



## TRATAMENTO DO PROBLEMA DE OTIMIZAÇÃO COM RESTRIÇÕES USANDO EVOLUÇÃO DIFERENCIAL

Daniel da Costa e Silva Coutinho

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Civil, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Civil.

Orientador: Beatriz de Souza Leite Pires de Lima

Rio de Janeiro  
Setembro de 2013

TRATAMENTO DO PROBLEMA DE OTIMIZAÇÃO COM RESTRIÇÕES  
USANDO EVOLUÇÃO DIFERENCIAL

Daniel da Costa e Silva Coutinho

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO  
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA  
(COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE  
DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE  
EM CIÊNCIAS EM ENGENHARIA CIVIL.

Examinada por:

---

Prof<sup>ª</sup>. Beatriz de Souza Leite Pires de Lima, D.Sc.

---

Prof. José Antonio Fontes Santiago, D.Sc.

---

Prof. Carl Horst Albrecht, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

SETEMBRO DE 2013

Coutinho, Daniel da Costa e Silva

Tratamento do Problema de Otimização com Restrições Usando Evolução Diferencial / Daniel da Costa e Silva Coutinho. – Rio de Janeiro: UFRJ/COPPE, 2013.

X, 65 p.: il.; 29,7 cm.

Orientador: Beatriz de Souza Leite Pires de Lima

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia Civil, 2013.

Referências Bibliográficas: p. 42-45.

1. Otimização. 2. Algoritmos Evolutivos. 3. Evolução Diferencial. 4. Tratamento de Restrições. I. Lima, Beatriz de Souza Leite Pires de. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Civil. III. Título.

## Agradecimentos

Aos meus pais pelo apoio e incentivo ao longo de toda a minha vida, à Professora Beatriz Lima pela orientação e carinho, à Cristiana de Siqueira Gonçalves pelo amor, motivação e paciência e aos amigos do LAMCSO pelo suporte e amizade.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

TRATAMENTO DO PROBLEMA DE OTIMIZAÇÃO COM RESTRIÇÕES  
USANDO EVOLUÇÃO DIFERENCIAL

Daniel da Costa e Silva Coutinho

Setembro/2013

Orientador: Beatriz de Souza Leite Pires de Lima

Programa: Engenharia Civil

Problemas de otimização são comumente encontrados em diversas áreas do conhecimento, e a busca por algoritmos para a solução desses problemas gerou inúmeras contribuições na literatura. Espaços de busca com restrições exigem metodologias específicas em adição aos algoritmos existentes para encontrar uma solução ótima e que não viole as restrições definidas. Nessa dissertação é apresentado um algoritmo baseado em Evolução Diferencial que utiliza uma estratégia em dois estágios para solucionar problemas de otimização com restrições. O algoritmo é executado em problemas conhecidos da literatura e seus resultados são comparados aos de estratégias renomadas.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

## CONSTRAINED OPTIMIZATION USING DIFFERENTIAL EVOLUTION

Daniel da Costa e Silva Coutinho

September/2013

Advisor: Beatriz de Souza Leite Pires de Lima

Department: Civil Engineering

Optimization problems are commonly found in many domains, and the search for optimization problem solvers has created countless contributions to scientific literature. Constrained search spaces need specific methods beside existing algorithms to find optimal solutions that do not violate the constraints. This dissertation presents a Differential Evolution based algorithm that uses a two-stage strategy for solving constrained optimization problems. The algorithm is run against known problems from scientific literature and its results are compared to those of well-known algorithms.

# Sumário

Índice de Figuras .....	ix
Índice de Tabelas .....	x
1 Introdução .....	1
1.1 Motivação e Objetivo .....	2
1.2 Organização do Trabalho .....	3
2 Otimização .....	4
2.1 Formulação dos Problemas .....	4
2.2 Métodos para a Solução .....	5
2.2.1 Métodos Determinísticos .....	5
2.2.2 Métodos Heurísticos .....	6
2.2.3 Métodos Inspirados na Natureza .....	7
3 Evolução Diferencial .....	8
3.1 Introdução .....	8
3.2 O Algoritmo .....	9
3.2.1 Inicialização de Vetores de Parâmetros .....	9
3.2.2 Mutaç�o com Vetores de Diferena .....	10
3.2.3 Recombinao .....	11
3.2.4 Seleo .....	13
3.3 Variaes do DE na Literatura .....	14
4 T�cnicas para o Tratamento de Restries .....	19

4.1	Introdução .....	19
4.2	Técnicas Empregadas na Literatura .....	20
4.3	Evolução Diferencial com Tratamento de Restrições.....	24
5	Metodologia Proposta.....	25
5.1	Introdução .....	25
5.2	Primeiro Estágio.....	25
5.2.1	Buscas Sucessivas .....	26
5.2.2	Avaliação de Soluções.....	27
5.2.3	Variante Proposta da Evolução Diferencial.....	29
5.3	Segundo Estágio .....	30
6	Estudos de Caso.....	32
6.1	Metodologia .....	32
6.2	Parâmetros do algoritmo .....	33
6.3	Resultados .....	34
7	Conclusão .....	40
7.1	Trabalhos Futuros .....	41
	Referências .....	42
	Anexo A.....	46

# Índice de Figuras

Figura 1: Efeito das restrições no espaço de busca. ....	5
Figura 2 - Etapas do Algoritmo DE.....	9
Figura 3 - Esquema de Mutação em Espaço 2D.....	11
Figura 4 - Pseudocódigo do primeiro estágio.....	26
Figura 5 - Ranking individual por restrição.....	29
Figura 6 - Seleção proporcional a avaliação.....	30

# Índice de Tabelas

Tabela 1 - Propriedades dos Problemas da G-Suite .....	33
Tabela 2 - Resultados do TSCH-RBDE na G Suite - 240.000 AFO .....	35
Tabela 3 - Resultados para comparação .....	37
Tabela 4 - Resultados para comparação - Continuação.....	38

# 1 Introdução

A otimização pode ser definida como o processo de tornar um sistema tão eficiente quanto possível. Esse conceito é tão inerente à sobrevivência humana, onde otimiza-se o aproveitamento de recursos escassos, quanto a própria sociedade moderna, onde sistemas complexos seriam inalcançáveis sem configurações ótimas.

Problemas que necessitam de otimização são comuns em diversas áreas do conhecimento e, quando modelados matematicamente, podem ser solucionados por métodos de otimização. O modelo matemático de problemas de otimização tem a forma de uma *função objetivo* e um *espaço de busca*.

Uma função objetivo tem como variáveis os aspectos do problema real que devem ser ajustados, e como resultado um valor quantitativo representando a qualidade desse conjunto de aspectos em relação à característica que se deseja otimizar (lucro, eficiência ou tempo, por exemplo).

O espaço de busca e suas propriedades são definidos pelas variáveis usadas na função objetivo. O espaço é  $D$ -dimensional, onde  $D$  é definido pelo número de variáveis, delimitado pelos limites inferiores e superiores dos valores e contínuo ou discreto, dependendo dos valores que podem ser assumidos por elas. Uma solução candidata de um problema de otimização é um ponto no espaço de busca, que representa um conjunto de valores para as variáveis.

Adicionalmente, problemas de otimização podem possuir *restrições* advindas de limitações do problema real (limitações físicas ou financeiras, por exemplo). Restrições dividem o espaço de busca em áreas viáveis e inviáveis, onde soluções candidatas viáveis são aquelas que respeitam a todas as restrições.

O objetivo do processo de otimização é encontrar a melhor configuração de valores viáveis para as variáveis do problema, de forma a maximizar ou minimizar a característica desejada - ou seja, encontrar a melhor solução viável, cujo valor resultante na função objetivo é o menor (*minimização*) ou maior (*maximização*) possível.

## **1.1 MOTIVAÇÃO E OBJETIVO**

A solução de problemas de otimização com restrições ainda é um processo computacional caro, e o tempo um recurso escasso para muitos projetos. Por essa razão existe uma demanda pelo desenvolvimento de algoritmos capazes de resolver problemas de otimização de forma rápida e satisfatória.

O objetivo desta dissertação é propor uma nova metodologia para o tratamento de restrições em problemas contínuos de otimização. Para este propósito, foi empregada uma estratégia em duas etapas, onde as restrições são solucionadas separadamente da função objetivo. O algoritmo Evolução Diferencial (STORN & PRICE, 1995) foi utilizado como base para as duas etapas por sua reconhecida performance e simplicidade de implementação, que facilitou customizações.

## **1.2 ORGANIZAÇÃO DO TRABALHO**

Este trabalho está organizado da seguinte forma: o capítulo dois aborda a formulação de problemas de otimização e as categorias de soluções disponíveis na literatura, enquanto o capítulo 3 apresenta de forma detalhada o algoritmo Evolução Diferencial (DE), utilizado na metodologia proposta, juntamente com as principais variantes disponíveis na literatura para a solução de problemas sem restrições. O capítulo 4 apresenta o conceito de restrições e algumas das principais estratégias de tratamento disponíveis na literatura, assim como suas aplicações com o algoritmo DE. O capítulo 5 descreve o método proposto como um todo e suas etapas, detalhando as alterações efetuadas no algoritmo DE para sua realização. O capítulo 6 apresenta os estudos executados com o algoritmo proposto e compara seus resultados aos de estratégias renomadas da literatura. Por fim, o capítulo 7 termina o trabalho com a conclusão.

## 2 Otimização

### 2.1 FORMULAÇÃO DOS PROBLEMAS

Para que seja possível utilizar um algoritmo para resolver um problema de otimização é necessário gerar um modelo matemático capaz de representá-lo corretamente. Modelos matemáticos de problemas de otimização são compostos por uma função objetivo que definirá o critério de desempenho de uma solução, e por um espaço de busca que definirá o escopo da busca e restrições. Modelos de problemas de otimização são normalmente apresentados no formato

$$\begin{aligned} & \text{Otimizar } f(\vec{X}) \\ & \vec{X} \in \mathfrak{R}^D \\ & \text{Sujeito a:} \\ & \begin{cases} g(\vec{X}) \leq 0 \\ h(\vec{X}) = 0 \end{cases} \end{aligned} \tag{1}$$

onde otimizar pode referir-se a maximizar ou minimizar,  $\vec{X}$  é um vetor contendo os parâmetros sendo otimizados,  $f(\vec{X})$  é a função objetivo que relacionará os parâmetros no vetor  $\vec{X}$  a uma medida de desempenho e  $g(\vec{X})$  e  $h(\vec{X})$  são, respectivamente, restrições de desigualdade e igualdade impostas a  $\vec{X}$ .

O conjunto de restrições de um problema de otimização adiciona complexidade ao espaço de busca tornando inviáveis todos os pontos que não se adequam as funções de restrição definidas. Caso o ponto ótimo da função objetivo se encontre em uma área inviável, a solução do problema de otimização será deslocada para o ponto de melhor

valor em área viável, e adicionalmente, ótimos locais podem ser criados nas fronteiras da área de inviabilidade, como ilustrado na Figura 1.

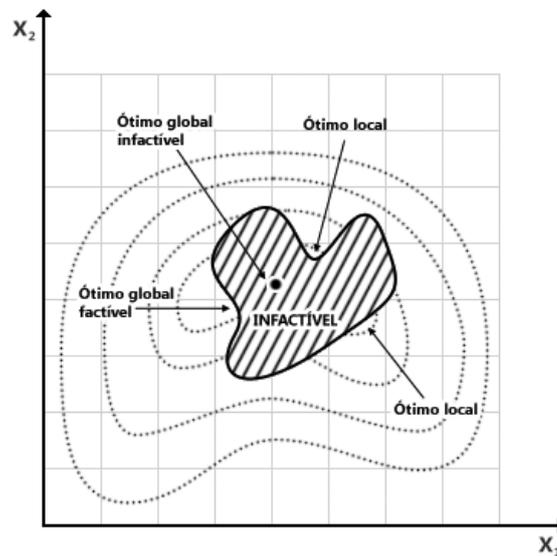


Figura 1: Efeito das restrições no espaço de busca.

## 2.2 MÉTODOS PARA A SOLUÇÃO

Existem hoje diversas metodologias para a solução de problemas de otimização que podem, fundamentalmente, ser divididas em duas categorias: determinísticas e heurísticas.

### 2.2.1 Métodos Determinísticos

Métodos determinísticos são métodos de solução de problemas de otimização matemáticos que podem ser utilizados em problemas nos quais as funções objetivo são contínuas e diferenciáveis ao longo do espaço de busca. Os métodos dessa categoria se enquadram em duas principais categorias da Programação Matemática de acordo com o

tipo de problemas que são capazes de solucionar: Programação Linear e Programação Não-Linear. A Programação Linear trata problemas com funções objetivo e de restrição lineares dos parâmetros otimizados, enquanto a Programação Não-Linear trata os problemas com funções não-lineares.

Como exemplos de métodos determinísticos pode-se citar o Simplex (DANTZIG & THAPA, 1997) para Programação Linear e o Método de Newton (ORTEGA & RHEINBOLDT, 2000) para Programação Não-Linear.

### **2.2.2 Métodos Heurísticos**

Os métodos heurísticos possuem como principal característica o uso do valor das avaliações da função objetivo para gerar informação sobre o problema e direcionar a busca em direção ao ótimo global.

O processo de otimização por heurísticas é feito de forma iterativa, e a busca normalmente exige um grande número de avaliações e conseqüentemente mais poder computacional do que os métodos determinísticos, porém sua execução é desacoplada do problema, permitindo uma melhor adequação a problemas diversos.

