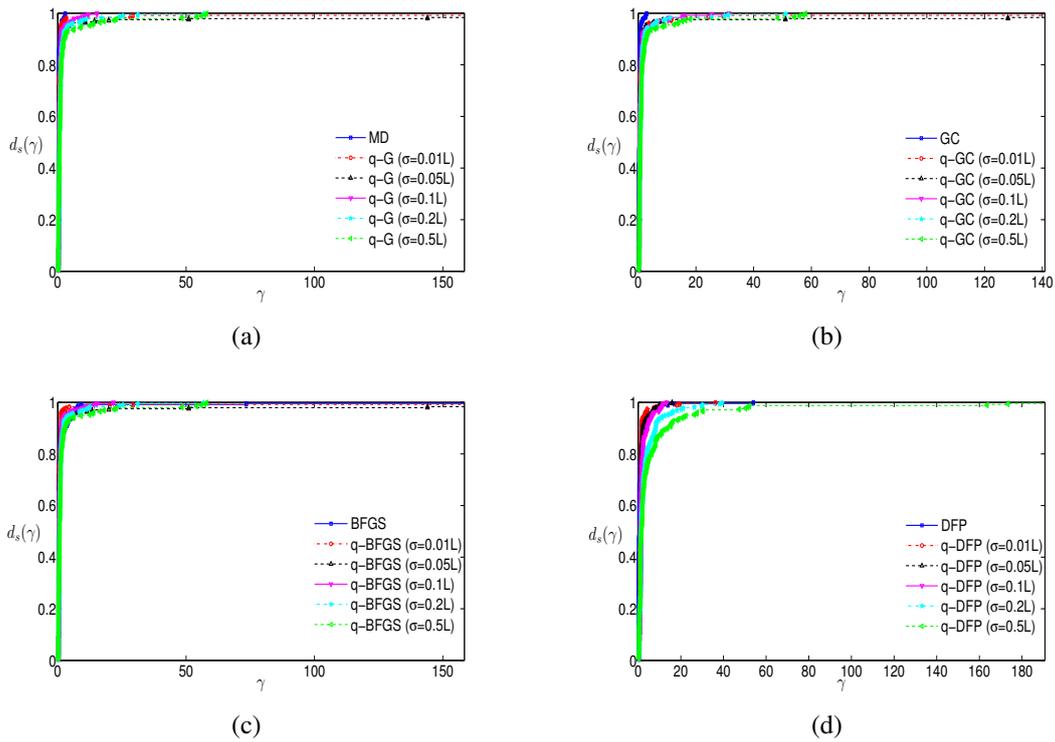


Figura 5.5 - Perfis de dados das  $q$ -versões e suas versões clássicas com vários valores de  $\sigma^{(0)}$  para 8 funções teste unimodais usando 30 diferentes pontos iniciais - (a) método da máxima descida versus método  $q$ -G, (b) método dos gradientes conjugados versus método  $q$ -GC, (c) método BFGS versus método  $q$ -BFGS e (d) método DFP versus método  $q$ -DFP.

mesma máquina.

Embora a comparação direta do tempo médio de execução com os outros métodos seja de difícil análise, devido ao uso de diferentes máquinas, ainda é possível perceber que os tempos médios obtidos nas 30 execuções pelos métodos convergentes baseados no vetor  $q$ -gradiente é da mesma ordem de grandeza, se não melhores, que os tempos médios de execução para os outros métodos.

Por fim, observa-se que se as funções fossem computacionalmente caras (por exemplo, a cada avaliação da função levar pelo menos alguns minutos), então o tempo médio das execuções seria elevado. Nesta situação, o que mais importa é o quão bem os algoritmos resolvem os problemas dado um valor limite de avaliações da função e isso pode ser visto a partir dos perfis de dados apresentados na Seção 5.3.