Entre os métodos desta categoria destacam-se a Busca Tabu (GLOVER, 1986) e a Busca em Vizinhaça Variável (MLADENOVIC & HANSEN, 1997).

O uso de heurísticas associadas gerou uma subclasse de métodos denominada meta-heurísticas, onde o grupo de métodos de otimização inspirados na natureza se destaca.

### **2.2.3 Métodos Inspirados na Natureza**

A computação inspirada na natureza é definida pelo uso de metáforas e modelos de sistemas naturais em algoritmos para a solução de problemas. O movimento tomou forma com a publicação do primeiro modelo matemático de um neurônio (MCCULLOCH & PITTS, 1943) e subsequente criação da primeira rede neural artificial (ROSENBLATT, 1957).

Utilizando as teorias de Darwin e Mendel, métodos de otimização baseados na evolução e seleção natural surgiram com o algoritmo Estratégia Evolutiva (RECHENBERG, 1971) e se popularizaram com o Algoritmo Genético (HOLLAND, 1975), culminando em uma nova classe de Algoritmos Evolutivos.

## 3 Evolução Diferencial

### 3.1 INTRODUÇÃO

A Evolução Diferencial (DE, do inglês *Differential Evolution*) é um algoritmo evolutivo apresentado por Rainer Storn e Kenneth Price (STORN & PRICE, 1995) que demonstrou grande eficiência no primeiro ICEO (*International Contest on Evolutionary Optimization*) - um concurso de otimização para algoritmos evolutivos organizado em conjunto com a *IEEE International Conference on Evolutionary Computation* (CEC) de 1996. Descrições detalhadas do algoritmo foram posteriormente apresentadas em periódicos (STORN & PRICE, 1997).

O algoritmo DE opera de forma essencialmente similar a outros algoritmos evolutivos. Uma população de indivíduos (soluções) é gerada aleatoriamente e sucessivamente melhorada em um processo iterativo até que um critério de parada predefinido seja alcançado.

Uma iteração, denominada geração em algoritmos evolutivos, é executada no DE através de uma operação de mutação repetida para cada indivíduo da população. Nesta operação, o indivíduo selecionado da população é denominado *vetor alvo*, e um novo indivíduo denominado *vetor doador* é criado adicionando-se uma diferença ponderada entre dois indivíduos da população a um terceiro indivíduo. O vetor doador é então re combinado com o vetor alvo para gerar um *vetor experimental*, que substituirá o vetor alvo na população caso seja melhor avaliado na função objetivo.

Este esquema adaptativo que extrai da população informação de distancias e direção para gerar desvios aleatórios apresenta uma excelente convergência. Algumas variantes deste esquema foram desenvolvidas e são apresentadas mais adiante. As próximas seções apresentam o esquema do algoritmo detalhadamente.

## 3.2 O ALGORITMO

O DE é um algoritmo de simples implementação e tem sua execução dividida em quatro etapas, apresentadas na Figura 2. Cada etapa é separadamente explicada nas subseções a seguir.

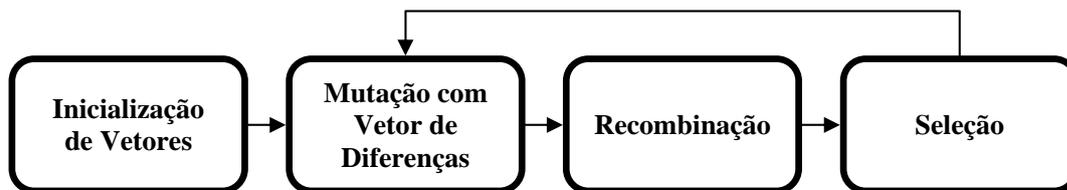


Figura 2 - Etapas do Algoritmo DE

### 3.2.1 Inicialização de Vetores de Parâmetros

O DE busca pelo ponto ótimo global em um espaço  $D$ -dimensional  $\mathfrak{R}^D$ . A busca é inicializada com uma população de  $NP$  vetores  $D$ -dimensionais de valores reais, cada um representando uma solução candidata para o problema de otimização. Uma vez que os vetores de parâmetros assumirão diferentes valores em cada geração, é assumida a seguinte notação para a representação de um vetor  $i$  da população em determinada geração  $G$ :

$$\vec{X}_{i,G} = [x_{1,i,G}, x_{2,i,G}, x_{3,i,G}, \dots, x_{D,i,G}]. \quad (2)$$

Como em outros algoritmos evolutivos, a população inicial do DE é gerada aleatoriamente, respeitando os limites inferiores e superiores dos parâmetros. Sendo estes limites definidos pelo problema na forma  $\vec{X}_{min} = [x_{1,min}, x_{2,min}, \dots, x_{D,min}]$  e  $\vec{X}_{max} = [x_{1,max}, x_{2,max}, \dots, x_{D,max}]$ , o componente  $j$  do vetor  $i$  será gerado como

$$x_{j,i,0} = x_{j,min} + rand_{i,j}[0,1] \cdot (x_{j,max} - x_{j,min}) \quad (3)$$

onde  $rand_{i,j}[0,1]$  é um número aleatório de distribuição uniforme entre 0 e 1, calculado individualmente para cada componente do vetor.

### 3.2.2 Mutaç o com Vetores de Diferen a

No contexto de algoritmos evolucion rios a muta o   conhecida como o processo de perturbar uma solu o candidata com um elemento aleat rio. No DE, para cada vetor alvo selecionado sequencialmente da popula o   gerado um vetor doador para ser usado posteriormente no processo de recombina o.

Na forma mais simples de muta o do DE, o vetor doador de um alvo de posi o  $i$  na popula o   criado atrav s da sele o aleat ria de tr s outros vetores da gera o atual:  $\vec{X}_{r_1^i,G}, \vec{X}_{r_2^i,G}, \vec{X}_{r_3^i,G}$ . A diferen a entre quaisquer dois desses vetores   multiplicada por um fator de pondera o  $F$  e posteriormente adicionada ao terceiro vetor para formar

o vetor doador. Esse processo é exemplificado na Figura 3 e pode ser representado como

$$\vec{V}_{i,G} = \vec{X}_{r_1,G} + F \cdot (\vec{X}_{r_2,G} - \vec{X}_{r_3,G}). \quad (4)$$

onde  $\vec{V}_{i,G}$  representa o vetor doador, criado individualmente para cada solução da geração. O valor de  $F$  é predefinido como parâmetro do algoritmo.

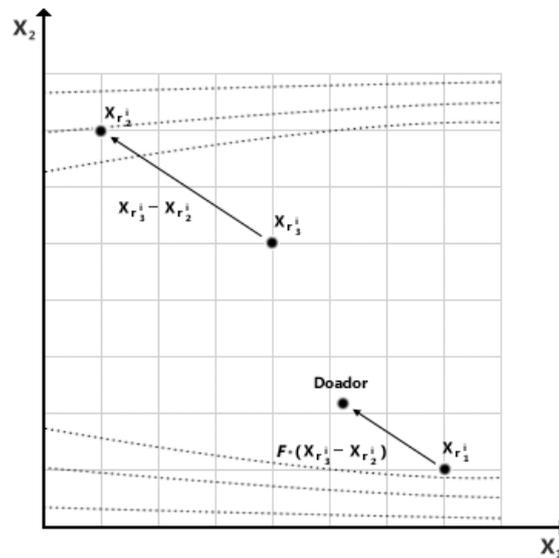


Figura 3 - Esquema de Mutação em Espaço 2D.

### 3.2.3 Recombinação

Para manter a diversidade da população, um operador de recombinação é aplicado para gerar o vetor experimental, onde o vetor doador troca seus componentes com o vetor alvo. O DE pode utilizar dois métodos de recombinação: exponencial e binomial.

Na recombinação exponencial um índice  $n$  é escolhido aleatoriamente dentre  $[1, D]$  para determinar o ponto de início da troca entre os componentes dos vetores alvo e doador. Um segundo valor  $L$  determina o número de componentes a serem trocados. Após a determinação dos valores de  $n$  e  $L$  o processo de recombinação para a geração de vetor experimental  $\vec{U}_{i,G}$  é definido como

$$u_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{se } j = \langle n \rangle_D, \langle n + 1 \rangle_D, \dots, \langle n + L - 1 \rangle_D \\ x_{j,i,G}, & \text{caso contrário} \end{cases} \quad (5)$$

onde  $\langle \ \rangle_D$  representa uma operação de módulo com divisor  $D$ . O valor de  $L$  é determinado de acordo com o seguinte pseudocódigo:

```

L = 0
FAÇA
    L = L + 1
ENQUANTO ((rand(0,1) ≤ Cr) E (L ≤ D))

```

onde  $Cr$  (do inglês *Crossover Rate*) é uma taxa de recombinação predefinida como parâmetro do algoritmo.

No caso da recombinação binomial, a troca é efetuada em cada um dos componentes sempre que um número aleatoriamente gerado entre 0 e 1 é menor ou igual ao valor de  $Cr$ . A recombinação binomial pode ser definida como

$$u_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{se } (rand_{i,j}[0,1] \leq Cr \text{ ou } j = j_{rand}) \\ x_{j,i,G}, & \text{caso contrário} \end{cases} \quad (6)$$

onde  $rand_{i,j}[0,1]$  é um número aleatório entre 0 e 1, calculado individualmente para cada componente do vetor, e  $j_{rand}$  é um número aleatório dentre  $[1, D]$ , calculado para cada vetor, representando o índice de um componente que será trocado independente do valor de  $Cr$  - garantindo assim que o vetor experimental receba ao menos um componente do vetor doador.

### 3.2.4 Seleção

Para manter o tamanho da população constante ao longo da otimização, uma operação de seleção é usada para determinar se o vetor experimental sobreviverá na próxima geração. A operação de seleção pode ser descrita como

$$\vec{X}_{i,G+1} = \begin{cases} \vec{U}_{i,G}, & \text{se } (f(\vec{U}_{i,G}) \leq f(\vec{X}_{i,G})) \\ \vec{X}_{i,G}, & \text{se } (f(\vec{U}_{i,G}) > f(\vec{X}_{i,G})) \end{cases} \quad (7)$$

onde  $f(\vec{X})$  é a função objetivo do problema sendo otimizado. Portanto, caso o vetor experimental possua uma melhor ou igual avaliação na função objetivo, ele substituirá o vetor alvo na população da próxima geração. A substituição de vetores com melhor avaliação garante populações melhores ou igualmente qualificadas ao longo das gerações, enquanto a substituição em caso de avaliações iguais faz com que o DE consiga mover a população em superfícies planas.

Por fim, o processo de otimização do DE utiliza critérios de parada similares aos outros métodos existentes na literatura, entre eles: quando é atingido um número fixo de gerações definido pela complexidade do problema, quando o melhor indivíduo da

população não muda após um número fixo de gerações, ou quando um valor de ótimo específico é alcançado na avaliação de uma solução pela função objetivo.

### 3.3 VARIAÇÕES DO DE NA LITERATURA

Storn e Price (PRICE & STORN, 2005, PRICE, 1999) apresentaram, juntamente com o algoritmo básico do DE, variantes explorando diferentes combinações dos métodos de recombinação, vetor receptor da diferença ponderada e número de vetores de diferença utilizados na mutação. Essas variantes são nomeadas no formato "*DE / vetor receptor / número de diferenças / método de recombinação*". Por exemplo, a variante básica previamente apresentada, usando recombinação exponencial, é denominada *DE/rand/1/exp*. Como o único processo diferente entre as variantes é a geração do vetor doador, as mesmas podem ser representadas da seguinte forma:

$$DE/best/1: \vec{V}_{i,G} = \vec{X}_{best,G} + F \cdot (\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) \quad (8)$$

$$DE/target-to-best/2: \vec{V}_{i,G} = \vec{X}_{i,G} + F \cdot (\vec{X}_{best,G} - \vec{X}_{i,G}) + F \cdot (\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) \quad (9)$$

$$DE/best/2: \vec{V}_{i,G} = \vec{X}_{best,G} + F \cdot (\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) + F \cdot (\vec{X}_{r_3^i,G} - \vec{X}_{r_4^i,G}) \quad (10)$$

$$DE/rand/2: \vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F \cdot (\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G}) + F \cdot (\vec{X}_{r_4^i,G} - \vec{X}_{r_5^i,G}) \quad (11)$$

onde  $r_1^i, r_2^i, r_3^i, r_4^i$  e  $r_5^i$  são índices escolhidos aleatoriamente entre  $[1, NP]$ , mutuamente exclusivos e diferentes do índice  $i$  do vetor alvo, e  $\vec{X}_{best,G}$  é o indivíduo com melhor avaliação na população da geração atual.

Somando as quatro variantes apresentadas acima com o *DE/rand/1* visto em detalhes neste capítulo, e multiplicando pelo número de variações da técnica de recombinação (binomial e exponencial), ao todo dez variantes do DE foram apresentadas por Storn e Price.

Além dos esquemas inicialmente propostos, diversas variantes do algoritmo básico foram propostas posteriormente. A seguir são apresentadas as variantes de melhor performance.

A variante *DE/current-to-rand/1*, apresentada por PRICE (1999), substitui os operadores de recombinação tradicionais (binomial ou exponencial) por uma combinação linear dos vetores doador e alvo, na forma de:

$$\vec{U}_{i,G} = \vec{X}_{i,G} + k_i \cdot (\vec{V}_{i,G} - \vec{X}_{i,G}) \quad (12)$$

onde  $k_i$  é o coeficiente de combinação, parâmetro que, de acordo com o autor, demonstra melhor eficiência quando tem seu valor escolhido aleatoriamente entre [0,1].

FAN & LAMPINEN (2003) propuseram um operador de mutação trigonométrico utilizando a avaliação dos vetores aleatoriamente selecionados para definir a influência de cada um no vetor doador. Após a seleção dos indivíduos aleatórios, como no algoritmo *DE/rand/1*, três coeficientes de ponderação são calculados para os indivíduos aleatórios como

$$p_1 = |f(\vec{X}_{r_1, G}^i)| / p' \quad (13a)$$

$$p_2 = |f(\vec{X}_{r_2, G}^i)| / p' \quad (13b)$$

$$p_3 = |f(\vec{X}_{r_3, G}^i)| / p' \quad (13c)$$

onde

$$p' = |f(\vec{X}_{r_1, G}^i)| + |f(\vec{X}_{r_2, G}^i)| + |f(\vec{X}_{r_3, G}^i)| \quad (13d)$$

e  $f()$  é a função objetivo. O processo de mutação trigonométrica pode então ser expresso como:

$$\begin{aligned} \vec{V}_{i, G} = & \frac{(\vec{X}_{r_1, G}^i + \vec{X}_{r_2, G}^i + \vec{X}_{r_3, G}^i)}{3} + (p_2 - p_1) \cdot (\vec{X}_{r_1, G}^i - \vec{X}_{r_2, G}^i) \\ & + (p_3 - p_2) \cdot (\vec{X}_{r_2, G}^i - \vec{X}_{r_3, G}^i) \\ & + (p_1 - p_3) \cdot (\vec{X}_{r_3, G}^i - \vec{X}_{r_1, G}^i) \text{ se } rand[0,1] \leq \Gamma \\ \vec{V}_{i, G} = & \vec{X}_{r_1, G}^i + F \cdot (\vec{X}_{r_2, G}^i - \vec{X}_{r_3, G}^i) \text{ caso contrário} \end{aligned} \quad (14)$$

onde  $\Gamma$  é a taxa de mutação trigonométrica, definida no intervalo  $[0,1]$  de forma que a mutação trigonométrica é usada com uma probabilidade de  $\Gamma$ , enquanto a mutação da variante  $DE/rand/l$  é usada com uma probabilidade de  $1-\Gamma$ . A taxa de mutação trigonométrica permite um equilíbrio entre a convergência rápida oferecida pelo novo operador e a busca do ótimo global, evitando convergências precoces para ótimos locais.

Uma versão foi proposta por RAHNAMAYAN, TIZHOOSH & SALAMA (2008) utilizando o aprendizado por opostos. TIZHOOSH (2005) define o oposto  $\check{x}$  de um número  $x$  no intervalo  $[a,b]$  como  $\check{x} = a + b - x$ . Dessa maneira, o oposto de um vetor  $\vec{X}$  é definido como

$$\check{X} = [\check{x}_1, \check{x}_2, \dots, \check{x}_D]. \quad (15)$$

No trabalho, vetores opostos são utilizados em três estágios diferentes do DE básico. Na geração da população inicial, uma população oposta é criada, e os  $NP$  melhores indivíduos do conjunto formado pelas duas populações são usados para iniciar a otimização. Esse mesmo processo pode se repetir ao fim de cada geração para compor a geração seguinte com uma probabilidade de 0.04%. Por fim, sempre que um novo melhor indivíduo é encontrado, um indivíduo oposto ao novo melhor é criado, e o melhor entre os dois é usado para substituir o antigo.

DAS, ABRAHAM, CHAKRABORTY & KONAR (2009) apresentaram uma variante do *DE/target-to-best/1* que utiliza apenas vetores na vizinhança do vetor alvo para a definição do doador. O trabalho considera a população como se conectada em topologia de anel, e os limites de uma vizinhança são definidos por um parâmetro  $k$  entre 1 e  $(NP-1)/2$  tal que a vizinhança de um vetor  $\vec{X}_{i,G}$  seja formada pelos vetores  $\vec{X}_{i-k,G}, \dots, \vec{X}_{i,G}, \dots, \vec{X}_{i+k,G}$ .

Devido à diferente aplicabilidade das diversas estratégias de mutação, QIN, HUANG & SUGANTHAN (2009) desenvolveram o algoritmo autoadaptativo SaDE,

que altera tanto os parâmetros do algoritmo quanto a estratégia de mutação utilizada ao longo do processo de otimização. As variantes *DE/rand/1/bin*, *DE/rand-to-best/2/bin*, *DE/rand/2/bin* e *DE/current-to-rand/1* foram escolhidas pelos autores para formar um conjunto de estratégias candidatas. Para cada vetor na população uma estratégia é escolhida de acordo com uma probabilidade de sucesso aprendida nas últimas *LP* gerações, sendo *LP* um parâmetro do algoritmo representando um período de aprendizado.

## 4 Técnicas para o Tratamento de Restrições

### 4.1 INTRODUÇÃO

Embora problemas reais frequentemente tenham limitações que são traduzidas por restrições em suas modelagens, a maioria dos métodos de otimização não foi desenvolvida visando resolver problemas com restrições. Para que esses algoritmos possam buscar soluções ótimas e viáveis é necessária a associação de alguma técnica para o tratamento das restrições.

Uma solução é considerada viável quando atende as condições impostas pelas equações e inequações de restrição de um problema. Uma solução inviável viola uma ou mais restrições, e pode ser qualificada pelo valor de suas violações. O cálculo do valor de violação de uma restrição  $k$  é definido como

$$v_k(\vec{X}) = \begin{cases} \max\{0, |h_k(\vec{X})| - \varepsilon\}, & \text{para restrições de igualdade} \\ \max\{0, g_k(\vec{X})\}, & \text{caso contrário} \end{cases} \quad (16)$$

onde  $h_k(\vec{X})$  e  $g_k(\vec{X})$  são os valores calculados para a função de restrição, e  $\varepsilon$  é um valor de tolerância para restrições de igualdade. Dessa forma, o valor absoluto da função descontado do valor de tolerância será considerado como violação em restrições de igualdade, e valores positivos serão considerados como violação em restrições de desigualdade.

## 4.2 TÉCNICAS EMPREGADAS NA LITERATURA

A maioria das técnicas para tratamento de restrições propostas na literatura utiliza alterações no valor de avaliação de uma solução e/ou métodos de comparação especializados para levar soluções paralelamente ao melhor valor na função objetivo e à viabilidade.

DEB (2000) sugere o uso de um método de comparação que prioriza soluções viáveis sobre soluções inviáveis na forma de

$$\vec{X}_1 < \vec{X}_2 \Leftrightarrow \begin{cases} f(\vec{X}_1) < f(\vec{X}_2), & \text{se } \vec{X}_1 \text{ e } \vec{X}_2 \text{ forem viáveis} \\ \vec{X}_{viável} < \vec{X}_{inviável}, & \text{se apenas um for viável} \\ \phi(\vec{X}_1) < \phi(\vec{X}_2), & \text{se } \vec{X}_1 \text{ e } \vec{X}_2 \text{ forem inviáveis} \end{cases} \quad (17)$$

onde  $\vec{X}_1 < \vec{X}_2$  define uma operação de precedência entre as soluções  $\vec{X}_1$  e  $\vec{X}_2$ , e  $\phi(\vec{X})$  é o valor ponderado da violação, obtido por

$$\phi(\vec{X}) = \frac{\sum_{k=1}^m (1/v_{max,k}) v_k(\vec{X})}{\sum_{k=1}^m 1/v_{max,k}} \quad (18)$$

onde  $v_{max,k}$  é o maior valor de violação encontrado para a restrição  $k$  até o momento.

BARBOSA & LEMONGE (2003) propuseram um método de penalização adaptativa (APM) consistindo em um processo de penalização ponderada de violações onde os coeficientes de ponderação são calculados automaticamente de acordo com o valor médio de violação na população. Assumindo um número  $m$  de restrições, o método avalia um indivíduo conforme

$$F(\vec{X}) = \begin{cases} f(\vec{X}), & \text{se } \vec{X} \text{ é viável} \\ \bar{f}(\vec{X}) + \sum_{k=1}^m w_k \cdot v_k(\vec{X}), & \text{caso contrário} \end{cases} \quad (19)$$

considerando

$$\bar{f}(\vec{X}) = \begin{cases} f(\vec{X}), & \text{se } f(\vec{X}) > \langle f(\vec{X}) \rangle \\ \langle f(\vec{X}) \rangle, & \text{caso contrário} \end{cases} \quad (20)$$

onde  $\langle f(\vec{X}) \rangle$  é a média dos valores na função objetivo da população atual, e  $w_k$  é o fator de ponderação para a restrição de índice  $k$ , calculado a cada geração como

$$w_k = |\langle f(\vec{X}) \rangle| \cdot \frac{\langle v_k(\vec{X}) \rangle}{\sum_{l=1}^m [\langle v_l(\vec{X}) \rangle]^2} \quad (21)$$

onde  $\langle v_l(\vec{X}) \rangle$  é o valor médio de violação da restrição de índice  $k$  na população.

Similarmente, o método de Penalidade Auto-Adaptativa (FARMANI & WRIGHT, 2003) calcula a ponderação de penalidades de acordo com a razão entre soluções viáveis e inviáveis na população. No método, a avaliação de uma solução é dada por

$$F(\vec{X}) = d(\vec{X}) + p(\vec{X}) \quad (19)$$

onde  $d(\vec{X})$  e  $p(\vec{X})$  são, respectivamente, valores de distância e penalidade. O valor de distância é definido como

$$d(\vec{X}) = \begin{cases} \phi(\vec{X}), & \text{se } r_f = 0 \\ \sqrt{f''(\vec{X})^2 + \phi(\vec{X})^2}, & \text{caso contrário} \end{cases} \quad (19)$$

onde  $r_f$  é definido por (número de soluções viáveis)/(tamanho da população) e  $f''(\vec{X}) = (f(\vec{X}) - f_{min})/(f_{max} - f_{min})$ , onde  $f_{max}$  e  $f_{min}$  são os valores máximo e mínimo da função objetivo na população. O valor de penalidade é definido como

$$\begin{aligned}
 p(\vec{X}) &= (1 - r_f)M(\vec{X}) + r_f N(\vec{X}) \\
 M(\vec{X}) &= \begin{cases} 0, & \text{se } r_f = 0 \\ \phi(\vec{X}), & \text{caso contrário} \end{cases} \\
 N(\vec{X}) &= \begin{cases} 0, & \text{se } \vec{X} \text{ é viável} \\ f''(\vec{X}), & \text{caso contrário.} \end{cases}
 \end{aligned} \tag{19}$$

A abordagem de HO & SHIMIZU (2007) utiliza a posição de soluções em rankings para amenizar diferenças de grandezas entre valores de violação de restrições e da função objetivo na forma de

$$F(\vec{X}) = \begin{cases} R_s + R_v, & \text{se não existirem viáveis na população} \\ R_s + R_v + R_f, & \text{caso contrário} \end{cases} \tag{19}$$

de forma que a busca contemple apenas a viabilidade até que um indivíduo viável seja encontrado.

A estratégia *Stochastic Ranking* (RUNARSSON & YAO, 2000) visa obter equilíbrio entre os valores da função objetivo e das violações de restrições de forma estocástica. Para tanto, um fator de probabilidade  $p_f$  é utilizado para determinar se uma comparação utilizará os valores na função objetivo ou os valores de violação de restrições, como

$$\vec{X}_1 < \vec{X}_2 \Leftrightarrow \begin{cases} f(\vec{X}_1) < f(\vec{X}_2), & \text{se } \vec{X}_1 \text{ e } \vec{X}_2 \text{ forem viáveis} \\ f(\vec{X}_1) < f(\vec{X}_2), & \text{se } \text{rand}[0,1] < p_f \\ \phi(\vec{X}_1) < \phi(\vec{X}_2), & \text{se } \text{rand}[0,1] \geq p_f. \end{cases} \quad (22)$$

A metodologia  $\varepsilon$ -constraint apresentada por TAKAHAMA & SAKAI (2006) faz uso de uma técnica de relaxação de restrições em que um valor de tolerância  $\varepsilon$  é adaptado na forma de

$$\begin{aligned} \varepsilon(0) &= \phi(\vec{X}_\theta) \\ \varepsilon(G) &= \begin{cases} \varepsilon(0) \cdot \left(1 - \frac{G}{T_C}\right)^{cp}, & 0 < G < T_C \\ 0, & G \geq T_C \end{cases} \end{aligned} \quad (23)$$

onde  $\phi(\vec{X}_\theta)$  é a média ponderada das violações do  $\theta$ -ésimo melhor indivíduo da população,  $\theta$  tem o valor de  $0.2NP$ ,  $T_C$  é a última geração em que a relaxação será usada, e  $cp$  é um fator da velocidade da redução da tolerância. O valor  $\varepsilon$  é utilizado nas comparações como

$$\vec{X}_1 < \vec{X}_2 \Leftrightarrow \begin{cases} f(\vec{X}_1) < f(\vec{X}_2), & \text{se } \phi(\vec{X}_1) < \varepsilon \text{ e } \phi(\vec{X}_2) < \varepsilon \\ f(\vec{X}_1) < f(\vec{X}_2), & \text{se } \phi(\vec{X}_1) = \phi(\vec{X}_2) \\ \phi(\vec{X}_1) < \phi(\vec{X}_2), & \text{em outro caso} \end{cases} \quad (24)$$

de forma que o valor na função objetivo é usado para a comparação sempre que ambos os indivíduos são viáveis (considerando  $\varepsilon$ ) ou equivalentemente inviáveis, enquanto o valor das violações é usado em outros casos.