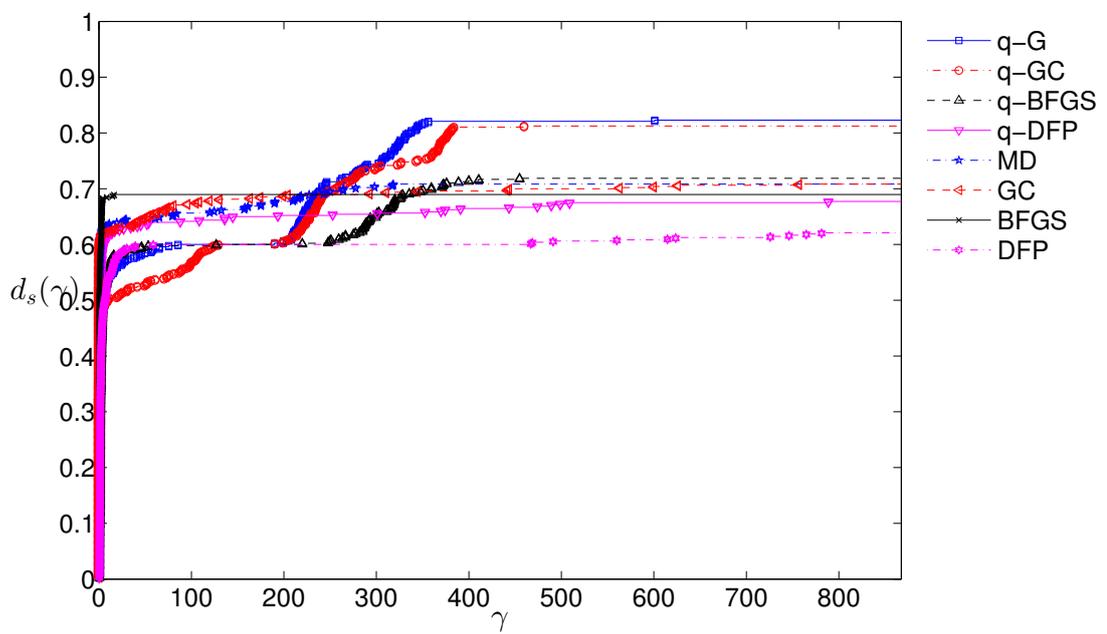
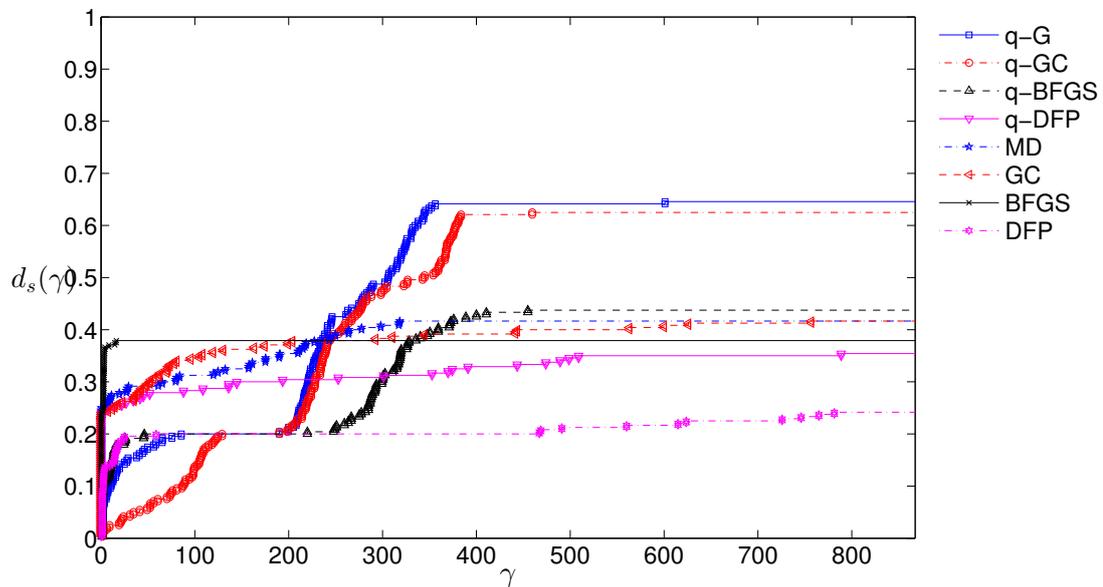
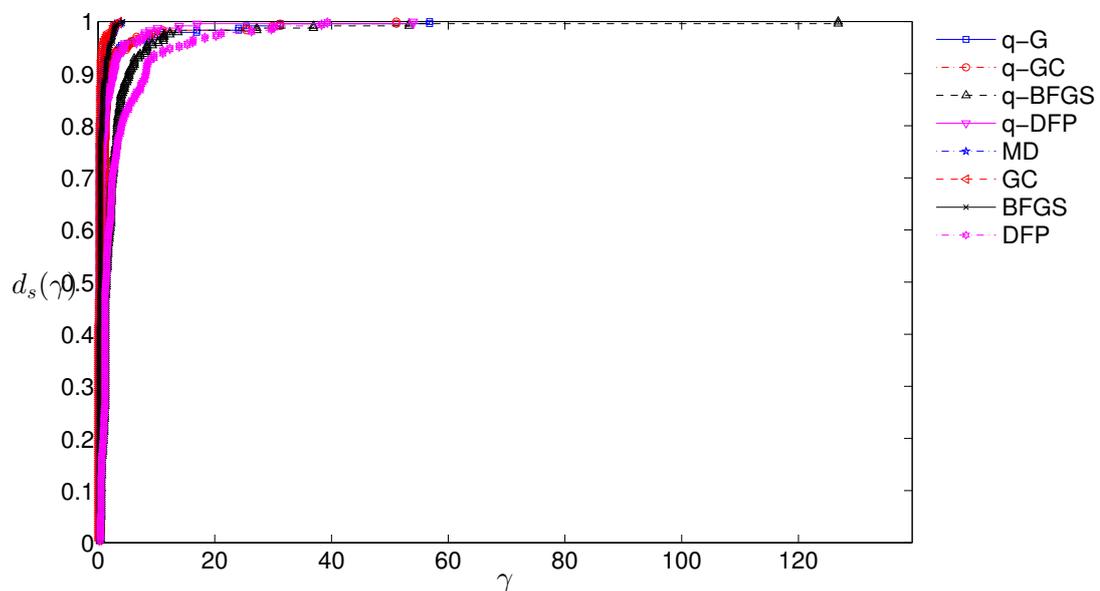


Figura 5.6 - Perfis de dados dos algoritmos propostos para 16 funções teste usando 30 diferentes pontos iniciais.



(a)



(b)

Figura 5.7 - Perfis de dados dos algoritmos propostos para 8 problemas multimodais (a) e 8 problemas unimodais (b) usando 30 diferentes pontos iniciais.

Tabela 5.4 - Tempo médio (em segundos) de 30 execuções dos algoritmos para as funções teste após 10.000 avaliações da função. As  $q$ -versões e suas versões clássicas foram executadas em um laptop com processador Intel(R) Core(TM) i5-2410M 2.30GHz com 6GB RAM e sistema operacional Ubuntu Linux 12.04.4 64-bit. Os outros métodos foram executados em um desktop Intel(R) Core(TM) i7-4770 CPU 3.4GHz 3.00GHz com 16GB RAM e sistema operacional Windows XP 64-bit.