Adicionalmente, o  $\varepsilon$ -constraint executa uma operação de reparo utilizando o gradiente das funções de restrição, objetivando levar soluções inviáveis para regiões de viabilidade.

### **4.3 EVOLUÇÃO DIFERENCIAL COM TRATAMENTO DE RESTRIÇÕES**

A seguir são citados alguns trabalhos da literatura que empregaram o algoritmo DE com técnicas para o tratamento de restrições.

O primeiro uso do DE para a solução de problemas de otimização com restrições foi proposto por STORN (1999) com uma estratégia de relaxação de restrições. LAMPINEM (2002) utilizou a o DE com Dominância de Pareto no espaço de restrições. O algoritmo  $\varepsilon$ -DE foi apresentado por TAKAHAMA & SAKAI (2006) obtendo grande sucesso na competição do CEC 2006. ZHANG, LUO & WANG (2008) utilizaram a estratégia *Stochastic Ranking* de RUNARSSON & YAO (2000) obtendo resultados competitivos. SILVA, BARBOSA & LEMONGE (2011) utilizaram o DE com o método de penalização adaptativa (APM) para resolver problemas de otimização em engenharia.

## 5 Metodologia Proposta

### 5.1 INTRODUÇÃO

O algoritmo proposto nesta dissertação, denominado TSCH-RBDE (*Two Stage Constraint Handling - Roulette Based Differential Evolution*), se propõe a solucionar problemas de otimização contínua com restrições através de uma estratégia de dois estágios utilizando o algoritmo *DE/rand/1/bin* como base.

O princípio da estratégia em dois estágios é reduzir a complexidade de problemas de otimização com restrições através da solução da função objetivo e das funções de restrição separadamente. O primeiro estágio da execução do algoritmo visa localizar múltiplas soluções viáveis de forma a preparar uma população para o segundo estágio, que utilizará a região viável delineada pelas soluções para localizar o ponto com o melhor valor na função objetivo.

### 5.2 PRIMEIRO ESTÁGIO

O primeiro estágio do algoritmo desenvolvido, TSCH-RBDE, tem o objetivo de localizar rapidamente múltiplas soluções viáveis para o problema a ser otimizado, obtendo o máximo de diversidade possível. A diversidade nas soluções viáveis é importante uma vez que uma população com pouca dispersão no espaço de busca propicia uma convergência precoce, possivelmente para um ótimo local. Uma população cobrindo o máximo possível da região viável no espaço de busca permite que no segundo estágio ocorra a convergência para a solução ótima. Para esse fim foram desenvolvidos uma nova metodologia de busca e uma variação do algoritmo DE.

### 5.2.1 Buscas Sucessivas

Algoritmos evolutivos funcionam através do movimento de convergência da população para uma solução, e como reflexo disso, a população final de uma execução possuirá pouca diversidade de soluções. A necessidade de gerar uma população de soluções diversas para delinear regiões viáveis foi resolvida através de execuções sucessivas do DE, acumulando soluções viáveis únicas até que seja atingido o tamanho da população do segundo estágio. O processo é apresentado na Figura 4.

```
INICIA_POPULACAO_LOCAL()
IGERACAO = N VIAVEIS = 0

FAÇA

    SE IGERACAO > MAXGERACAO OU N VIAVEIS >= 1 ENTÃO
        INICIA_POPULACAO_LOCAL()
        IGERACAO = N VIAVEIS = 0
    FIM SE

    EXECUTA_GERACAO_DE()

    PARA CADA INDIVIDUO EM POPULACAO_LOCAL FAÇA
        SE INDIVIDUO É VIÁVEL E NÃO EXISTE NA POPULAÇÃO GLOBAL
            SALVA_POPULACAO_GLOBAL(INDIVIDUO)
            N VIAVEIS = N VIAVEIS +1
        FIM SE
    FIM PARA

    IGERACAO = IGERACAO + 1
    SE POPULACAO_GLOBAL >= TAM_POP_ESTAGIO_2 ENTÃO
        SAIR FAÇA
    FIM SE

FIM FAÇA
```

Figura 4 - Pseudocódigo do primeiro estágio

O algoritmo DE é executado múltiplas vezes durante o primeiro estágio, tendo como critério de parada um número máximo de gerações preestabelecido e a localização de ao menos uma solução viável. Uma vez que um dos critérios de parada é encontrado, soluções viáveis que não tenham sido previamente localizadas são adicionadas a população global (do segundo estágio) e a população local é reiniciada aleatoriamente, efetivamente iniciando uma nova execução. O critério de parada por número máximo de gerações evita que o algoritmo gaste excessivas avaliações da função objetivo em execuções malsucedidas, e a limitação de soluções viáveis permite que uma próxima execução localize novas soluções viáveis em regiões diferentes. Essas execuções são repetidas sucessivamente até que a população do segundo estágio esteja completa.

### 5.2.2 Avaliação de Soluções

Para tornar possível e rápida a busca de soluções viáveis, uma metodologia de avaliação diferenciada foi desenvolvida para o primeiro estágio do algoritmo.

Baseando-se na abordagem sugerida por HO & SHIMIZU (2007) onde um somatório de posições em *rankings* é usado para amenizar diferenças de grandezas entre valores utilizados para a avaliação, a metodologia proposta avalia as soluções considerando

$$f(\vec{X}) = R_s + R_v + R_{u_i} + R_{u_{i+1}} + \dots R_{u_m} \quad (22)$$

onde  $R_s$  é o posicionamento da solução  $\vec{X}$  em um ranking por somatório dos valores de violação de cada restrição,  $R_v$  é seu posicionamento em um ranking por número total de violações e  $R_{u_{i..n}}$  são seus posicionamentos em rankings individuais para cada restrição.

Os rankings individuais por restrição são utilizados para valorizar soluções proeminentes em uma ou mais restrições, independentemente de suas violações nas restantes - uma vez que sua presença na população pode gerar recombinações importantes para alcançar uma solução viável. Os valores usados para a geração desses rankings são obtidos na forma de

$$V_{\vec{X},k,G} = \begin{cases} |dev(P_{\vec{X},k,G})|, & \text{se } dev(P_{\vec{X},k,G}) > std(P_{k,G}) \text{ e } dev(P_{\vec{X},k,G}) < 0 \\ 0, & \text{caso contrário} \end{cases} \quad (23)$$

onde  $dev(P_{\vec{X},k,G})$  é o desvio do indivíduo  $\vec{X}$  em relação ao valor médio de violação da restrição  $k$  na população  $P$ , e  $std(P_{k,G})$  é o desvio padrão nos valores da restrição  $k$  na população  $P$ , de forma que um indivíduo que possua um desvio significativo para melhor na restrição receba uma pontuação. Estes valores são ilustrados na Figura 5.

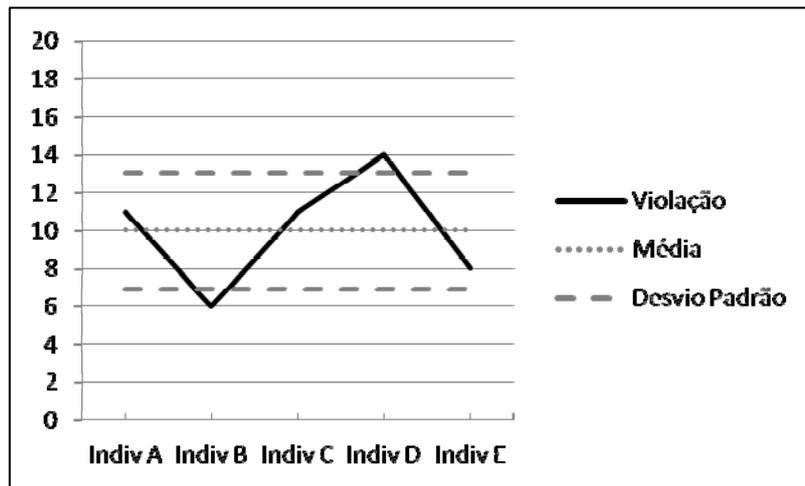


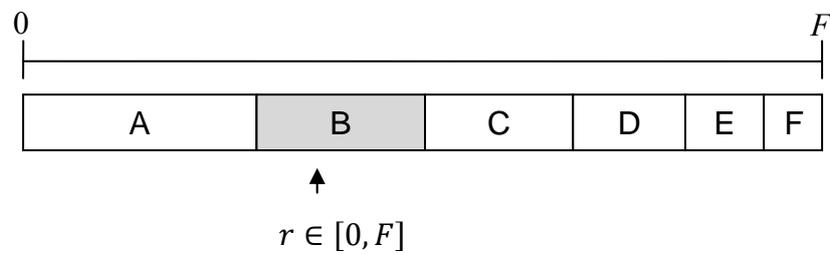
Figura 5 - Valores do ranking individual por restrição

### 5.2.3 Variante Proposta da Evolução Diferencial

O algoritmo DE básico utiliza soluções aleatoriamente escolhidas da população para a geração do vetor doador durante o processo de mutação. O valor da avaliação da solução é usado apenas no processo de seleção, quando vetores experimentais de melhor avaliação substituem o vetor alvo na geração seguinte. Essa metodologia não demonstrou velocidade de convergência suficiente para ser aplicada no processo de buscas sucessivas - e também faz pouco uso do modelo de avaliação implementado.

Para acelerar a evolução sem perder a diversidade foi utilizada no DE uma técnica de seleção proporcional à avaliação - popularmente denominada roleta. O uso da roleta para a seleção de soluções traz para o DE o conceito de pressão de seleção, comum em algoritmos genéticos, e permite que soluções com melhor avaliação sejam escolhidas com mais frequência para o processo de mutação.

O processo oferece a cada solução uma chance de seleção proporcional a sua contribuição para a avaliação geral da população. O processo é executado organizando a população de forma decrescente pela avaliação e gerando valores de avaliação acumulada para cada solução. A avaliação acumulada é calculada como a soma da avaliação da solução com a avaliação das soluções posicionadas anteriormente. Para selecionar uma solução, um número aleatório  $R$  entre 0 e o valor total acumulado é escolhido. A primeira solução cuja avaliação acumulada for maior do que  $R$  será selecionada. O processo é ilustrado na Figura 6.



**Figura 6 - Seleção proporcional a avaliação**

Na variante desenvolvida o processo de seleção por roleta é utilizado para selecionar o vetor alvo e os três vetores adicionais para a geração do doador, que é recombinado utilizando a variante binomial do DE básico.

### 5.3 SEGUNDO ESTÁGIO

O segundo estágio do algoritmo executa uma variante básica do algoritmo DE com mutação aleatória e recombinação binomial (*DE/rand/1/bin*). O uso do algoritmo básico ao invés da variante proposta (DE baseado em roleta) é fundamentado no fato de

que a robustez na solução é preferível em relação à rapidez de convergência no caso de uma única execução.

Apenas duas alterações foram feitas no algoritmo básico para o segundo estágio: uma foi a remoção da etapa da geração da população inicial, que na variante é predefinida pela população viável resultante do primeiro estágio, e a segunda foi a limitação da busca a soluções viáveis.

A busca foi limitada a soluções viáveis através de uma técnica simples de 'pena de morte' imposta na comparação entre indivíduos, onde soluções inviáveis são sempre consideradas inferiores a soluções viáveis, enquanto soluções viáveis são comparadas pelo seu valor na função objetivo. Dessa maneira, nenhuma solução inviável será aceita no processo de seleção, mantendo a população viável ao longo das gerações.

## 6 Estudos de Caso

Para a validação do algoritmo proposto foram executadas diversas análises utilizando funções de *benchmark* disponíveis na literatura para otimização de problemas com restrições. Estas funções tem servido como comparação na competição do CEC (LIANG *et al.*, 2006) entre os diferentes métodos de otimização apresentados pelos pesquisadores da área. Os resultados foram comparados com os de métodos recentes e competitivos.

### 6.1 METODOLOGIA

O algoritmo foi testado no conjunto de funções com restrições denominado *G Suite* (LIANG *et al.*, 2006) e descrito no Anexo A. As principais características destas funções são apresentadas na Tabela 1.

Para o fim de uma comparação de diferentes estratégias com uma mesma metodologia foram utilizados os resultados das estratégias *Superiority of Feasible Solutions* (DEB, 2000), *Self-Adaptive Penalty* (FARMANI & WRIGHT, 2003),  *$\epsilon$ -Constraint* (TAKAHAMA & SAKAI, 2006) e *Stochastic Ranking* (RUNARSSON & YAO, 2000) conforme sintetizados em MALLIPEDDI & SUGANTHAN (2010). As estratégias são identificadas nas tabelas através das siglas SF, SP, EC e SR, respectivamente. A metodologia de execução utilizada neste trabalho é a mesma empregada no referido artigo: cada função foi executada 30 vezes com um limite de 240.000 avaliações da função objetivo (AFO). Um valor de tolerância igual a 0.0001 foi adotado para as restrições de igualdade.