Função	MD	GC	BFGS	DFP	$q$ -G	$q$ -GC	$q$ -BFGS	$q$ -DFP	CMA-ES	CRS2-LM	IPOPT	ISRES	Nelder-Mead
Ackley	6,87	4,19	9,19	7,98	6,42	1,66	1,90	1,89	3,12	7,11	8,71	6,71	13,21
Griewank	11,64	5,17	13,40	8,66	1,81	1,78	1,87	1,90	3,58	6,86	3,75	6,70	12,76
Levy	5,04	4,28	9,90	7,48	1,68	1,74	1,67	1,66	3,17	7,07	2,29	6,72	11,96
Mich	9,99	5,79	10,10	8,84	1,98	2,05	1,94	1,93	2,94	6,88	2,54	6,74	14,25
Rastrigin	7,68	3,63	8,55	7,95	1,62	1,58	1,65	1,61	3,23	6,90	7,34	6,69	13,42
Brown	3,20	2,67	2,83	5,03	1,31	1,34	1,35	1,36	3,27	7,15	3,23	6,73	11,94
BroydenBanded	10,26	5,63	12,80	11,02	2,15	2,18	1,97	1,98	3,62	7,17	7,49	6,70	12,05
BroydenTriDiagonal	2,27	2,37	2,30	2,89	1,25	1,27	1,30	1,28	3,14	7,21	3,55	6,72	11,20
DiscreteBoundary	3,05	2,70	6,24	5,84	1,35	1,43	1,38	1,38	3,28	7,02	2,40	6,73	12,39
DiscreteIntegral	11,52	6,11	14,50	11,22	2,14	2,13	2,04	2,05	4,14	7,54	10,53	6,79	12,50
PenaltyOne	3,63	3,21	6,24	4,42	1,30	1,33	1,35	1,34	2,45	6,96	3,13	6,72	12,84
Rosenbrock	2,42	2,22	4,18	3,79	1,30	1,28	1,36	1,34	2,76	6,78	2,13	6,73	11,12
Trigonometric	35,21	17,20	50,20	24,13	5,53	5,03	4,70	4,51	3,79	7,42	4,28	6,75	12,64
F01	2,79	3,31	4,11	3,94	1,18	1,34	1,45	1,46	20,57	25,18	28,32	25,22	26,30
F02	3,97	3,04	3,61	3,61	1,44	1,50	1,61	1,62	20,38	25,39	17,75	25,45	27,03
F03	8,57	8,24	9,48	9,44	2,54	2,70	3,09	3,01	28,10	32,91	21,00	32,56	32,51

Função	MD	GC	BFGS	DFP	$q$ -G	$q$ -GC	$q$ -BFGS	$q$ -DFP	CMA-ES	CRS2-LM	IPOPT	ISRES	Nelder-Mead
F04	3,30	3,27	3,62	3,51	1,56	1,71	1,74	1,91	2,68	25,78	14,77	25,80	28,55
F05	7,77	5,98	12,50	12,82	1,78	1,91	2,21	2,17	20,40	25,69	15,85	25,80	26,56
F06	2,47	2,17	3,07	2,98	1,24	1,33	1,71	1,67	20,10	25,23	17,13	25,31	25,65
F07	5,13	5,33	5,37	5,35	2,26	2,39	2,51	2,46	28,07	32,64	23,40	32,39	32,27
F08	6,06	6,06	6,63	6,85	1,82	1,97	2,32	2,22	28,33	32,36	21,40	32,80	33,65
F09	9,57	5,50	10,12	8,35	1,83	2,02	2,22	2,21	20,39	25,45	23,10	25,42	26,40
F10	11,52	6,61	10,61	9,21	2,03	2,38	2,45	2,46	27,95	36,85	27,37	32,28	33,41
F11	446,00	442,00	547,20	550,01	47,50	50,50	48,73	49,41	28,23	33,06	22,41	33,21	35,78
F12	51,10	46,80	50,71	57,70	11,00	10,80	10,91	11,20	21,18	26,33	18,58	26,21	26,64
F13	6,92	4,74	6,57	6,86	1,62	1,77	2,04	2,04	20,75	25,77	16,23	25,70	27,16
F14	7,27	7,40	6,79	6,61	1,93	2,11	2,36	2,36	28,06	32,94	22,34	32,94	39,69



## 6 APLICAÇÃO A DOIS PROBLEMAS COMPLEXOS DE OTIMIZAÇÃO

Este capítulo apresenta dois problemas considerados difíceis quanto à sua minimização. Um problema de minimização quadrática não-convexa com mínimo global conhecido por construção e um problema suave na redução de reflexão com um mínimo global que pode ser calculado diretamente. Os problemas são de dimensionalidades semelhantes e possuem um grande número de mínimos locais, porém, as características de cada problema são diferentes. Ambos os problemas podem ser encontrados em [Easterling et al. \(2013\)](#).

O problema de minimização quadrática não-convexa possui 57 dimensões e é uma reformulação de um problema de programação inteira, onde a função é cuidadosamente construída para conter um grande número de mínimos locais e um único mínimo global. O problema de aniquilação da onda é um problema de minimização suave com 56 dimensões. No contexto geral, esses dois problemas são úteis para comparar o desempenho dos algoritmos de otimização em problemas moderadamente grandes, qualitativamente diferentes e de complexidade elevada ([EASTERLING et al., 2013](#)).

Os métodos baseados no vetor  $q$ -gradiente foram comparados com seis algoritmos de otimização para cada problema: um algoritmo de aproximação estocástica com perturbações simultâneas (SPSA), duas implementações paralelas de um esquema de recozimento simulado (SPAN e SA), uma implementação paralela do algoritmo DIRECT, um método de ponto interior encontrado comercialmente em um pacote de otimização (KNITRO), e um algoritmo estocástico quase-Newton (QNSTOP). Os detalhes de cada método podem ser encontrados em [Easterling et al. \(2013\)](#).

### 6.1 Minimização de Função Quadrática Não-Convexa

De acordo com [Easterling et al. \(2013\)](#), o problema de minimização de uma função quadrática não-convexa com restrições laterais é dado por

$$(P_b) : \min\{P(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A \mathbf{x} - f^T \mathbf{x} : \mathbf{x} \in \mathbf{X}_b\}, \quad (6.1)$$

onde

$$\mathbf{X}_b = \{\mathbf{x} \in \mathbb{R}^n \mid -1 \leq \mathbf{x}_i \leq 1, \forall_i = 1, \dots, n\}. \quad (6.2)$$

Substituindo o conjunto viável  $\mathbf{X}_b$  pelos seus vértices

$$\delta\mathbf{X}_b = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} \in \{-1, 1\}^n\}, \quad (6.3)$$

resulta no problema de programação inteira

$$(P_{ip}) : \min\{P(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A \mathbf{x} - \mathbf{x}^T f : \mathbf{x} \in \delta \mathbf{X}_b\}. \quad (6.4)$$