**Tabela 1 - Propriedades dos Problemas da G-Suite**

Função	$n$	Classificação de $f$	$\rho$ (%)	$LI$	$NI$	$LE$	$NE$	$a$
G01	13	Quadrática	0.0111	9	0	0	0	6
G02	20	Não linear	99.8474	0	2	0	0	1
G03	10	Polinomial	0	0	0	0	1	1
G04	5	Quadrática	52.123	0	6	0	0	2
G05	4	Cúbica	0	2	0	0	3	3
G06	2	Cúbica	0.0066	0	2	0	0	2
G07	10	Quadrática	0.0003	3	5	0	0	6
G08	2	Não linear	0.856	0	2	0	0	0
G09	7	Polinomial	0.5121	0	4	0	0	2
G10	8	Linear	0.001	3	3	0	0	6
G11	2	Quadrática	0	0	0	0	1	1
G12	3	Quadrática	4.7713	0	1	0	0	0
G13	5	Não linear	0	0	0	0	3	3
G14	10	Não linear	0	0	0	3	0	3
G15	3	Quadrática	0	0	0	1	1	2
G16	5	Não linear	0.0204	4	34	0	0	4
G17	6	Não linear	0	0	0	0	4	4
G18	9	Quadrática	0	0	12	0	0	4
G19	15	Não linear	33.4761	0	5	0	0	–
G20	24	Linear	0	0	6	2	12	–
G21	7	Linear	0	0	1	0	5	6
G22	22	Linear	0	0	1	8	11	–
G23	9	Linear	0	0	2	3	1	–
G24	2	Linear	79.6556	0	2	0	0	2

$n$  é o número de parâmetros,  $\rho$  é a razão estimada entre a região viável e o espaço de busca,  $LI$  e  $NI$  são o número de restrições lineares e não lineares de desigualdade,  $LE$  e  $NE$  são o número de restrições lineares e não lineares de igualdade e  $a$  é o número de restrições ativas no ponto ótimo considerando uma tolerância de  $\delta = 0.0001$  para restrições de igualdade ( $LE$  e  $NE$ ).

Os resultados apresentados foram gerados com os valores na função objetivo das melhores soluções viáveis resultantes das 30 execuções. Como soluções viáveis não foram encontradas por nenhuma estratégia para as funções G20 e G22, essas funções não estão incluídas na tabela de resultados.

## 6.2 PARÂMETROS DO ALGORITMO

O algoritmo proposto utiliza diferentes instâncias do algoritmo de Evolução Diferencial para o primeiro e segundo estágios, e parâmetros diferentes foram definidos para cada um dos estágios de forma a obter os resultados apresentados.

Para o primeiro estágio, foi utilizado um número máximo de 400 gerações por execução. O tamanho da população é de 5 indivíduos, a taxa de recombinação  $Cr$  foi definida como 0.5 e o fator de ponderação  $F$  para a mutação é escolhido aleatoriamente entre 0.7 e 0.9.

No segundo estágio, foi empregado um tamanho da população de 32 indivíduos, a taxa de recombinação  $Cr$  é 0.5 e o fator de ponderação  $F$  para mutação escolhido aleatoriamente entre 0.1 e 0.9.

### **6.3 RESULTADOS**

A Tabela 2 apresenta os resultados gerais do algoritmo desenvolvido neste trabalho para as funções da *G Suite* juntamente com os valores conhecidos na função objetivo para as soluções ótimas (2ª coluna - ótimo). Os resultados obtidos são apresentados como o valor na função objetivo da melhor solução encontrada, a média, o pior e o desvio padrão nas 30 rodadas executadas. A última coluna apresenta o valor  $V$  que representa a porcentagem de soluções viáveis nas execuções. Os valores em negrito são os resultados encontrados pelo algoritmo desenvolvido (TSCH-RDBE) que alcançaram os ótimos globais conhecidos.

Tabela 2 - Resultados do TSCH-RBDE na G Suite - 240.000 AFO

<i>f</i>	Ótimo	Melhor	Média	Pior	Desv.Padrão	V
<b>G01</b>	-15.0000	<b>-15.0000</b>	<b>-15.0000</b>	<b>-15.0000</b>	0.0000	100%
<b>G02</b>	-0.8036	<b>-0.8036</b>	-0.7894	-0.7484	0.0772	100%
<b>G03</b>	-1.0005	<b>-1.0005</b>	<b>-1.0005</b>	<b>-1.0005</b>	0.0000	100%
<b>G04</b>	-30665.5387	<b>-30665.5387</b>	<b>-30665.5387</b>	<b>-30665.5387</b>	0.0000	100%
<b>G05</b>	5126.4967	<b>5126.4967</b>	5207.6101	5577.5467	119.1776	100%
<b>G06</b>	-6961.8139	<b>-6961.8139</b>	<b>-6961.8139</b>	<b>-6961.8139</b>	0.0000	100%
<b>G07</b>	24.3062	24.3073	24.3255	24.4673	0.1610	100%
<b>G08</b>	-0.0958	<b>-0.0958</b>	<b>-0.0958</b>	<b>-0.0958</b>	0.0000	100%
<b>G09</b>	680.6301	<b>680.6301</b>	<b>680.6301</b>	680.6304	0.0005	100%
<b>G10</b>	7049.2480	7051.2987	7100.0657	7250.9751	343.9635	100%
<b>G11</b>	0.7499	<b>0.7499</b>	<b>0.7499</b>	<b>0.7499</b>	0.0000	100%
<b>G12</b>	-1.0000	<b>-1.0000</b>	<b>-1.0000</b>	<b>-1.0000</b>	0.0000	100%
<b>G13</b>	0.0539	0.2982	0.8124	1.9681	1.6659	100%
<b>G14</b>	-47.7649	-	-	-	-	0%
<b>G15</b>	961.7150	<b>961.7150</b>	964.1436	967.8495	10.5722	100%
<b>G16</b>	-1.9052	<b>-1.9052</b>	<b>-1.9052</b>	<b>-1.9052</b>	0.0000	100%
<b>G17</b>	8853.5397	-	-	-	-	0%
<b>G18</b>	-0.8660	<b>-0.8660</b>	-0.8150	-0.6748	0.4628	100%
<b>G19</b>	32.6556	32.9554	33.6041	35.5863	3.4074	100%
<b>G21</b>	193.7245	-	-	-	-	0%
<b>G23</b>	-400.0551	-	-	-	-	0%
<b>G24</b>	-5.5080	<b>-5.5080</b>	<b>-5.5080</b>	<b>-5.5080</b>	0.0000	100%

Observa-se na tabela que o TSCH-RBDE apresentou soluções ótimas em todas as rodadas para nove das funções testadas. A solução ótima foi localizada ao menos uma vez em quatorze das vinte e quatro funções.

Embora tenham sido encontradas soluções viáveis em todas as execuções da função G13, os valores das soluções na função objetivo ficaram muito distantes do valor ótimo, e o algoritmo não foi capaz de encontrar soluções viáveis nas funções G14, G17, G21 e G23.

Foi observado durante as execuções que a finalização do primeiro estágio para a função G13, que implica na localização de 32 indivíduos viáveis que formarão a população do estágio seguinte, aconteceu com poucas avaliações da função objetivo - em média 30.000, mas que o segundo estágio não foi capaz de evoluir as soluções para bons valores na função objetivo. Dessa observação supõe-se que uma ou mais regiões viáveis de fácil localização dominem a maioria dos indivíduos localizados no primeiro estágio, impedindo a localização do ótimo global no segundo.

As funções que apresentaram problemas com o algoritmo desenvolvido, G13, G14, G17, G21 e G23 possuem em comum um número de ao menos três restrições de igualdade e uma região viável muito pequena - o que se deduz ser a razão para a performance inferior do algoritmo nas mesmas.

Os resultados do algoritmo são apresentados juntamente com os resultados das outras estratégias na Tabela 3 para facilitar uma comparação direta.

Tabela 3 - Resultados para comparação

f		SF	SP	EC	SR	TSCH-RBDE
<b>G01</b>	Melhor	-14.9999	-15.0000	-14.9998	-14.9996	<b>-15.0000</b>
	Média	-14.9999	14.9999	-14.9996	-14.9991	<b>-15.0000</b>
	Pior	-14.9998	14.9999	-14.9995	-14.9985	<b>-15.0000</b>
	Desv.P.	0.0000	0.0000	0.0001	0.0003	<b>0.0000</b>
<b>G02</b>	Melhor	-0.8036	-0.8036	-0.8036	-0.8036	<b>-0.8036</b>
	Média	0.7994	-0.7961	-0.7988	-0.7925	-0.7894
	Pior	<b>-0.7926</b>	-0.7853	-0.7926	-0.7783	-0.7484
	Desv.P.	<b>0.0054</b>	0.0059	0.0051	0.0082	0.0772
<b>G03</b>	Melhor	-1.0005	-1.0005	-1.0005	-1.0005	<b>-1.0005</b>
	Média	-1.0005	-1.0005	-1.0005	-1.0005	<b>-1.0005</b>
	Pior	-1.0005	-1.0005	-1.0005	-1.0005	<b>-1.0005</b>
	Desv.P.	0.0000	0.0000	0.0000	0.0000	<b>0.0000</b>
<b>G04</b>	Melhor	-30665.5387	-30665.5387	-30665.5387	-30665.5387	<b>-30665.5387</b>
	Média	-30665.5387	-30665.5387	-30665.5387	-30665.5387	<b>-30665.5387</b>
	Pior	-30665.5387	-30665.5387	-30665.5387	-30665.5387	<b>-30665.5387</b>
	Desv.P.	0.0000	0.0000	0.0000	0.0000	<b>0.0000</b>
<b>G05</b>	Melhor	5126.4969	5126.4967	5126.4967	5126.4969	<b>5126.4967</b>
	Média	5161.5388	5127.7182	5126.5058	5158.3317	5207.6101
	Pior	5485.1800	5134.6751	5126.6048	5329.3866	5577.5467
	Desv.P.	78.0000	2.3000	<b>0.0220</b>	60.0000	119.1776
<b>G06</b>	Melhor	-6961.8139	-6961.8139	-6961.8139	-6961.8139	<b>-6961.8139</b>
	Média	-6961.8139	-6961.8139	-6961.8139	-6961.8139	<b>-6961.8139</b>
	Pior	-6961.8139	-6961.8139	-6961.8139	-6961.8139	<b>-6961.8139</b>
	Desv.P.	0.0000	0.0000	0.0000	0.0000	<b>0.0000</b>
<b>G07</b>	Melhor	24.3064	24.3062	24.3066	24.3063	24.3073
	Média	24.3093	24.3071	24.3096	24.3097	24.3255
	Pior	24.3199	24.3120	24.3290	24.3253	24.4673
	Desv.P.	0.0031	0.0014	0.0001	0.0047	0.1610
<b>G08</b>	Melhor	-0.0958	-0.0958	-0.0958	-0.0958	<b>-0.0958</b>
	Média	-0.0958	-0.0958	-0.0958	-0.0958	<b>-0.0958</b>
	Pior	-0.0958	-0.0958	-0.0958	-0.0958	<b>-0.0958</b>
	Desv.P.	0.0000	0.0000	0.0000	0.0000	<b>0.0000</b>
<b>G09</b>	Melhor	680.6301	680.6301	680.6301	680.6301	680.6301
	Média	680.6301	680.6306	680.6301	680.6301	680.6301
	Pior	680.6301	680.6310	680.6301	680.6301	680.6304
	Desv.P.	0.0000	0.0001	0.0000	0.0000	0.0005
<b>G10</b>	Melhor	7054.1548	7049.2566	7051.2552	7055.6236	7051.2987
	Média	7123.6326	7051.6584	7094.3033	7155.6031	7100.0657
	Pior	7242.9482	7080.8116	7213.3871	7428.5722	7250.9751
	Desv.P.	53.0000	5.7000	37.5000	80.0000	343.9635
<b>G11</b>	Melhor	0.7499	0.7499	0.7499	0.7499	0.7499
	Média	0.7499	0.7499	0.7499	0.7499	0.7499
	Pior	0.7499	0.7499	0.7499	0.7499	0.7499
	Desv.P.	0.0000	0.0000	0.0000	0.0000	0.0000
<b>G12</b>	Melhor	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000
	Média	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000
	Pior	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000
	Desv.P.	0.0000	0.0000	0.0000	0.0000	0.0000