Usando a teoria de dualidade canônica de Gao (2000b), Gao (2000a), o problema de programação inteira ( $P_{ip}$ ) pode ser reformulado como um problema de minimização irrestrito não-convexa, dado por:

$$(P_{ip}^d) : \min\{Q(\varphi) = \frac{1}{2}\varphi^T \varphi - \sum_{i=1}^n |f_i + (B^T \varphi)_i| : \varphi \in \mathbb{R}^m\}, \quad (6.5)$$

onde  $\varphi = (\varphi_1, \dots, \varphi_m)$ ,  $f = (f_1, \dots, f_n)$ , e a matriz real  $B_{m \times n}$  está relacionado a A. Aqui  $m = 57$  e  $n = 190$ ,

$$\hat{B} = \begin{bmatrix} 1 & -1 & 0 & -1 & 2 & 0 & 1 & -2 & 1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 0 & -2 & 2 & 0 & 1 \\ 2 & 2 & -1 & -1 & 2 & -2 & 0 & 0 & -1 & 1 \end{bmatrix},$$

$$\hat{f} = 10^{-2} \begin{bmatrix} 1.491803633709836, 3.0717213019723066, \\ 5.246230264266409, -6.718373452055033, \\ 3.969549763760797, 7.502845410079123, \\ 5.622108089244097, -1.9585631018739558, \\ -2.729844702016424, 8.26721052052138 \end{bmatrix},$$

$$B = I_{19 \times 19} \otimes \hat{B} \quad \text{e} \quad f = e_{19} \otimes \hat{f},$$

onde  $e_{19} = (1, \dots, 1) \in \mathbb{R}^{19}$ . Este problema tem exatamente  $2^{19}$  mínimos locais conhecidos e um único mínimo global  $Q(\varphi^{(1)})$  localizado em

$$\varphi^{(1)} = e_{19} \otimes (6 \quad -4 \quad 12).$$

Todos os mínimos locais estão 0.5% dentro do mínimo global  $Q(\varphi^{(1)}) = -1866,01$ .

## 6.2 Problema de Aniquilação de Onda

O problema de aniquilação de onda teve como base o desenvolvimento de um método para a produção de um revestimento de espessura total  $T$ , distribuído uniformemente em  $n$  camadas variando entre dois meios, com o objetivo de reduzir a reflexão de ondas sobre uma frequência  $[\Omega_0, \Omega_1]$  de um desses meios. Esse processo é conhecido como uma aplicação acústica em [Hager and Wang \(2000\)](#), porém, pode ser facilmente adaptado ao eletromagnetismo ou quaisquer outros fenômenos regidos por variantes da equação de onda linear.

Um componente crucial neste processo está em determinar os revestimentos específicos,  $n$ , de tal forma que a reflexão seja  $r = 0$  na frequência ([EASTERLING et al., 2013](#))

$$\omega_k = \Omega_0 + \left( \frac{k-1}{n-1} \right) (\Omega_1 - \Omega_0), \quad (6.6)$$

para  $k = 1, 2, \dots, n$ , onde o valor complexo

$$r(n, \rho, \kappa, \omega_k) = \frac{(\Gamma_-, \rho_1) \prod_{j=1}^n A_j \begin{pmatrix} -1 \\ 1 \end{pmatrix}}{(\Gamma_-, \rho_1) \prod_{j=1}^n A_j \begin{pmatrix} 1 \\ 1 \end{pmatrix}}, \quad (6.7)$$

sendo

$$A_j = \begin{pmatrix} \rho_j e_j^+ & \rho_{j+1} e_j^- \\ \rho_j e_j^- & \rho_{j+1} e_j^+ \end{pmatrix}, \quad \rho_{n+1} = \Gamma_+, \quad e_j^\pm = \exp\left(\frac{2\rho_j \Delta x \omega_k i}{\kappa_j}\right) \pm 1,$$

onde  $i = \sqrt{-1}$ ,  $\Gamma_+$  e  $\Gamma_-$  são as impedâncias dos espaços intermediários do revestimento,  $\Delta x = T/n$ , e  $\rho_j$  e  $\kappa_j$  são a impedância e a rigidez da camada  $j$ , respectivamente.

De acordo com [Easterling et al. \(2013\)](#), uma função objetivo  $f = \|r\|^2$  pode ser formada por meio da observação de que o produto interno do vetor complexo  $(r(\omega_1), \dots, r(\omega_n))$  com si mesmo, produz um valor real conhecido com um mínimo global zero.

Ao escolher  $n = 28$ , um problema de 56 variáveis reais é construído, a impedância e a rigidez das camadas  $n$  são utilizadas como argumentos de  $r$ , enquanto as frequências  $\omega_k$  são determinadas diretamente de  $n$ . Assim, como definido em [Easterling et al. \(2013\)](#),

$\Gamma_+ = 1$  e  $\Gamma_- = 28,14776$ ,  $T = 1m$ , e  $\Omega_0 = 0,09091Hz$  enquanto  $\Omega_1 = 10\Omega_0$ .

### 6.3 Resultados Numéricos

Os resultados numéricos dos algoritmos: SA, SPAN, SPSA, DIRECT, KNITRO, QNSTOP,  $q$ -G,  $q$ -GC,  $q$ -BFGS e  $q$ -DFP com perturbações gaussianas, foram obtidos a partir de 50 execuções, de cada algoritmo, utilizando pontos iniciais selecionados em um hiper-cubo latino<sup>1</sup>, com base nos limites de cada problema. Para o problema de minimização quadrática não-convexa, os limites de cada variável são  $\pm 41.569$ . Para o problema de aniquilação da onda, o limite inferior é 0 e o limite superior é 40. O critério de parada é um número de avaliações da função objetivo fixado em  $10^6$ . A definição dos parâmetros para os métodos baseados no vetor  $q$ -gradiente foram os mesmos já apresentados no capítulo anterior, isto é,  $\sigma^{(0)} = 0,2 \times L$  (para os métodos  $q$ -G e  $q$ -GC) e  $\sigma^{(0)} = 0,05 \times L$  (para os métodos  $q$ -BFGS e  $q$ -DFP),  $\alpha^{(0)} = 0,1 \times L$ ,  $\beta = 0,999$ ,  $\theta = 0,2 \times L$  utilizado a cada 10 avaliações da função objetivo, onde  $L$  é a maior distância dentro do espaço de busca definido por  $L = \sqrt{\sum_{i=1}^n (\mathbf{x}_{supi} - \mathbf{x}_{infi})^2}$ .

As Tabelas 6.1 e 6.2 apresentam o mínimo, máximo, primeiro, segundo e terceiro quartil do valor da função objetivo de cada um dos algoritmos, para o problema de minimização quadrática não-convexa e o problema de aniquilação da onda, respectivamente. Os melhores valores encontrados para essas funções também são apresentados.

Tabela 6.1 - Resultados para o problema de minimização quadrática não-convexa. Melhor valor encontrado:  $-1866,01$ .

	Mínimo	1° quartil	2° quartil	3° quartil	Máximo
DIRECT	-1864,32	-1863,03	-1862,21	-1861,79	-1860,00
SA (naive)	-1146,16	-1110,06	-1095,66	-1084,64	-1030,77
SPAN	-1861,53	-1859,93	-1859,20	-1858,69	-1857,43
SPSA	253,30	604,44	688,00	759,65	893,00
KNITRO	-1864,74	-1827,05	-1825,67	-1808,06	-1609,60
QNSTOP	-1863,90	-1862,63	-1862,21	-1861,37	-1860,52
$q$ -G	-1863,90	-1862,21	-1861,37	-1854,63	-1834,68
$q$ -GC	-1863,90	-1862,63	-1862,21	-1861,79	-1860,10
$q$ -BFGS	-1863,90	-1862,63	-1681,79	-1861,37	-1841,41
$q$ -DFP	-1819,21	-1752,96	-1715,86	-1682,87	-1561,74

<sup>1</sup>O hiper-cubo latino é uma técnica de amostragem estratificada, a qual garante a representação de cada variável ao longo de todo o domínio considerado. Tal técnica divide o domínio de cada variável em intervalos com probabilidades iguais de sorteio e seleciona um valor aleatório pertencente a cada um dos intervalos, que são depois permutados aleatoriamente, gerando a amostragem final (LOH et al., 1996).

Tabela 6.2 - Resultados para o problema de aniquilação da onda. Melhor valor encontrado: 0.

	Mínimo	1° quartil	2° quartil	3° quartil	Máximo
DIRECT	$8,19 * 10^{-7}$	$1,02 * 10^{-3}$	$5,76 * 10^{-3}$	$5,74 * 10^{-2}$	$2,7 * 10^{-1}$
SA (naive)	26,87	27,26	27,36	27,53	27,76
SPAN	2,71	3,35	25,20	26,25	26,62
SPSA	12,94	523,35	2902,51	8031,26	206193,00
KNITRO	27,09	28,00	28,00	28,00	28,00
QNSTOP	26,64	27,10	27,19	27,30	27,48
q-G	26,37	27,14	27,25	27,35	27,63
q-GC	26,21	27,14	27,28	27,57	27,92
q-BFGS	26,17	27,16	27,26	27,50	27,98
q-DFP	26,61	27,10	27,22	27,32	28,00

Para o problema de minimização quadrática não-convexa (Tabela 6.1), os métodos baseados no vetor  $q$ -gradiente tiveram um desempenho semelhante, com exceção do método  $q$ -DFP. No geral, dentre os métodos baseados no vetor  $q$ -gradiente, o método  $q$ -GC foi o que mais se aproximou do método DIRECT, que obteve o melhor resultado dentre os métodos listados. Além do mais, mesmo sendo um método recente, o método  $q$ -GC conseguiu um desempenho melhor quando comparados a métodos antigos da área de otimização, tais como SA, SPAN e SPSA.

Para o problema de aniquilação da onda (Tabela 6.2), os métodos baseados no vetor  $q$ -gradiente possuem um desempenho semelhante aos métodos SA, KNITRO e QNSTOP e um desempenho superior a partir do primeiro quartil quando comparado ao método SPSA. Resultados melhores poderiam ser obtidos com o ajuste adequado dos parâmetros para essa função específica, uma vez que os parâmetros estão fixos para todas as funções analisadas neste trabalho.

De uma forma geral, os resultados dos métodos baseados no vetor  $q$ -gradiente se mostraram competitivos com os outros otimizadores (com exceção do DIRECT), demonstrando potencial para a resolução de problemas de minimização complexos.



## 7 CONCLUSÕES

Este trabalho ampliou o estudo desenvolvido por Soterroni (2012) e colaboradores quanto à utilização do  $q$ -cálculo na área de otimização por meio do vetor  $q$ -gradiente da função objetivo em métodos clássicos da literatura. O método  $q$ -G nada mais é que uma generalização do método da máxima descida com o uso do vetor  $q$ -gradiente como direção de busca. O método  $q$ -G foi extensivamente comparado com os algoritmos evolutivos em funções teste da literatura e obteve bom desempenho, sobretudo para as funções com diversos extremos locais (SOTERRONI et al., 2015). Dessa forma, este trabalho apresentou outras  $q$ -versões, sendo uma  $q$ -versão do método dos gradientes conjugados de Fletcher e Reeves, denominado método  $q$ -GC e duas  $q$ -versões dos métodos quase-Newton, denominados método  $q$ -BFGS e método  $q$ -DFP, generalizações dos métodos de Broyden-Fletcher-Goldfarb-Shanno e Davidon-Fletcher-Powell, respectivamente. Em todas as generalizações, no limite quando  $q$  tende a 1, as suas versões clássicas são retomadas, isto é, o método  $q$ -G retorna ao método da máxima descida, o método  $q$ -GC retorna ao método dos gradientes conjugados e os métodos  $q$ -BFGS e  $q$ -DFP retornam aos métodos quase-Newton clássicos.

O vetor  $q$ -gradiente usado nas  $q$ -versões para calcular a direção de busca, fornece a esses algoritmos um mecanismo eficaz para escapar de mínimos locais. O processo de busca muda gradualmente de busca global no início para busca local no final do procedimento iterativo. Essa transição é controlada por quatro parâmetros livres, os quais equilibram o compromisso entre busca global e busca local. Além disso, perturbações gaussianas são usadas em uma subsequência de iterações para garantir a convergência dos métodos para o mínimo global em um sentido probabilístico.