Tabela 4 - Resultados para comparação - Continuação

f		SF	SP	EC	SR	TSCH-RBDE
<b>G13</b>	Melhor	0.0539	0.0539	0.0539	0.0539	0.2982
	Média	0.0539	0.0539	0.0539	0.0539	0.8124
	Pior	0.0539	0.0539	0.0539	0.0539	1.9681
	Desv.P.	0.0000	0.0000	0.0000	0.0000	1.6659
<b>G14</b>	Melhor	-46.7488	-47.7582	-47.2746	-47.0015	-
	Média	-45.3994	-47.6078	-46.5188	-45.1402	-
	Pior	-43.7368	-47.0345	-45.4870	-43.2758	-
	Desv.P.	0.7900	0.1700	0.4700	0.8500	-
<b>G15</b>	Melhor	961.7150	961.7150	961.7150	961.7150	961.7150
	Média	961.7150	961.7150	961.7150	961.7150	964.1436
	Pior	961.7150	961.7150	961.7150	961.7150	967.8495
	Desv.P.	0.0000	0.0000	0.0000	0.0000	10.5722
<b>G16</b>	Melhor	-1.9052	-1.9052	-1.9052	-1.9052	-1.9052
	Média	-1.9052	-1.9052	-1.9052	-1.9052	-1.9052
	Pior	-1.9052	-1.9052	-1.9052	-1.9052	-1.9052
	Desv.P.	0.0000	0.0000	0.0000	0.0000	0.0000
<b>G17</b>	Melhor	8860.8369	8872.8260	8853.5603	8871.0573	-
	Média	8942.0714	8916.7643	8853.8194	8935.0337	-
	Pior	8989.0609	8962.8508	8854.2410	9276.6484	-
	Desv.P.	30.0000	28.0000	0.1900	76.0000	-
<b>G18</b>	Melhor	-0.8660	-0.8661	-0.8660	-0.8660	-0.8660
	Média	-0.8660	-0.8648	-0.8660	-0.8660	-0.8150
	Pior	-0.8660	-0.8589	-0.8660	-0.8660	-0.6748
	Desv.P.	0.0000	0.0022	0.0000	0.0000	0.4628
<b>G19</b>	Melhor	32.8928	32.7005	32.8066	32.7902	32.9554
	Média	33.2849	32.8334	33.2224	33.3392	33.6041
	Pior	33.9365	33.1570	34.0538	33.7422	35.5863
	Desv.P.	0.2500	0.0001	0.2700	0.2700	3.4074
<b>G21</b>	Melhor	193.7491	195.3702	193.7290	193.7300	-
	Média	225.4789	241.7966	234.7433	206.1255	-
	Pior	294.1277	330.4669	275.8960	261.9906	-
	Desv.P.	41.0000	39.0000	26.6000	26.3000	-
<b>G23</b>	Melhor	-380.3077	-324.5844	-384.9543	-371.1396	-
	Média	-346.5745	-244.5554	-347.2257	-339.6890	-
	Pior	-306.4070	-137.5764	-289.2883	-312.6974	-
	Desv.P.	26.0000	55.5000	23.0000	18.0000	-
<b>G24</b>	Melhor	-5.5080	-5.5080	-5.5080	-5.5080	-5.5080
	Média	-5.5080	-5.5080	-5.5080	-5.5080	-5.5080
	Pior	-5.5080	-5.5080	-5.5080	-5.5080	-5.5080
	Desv.P.	0.0000	0.0000	0.0000	0.0000	0.0000

A comparação mostra que o algoritmo proposto obteve o melhor resultado para a função G01 e que todas as estratégias obtiveram sucesso em oito dos problemas (G03, G04, G06, G08, G11, G12, G16 e G24).

Nas funções G13 e G15, o TSCH-RDBE não obteve sucesso, sendo o único a não conseguir resultados ótimos. A função G15 possui duas restrições de igualdade e uma região viável pequena - o que volta a demonstrar o problema previamente colocado para as funções G13, G14, G17, G21 e G23.

Na função G10 o algoritmo obteve resultados superiores aos encontrados pelas estratégias *Superiority of Feasible Solutions* (DEB, 2000) e *Stochastic Ranking* (RUNARSSON & YAO, 2000), enquanto no restante dos problemas o algoritmo apresenta uma performance equivalente ou ligeiramente pior.

## 7 Conclusão

O algoritmo proposto nesta dissertação surgiu de forma experimental, fruto de observações, discussões e ponderações sobre assuntos da disciplina de otimização. Sua realização define uma nova estratégia de solução para problemas de otimização com restrições, essencialmente diferente dos métodos atualmente disponíveis na literatura.

A comparação do algoritmo com estratégias testadas e aprovadas pela comunidade científica demonstrou uma performance competitiva, mas que ainda pode ser melhorada.

Nos estudos realizados foi possível observar que o primeiro estágio do algoritmo apresentou grande agilidade na busca de soluções viáveis. Mesmo em algumas funções consideradas difíceis, um número de 32 soluções viáveis foi gerado dentro do limite de 240.000 avaliações da função objetivo, permitindo ainda a execução do segundo estágio nas avaliações restantes.

Estudos adicionais demonstraram melhores resultados para as funções onde não foram encontradas soluções viáveis através da definição de um valor mais alto para o limite de gerações no primeiro estágio, porém a mudança desse parâmetro sem um aumento proporcional no número máximo de avaliações causa degradação na performance geral, uma vez que reduz as avaliações disponíveis para o segundo estágio.

Embora soluções de valor ótimo na função objetivo tenham sido localizados na região viável delimitada para o segundo estágio em diversos problemas, alguns

apresentaram pouca melhora na população durante a execução do algoritmo. É deduzido que espaços de busca com áreas viáveis pequenas e distantes causem isolamento de soluções viáveis, evitando a convergência e tornando a evolução de indivíduos por diferenças entre vetores pouco efetiva.

## **7.1 TRABALHOS FUTUROS**

Como desenvolvimentos futuros do algoritmo apresentado nessa dissertação, propõe-se:

- O uso de auto adaptação nos parâmetros do algoritmo DE e no número máximo de gerações do primeiro estágio, visando uma melhor performance perante funções de restrição de problemas diversos.
- A hibridização da estratégia utilizando um algoritmo menos dependente de convergência da população para o segundo estágio, possivelmente algum método determinístico.
- Estudos comparativos utilizando as metodologias e métricas da competição do CEC.

## Referências

BARBOSA, H.J.C., LEMONGE, A.C.C., "A new adaptive penalty scheme for genetic algorithms", in *Information Sciences*, 156, nos. 3–4, pp. 215–251, 2003.

DANTZIG, G. B.; THAPA, M.N., *Linear programming: 1: Introduction*, Springer, 1997.

DAS, S., ABRAHAM, A., CHAKRABORTY, U.K., KONAR, A., "Differential evolution using a neighborhood based mutation operator", in *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526–553, Jun. 2009.

DEB, K., "An efficient constraint handling method for genetic algorithms", in *Computer Methods in Applied Mechanics and Engineering*, vol. 186, nos. 2–4, pp. 311–338, Jun. 2000.

FAN, H.Y., LAMPINEN, J., "A trigonometric mutation operation to differential evolution", in *Journal of Global Optimization*, vol. 27, no. 1, pp. 105–129, 2003.

FARMANI, R., WRIGHT, J.A., "Self-adaptive fitness formulation for constrained optimization", in *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 5, pp. 445–455, Oct. 2003.

GLOVER, F., "Future paths for integer programming and links to artificial intelligence", *Computers & Operations Research*, vol. 13, no. 5, pp. 533-549, 1986.

HO, P.Y., SHIMIZU, K., "Evolutionary Constrained Optimization Using an Addition of Ranking Method and a Percentage-based Tolerance Value Adjustment Scheme", *Information Sciences*, v. 177, n. 14, pp. 2985-3004, 2007.

HOLLAND, J.H., *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. 1 ed, Ann Arbor, University of Michigan Press, 1975

LIANG, J.J., RUNARSSON, T.P., MEZURA-MONTES, E. *et al.*, “Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-parameter Optimization”, *Journal of Applied Mechanics*, v. 41, 2006.

MALLIPEDDI, R., SUGANTHAN, P., “Ensemble of constraint handling techniques”, in *IEEE Transactions on Evolutionary Computation*, vol. 14, pp.561 -579, 2010.

MCCULLOCH, W.S., PITTS, W.H., “A logical calculus of the ideas immanent in nervous activity”, in *Bulletin of Mathematical Biophysics*, 5, 115-133, 1943.

MLADENOVIĆ, N., HANSEN, P., “Variable neighborhood search”, *Computers & Operations Research*, vol. 24, no. 11, pp. 1097-1100, 1997.

ORTEGA, J.M.; RHEINBOLDT, W.C., *Iterative solution of nonlinear equations in several variables*, Siam, 2000.

PRICE, K.V., STORN, R., “Differential evolution: A simple evolution strategy for fast optimization”, *Dr. Dobb's Journal*, vol. 22, no. 4, pp. 18–24, 1997.

PRICE, K.V., "An introduction to differential evolution", in *New Ideas in Optimization*, D. Corne, M. Dorigo, and V. Glover, Eds. London, U.K.: McGraw-Hill, , pp. 79–108, 1999.

PRICE, K.V., STORN, R., LAMPINEN, J., *Differential Evolution—A Practical Approach to Global Optimization*. Berlin, Germany: Springer, 2005.

QIN, A.K., HUANG, V.L., SUNGANTHAN, P.N., “Differential evolution algorithm with strategy adaptation for global numerical optimization”, in *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, Apr. 2009.

RAHNAMAYAN, S., TIZHOOSH, H.R., SALAMA, M.M.A., “Opposition based differential evolution,” in *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64–79, Feb. 2008.

RECHENBERG, I., *Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. PhD thesis, 1971.

ROSENBLATT, F., *The Perceptron - a Perceiving and Recognizing Automaton*,. 85-Tech. Rep. 460-1, Cornell Aeronautical Laboratory, USA, 1957

RUNARSSON, T.P., YAO, X., “Stochastic Ranking for Constrained Evolutionary Optimization”, in *IEEE Transactions on Evolutionary Computation*, v. 4, n. 3, pp. 284-29, 2000.

SILVA, E.K., BARBOSA, H.J.C., LEMONGE, A.C.C., "An adaptive constraint handling technique for differential evolution with dynamic use of variants in engineering optimization", in *Optimization and Engineering*, 12, pp. 31–54, 2011.

STORN, R., PRICE, K.V., *Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces*, Tech. Rep. TR-95-012, ICSI, USA, 1995.

STORN, R., PRICE, K.V., “Minimizing the real functions of the ICEC 1996 contest by differential evolution”, in *Proceedings of IEEE International Conference on Evolutionary Computation*, 1996, pp. 842–844, 1996.

STORN, R., PRICE, K.V., “Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces”, in *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

STORN, R., "System Design by Constraint Adaptation and Differential Evolution", in *IEEE Transactions on Evolutionary Computation*, no. 3, pp. 22–34, 1999.

TAKAHAMA, T., SAKAI, S., “Constrained optimization by the  $\epsilon$  constrained differential evolution with gradient-based mutation and feasible elites”, in *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 308–315, Jul. 2006.

TIZHOOSH, H.R., “Opposition-Based Learning: A New Scheme for Machine Intelligence”, in *International Conference on Computational Intelligence for Modeling Control and Automation*, Vol. I, pp. 695-701, Vienna, Austria, 2005.

ZHANG, M., LUO, W., WANG, X., “Differential evolution with dynamic stochastic selection for constrained optimization”, in *Information Sciences*, vol. 178, no. 15, pp. 3043–3074, Aug. 2008.

ZHANG, J., SANDERSON, A.C., “JADE: Adaptive differential evolution with optional external archive”, in *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, Oct. 2009.

# Anexo A

## G1

minimize:

$$f(\vec{x}) = 5 \sum_{i=1}^4 x_i - 5 \sum_{i=1}^4 x_i^2 - \sum_{i=5}^{13} x_i$$

sujeito a:

$$g_1(\vec{x}) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0$$

$$g_2(\vec{x}) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0$$

$$g_3(\vec{x}) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0$$

$$g_4(\vec{x}) = -8x_1 + x_{10} \leq 0$$

$$g_5(\vec{x}) = -8x_2 + x_{11} \leq 0$$

$$g_6(\vec{x}) = -8x_3 + x_{12} \leq 0$$

$$g_7(\vec{x}) = -2x_4 - x_5 + x_{10} \leq 0$$

$$g_8(\vec{x}) = -2x_6 - x_7 + x_{11} \leq 0$$

$$g_9(\vec{x}) = -2x_8 - x_9 + x_{12} \leq 0$$

onde:

$$0 \leq x_i \leq 1 \quad (i = 1, \dots, 9, 13)$$

$$0 \leq x_i \leq 100 \quad (i = 10, 11, 12)$$

## G2

minimize:

$$f(\vec{x}) = - \left| \frac{\sum_{i=1}^n \cos^4(x_i) - 2 \prod_{i=1}^n \cos^2(x_i)}{\sqrt{\sum_{i=1}^n i x_i^2}} \right|$$