A convergência dos métodos baseados no vetor  $q$ -gradiente segue o framework GARS (*Generalized Adaptive Random Search*), desenvolvido por Regis (2010), o qual impõe algumas condições na distribuição de probabilidade que geram os vetores aleatórios em uma subsequência de iterações para garantir a convergência. Essa estratégia permite que o framework GARS seja aplicado a uma ampla gama de algoritmos estocásticos para otimização global, incluindo aqueles que combinam estratégias de busca estocástica e determinística e que realizam busca local e global, como no caso das  $q$ -versões aqui desenvolvidas.

Além de apresentar a prova de convergência para os métodos baseados no vetor  $q$ -gradiente, este trabalho também apresentou uma análise do desempenho computacional, comparando as  $q$ -versões convergentes com suas versões clássicas e com outros métodos amplamente utilizados na literatura de otimização, tais como os algoritmos evolutivos

CMA-ES e ISRES, uma variante da busca aleatória controlada, denominada CRS2-LM, um algoritmo baseado em derivadas por diferenças finitas, denominado IPOPT e o método de busca direta de Nelder-Mead. Em geral, os resultados mostraram que os métodos convergentes baseados no vetor  $q$ -gradiente são bastante promissores e competitivos, especialmente em problemas com vários mínimos locais. Dentre as quatro  $q$ -versões convergentes desenvolvidas neste trabalho, as que obtiveram um melhor desempenho foram os métodos  $q$ -G e  $q$ -GC com perturbações gaussianas.

As  $q$ -versões também foram aplicadas em dois problemas complexos de otimização e os resultados mostraram a viabilidade de seu uso em problemas de difícil solução.

Trabalhos futuros incluem o aprimoramento das  $q$ -versões convergentes com a introdução de restrições mais elaboradas e novas estratégias de obtenção do tamanho do passo na direção do vetor  $q$ -gradiente. Uma vez que o número de avaliações da função objetivo aumenta com a dimensionalidade do problema em questão, estratégias para diminuir o custo computacional dos métodos baseados no vetor  $q$ -gradiente, quando aplicados a problemas com alta dimensionalidade, também serão válidas.

Parte deste trabalho foi publicado no European Journal of Operational Research (EJOR) (GOUVÊA et al., 2016).

## REFERÊNCIAS BIBLIOGRÁFICAS

BACK, T.; HAMMEL, U.; SCHWEFEL, H. P. Evolutionary computation: comments on the history and current state. **IEEE Transactions on Evolutionary Computation**, v. 1, p. 3–17, 1997. 21

BARR, R.; GOLDEN, B.; KELLY, J.; RESENDE, M.; STEWART, W. Designing and reporting on computational experiments with heuristic methods. **Journal of Heuristics**, v. 1, n. 1, p. 9–32, 1995. 55

BLUM, C.; AGUILERA, M. J. B.; ROLI, A.; SAMPELS, M. (Ed.). **Hybrid metaheuristics: an emerging approach to optimization**. 1st. ed. [S.l.]: Springer, 2008. (Studies in Computational Intelligence, v. 114). ISBN 354078294X, 9783540782940. 2

BOOKER, A. J.; JR., J. D.; FRANK, P. D.; SERAFINI, D. B.; TORCZON, V. J.; TROSSET, M. W. **A rigorous framework for optimization of expensive functions by surrogates**. Houston, 1998. 1-24 p. 25

CHAUNDY, T. W. Frank hilton jackson. **Journal of the London Mathematical Society**, s1-37, n. 1, p. 126–128, 1962. 3

CHELOUAH, R.; SIARRY, P. A hybrid method combining continuous tabu search and Nelder-Mead simplex algorithms for the global optimization of multimimima functions. **European Journal of Operational Research**, v. 161, n. 3, p. 636–654, 2005. 2

COLLETTE, Y.; SIARRY, P. **Multiobjective optimization: Principles and case studies**. 1. ed. New York: Springer-Verlag, 2003. 6

CONN, A. R.; SCHEINBERG, K.; TOINT, P. L. On the convergence of derivative-free methods for unconstrained optimization. In: **POWELL, M. J. D. (e.). Approximation theory and optimization**. Cambridge, UK: Cambridge University Press, 1996. p. 83–108. 25

CONN, A. R.; SCHEINBERG, K.; VICENTE, L. N. **Introduction to derivative-free optimization**. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2009. (MPS/SIAM Series on Optimization). ISBN 0-89871-460-5. 33, 34

CURRIE, J.; WILSON, D. I. OPTI: Lowering the Barrier Between Open Source Optimizers and the Industrial MATLAB User. In: SAHINIDIS, N.; PINTO, J. (Ed.). **Foundations of Computer-Aided Process Operations**. Savannah, Georgia, USA: [s.n.], 2012. 55

DARWIN, C. On the origins of species by means of natural selection. **London: Murray**, p. 247, 1859. 21

DRÉO A. PÉTROWSKI, P. S. J.; TAILLARD, E. **Metaheuristics for hard optimization: Methods and case studies**. Berlin: Springer-Verlag, 2006. 24

EASTERLING, D. R.; WATSON, L. T.; MADIGAN, M. L.; CASTLE, B. S.; TROSSET, M. W. Parallel deterministic and stochastic global minimization of functions with very many minima. **Computational Optimization and Applications**, Springer US, New York, v. 57, n. 2, p. 469–492, 2013. 77, 79

ELSTER, C.; NEUMAIER, A. A grid algorithm for bound constrained optimization of noisy functions. **IMA Journal of Numerical Analysis**, v. 15, n. 4, p. 585–608, 1995. 64

ERNST, T. **The history of q-calculus and a new method**. [S.l.], 2000. 31

\_\_\_\_\_. A method for q-calculus. **Journal of Nonlinear Mathematical Physics**, v. 10, n. 4, p. 487–525, 2003. 3

\_\_\_\_\_. **A comprehensive treatment of q-calculus**. [S.l.]: Birkhauser Basel, 2012. 31, 32

FANG, K. T.; ZHANG, Y. T. **Generalized multivariate analysis**. Beijing and Berlin: Science Press and Springer-Verlag, 1990. 50

FLETCHER, R.; REEVES, C. M. Function minimization by conjugate gradients. **Computer Journal**, v. 7, p. 149–154, 1964. 18, 42, 55

FOGEL, L. J.; OWENS, A. J.; WALSH, M. J. **Artificial intelligence through simulated evolution**. New York: John Wiley and Sons, 1966. 21

GAO, D. Canonical dual transformation method and generalized triality theory in nonsmooth global optimization. **Journal of Global Optimization**, v. 17 (1/4), p. 127–160, 2000. 78

\_\_\_\_\_. Duality principles in nonconvex systems: Theory, methods, and applications. **Kluwer Academic**, Norwell, p. 472, 2000. 78

GAO, F.; HAN, L. Implementing the nelder-mead simplex algorithm with adaptive parameters. **Computational Optimization and Applications**, Springer, v. 51, n. 1, p. 259–277, 2012. 29

GENDREAU, M.; POTVIN, J.-Y. (Ed.). **Handbook of metaheuristics**. 2. ed. [S.l.]: Springer US, 2010. (International Series in Operations Research and Management Science, v. 146). 21

GOLDBARG, M. C.; LUNA, H. P. L. **Otimização combinatória e programação linear: modelos e algoritmos**. 2. ed. Rio de Janeiro: Campus/Elsevier, 2005. 1

GOLDBERG, D. E. **Genetic algorithms in search, optimization, and machine learning**. New York: Addison-Wesley Publishing, 1989. 21

GOLDEN, B. L.; WASIL, E. A. Metaheuristics: introduction. In: PARDALOS, P. M.; RESENDE, M. G. C. (Ed.). **Handbook of applied optimization**. New York: Oxford University Press, 2002. p. 123–138. 23

GOUVÊA, E. J. C.; REGIS, R.; SOTERRONI, A. C.; SCARABELLO, M.; RAMOS, F. M. Global optimization using  $q$ -gradients. **European Journal of Operational Research**, v. 251, p. 727–738, 2016. 84

HAGER, R. R. W.; WANG, D. The wave annihilation technique and the design of nonreflective coatings. **SIAM Journal on Applied Mathematics**, v. 60 (4), p. 1388–1424, 2000. 79

HANSEN, P.; MLADENOVIĆ, N. Variable neighborhood search. **Computers & Operations Research**, v. 24, n. 11, p. 1097–1100, 1997. 23, 55

\_\_\_\_\_. Variable neighborhood search: Principles and applications. **European Journal of Operational Research**, v. 130, n. 3, p. 449–467, 2001. 2

HEDAR, A.-R.; FUKUSHIMA, M. Tabu Search directed by direct search methods for nonlinear global optimization. **European Journal of Operational Research**, v. 170, n. 2, p. 329–349, 2006. 2

HERRERA, F.; LOZANO, M.; MOLINA, D. Continuous scatter search: An analysis of the integration of some combination methods and improvement strategies. **European Journal of Operational Research**, v. 169, n. 2, p. 450–476, 2006. 2

HESTENES, M. R. **Conjugate direction methods in optimization**. University of California, USA: Springer-Verlag, 1980. 17

HOLLAND, J. H. **Adaptation in natural and artificial systems**. Cambridge: MIT Press, 1992. 21

IZMAILOV, A.; SOLODOV, M. **ção: Métodos computacionais**. Rio de Janeiro: IMPA, 2007. 11, 12, 17, 19, 20

JACKSON, F. H. A generalization of the functions  $\gamma(n)$  and  $x^n$ . **Proc. Roy Soc. London**, v. 74, p. 64–72, 1904. 31

\_\_\_\_\_. On q-functions and a certain difference operator. **Earth and Environmental Science Transactions of the Royal Society of Edinburgh**, v. 46, n. 2, p. 253–281, 1909. 3, 31, 32

\_\_\_\_\_. On q-definite integrals. **Quart. J. Pure and Appl. Math.**, v. 41, p. 193–203, 1910. 31

\_\_\_\_\_. q-difference equations. **American Journal of Mathematics**, v. 32, p. 307–314, 1910. 31

JOHNSON, S. G. **The NLOpt nonlinear-optimization package**. 2008.  
<http://ab-initio.mit.edu/nlopt>. 55

KAELO, M. A. P. Some variants of the controlled random search algorithm for global optimization. **Journal of Optimization Theory and Applications**, v. 130 (2), p. 253–264, 2006. 55

KIRKPATRICK, S.; JR., C. D. G.; VECCHI, M. P. Optimizing by simulated annealing. **Science**, v. 220, n. 4598, p. 671–680, 1983. 23

LACERDA, E. G. M.; CARVALHO, A. C. P. L. F. de. Introdução aos algoritmos genéticos. In: CONGRESSO NACIONAL DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 19, 1999, Rio de Janeiro. **Proceedings...** Rio de Janeiro: Pontifícia Universidade Católica do Rio de Janeiro, 1999. v. 2, p. 51–126. 23

LAGARIAS, J. C.; REEDS, J. A.; WRIGHT, M. H.; WRIGHT, P. E. Convergence properties of the nelder–mead simplex method in low dimensions. **SIAM Journal on optimization**, SIAM, v. 9, n. 1, p. 112–147, 1998. 26, 27

LOH, W.-L. et al. On latin hypercube sampling. **The annals of statistics**, Institute of Mathematical Statistics, v. 24, n. 5, p. 2058–2080, 1996. 80

LUCIDI, S.; SCIANDRONE, M. On the global convergence of derivative-free methods for unconstrained minimization. **SIAM Journal on Optimization**, v. 13, p. 97–116, 2002. 25

LUENBERGER, D. G. **Linear and nonlinear programming**. 2. ed. Massachusetts: Addison-Wesley, 1984. 5, 15, 17, 18, 19, 20

MARAZZI, M.; NOCEDAL, J. Wedge trust region methods for derivative free optimization. **Mathematical Programming**, v. 91, p. 289–305, 2002. 64