*sujeito a:*

$$g_1(\vec{x}) = 0.75 - \prod_{i=1}^n x_i \leq 0$$

$$g_2(\vec{x}) = \sum_{i=1}^n x_i - 7.5n \leq 0$$

*onde:*

$$n = 20$$

$$0 \leq x_i \leq 10 \quad (i = 1, \dots, n)$$

### G3

*minimize:*

$$f(\vec{x}) = -(\sqrt{n})^n \prod_{i=1}^n x_i$$

*sujeito a:*

$$h_1(\vec{x}) = \sum_{i=1}^n x_i^2 - 1 = 0$$

*onde:*

$$n = 10$$

$$0 \leq x_i \leq 1 \quad (i = 1, \dots, n)$$

### G4

*minimize:*

$$f(\vec{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$$

*sujeito a:*

$$g_1(\vec{x}) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \leq 0$$

$$g_2(\vec{x}) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0$$

$$g_3(\vec{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leq 0$$

$$g_4(\vec{x}) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0$$

$$g_5(\vec{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0$$

$$g_6(\vec{x}) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0$$

onde:

$$78 \leq x_1 \leq 102$$

$$33 \leq x_2 \leq 45$$

$$27 \leq x_i \leq 45 \quad (i = 3, 4, 5)$$

## G5

minimize:

$$f(\vec{x}) = 3x_1 + 0.000001x_1^3 + 2x_2 + \left(\frac{0.000002}{3}\right)x_2^3$$

sujeito a:

$$g_1(\vec{x}) = -x_4 + x_3 - 0.55 \leq 0$$

$$g_2(\vec{x}) = -x_3 + x_4 - 0.55 \leq 0$$

$$h_3(\vec{x}) = 1000 \sin(-x_3 - 0.25) + 1000 \sin(-x_4 - 0.25) + 894.8 - x_1 = 0$$

$$h_4(\vec{x}) = 1000 \sin(x_3 - 0.25) + 1000 \sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0$$

$$h_5(\vec{x}) = 1000 \sin(x_4 - 0.25) + 1000 \sin(x_4 - x_3 - 0.25) + 1294.8 = 0$$

onde:

$$0 \leq x_i \leq 1200 \quad (i = 1, 2)$$

$$-0.55 \leq x_i \leq 0.55 \quad (i = 3, 4)$$

**G6**

*minimize:*

$$f(\vec{x}) = (x_1 - 10)^3 + (x_2 - 20)^3$$

*sujeito a:*

$$g_1(\vec{x}) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0$$

$$g_2(\vec{x}) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0$$

*onde:*

$$13 \leq x_1 \leq 100$$

$$0 \leq x_2 \leq 100$$

**G7**

*minimize:*

$$f(\vec{x}) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 \\ + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$$

*sujeito a:*

$$g_1(\vec{x}) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0$$

$$g_2(\vec{x}) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0$$

$$g_3(\vec{x}) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0$$

$$g_4(\vec{x}) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0$$

$$g_5(\vec{x}) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0$$

$$g_6(\vec{x}) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0$$

$$g_7(\vec{x}) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \leq 0$$

$$g_8(\vec{x}) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0$$

*onde:*

$$-10 \leq x_i \leq 10 \quad (i = 1, \dots, 10)$$

**G8**

minimize:

$$f(\vec{x}) = -\frac{\sin^3(2\pi x_1) \sin(2\pi x_2)}{x_1^3(x_1 + x_2)}$$

sujeito a:

$$g_1(\vec{x}) = x_1^2 - x_2 + 1 \leq 0$$

$$g_2(\vec{x}) = 1 - x_1 + (x_2 - 4)^2 \leq 0$$

onde:

$$0 \leq x_i \leq 10 \quad (i = 1, 2)$$

**G9**

minimize:

$$f(\vec{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 + 4x_6x_7 \\ - 10x_6 - 8x_7$$

sujeito a:

$$g_1(\vec{x}) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 0$$

$$g_2(\vec{x}) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0$$

$$g_3(\vec{x}) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0$$

$$g_4(\vec{x}) = 4x_1^2 + x_2^2 + 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0$$

onde:

$$-10 \leq x_i \leq 10 \quad (i = 1, \dots, 7)$$

**G10**

*minimize:*

$$f(\vec{x}) = x_1 + x_2 + x_3$$

*sujeito a:*

$$g_1(\vec{x}) = -1 + 0.0025(x_4 + x_6) \leq 0$$

$$g_2(\vec{x}) = -1 + 0.0025(x_5 + x_7 - x_4) \leq 0$$

$$g_3(\vec{x}) = -1 + 0.01(x_8 - x_5) \leq 0$$

$$g_4(\vec{x}) = -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \leq 0$$

$$g_5(\vec{x}) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0$$

$$g_6(\vec{x}) = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \leq 0$$

*onde:*

$$100 \leq x_1 \leq 10000$$

$$1000 \leq x_i \leq 10000 \quad (i = 2, 3)$$

$$10 \leq x_i \leq 1000 \quad (i = 4, \dots, 8)$$

**G11**

*minimize:*

$$f(\vec{x}) = x_1^2 + (x_2 - 1)^2$$

*sujeito a:*

$$h(\vec{x}) = x_2 - x_1^2 = 0$$

*onde:*

$$-1 \leq x_i \leq 1 \quad (i = 1, 2)$$

**G12**

*minimize:*

$$f(\vec{x}) = -(100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2)/100$$

*sujeito a:*

$$g(\vec{x}) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \leq 0$$

*onde:*

$$0 \leq x_i \leq 10 \quad (i = 1, 2, 3)$$

$$p, q, r = 1, \dots, 9$$

Uma solução é viável caso existam valores para  $p, q$  e  $r$  que respeitem a restrição.

**G13**

*minimize:*

$$f(\vec{x}) = e^{x_1 x_2 x_3 x_4 x_5}$$

*sujeito a:*

$$h_1(\vec{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$$

$$h_2(\vec{x}) = x_2 x_3 - 5 x_4 x_5 = 0$$

$$h_3(\vec{x}) = x_1^3 + x_2^3 + 1 = 0$$

*onde:*

$$-2.3 \leq x_i \leq 2.3 \quad (i = 1, 2)$$

$$-3.2 \leq x_i \leq 3.2 \quad (i = 3, 4, 5)$$

**G14**

*minimize:*

$$f(\vec{x}) = \sum_{i=1}^{10} x_i \left( c_i + \ln \frac{x_i}{\sum_{j=1}^{10} x_j} \right)$$

*sujeito a:*

$$h_1(\vec{x}) = x_1 + 2x_2 + 2x_3 + x_6 + x_{10} - 2 = 0$$

$$h_2(\vec{x}) = x_4 + 2x_5 + x_6 + x_7 - 1 = 0$$

$$h_3(\vec{x}) = x_3 + x_7 + x_8 + 2x_9 + x_{10} - 1 = 0$$

*onde:*

$$0 \leq x_i \leq 10 \quad (i = 1, \dots, 10)$$

$$c_1 = -6.089$$

$$c_2 = -17.164$$

$$c_3 = -34.054$$

$$c_4 = -5.914$$

$$c_5 = -24.721$$

$$c_6 = -14.986$$

$$c_7 = -21.1$$

$$c_8 = -10.708$$

$$c_9 = -26.662$$

$$c_{10} = -22.179$$

**G15***minimize:*

$$f(\vec{x}) = 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3$$

*sujeito a:*

$$h_1(\vec{x}) = x_1^2 + x_2^2 + x_3^2 - 25 = 0$$

$$h_2(\vec{x}) = 8x_1 + 14x_2 + 7x_3 - 56 = 0$$

*onde:*

$$0 \leq x_i \leq 10 \quad (i = 1, 2, 3)$$

**G16***minimize:*

$$\begin{aligned} f(\vec{x}) = & 0.000117y_{14} + 0.1365 + 0.00002358y_{13} + 0.000001502y_{16} \\ & + 0.0321y_{12} + 0.004324y_5 + 0.0001 \frac{c_{15}}{c_{16}} + 37.48 \frac{y_2}{c_{12}} \\ & - 0.0000005843y_{17} \end{aligned}$$

*sujeito a:*

$$g_1(\vec{x}) = \frac{0.28}{0.72}y_5 - y_4 \leq 0$$

$$g_2(\vec{x}) = x_3 - 1.5x_2 \leq 0$$

$$g_3(\vec{x}) = 3496 \frac{y_2}{c_{12}} - 21 \leq 0$$

$$g_4(\vec{x}) = 110.6 + y_1 - \frac{62212}{c_{17}} \leq 0$$

$$g_5(\vec{x}) = 213.1 - y_1 \leq 0$$

$$g_6(\vec{x}) = y_1 - 405.23 \leq 0$$

$$g_7(\vec{x}) = 17.505 - y_2 \leq 0$$

$$g_8(\vec{x}) = y_2 - 1053.6667 \leq 0$$

$$g_9(\vec{x}) = 11.275 - y_3 \leq 0$$

$$g_{10}(\vec{x}) = y_3 - 35.03 \leq 0$$

$$g_{11}(\vec{x}) = 214.228 - y_4 \leq 0$$

$$g_{12}(\vec{x}) = y_4 - 665.585 \leq 0$$

$$g_{13}(\vec{x}) = 7.458 - y_5 \leq 0$$

$$g_{14}(\vec{x}) = y_5 - 584.463 \leq 0$$

$$g_{15}(\vec{x}) = 0.961 - y_6 \leq 0$$

$$g_{16}(\vec{x}) = y_6 - 265.916 \leq 0$$

$$g_{17}(\vec{x}) = 1.612 - y_7 \leq 0$$

$$g_{18}(\vec{x}) = y_7 - 7.046 \leq 0$$

$$g_{19}(\vec{x}) = 0.146 - y_8 \leq 0$$

$$g_{20}(\vec{x}) = y_8 - 0.222 \leq 0$$

$$g_{21}(\vec{x}) = 107.99 - y_9 \leq 0$$

$$g_{22}(\vec{x}) = y_9 - 273.366 \leq 0$$

$$g_{23}(\vec{x}) = 922.693 - y_{10} \leq 0$$

$$g_{24}(\vec{x}) = y_{10} - 1286.105 \leq 0$$

$$g_{25}(\vec{x}) = 926.832 - y_{11} \leq 0$$

$$g_{26}(\vec{x}) = y_{11} - 1444.046 \leq 0$$

$$g_{27}(\vec{x}) = 18.766 - y_{12} \leq 0$$

$$g_{28}(\vec{x}) = y_{12} - 537.141 \leq 0$$

$$g_{29}(\vec{x}) = 1072.163 - y_{13} \leq 0$$

$$g_{30}(\vec{x}) = y_{13} - 3247.039 \leq 0$$

$$g_{31}(\vec{x}) = 8961.448 - y_{14} \leq 0$$

$$g_{32}(\vec{x}) = y_{14} - 26844.086 \leq 0$$

$$g_{33}(\vec{x}) = 0.063 - y_{15} \leq 0$$

$$g_{34}(\vec{x}) = y_{15} - 0.386 \leq 0$$

$$g_{35}(\vec{x}) = 71084.33 - y_{16} \leq 0$$

$$g_{36}(\vec{x}) = -140000 + y_{16} \leq 0$$

$$g_{37}(\vec{x}) = 2802713 - y_{17} \leq 0$$

$$g_{38}(\vec{x}) = y_{17} - 12146108 \leq 0$$

onde:

$$704.4148 \leq x_1 \leq 906.3855$$

$$68.6 \leq x_2 \leq 288.88$$

$$0 \leq x_3 \leq 134.75$$

$$193 \leq x_4 \leq 287.0966$$

$$25 \leq x_5 \leq 84.1988$$

$$y_1 = x_2 + x_3 + 41.6$$

$$c_1 = 0.024x_4 - 4.62$$

$$y_2 = \frac{12.5}{c_1} + 12$$

$$c_2 = 0.0003535x_1^2 + 0.5311x_1 + 0.08705y_2x_1$$

$$c_3 = 0.052x_1 + 78 + 0.002377y_2x_1$$

$$y_3 = \frac{c_2}{c_3}$$

$$y_4 = 19y_3$$

$$c_4 = 0.04782(x_1 - y_3) + \frac{0.1956(x_1 - y_3)^2}{x_2} + 0.6376y_4 + 1.594y_3$$

$$c_5 = 100x_2$$

$$c_6 = x_1 - y_3 - y_4$$

$$c_7 = 0.950 - \frac{c_4}{c_5}$$

$$y_5 = c_6 c_7$$

$$y_6 = x_1 - y_5 - y_4 - y_3$$

$$c_8 = (y_5 + y_4)0.995$$

$$y_7 = \frac{c_8}{y_1}$$

$$y_8 = \frac{c_8}{3798}$$

$$c_9 = y_7 - \frac{0.0663y_7}{y_8} - 0.3153$$

$$y_9 = \frac{96.82}{c_9} + 0.321y_1$$

$$y_{10} = 1.29y_5 + 1.258y_4 + 2.29y_3 + 1.71y_6$$

$$y_{11} = 1.71x_1 - 0.452y_4 + 0.580y_3$$

$$c_{10} = \frac{12.3}{752.3}$$

$$c_{11} = (1.75y_2)(0.995x_1)$$

$$c_{12} = 0.995y_{10} + 1998$$

$$y_{12} = c_{10}x_1 + \frac{c_{11}}{c_{12}}$$

$$y_{13} = c_{12} - 1.75y_2$$

$$y_{14} = 3623 + 64.4x_2 + 58.4x_3 + \frac{146312}{y_9 + x_5}$$

$$c_{13} = 0.995y_{10} + 60.8x_2 + 48x_4 - 0.1121y_{14} - 5095$$

$$y_{15} = \frac{y_{13}}{c_{13}}$$

$$y_{16} = 148000 - 331000y_{15} + 40y_{13} - 61y_{15}y_{13}$$

$$c_{14} = 2324y_{10} - 28740000y_2$$

$$y_{17} = 14130000 - 1328y_{10} - 531y_{11} + \frac{c_{14}}{c_{12}}$$

$$c_{15} = \frac{y_{13}}{y_{15}} - \frac{y_{13}}{0.52}$$

$$c_{16} = 1.104 - 0.72y_{15}$$

$$c_{17} = y_9 + x_5$$

## G17

minimize:

$$f(\vec{x}) = f(x_1) + f(x_2)$$

$$f_1(x_1) = \begin{cases} 30x_1, & 0 \leq x_1 < 300 \\ 31x_1, & 300 \leq x_1 < 400 \end{cases}$$

$$f_2(x_2) = \begin{cases} 28x_2, & 0 \leq x_2 < 100 \\ 29x_2, & 100 \leq x_2 < 200 \\ 30x_2, & 200 \leq x_2 < 1000 \end{cases}$$