- MARTÍ, R.; LAGUNA, M.; GLOVER, F. Principles of scatter search. **European Journal of Operational Research**, v. 169, n. 2, p. 359–372, 2006. [2](#)
- METROPOLIS, N.; ROSENBLUTH, A. W.; ROSENBLUTH, M. N.; TELLER, A. H. Equation of state calculations by fast computing machines. **The Journal of Chemical Physics**, v. 21, n. 6, p. 1087–1092, 1953. [23](#)
- MORÉ, J. J.; GARBOW, B. S.; HILLSTROM, K. E. Testing unconstrained optimization software. **ACM Transactions on Mathematical Software**, v. 7, n. 1, p. 17–41, 1981. [53](#)
- MORÉ, J. J.; WILD, S. M. Benchmarking derivative-free optimization algorithms. **SIAM Journal on Optimization**, v. 20, n. 1, p. 172–191, 2009. [56](#), [63](#), [64](#)
- NELDER, J.; MEAD, R. A simplex method for function minimization. **The Computer Journal**, Br Computer Soc, v. 7, n. 4, p. 308, 1965. [25](#), [26](#), [55](#)
- NOCEDAL, J.; WRIGHT, S. J. **Numerical optimization**. 2. ed. New York: Springer-Verlag New York, Inc, 2006. [7](#), [8](#), [13](#), [14](#), [17](#), [36](#)
- PILLO, G. D.; PALAGI, L. Nonlinear programming: Introduction. In: PARDALOS, P. M.; RESENDE, M. G. C. (Ed.). **Handbook of applied optimization**. New York: Oxford University Press, 2002. p. 263–268. [13](#), [19](#), [20](#), [21](#), [36](#)
- RANGAIAH, G. P. **Stochastic global optimization - techniques and applications in chemical engineering**. Singapore: World Scientific Publishing, 2010. [23](#)
- RECHENBERG, I. (Ed.). **Evolutions strategie: optimierung technischer systeme nach prinzipien der biologischen evolution**. [S.l.]: Frommann-Holzboog, 1973. [21](#)
- REGIS, R. G. Convergence guarantees for generalized adaptive stochastic search methods for continuous global optimization. **European Journal of Operational Research**, v. 207, n. 3, p. 1187–1202, 2010. [49](#), [50](#), [51](#), [52](#), [83](#)
- \_\_\_\_\_. The calculus of simplex gradients. **Optimization Letters**, v. 9, n. Issue 5, p. 845–865, 2015. [33](#), [34](#)
- RUNARSSON, T. P.; YAO, X. Search biases in constrained evolutionary optimization. **IEEE Transactions on Systems, Man, and Cybernetics, Part C**, v. 35, n. 2, p. 233–243, 2005. [23](#), [55](#)
- SOLANO, H. L. **Introducción a la teoría de probabilidad**. Barranquilla, Col.: Editorial Universidad del Norte, 2014. [49](#)

- SOONG, T. T. **Random differential equations in science and engineering**. [S.l.]: Elsevier Science, 1973. (Mathematics in Science and Engineering). ISBN 9780080956121. 51
- SOTERRONI, A. C. **O método do  $q$ -gradiente para otimização global**. PhD Thesis (Tese de Doutorado em Computação Aplicada) — INPE, São José dos Campos, 2012. 3, 32, 33, 35, 37, 83
- SOTERRONI, A. C.; GALSKI, R. L.; RAMOS, F. M. The  $q$ -gradient vector for unconstrained continuous optimization problems. In: HU, B.; MORASCH, K.; PICKL, S.; SIEGLE, M. (Ed.). **Operations Research Proceedings 2010**. [S.l.]: Springer Berlin Heidelberg, 2011. p. 365–370. 2, 3, 35
- \_\_\_\_\_. The  $q$ -gradient method for global optimization. **arXiv:1209.2084**, math.OC, 2012. 3, 35
- \_\_\_\_\_. The  $q$ -gradient method for continuous global optimization. **11th Internacional Conference of Numerical Analysis and Applied Mathematics**, v. 1558, p. 2389–2393, 2013. 3, 35
- SOTERRONI, A. C.; GALSKI, R. L.; SCARABELLO, M. C.; RAMOS, F. M. The  $q$ -g method: A  $q$ -version of the steepest descent method for global optimization. **SpringerPlus**, n. 4, 2015. 3, 32, 33, 35, 37, 38, 55, 83
- SOUZA, M.; DINIZ-EHRHARDT, M. **Otimização e análise convexa: aspectos teóricos e aplicações**. [S.l.]: Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica, 2011. (Relatório de pesquisa). 7
- SPALL, J. C. **Introduction to stochastic search and optimization**. Hoboken, New Jersey: John Wiley and Sons, 2003. 21
- SPEARS, W. M. **Evolutionary algorithms : the role of mutation and recombination**. Berlin, Heidelberg, New York: Springer, 2000. (Natural computing series). ISBN 3-540-66950-7. 21
- SPENDLEY, W.; HEXT, G. R.; HIMSWORTH, F. R. Sequential application for simplex designs in optimisation and evolutionary operation. **Technometrics**, v. 4, p. 441–461, 1962. 25
- SUGANTHAN, P.; HANSEN, N.; LIANG, J. J.; DEB, K.; CHEN, Y. P.; AUGER, A.; TIWARI, S. **Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization**: Technical report. [S.l.], 2005. 53

SUN, W.; YUAN, Y.-X. **Optimization theory and methods: nonlinear programming**. 1. ed. New York: Springer Optimization and Its Applications, 2006. 6, 10, 15, 16, 19, 20

TAO, T. **An introduction to measure theory**. Los Angeles, CA: University of California, 2011. 51

VANDERPLAATS, G. N. **Numerical optimization techniques for engineering design: with applications**. New York: McGraw-Hill, Inc, 1984. 1, 9, 10, 12, 14

WÄCHTER, A.; BIEGLER, L. T. 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, may 2006. 55

WRIGHT, M. H. Direct search methods: Once scorned, now respectable. In: PROCEEDINGS OF THE 1995 DUNDEE BIENNIAL CONFERENCE IN NUMERICAL ANALYSIS. **Numerical analysis 1995**. Harlow, UK: Addison-Wesley, Longman: In D. F. GriOths, G. A. Watson (Eds.), 1995. p. 191–208. 26

YANG, X.-S. **Engineering optimization: an introduction with metaheuristic applications**. [S.l.]: John Wiley & Sons, 2010. 24

ZABINSKY, Z. B. **Stochastic adaptive search for global optimization**. [S.l.]: Kluwer Academics Publishers, 2003. (Nonconvex optimization and its applications, v. 72). 2