sujeito a:

$$h_1(\vec{x}) = -x_1 + 300 - \frac{x_3x_4}{131.078} \cos(1.48477 - x_6) + \frac{0.90798x_3^2}{131.078} \cos(1.47588) = 0$$

$$h_2(\vec{x}) = -x_2 - \frac{x_3x_4}{131.078} \cos(1.48477 + x_6) + \frac{0.90798x_4^2}{131.078} \cos(1.47588) = 0$$

$$h_3(\vec{x}) = -x_5 - \frac{x_3x_4}{131.078} \sin(1.48477 + x_6) + \frac{0.90798x_4^2}{131.078} \sin(1.47588) = 0$$

$$h_4(\vec{x}) = 200 - \frac{x_3x_4}{131.078} \sin(1.48477 - x_6) + \frac{0.90798x_3^2}{131.078} \sin(1.47588) = 0$$

onde:

$$0 \leq x_1 \leq 400$$

$$0 \leq x_2 \leq 1000$$

$$340 \leq x_i \leq 420 \quad (i = 3, 4)$$

$$-1000 \leq x_5 \leq 1000$$

$$0 \leq x_6 \leq 0.5236$$

**G18**

*minimize:*

$$f(\vec{x}) = -0.5(x_1x_4 - x_2x_3 + x_3x_9 - x_5x_9 + x_5x_8 - x_6x_7)$$

*sujeito a:*

$$g_1(\vec{x}) = x_3^2 + x_4^2 - 1 \leq 0$$

$$g_2(\vec{x}) = x_9^2 - 1 \leq 0$$

$$g_3(\vec{x}) = x_5^2 + x_6^2 - 1 \leq 0$$

$$g_4(\vec{x}) = x_1^2 + (x_2 - x_9)^2 - 1 \leq 0$$

$$g_5(\vec{x}) = (x_1 - x_5)^2 + (x_2 - x_6)^2 - 1 \leq 0$$

$$g_6(\vec{x}) = (x_1 - x_7)^2 + (x_2 - x_8)^2 - 1 \leq 0$$

$$g_7(\vec{x}) = (x_3 - x_5)^2 + (x_4 - x_6)^2 - 1 \leq 0$$

$$g_8(\vec{x}) = (x_3 - x_7)^2 + (x_4 - x_8)^2 - 1 \leq 0$$

$$g_9(\vec{x}) = x_7^2 + (x_8 - x_9)^2 - 1 \leq 0$$

$$g_{10}(\vec{x}) = x_2x_3 - x_1x_4 \leq 0$$

$$g_{11}(\vec{x}) = -x_3x_9 \leq 0$$

$$g_{12}(\vec{x}) = x_5x_9 \leq 0$$

$$g_{13}(\vec{x}) = x_6x_7 - x_5x_8 \leq 0$$

*onde:*

$$-10 \leq x_i \leq 10 \quad (i = 1, \dots, 8)$$

$$0 \leq x_9 \leq 20$$

**G19**

minimize:

$$f(\vec{x}) = \sum_{j=1}^5 \sum_{i=1}^5 c_{ij} x_{(10+i)} x_{(10+j)} + 2 \sum_{j=1}^5 d_j x_{(10+j)}^3 - \sum_{i=1}^{10} b_i x_i$$

sujeito a:

$$g_j(\vec{x}) = -2 \sum_{i=1}^5 c_{ij} x_{(10+i)} - 3d_j x_{(10+j)}^2 - e_j + \sum_{i=1}^{10} a_{ij} x_i \leq 0, j = 1, \dots, 5$$

onde:

$$0 \leq x_i \leq 10 \quad (i = 1, \dots, 15)$$

$$\vec{b} = [-40, -2, -0.25, -4, -4, -1, -40, -60, 5, 1]$$

<b>j</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b><math>e_j</math></b>	-15	-27	-36	-18	-12
<b><math>c_{1j}</math></b>	30	-20	-10	32	-10
<b><math>c_{2j}</math></b>	-20	39	-6	-31	32
<b><math>c_{3j}</math></b>	-10	-6	10	-6	-10
<b><math>c_{4j}</math></b>	32	-31	-6	39	-20
<b><math>c_{5j}</math></b>	-10	32	-10	-20	30
<b><math>d_j</math></b>	4	8	10	6	2
<b><math>a_{1j}</math></b>	-16	2	0	1	0
<b><math>a_{2j}</math></b>	0	-2	0	0.4	2
<b><math>a_{3j}</math></b>	-3.5	0	2	0	0
<b><math>a_{4j}</math></b>	0	-2	0	-4	-1
<b><math>a_{5j}</math></b>	0	-9	-2	1	-2.8
<b><math>a_{6j}</math></b>	2	0	-4	0	0
<b><math>a_{7j}</math></b>	-1	-1	-1	-1	-1
<b><math>a_{8j}</math></b>	-1	-2	-3	-2	-1
<b><math>a_{9j}</math></b>	1	2	3	4	5
<b><math>a_{10j}</math></b>	1	1	1	1	1

Dados para o problema G19

**G20**

minimize:

$$f(\vec{x}) = \sum_{i=1}^{24} a_i x_i$$

sujeito a:

$$g_i(\vec{x}) = \frac{(x_i + x_{(i+12)})}{\sum_{j=1}^{24} x_j + e_i} \leq 0, i = 1, 2, 3$$

$$g_i(\vec{x}) = \frac{(x_{(i+3)} + x_{(i+15)})}{\sum_{j=1}^{24} x_j + e_i} \leq 0, i = 4, 5, 6$$

$$h_i(\vec{x}) = \frac{x_{(i+12)}}{b_{(i+12)} \sum_{j=13}^{24} \frac{x_j}{b_j}} - \frac{c_i x_i}{40 b_i \sum_{j=1}^{12} \frac{x_j}{b_j}} = 0, i = 1, \dots, 12$$

$$h_{13}(\vec{x}) = \sum_{i=1}^{24} x_i - 1 = 0$$

$$h_{14}(\vec{x}) = \sum_{i=1}^{12} \frac{x_i}{d_i} + k \sum_{i=13}^{24} \frac{x_i}{b_i} - 1.671 = 0$$

onde:

$$0 \leq x_i \leq 10 \quad (i = 1, \dots, 24)$$

$$k = (0.7302)(530) \left( \frac{14.7}{40} \right)$$

i	$a_i$	$b_i$	$c_i$	$d_i$	$e_i$
1	0.0693	44.094	123.7	31.244	0.1
2	0.0577	58.12	31.7	36.12	0.3
3	0.05	58.12	45.7	34.784	0.4
4	0.2	137.4	14.7	92.7	0.3
5	0.26	120.9	84.7	82.7	0.6
6	0.55	170.9	27.7	91.6	0.3
7	0.06	62.501	49.7	56.708	
8	0.1	84.94	7.1	82.7	
9	0.12	133.425	2.1	80.8	

<b>10</b>	0.18	82.507	17.7	64.517	
<b>11</b>	0.1	46.07	0.85	49.4	
<b>12</b>	0.09	60.097	0.64	49.1	
<b>13</b>	0.0693	44.094			
<b>14</b>	0.0577	58.12			
<b>15</b>	0.05	58.12			
<b>16</b>	0.2	137.4			
<b>17</b>	0.26	120.9			
<b>18</b>	0.55	170.9			
<b>19</b>	0.06	62.501			
<b>20</b>	0.1	84.94			
<b>21</b>	0.12	133.425			
<b>22</b>	0.18	82.507			
<b>23</b>	0.1	46.07			
<b>24</b>	0.09	60.097			

Dados para o problema G20

## G21

*minimize:*

$$f(\vec{x}) = x_1$$

*sujeito a:*

$$g_1(\vec{x}) = -x_1 + 35x_2^{0.6} + 35x_3^{0.6} \leq 0$$

$$h_1(\vec{x}) = -300x_3 + 7500x_5 - 75000x_6 - 25x_4x_5 + 25x_4x_6 + x_3x_4 = 0$$

$$h_2(\vec{x}) = -100x_2 + 155.365x_4 + 2500x_7 - x_2x_4 - 25x_4x_7 - 15536.5 = 0$$

$$h_3(\vec{x}) = -x_5 + \ln(-x_4 + 900) = 0$$

$$h_4(\vec{x}) = -x_6 + \ln(x_4 + 300) = 0$$

$$h_5(\vec{x}) = -x_7 + \ln(-2x_4 + 700) = 0$$

*onde:*

$$0 \leq x_1 \leq 1000$$

$$0 \leq x_i \leq 40 \quad (i = 2, 3)$$

$$100 \leq x_4 \leq 300$$

$$6.3 \leq x_5 \leq 6.7$$

$$5.9 \leq x_6 \leq 6.4$$

$$4.5 \leq x_7 \leq 6.25$$

**G22**

*minimize:*

$$f(\vec{x}) = x_1$$

*sujeito a:*

$$g_1(\vec{x}) = -x_1 + x_2^{0.6} + x_3^{0.6} + x_4^{0.6} \leq 0$$

$$h_1(\vec{x}) = x_5 - 100000x_8 + 1 \times 10^7 = 0$$

$$h_2(\vec{x}) = x_6 + 100000x_8 - 100000x_9 = 0$$

$$h_3(\vec{x}) = x_7 + 100000x_9 - 5 \times 10^7 = 0$$

$$h_4(\vec{x}) = x_5 + 100000x_{10} - 3.3 \times 10^7 = 0$$

$$h_5(\vec{x}) = x_6 + 100000x_{11} - 4.4 \times 10^7 = 0$$

$$h_6(\vec{x}) = x_7 + 100000x_{12} - 6.6 \times 10^7 = 0$$

$$h_7(\vec{x}) = x_5 - 120x_2x_{13} = 0$$

$$h_8(\vec{x}) = x_6 - 80x_3x_{14} = 0$$

$$h_9(\vec{x}) = x_7 - 40x_4x_{15} = 0$$

$$h_{10}(\vec{x}) = x_8 - x_{11} + x_{16} = 0$$

$$h_{11}(\vec{x}) = x_9 - x_{12} + x_{17} = 0$$

$$h_{12}(\vec{x}) = -x_{18} + \ln(x_{10} - 100) = 0$$

$$h_{13}(\vec{x}) = -x_{19} + \ln(x_8 + 300) = 0$$

$$h_{14}(\vec{x}) = -x_{20} + \ln(x_{16}) = 0$$

$$h_{15}(\vec{x}) = -x_{21} + \ln(x_9 + 400) = 0$$

$$h_{16}(\vec{x}) = -x_{22} + \ln(x_{17}) = 0$$

$$h_{17}(\vec{x}) = -x_8 - x_{10} + x_{13}x_{18} - x_{13}x_{19} + 400 = 0$$

$$h_{18}(\vec{x}) = x_8 - x_9 - x_{11} + x_{14}x_{20} - x_{14}x_{21} + 400 = 0$$

$$h_{19}(\vec{x}) = x_9 - x_{12} - 4.60517x_{15} + x_{15}x_{22} + 100 = 0$$

onde:

$$0 \leq x_1 \leq 20000$$

$$0 \leq x_i \leq 1 \times 10^6 \quad (i = 2, 3, 4)$$

$$0 \leq x_i \leq 4 \times 10^7 \quad (i = 5, 6, 7)$$

$$100 \leq x_8 \leq 299.99$$

$$100 \leq x_9 \leq 399.99$$

$$100.01 \leq x_{10} \leq 300$$

$$100 \leq x_{11} \leq 400$$

$$100 \leq x_{12} \leq 600$$

$$0 \leq x_i \leq 500 \quad (i = 13, 14, 15)$$

$$0.01 \leq x_{16} \leq 300$$

$$0.01 \leq x_{17} \leq 400$$

$$-4.7 \leq x_i \leq 6.25 \quad (i = 18, \dots, 22)$$

## G23

minimize:

$$f(\vec{x}) = -9x_5 - 15x_8 + 6x_1 + 16x_2 + 10(x_6 + x_7)$$

sujeito a:

$$g_1(\vec{x}) = x_9x_3 + 0.02x_6 - 0.025x_5 \leq 0$$

$$g_2(\vec{x}) = x_9x_4 + 0.02x_7 - 0.015x_8 \leq 0$$

$$h_1(\vec{x}) = x_1 + x_2 - x_3 - x_4 = 0$$

$$h_2(\vec{x}) = 0.03x_1 + 0.01x_2 - x_9(x_3 + x_4) = 0$$

$$h_3(\vec{x}) = x_3 + x_6 - x_5 = 0$$

$$h_4(\vec{x}) = x_4 + x_7 - x_8 = 0$$

onde:

$$0 \leq x_i \leq 300 \quad (i = 1, 2, 6)$$

$$0 \leq x_i \leq 100 \quad (i = 3, 5, 7)$$

$$0 \leq x_i \leq 200 \quad (i = 4, 8)$$

$$0.01 \leq x_9 \leq 0.03$$

## G24

minimize:

$$f(\vec{x}) = -x_1 - x_2$$

sujeito a:

$$g_1(\vec{x}) = -2x_1^4 + 8x_1^3 - 8x_1^2 + x_2 - 2 \leq 0$$

$$g_2(\vec{x}) = -4x_1^4 + 32x_1^3 - 88x_1^2 + 96x_1 + x_2 - 36 \leq 0$$

onde:

$$0 \leq x_1 \leq 3$$

$$0 \leq x_2 \leq 4$$